

calculabilit 

DEGR S TURING ←

TH RIE ALGORITHMIQUE DE L'AL ATOIRE ←

MATH MATIQUES   REBOURS ←

HYPERCALCULABILIT  ←

Beno t Monin
Ludovic Patey



C&M

TABLEAU NOIR

Tableau Noir

- 101. — Rached Mneimné. *Réduction des endomorphismes*
- 102. — Marc Hindry. *Arithmétique*
- 103. — Jean-Denis Eiden. *Géométrie analytique classique*
- 104. — Denis Choimet et Hervé Queffélec. *Analyse mathématique. – Grands théorèmes du vingtième siècle*
- 105. — Pascal Boyer. *Algèbre et géométries*
- 106. — Patrick Dehornoy. *La théorie des ensembles*
- 107. — Benoît Monin et Ludovic Patey. *Calculabilité*
- 108. — Gilles Godefroy. *Introduction aux méthodes de Baire* (à paraître)

Benoît Monin et Ludovic Patey

Calculabilité

Aléatoire, mathématiques à rebours
et hypercalculabilité



Calvage & Mounet



BENOÎT MONIN est maître de conférences à l'université de Créteil. Ses travaux portent sur la calculabilité classique, l'aléatoire algorithmique et l'hypercalculabilité.

benoit.monin@computability.fr
<https://www.lacl.fr/%7Ebenoit.monin/>

LUDOVIC PATEY est chargé de recherche à l'université de Paris. Ses travaux portent sur la calculabilité classique et les mathématiques à rebours.

ludovic.patey@computability.fr
<https://ludovicpatey.com/>

Mathematics Subject Classification (2010) – Primary :

03-Dxx Computability and recursion theory
03D30 Other degrees and reducibilities in computability and recursion theory
03D60 Computability and recursion theory on ordinals, admissible sets, etc.
03D80 Applications of computability and recursion theory
03D10 Turing machines and related notions
03D20 Recursive functions and relations, subrecursive hierarchies
03D25 Recursively (computably) enumerable sets and degrees
03D32 Algorithmic randomness and dimension
03D55 Hierarchies of computability and definability
03D78 Computation over the reals, computable analysis
03B30 Foundations of classical theories (including reverse mathematics)
68-Qxx Theory of computing
68Q05 Models of computation (Turing machines, etc.)
68Q30 Algorithmic information theory (Kolmogorov complexity, etc.)
03B10 Classical first-order logic
03C07 Basic properties of first-order languages and structures
03F35 Second- and higher-order arithmetic and fragments
03F40 Gödel numberings and issues of incompleteness
03F60 Constructive and recursive analysis

ISBN 978-2-91-635296-1



∞ Imprimé sur papier permanent

© Calvage & Mounet, Paris, 2022

à nos familles

Table des matières

Préambule

Remerciements

1. Introduction

1-1. Qu'est-ce qu'une fonction calculable ?	4
1-2. Quelles sont les fonctions incalculables ?	5
1-3. Motivations	6
1-4. Panorama de la calculabilité	7

2. Infinis de Cantor

2-1. Équipotence et subpotence	15
2-2. Théorème de Cantor–Bernstein	16
2-3. Ensembles dénombrables	18
2-4. Argument diagonal de Cantor	21
2-5. Réels non calculables	23
2-6. Espace de Cantor	24

I Calculabilité classique 29

3. Fondements de la calculabilité

3-1. Fonctions calculables	31
3-2. Ensembles calculables	36
3-3. Programme universel	37
3-4. Théorème SMN	38
3-5. Lemme de remplissage	40
3-6. Théorème du point fixe de Kleene	41
3-7. Ensembles calculatoirement énumérables	44

4. Degrés Turing	
4-1. Les chaînes finies	51
4-2. Calcul avec oracle	52
4-3. Relativisation des preuves	54
4-4. Propriété de l'usage	56
4-5. Degrés Turing	57
4-6. Saut Turing	60
4-7. Calculabilité à la limite	61
4-8. Méthode des extensions finies	66
4-9. Degrés low	71
4-10. Degrés high	74
5. Hiérarchie arithmétique	
5-1. Propriétés élémentaires	82
5-2. Hiérarchie arithmétique et calculabilité	87
5-3. Relativisation à un oracle	88
5-4. Degrés many-one	89
5-5. Théorème de Post	91
5-6. Théorème de Rice	93
5-7. Codes arithmétiques	94
6. La thèse de Church-Turing	
6-1. L'Entscheidungsproblem et la quête du Graal	99
6-2. Thèse de Church-Turing	104
6-3. Étude détaillée des fonctions récursives	107
7. Immunité et croissance de fonction	
7-1. Ensembles immunes	130
7-2. Fonctions DNC	132
7-3. Critère de complétude d'Arslanov	136
7-4. Fonctions hyperimmunes	138
7-5. Degrés calculatoirement dominés	140
7-6. Théorème de domination de Martin	147
7-7. Degrés High ou DNC	151
8. Classes Π_1^0 et degrés PA	
8-1. Arbres binaires	154
8-2. Topologie sur l'espace de Cantor	158
8-3. Classes Π_1^0	165
8-4. Théorèmes de base	168
8-5. Bases pour les classes Π_1^0 parfaites	172
8-6. Degrés PA	174
8-7. Arbres à branchement fini	179

9. Interlude formel	
9-1. Un peu d'histoire : la crise des fondements	189
9-2. La logique du premier ordre	195
9-3. Théorèmes d'incomplétudes de Gödel	212
9-4. Système ZFC	223
10. Forcing de Cohen	
10-1. Formules de l'arithmétique du second ordre	232
10-2. Forcing Σ_1^0/Π_1^0	234
10-3. Généricité effective	243
10-4. Forcing Σ_n^0/Π_n^0	258
10-5. Ensembles arbitrairement génériques	266
11. Forcing effectif	
11-1. Fondements du forcing	273
11-2. Relation de forcing	276
11-3. Forcing avec des arbres	279
11-4. Complexité calculatoire et question de forcing	283
12. La quête de degrés naturels	
12-1. Trois problèmes indécidables emblématiques	298
12-2. Approche pour la naturalité des degrés Turing	301
12-3. Problèmes de masse	305
13. Méthode de priorité et degrés c. e.	
13-1. Degrés c. e.	310
13-2. Méthode de permission	311
13-3. Méthode de priorité Σ_1^0 (à blessure finie)	312
13-4. Méthode de priorité Σ_2^0	320
13-5. Méthode de priorité Π_2^0 (à blessure infinie)	323
14. Structure des degrés Turing	
14-1. Degrés minimaux	336
14-2. Nature de $\mathcal{D}$	342
14-3. Universalité de $\mathcal{D}$	347
14-4. Théorie du premier ordre de $\mathcal{D}$	352
14-5. Structure des degrés c. e.	360

II Aléatoire algorithmique

363

15. Introduction

16. Complexité de Kolmogorov et nombres aléatoires	
16-1. Complexité de Kolmogorov	370
16-2. Nombres aléatoires à la Chaitin/Levin	379
16-3. Caractérisation de K	385
16-4. Ensembles K -triviaux	389
17. Boréliens, mesure et calculabilité	
17-1. Un peu d'histoire	395
17-2. Premières intuitions sur la mesure	399
17-3. Classes boréliennes	402
17-4. Mesure de Lebesgue	407
18. Aléatoire au sens de Martin-Löf	
18-1. Intuitions et définitions	415
18-2. Les aléatoires de Martin-Löf et de Chaitin/Levin coïncident	418
18-3. Aléatoire et degré Turing	419
18-4. Aléatoire et degré DNC	423
19. Autres notions d'aléatoire	
19-1. Les fortement MLR	427
19-2. Relativisation de l'aléatoire	432
19-3. Les 2-aléatoires	436
19-4. Aléatoires incomplets	439
20. Les K-triviaux	
20-1. Lowness et bases pour l'aléatoire	445
20-2. Le processus d'or	450
20-3. Caractérisation des K -triviaux c. e.	462
20-4. Une nouvelle preuve de K -trivial implique low-pour- K	470

III Mathématiques à rebours 473

21. Introduction	
21-1. Quête des axiomes optimaux	475
21-2. Comparaison des théorèmes	479
22. Arithmétique du second ordre	
22-1. Langage de Z_2	482
22-2. La théorie Z_2	484
22-3. Sémantiques de l'arithmétique du second ordre	486
22-4. Formaliser l'analyse dans Z_2	491
22-5. RCA_0 ou les mathématiques calculables	495
22-6. ACA_0 et la hiérarchie arithmétique	501

22-7. WKL_0 et l'argument de compacité	505
22-8. Systèmes plus puissants	511
23. Induction et conservation	
23-1. Fonctions RCA_0 -prouvablement calculables	520
23-2. Sous-systèmes faibles de PA	522
23-3. Hiérarchies d'induction	528
23-4. Fonctions primitives récursives et RCA_0	535
23-5. Le schéma de compréhension bornée	539
23-6. Théorèmes de conservation	542
23-7. Programme de Hilbert	553
24. Réductions calculatoires	
24-1. ω -réduction	560
24-2. Réduction calculatoire	565
24-3. Réduction Weihrauch	567
24-4. Jeux de réduction	571
24-5. Réductions fortes	573
25. Théorème de Ramsey	
25-1. Aperçu général	578
25-2. Théorème de Ramsey dans la hiérarchie arithmétique	581
25-3. Principe infini des tiroirs	593
25-4. Théorème de Ramsey pour les paires	612

IV Hypercalculabilité 625

26. Introduction	
26-1. Motivations	628
26-2. Panorama de l'hypercalculabilité	630
26-3. Correspondance avec la calculabilité classique	631
27. Nombres transfinis	
27-1. Motivation : itérations calculables du saut	635
27-2. Ordinaux	639
27-3. Induction et récurrence transfinie	647
27-4. Ordinaux dénombrables et indénombrables	653
27-5. Ordinaux effectifs	657
27-6. Relativisation	664

28. Ensembles hyperarithmétiques	
28-1. Hiérarchie de Kleene	667
28-2. Les singletons Π_2^0	675
28-3. Relativisation	677
28-4. Hiérarchie borélienne effective	678
29. Au delà des hyperarithmétiques	
29-1. Un peu d'histoire : l'école de Moscou	683
29-2. Quantifications du second ordre	688
29-3. Les Π_1^1 et les bons ordres	693
29-4. Analogies entre ensembles Π_1^1 et ensembles c.e.	697
29-5. Théorème d'équivalence de Kleene/Souslin	700
29-6. Autres théorèmes de majoration	707
29-7. Réduction hyperarithmétique	709
30. Classes Σ_1^1 et Π_1^1	
30-1. Représentation canonique des classes Σ_1^1	711
30-2. Théorèmes de base pour les classes Σ_1^1	713
30-3. L'hypothèse du continu pour les classes Σ_1^1	717
30-4. Quelques classes Π_1^1 emblématiques	720
30-5. Étude d'une classe Π_1^1 très spéciale	723
30-6. Les singletons Π_1^1	728
31. Les systèmes ATR_0 et $\Pi_1^1\text{-CA}_0$	
31-1. Définitions	733
31-2. ATR_0 et $\Pi_1^1\text{-CA}_0$ en hypercalculabilité	736
31-3. HYP n'est pas modèle de ATR_0	737
31-4. Codes d'ordinaux non standard	741
31-5. Séparation entre ATR_0 et $\Pi_1^1\text{-CA}_0$	747
A. Correction des exercices	
Bibliographie	797
Notations	813
Index	819

Préambule

Ce livre est une introduction à la théorie de la calculabilité ainsi qu'à trois de ses ramifications principales, à savoir la théorie algorithmique de l'aléatoire, les mathématiques à rebours et l'hyperc calculabilité. Il s'agit d'un ouvrage principalement destiné aux étudiants et enseignants en master de recherche en informatique et mathématiques, ainsi qu'aux chercheurs désirant acquérir une solide connaissance en calculabilité.

Raison d'être du livre

Cet ouvrage trouve ses origines dans plusieurs constats.

- ▷ La littérature francophone sur la calculabilité classique est quasiment inexistante. Il existe quelques introductions à la calculabilité en français qui s'intègrent dans le cadre d'une présentation de la théorie de la complexité, comme l'excellent ouvrage d'Olivier Carton *Langages formels, calculabilité, et complexité* [28]. Les modèles de calcul sont alors présentés comme une base solide pour développer une théorie de la complexité algorithmique. Les résultats de calculabilité à proprement parler dépassent rarement l'indécidabilité du problème de l'arrêt et la définition de la hiérarchie arithmétique.
- ▷ Il existe de nombreux livres de référence en anglais sur la calculabilité (*Classical Recursion Theory : The Theory of Functions and Sets of Natural Numbers* de Piergiorgio Odifreddi [169], *Computability Theory* de Barry Cooper [41] ou encore *Turing computability : Theory and Applications* de Robert Soare [208]). Concernant la théorie algorithmique de l'aléatoire, on citera *Computability and Randomness* d'André Nies [167], et *Algorithmic Randomness and Complexity* de Rodney

Downey et Denis Hirschfeldt [50]. En mathématiques à rebours, la référence historique est *Subsystems of Second Order Arithmetic* de Stephen Simpson [203]. On mentionnera l'ouvrage plus récent de Denis Hirschfeldt, *Slicing the Truth* [87], et l'ouvrage en cours de rédaction, *Reverse Mathematics* de Damir Dzhafarov et Carl Mummert. Enfin, en hypercalculabilité, les deux références sont *Higher recursion theory* de Gerald Sacks [192] et *Recursion Theory : Computational Aspects of Definability* de Chi Tat Chong et Liang Yu [35]. Chacun de ces ouvrages présente l'état de l'art de la recherche pour un sous-domaine spécifique de la calculabilité, mais il n'existe pas un unique livre qui organise une présentation cohérente de ces différents aspects.

- ▷ Le calculabilité est un sujet d'étude extrêmement vaste, comme le laisse entrevoir la profusion d'articles sur le sujet et la taille des ouvrages de référence en anglais. Cependant, ce domaine reste très peu représenté en France, aussi bien du point de vue de la recherche que de l'enseignement. Peu de cours de calculabilité sont proposés en master, et leur contenu tend à véhiculer l'idée fausse que la calculabilité est une théorie des modèles de calcul. Cet ouvrage a pour ambition de donner ses lettres de noblesse à la calculabilité en France en donnant un petit panorama de sa grande richesse, encore trop méconnue.

Organisation du livre

Ce livre est structuré en quatre grandes parties, à savoir la calculabilité classique, l'aléatoire algorithmique, les mathématiques à rebours et l'hypercalculabilité.

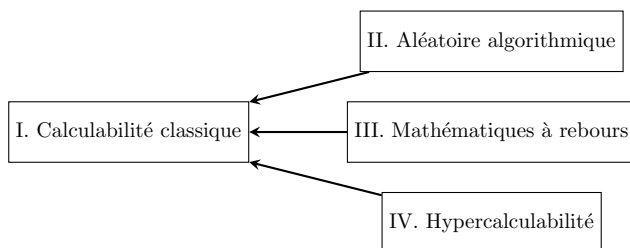
Plan général

- ▷ La *calculabilité classique* est l'étude des degrés Turing, autrement dit la puissance calculatoire des ensembles d'entiers naturels. Elle constitue le cœur historique de la calculabilité, et le socle épistémique sur lequel s'appuient les trois parties suivantes.
- ▷ L'*aléatoire algorithmique* utilise la calculabilité classique pour donner un cadre effectif à la théorie de la mesure, qui permet d'étudier individuellement les suites de bits dites « aléatoires ». Les hiérarchies induites par la calculabilité classique permettent de définir différents niveaux d'aléa, à la lumière desquels on peut, par exemple, réexaminer la signification de tel ou tel théorème probabiliste stipulant qu'une propriété est vraie presque partout.
- ▷ Les *mathématiques à rebours* forment un programme sur les fondements des mathématiques, qui vise à identifier les axiomes nécessaires pour

prouver les théorèmes mathématiques de la vie de tous les jours¹. Elles reposent sur une théorie de base, RCA_0 , dont les axiomes capturent les mathématiques « calculables » grâce à une correspondance entre la calculabilité et la définissabilité par des formules logiques.

- ▷ L'*hypercalculabilité* étend la notion de calculabilité à un cadre plus général rejoignant la théorie des ensembles. De la même manière que les opérations arithmétiques élémentaires s'étendent aux ordinaux, les machines de Turing peuvent étendre leur temps de calcul, des entiers aux ordinaux, et manipuler ainsi de plus grandes classes de réels. Il s'agit de l'approche par « modèle de calcul » de l'hypercalculabilité, qui correspond, comme pour la calculabilité classique, à certaines classes logiques.

Les trois dernières parties s'appuient toutes fortement sur la calculabilité classique, mais sont relativement indépendantes entre elles, et peuvent être pour l'essentiel, lues dans un ordre quelconque :



Dépendances entre les quatre grandes parties du livre

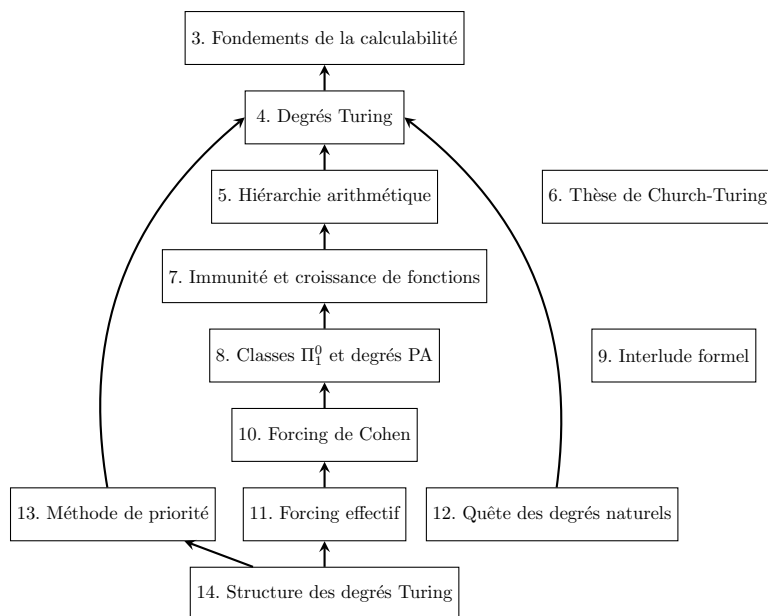
Notons tout de même la notion de « classe Borélienne » introduite dans la partie II, et fondamentale pour la compréhension de la partie IV.

Calculabilité classique

La calculabilité classique a un rôle prépondérant en ce qu'elle fixe un cadre formel et une série d'outils qui serviront à développer les parties suivantes. Il convient donc de s'attarder sur la première partie et de détailler les dépendances de ses chapitres. Les chapitres fondamentaux sont principalement à lire linéairement, avec les exceptions suivantes : les chapitres 6 et 9 peuvent être lus indépendamment des autres, mais seront cependant utiles pour aborder sereinement la partie III du livre, sur les mathématiques à rebours. Les chapitres 12 et 14 ne seront pas absolument nécessaires pour la compréhension des parties suivantes, et visent à prendre un peu de recul

1. De la vie de tous les jours du mathématicien, s'entend.

sur notre travail. Le chapitre 12 — moins technique — le fera à travers l'examen d'une question spécifique, à la frontière de la philosophie ; et le chapitre 14 à travers une étude plus abstraite de la structure des degrés Turing, et la présentation de quelques-unes des grandes questions ouvertes du domaine.



Dépendances entre les chapitres de la calculabilité classique

Comment lire ce livre

Pour les enseignants

Ce livre peut servir de support pour un cours d'introduction à la calculabilité de niveau master, ainsi que pour des cours thématiques plus avancés, parlant de théorie algorithmique de l'aléatoire, de mathématiques à rebours et d'hypercalculabilité. Les sujets abordés vont bien au-delà des connaissances en calculabilité que l'on peut attendre d'un étudiant en master. Nous allons donc proposer un plan de cours contenant les notions incontournables.

L'équivalence entre les modèles de calcul forme un socle robuste pour le développement de la calculabilité. Cependant, les preuves peuvent sembler

assez longues et fastidieuses. De nos jours, avec la démocratisation des ordinateurs, on peut s'attendre à ce que les étudiants possèdent une certaine intuition de ce qu'est un algorithme, et il semble préférable de partir de cette intuition pour faire les premiers développements afin d'éviter un formalisme relativement lourd. Nous recommandons d'aborder l'équivalence des modèles de calcul, notamment entre les fonctions générales récursives et les machines de Turing, à travers une séance de travaux dirigés, où les étudiants auront l'occasion de manipuler les formalismes en définissant des fonctions de plus en plus compliquées pour se convaincre que ces définitions permettent de capturer tous les algorithmes.

Nous invitons nos lecteurs à suivre les développements des différents chapitres 3, 4, 5, 7, 8, 10, 13 dans cet ordre. Le chapitre 3 établit les premiers théorèmes fondamentaux de la calculabilité sur la base de l'intuition que l'on a des algorithmes. Le chapitre 4 définit la notion de machine à oracle, et de degré Turing. On y trouve les définitions les plus centrales de la calculabilité, comme le saut de Turing, et la méthode des extensions finies qui est une technique très puissante pour prouver l'existence de certains degrés Turing. Le chapitre 5 sur la hiérarchie arithmétique établit un lien essentiel entre la puissance calculatoire d'ensembles d'entiers et leur définissabilité par des formules de l'arithmétique, à travers le théorème de Post.

Avec le chapitre 7, on commence l'étude de différentes propriétés calculatoires fondamentales, comme la notion de degré hyperimmune, et ses liens avec l'existence de fonctions à croissance rapide. Cette étude est poursuivie dans le chapitre 8 sur les classes Π_1^0 , où l'on définit la notion de degré PA qui est une notion centrale en calculabilité. On la retrouve tout au long de ce livre, notamment en aléatoire algorithmique et en mathématiques à rebours.

Le chapitre 10 introduit une technique fondamentale de la calculabilité, à savoir le forcing, présentée ici comme une élaboration de la méthode des extensions finies du chapitre 4. Ce chapitre peut également servir comme le premier niveau d'une compréhension graduelle de la technique générale du forcing de la théorie des ensembles.

Le chapitre 13 introduit finalement les méthodes de priorités, une autre technique fondamentale de la calculabilité, qui permet notamment de montrer l'existence de certains degrés calculatoirement énumérables.

Pour les étudiants

Les compétences requises pour comprendre les notions présentées dans cet ouvrage sont celles d'une première année de licence en informatique ou mathématiques. Il est nécessaire pour l'essentiel de comprendre le langage

mathématique usuel (variables, quantifications, etc.), d'avoir quelques notions élémentaires de logique (preuve par contraposée, preuve par l'absurde, etc.), et de comprendre les éléments du corpus mathématique de base (comprendre ce qu'est une bijection, ce qu'est une intersection entre deux ensembles, les fonctions puissance et logarithme, etc.). En plus de cela, une expérience même sommaire en programmation, ou à défaut la compréhension de ce qu'est un algorithme, est également nécessaire pour aborder sereinement la lecture de ce livre.

Les mathématiques que nous utiliserons et qui ne sont pas enseignées dans une première année de licence seront introduites et expliquées au fur et à mesure des besoins (notions de base de topologie ou de théorie de la mesure par exemple). Ayant établi cela, notons tout de même que le degré d'élaboration des preuves, ainsi que la technicité de certains concepts, rendront sans doute cet ouvrage difficile à aborder sans une certaine maturité mathématique.

Les techniques développées en calculabilité sont assez différentes de celles apprises à travers un cursus mathématique standard. Cette particularité de la calculabilité est une force et rend cette discipline plus accessible puisqu'elle est peu sensible aux lacunes que l'on peut avoir développé au cours de son cursus (ou de son absence de cursus). En revanche, cette différence peut également déstabiliser l'étudiant car cela demande de se créer un univers conceptuel. Il va sans dire qu'en l'absence de professeur, il est d'autant plus essentiel de faire les exercices proposés au cours du livre pour bien intégrer les concepts. Les solutions sont disponibles à la fin du livre.

La taille de cet ouvrage peut s'avérer décourageante pour un étudiant désireux de faire ses premiers pas en calculabilité. Nous rappelons que ce livre couvre des connaissances allant bien au-delà de ce que l'on attend d'un étudiant en master. Nous recommandons donc aux autodidactes de suivre la suggestion de cours de la section précédente, destinée aux enseignants.

Pour les chercheurs

Cet ouvrage est une introduction à la calculabilité et à plusieurs de ses branches principales. Il ne faut cependant pas s'arrêter à l'aspect introductif de cet ouvrage, car la plupart des résultats présentés correspondent à l'état de l'art de la recherche. Ce livre s'adresse donc aussi bien aux chercheurs de domaines connexes désireux d'acquérir de solides connaissances en calculabilité, qu'aux doctorants et chercheurs se destinant à la recherche en calculabilité. En effet, les techniques et concepts de ce livre rendent directement accessibles les articles de recherche du domaine.

Exercices

Les chapitres sont parsemés d'exercices de difficulté variable, dont la correction est donnée à la fin du livre. On ne rappellera jamais assez l'importance de faire des exercices pour bien assimiler les notions présentées dans les chapitres. L'intuition des concepts se crée en les manipulant sous toutes leurs formes. La difficulté des exercices est indiquée à l'aide d'un système d'étoiles (★) allant de 0 à 3 : un exercice sans étoile est une application directe des définitions, tandis qu'un exercice à deux étoiles demande une profonde maîtrise des concepts pour être résolu. On trouvera également quelques exercices à trois étoiles, qui sont d'un niveau « recherche ».

À travers cet ouvrage, nous allons présenter beaucoup de propriétés calculatoires sur les ensembles d'entiers ou sur d'autres structures plus complexes. Outre les exercices donnés, il est important de faire preuve d'une curiosité intellectuelle consistant à chercher systématiquement comment se combinent ces propriétés, savoir si l'on peut construire des objets satisfaisant plusieurs d'entre elles simultanément, et ainsi de suite. De même, lorsque les théorèmes possèdent des hypothèses, il est utile de chercher des contre-exemples sans ces hypothèses, afin de mieux comprendre leur nécessité ainsi que leur usage dans la preuve.

Errata

Il est impossible d'écrire un livre de cette taille sans laisser se glisser un certain nombre de coquilles. Cet ouvrage ne dérogera probablement pas à cette règle. Nous maintiendrons une liste des coquilles sur la page web des auteurs. Vous pouvez signaler les erreurs à l'une des adresses suivantes :

`benoit.monin@computability.fr`

ou

`ludovic.patey@computability.fr`

Remerciements

Notre tout premier remerciement va à l'Agence nationale de la recherche, qui a en grande partie financé la collaboration entre les deux auteurs, notamment pendant l'écriture du livre, dans le cadre du projet « Aspects Calculatoires des Théorèmes Combinatoires – ACTC »

<https://anr.fr/Projet-ANR-19-CE48-0012>

Plusieurs résultats présentés dans ce livre, notamment le théorème 25-3.23 sur le principe des tiroirs ou la preuve simplifiée du théorème 25-3.24 de Liu proviennent d'articles financés par ce même projet.

Nous tenons également à remercier nos équipes et organismes de rattachement, qui nous ont fourni un vivier d'émulation intellectuelle ainsi qu'un soutien moral et financier. Pendant la rédaction de cet ouvrage, Monin était maître de conférences dans le Laboratoire d'Algorithmique, Complexité et Logique de l'Université Paris-Est Créteil et Patey chargé de recherche au CNRS, dans l'équipe Algèbre, Géométrie, Logique de l'Institut Camille Jordan à Villeurbanne.

De nombreux collègues nous ont généreusement aidé à améliorer la qualité de cet ouvrage, en donnant un regard critique sur son contenu scientifique et pédagogique, fort de leurs expertises respectives, ou bien en signalant les erreurs typographiques qui se sont inévitablement glissées dans le livre. Nous remercions donc Paul-Elliot Anglès d'Auriac, Sébastien Tavenas, Pascal Vanier, Mathieu Hoyrup, Benjamin Hellouin, Laurent Bienvenu, Bruno Durand, Denis Kuperberg, Julien Cervelle, Damir Dzhafarov, Loïc Gassmann, William Gaudelier, Denis Hirschfeldt, Quentin Le Houerou, Alexander Shen, Keita Yokoyama, Adrien Deloro, Pascal Monin et Shahin Amini.

Le contenu scientifique du livre est avant tout l'œuvre de la communauté de la calculabilité. Les auteurs ont forgé leurs intuitions en lisant les ouvrages

de leurs prédécesseurs et en ajoutant leur pierre à ce magnifique édifice intellectuel. Nous tenons à remercier nos collègues au niveau international pour les collaborations et les visites mutuelles ayant permis d'améliorer notre compréhension du sujet.

Nous tenons également à remercier Laurent Bienvenu, qui par son travail et à travers la direction de nos thèses respectives, a su nous transmettre sa passion, et a largement contribué à introduire la calculabilité en France.

La rédaction d'un ouvrage de cette ampleur prend beaucoup de temps et d'énergie, et n'aurait pu avoir lieu sans le soutien moral de nos familles et amis.

Nous tenons enfin à remercier les éditions Calvage et Mounet pour leur confiance, leur soutien et leur travail éditorial, et particulièrement Rached Mneimné pour sa relecture minutieuse du livre.

Chapitre 1

Introduction

Le savant n'étudie pas la nature parce que cela est utile ; il l'étudie parce qu'il y prend plaisir et il y prend plaisir parce qu'elle est belle. Si la nature n'était pas belle, elle ne vaudrait pas la peine d'être connue, la vie ne vaudrait pas la peine d'être vécue. Je ne parle pas ici, bien entendu, de cette beauté qui frappe les sens, de la beauté des qualités et des apparences ; non que j'en fasse fi, loin de là, mais elle n'a rien à faire avec la science ; je veux parler de cette beauté plus intime qui vient de l'ordre harmonieux des parties, et qu'une intelligence pure peut saisir.

Science et méthode, Henri Poincaré

Qu'est-ce que la calculabilité ? On considère classiquement la calculabilité comme l'un des quatre piliers de la logique, aux côtés de la théorie des ensembles, la théorie des modèles et la théorie de la preuve. Le domaine s'est au départ forgé sur la question de ce qui caractérise les fonctions $f : \mathbb{N} \rightarrow \mathbb{N}$ dont les valeurs peuvent être obtenues par un processus purement *mécanisable* ou *algorithmique*, en un temps fini, bien qu'arbitrairement grand. Nous dirons que de telles fonctions sont *effectivement calculables*. Bien avant l'apparition des premiers ordinateurs, la calculabilité a fondé sa base théorique sur un constat — ou plutôt un miracle — à savoir l'existence d'une définition robuste, consensuelle et indépendante de tout formalisme, de la notion épistémologique de fonction effectivement calculable.

La question initiale — à savoir « Qu'est-ce qu'une fonction calculable ? » — ayant obtenu une réponse satisfaisante, l'étude s'est naturellement portée vers la question de savoir, parmi les fonctions naturelles, lesquelles sont calculables et lesquelles ne le sont pas. Par la suite, le domaine a connu un développement considérable grâce à la notion de calculabilité relative, la question n'étant plus de déterminer si une fonction est calculable ou non, mais d'identifier la puissance calculatoire intrinsèque à cette fonction, à travers des questions comme « Si cette fonction était calculable, quelles autres fonctions pourrait-on calculer ? »

Plus récemment, le sujet d'étude s'est étendu à de très nombreux objets mathématiques — par exemple des structures algébriques ou des sous-ensembles de $\mathbb{R}$ — et a donné de nombreuses ramifications. Nous verrons notamment dans ce livre que la calculabilité sert de fondement robuste à la théorie algorithmique de l'aléatoire, et aux mathématiques à rebours, dont les objets d'études sont les théorèmes mathématiques eux-mêmes.

De nos jours, l'appellation « calculabilité » pour un domaine qui étudie des objets mathématiques arbitraires, dont la plupart ne sont pas calculables, peut sembler étonnante, voire un reliquat de son sujet d'étude historique. En vérité, ce nom est toujours judicieux, mais sa signification a changé : le terme *calculabilité* ne porte plus sur le sujet de l'étude, mais sur l'angle sous lequel le sujet est abordé. Une définition moderne de la calculabilité en une phrase pourrait être : **la calculabilité est l'étude des mathématiques sous le prisme de leur complexité calculatoire.**

1. Qu'est-ce qu'une fonction calculable ?

La principale difficulté de cette question réside dans l'obtention d'une classe de fonctions suffisamment robuste pour ne pas dépendre du modèle d'ordinateur, du choix du langage de programmation, des progrès technologiques, ou de l'avancée de la connaissance de manière générale.

Avec l'avènement des ordinateurs, la notion d'algorithme s'est peu à peu ancrée dans la culture scientifique. Toute personne ayant déjà eu un premier contact avec la programmation se sera déjà formée une bonne idée de ce qu'est une tâche automatisable. Forts de notre connaissance de l'informatique, la définition suivante viendrait naturellement : « Une fonction est effectivement calculable si elle possède un algorithme, autrement dit si elle peut être programmée dans un langage suffisamment expressif, et exécutée par un ordinateur suffisamment puissant. »

Cette définition, si elle a l'avantage d'être en adéquation avec notre intuition, ne fournit pas un cadre suffisamment formel pour raisonner sur la classe des fonctions calculables. Une seconde approche consisterait à

fixer un ordinateur et un langage de programmation étalon, et définir une fonction comme calculable si elle est programmable dans ce langage, et exécutable par cet ordinateur en temps fini, à l'aide de suffisamment de mémoire. Si l'on ne se préoccupe pas de la rapidité d'exécution, ni de l'espace mémoire nécessaire, il apparaît rapidement que cette définition coïncide avec la précédente. En effet, la puissance et la mémoire des ordinateurs augmentent au gré des progrès technologiques, et permettent donc d'exécuter plus rapidement les programmes, mais n'augmentent pas pour autant la classe des fonctions calculables. Même les ordinateurs basés sur de nouveaux paradigmes de calcul, comme les ordinateurs quantiques ou biologiques, sont simulables — au prix d'un surcoût d'espace et de temps exponentiel — par des ordinateurs classiques, et ne changent donc pas la classe des fonctions calculables. Quant aux langages de programmation, l'existence de systèmes d'exploitation et d'interpréteurs permettent de se convaincre aisément que les principaux d'entre eux tels que C++, Java ou Python, permettent de programmer — plus ou moins élégamment — les mêmes fonctions mathématiques. Cela montre donc empiriquement une certaine robustesse dans la définition de la classe des fonctions programmables.

Un problème subsiste : quelle est la garantie que les ordinateurs actuels représentent la limite de ce qui est automatisable, ou calculable par un être humain ? Qui nous dit qu'avec les progrès de la science, nous ne découvrirons pas un nouveau paradigme de calcul ou une nouvelle manière de raisonner permettant de considérer comme calculable une plus large classe de fonctions ? C'est là le sujet d'une longue quête fondationnelle débutée au XX^e siècle, et aboutissant à la fameuse thèse de Church-Turing en 1936, que nous présenterons dans le chapitre 6.

2. Quelles sont les fonctions incalculables ?

Au regard de notre définition précédente, pour l'instant très informelle, la plupart — si ce n'est la totalité — des fonctions mathématiques utilisées au quotidien sont calculables : l'addition, la multiplication, la fonction $(n, m) \mapsto n^m$, la fonction qui à n associe le n -ième nombre premier, ou encore celle qui calcule le plus grand diviseur commun de deux entiers naturels, sont toutes calculables. On peut rajouter à cette liste des exemples moins triviaux : la fonction qui prend un programme informatique écrit en C++ et détermine si le programme est syntaxiquement correct — c'est ce que fait entre autres choses un compilateur pour C++ — ou encore celle qui renvoie la n -ième décimale de π , $\sqrt{2}$ ou du nombre d'or — chacun de ces nombres est la somme d'une suite calculable de rationnels de convergence

suffisamment rapide — ou pour finir celle qui à n associe le nombre de parties possibles que l'on peut faire au jeu de go sur un plateau — appelé aussi *goban* — de taille $n \times n$. Ce dernier exemple illustre en particulier le fait suivant : on ne s'occupe pas en calculabilité du temps que prend un calcul. Seule l'existence d'un algorithme nous importe. Dans le cas du nombre de parties au jeu de go, l'algorithme en question repose sur une idée simple ; il « suffit » de lister toutes les parties possibles et de les compter. Cependant, le temps d'exécution d'un tel algorithme est tellement grand que cela le rend impossible à utiliser en pratique pour $n > 2$ ⁽¹⁾. Pour $n = 19$, qui est la taille d'un goban standard, ce nombre est compris entre $10^{10^{48}}$ et $10^{10^{171}}$ [225], ce qui fait clairement trop de parties à compter même si tous les ordinateurs de monde s'y attelaient pour un milliard d'années...

Même si la fonction de multiplication par 2 nous est, en un sens, bien plus accessible que celle qui compte le nombre de parties au jeu de go, il existe un algorithme qui calcule chacune d'entre elles. Ces deux fonctions ne sont donc pas différentes l'une de l'autre du point de vue de la calculabilité : elles sont toutes les deux calculables, et nous allons principalement nous intéresser aux fonctions qui *ne le sont pas*, c'est-à-dire les fonctions dont les valeurs *ne peuvent pas* être obtenues par un processus purement mécanisable ou algorithmique. La simple existence de telles fonctions n'est pas une évidence en soi, et l'une des premières tâches à laquelle nous nous attellerons sera d'en montrer l'existence. Cela sera fait dans le chapitre suivant via l'argument diagonal de Cantor. Nous donnerons ensuite tout au long du livre de très nombreux exemples de telles fonctions, la plus connue d'entre elles étant sans doute le *problème de l'arrêt*, défini comme la fonction qui prend en entrée un programme, et détermine si son exécution va s'arrêter, en temps fini nécessairement. Nous verrons que le problème de l'arrêt n'est pas calculable ; et il est important de comprendre qu'il s'agit bien ici d'une impossibilité théorique et fondamentale, qui ne dépend pas de la puissance ou vitesse de calcul des ordinateurs. L'incalculabilité du problème de l'arrêt n'est pas due à une ignorance de son algorithme qui pourrait être un jour découvert, mais bien à une impossibilité absolue, car l'existence d'un tel algorithme entraînerait un paradoxe.

3. Motivations

La calculabilité porte principalement sur l'étude des fonctions — ou objets mathématiques plus généraux — incalculables. Il est légitime de se demander si une telle étude est bien raisonnable. Si même certaines fonctions calculables nous sont inaccessibles — comme le nombre de parties possibles

1. Il y a déjà 386 356 909 593 parties possibles sur un goban de taille 2×2 [225] !

au jeu de go — alors à quoi bon se donner la peine de réfléchir sur des fonctions *encore plus inaccessibles* ?

Une première motivation pour notre étude est d'ordre exploratoire. Il existe des objets inaccessibles, essayons d'en explorer l'univers. Simplement parce qu'il est là, et par curiosité sur les mystères qu'il renferme. Nos efforts seront récompensés par une série de théorèmes d'une très grande profondeur. Quiconque se plonge avec sérieux dans les développements de ce livre, une fois peut-être passé quelques difficultés d'adaptation inhérentes à toute discipline scientifique, verra un monde d'une richesse stupéfiante prendre vie dans son esprit, avec sa faune et sa flore, ses règles et mécanismes. La calculabilité est caractérisée par la nature très dynamique de ses preuves, chacune d'elles en offrant un aperçu sur le fonctionnement détaillé d'un fragment de la machinerie titanesque qui anime cet univers.

Une deuxième motivation survient tout simplement par nécessité. Les Pythagoriciens se sont retrouvés contraints et forcés d'admettre l'existence de mesures irrationnelles, comme la diagonale d'un carré de côté 1, ce qui allait à l'encontre de leur compréhension du monde, qu'ils pensaient explicable uniquement en se fondant sur les rapports entre nombres entiers. Mais si l'on admet l'existence des entiers, et l'existence du carré, on est forcé d'admettre celle de quantités *incommensurables*, que l'on appelle aujourd'hui irrationnelles, comme $\sqrt{2}$. De la même manière, nous verrons que si l'on admet l'existence des objets calculables, on est forcé aussi d'admettre l'existence d'objets qui ne le sont pas, et qui apparaissent malgré tout naturellement dans toute une série de situations.

Une dernière motivation enfin est d'ordre pratique. La calculabilité, via la compréhension qu'elle donne des objets incalculables, a obtenu des succès majeurs en fournissant un cadre formel pour l'étude de questions à la frontière entre science et philosophie. Nous en verrons deux : la recherche de la définition d'objets aléatoires avec la partie II, et la compréhension de ce que signifie *la force* d'un théorème, notamment par rapport à un autre, avec la partie III. La partie IV de ce livre amènera quant à elle la calculabilité vers la frontière qu'elle partage avec la théorie des ensembles.

4. Panorama de la calculabilité

La calculabilité peut se décomposer en plusieurs sous-domaines, qui s'appuient tous sur la même notion robuste de fonction effectivement calculable.

4.1. Domaines couverts par ce livre

Cet ouvrage est décomposé en quatre parties, chacune d'entre elles couvrant une ramification de la calculabilité : la calculabilité classique, l'aléatoire algorithmique, les mathématiques à rebours, et l'hypercalculabilité.

Calculabilité classique

Comme nous l'avons spécifié, la calculabilité porte avant tout sur des objets que l'on ne peut pas calculer. La calculabilité classique se concentre sur les fonctions $f : \mathbb{N} \rightarrow \mathbb{N}$ ainsi que sur les ensembles d'entiers $E \subseteq \mathbb{N}$. Remarquons qu'un tel ensemble peut aussi être représenté par sa fonction caractéristique $\chi_E : \mathbb{N} \rightarrow \mathbb{N}$ définie par $\chi_E(x) = 1$ si $x \in E$, et $\chi_E(x) = 0$ sinon.

Les développements de la calculabilité classique s'articulent autour d'un outil fondamental qui nous permettra de comparer ou encore de mesurer le *degré d'incalculabilité* d'une fonction, appelé aussi *degré d'insolubilité* ou encore *degré Turing*, en référence au mathématicien Alan Turing qui introduisit la notion. Fixons une fonction non calculable $g : \mathbb{N} \rightarrow \mathbb{N}$. Il est naturel de se demander « Si j'étais capable de calculer g , quelles autres fonctions pourrais-je calculer ? » On dit qu'une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ est *calculable relativement à g* (ou g -calculable) s'il existe un algorithme permettant de calculer f dans un langage de programmation étendu, où l'on aurait rajouté la fonction g comme primitive : une instruction spéciale nous permet d'appeler la fonction g sur un paramètre n dans notre programme, comme si elle existait réellement, et d'en récupérer le résultat. Si f est g -calculable, rien ne nous indique comment calculer g , mais si l'on disposait d'un « oracle » nous permettant de calculer les valeurs de g , il serait possible de calculer les valeurs de f .

Cette notion de calculabilité relative nous permet de définir un pré-ordre partiel entre les fonctions, notant $f \leq_T g$ si la fonction f est g -calculable. Il s'agit de la *réduction Turing*. Différentes fonctions peuvent porter la même puissance calculatoire, au sens où elles sont mutuellement calculables. On définit donc le *degré Turing* d'une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ l'ensemble $\deg_T f$ de toutes les fonctions g telles que $f \leq_T g$ et $g \leq_T f$. La notion de degré Turing représente une puissance calculatoire, au sens où deux fonctions de même degré Turing sont indistinguables du point de vue de la calculabilité. Le pré-ordre partiel sur les fonctions induit un ordre partiel sur les degrés Turing.

La calculabilité classique porte principalement sur l'étude des degrés Turing munis de la relation d'ordre partielle définie ci-dessus. Existe-t-il une infinité de puissances calculatoires ? Sont-elles linéairement ordonnées ? Plus généralement, quelles sont les propriétés de cet ordre partiel ? Il s'avère que cette structure est extrêmement riche et complexe, comme nous aurons l'occasion de le voir.

Aléatoire algorithmique

La théorie classique des probabilités étudie les phénomènes probabilistes, modélisés avec succès via la notion de mesure qui sert à définir formellement

les lois de probabilité. Cette théorie n'a en revanche pas les outils nécessaires — et ce n'est pas là son objectif — pour parler d'objets aléatoires *individuellement*. C'est ce point précis que la théorie algorithmique de l'aléatoire se propose d'éclaircir, en s'appuyant sur la calculabilité. Voyons à travers un exemple de quoi il s'agit.

Représentons un nombre réel $R \in [0, 1]$ par son développement binaire, de la forme $R = 0.b_0b_1b_2b_3 \dots$ où $(b_n)_{n \in \mathbb{N}}$ est une suite de bits. Supposons que le réel R soit obtenu en tirant ses bits *au hasard* par une suite de tirage à pile ou face. On suppose bien entendu que chaque tirage est *équiprobable* : nous avons une chance sur deux d'obtenir pile et une chance sur deux d'obtenir face. L'intuition nous dicte que le réel R ainsi obtenu est *aléatoire*. Que cela signifie-t-il exactement ? On ne s'attend par exemple pas à n'obtenir que des « pile » sur les cent mille premiers lancers : si chaque tirage est équiprobable, cela ne peut arriver, ou en tout cas avec une probabilité tellement faible que l'on peut la considérer comme négligeable. On ne s'attend pas non plus à obtenir deux fois plus de « pile » que de « face ». Là encore, la probabilité que cela arrive sur cent mille tirages est tellement faible que l'on supposera les tirages biaisés plutôt que d'être témoin d'un événement si improbable. On peut de fait identifier une première propriété que l'on est en droit d'attendre d'une suite de tirages équiprobables : la suite obtenue devrait respecter la loi des grands nombres, c'est-à-dire que les nombres de tirages « pile » et « face » doivent à peu près être les mêmes.

Cela est-il pour autant suffisant ? Supposons à présent par exemple que sur chaque tirage numéro n , si n est un nombre premier on obtient systématiquement un « pile ». Dans l'hypothèse où une certaine obsession des nombres premiers nous conduirait à remarquer ce curieux phénomène, nous serons là encore en face d'une énigme — un peu absurde — et l'on sera amené à penser que d'une manière ou d'une autre, quelque chose d'anormal est en train de se produire. Mais prenons encore plus de recul. Au fond, et peu importe la suite de bit obtenue, on peut identifier des nombres $n_1 < n_2 < n_3 < \dots$ tels que les tirages numéros $n_1, n_2, n_3, \dots$ sont tous des tirages « pile ». Dans le cas où notre suite $n_1, n_2, n_3, \dots$ contient les nombres premiers, cela nous semble relever d'un « bug probabiliste », mais pourquoi cela devrait-il être acceptable si $n_1, n_2, n_3, \dots$ sont des nombres entiers quelconques ? C'est là que la calculabilité entre en jeu, et va nous permettre de formaliser précisément les propriétés que devrait avoir — en accord avec notre intuition humaine — une suite de bits aléatoire.

Mathématiques à rebours

La notion de *théorème* est relative à un système d'axiomes. Lorsque l'on omet de mentionner le système de référence, il est communément admis que l'on se réfère au système de Zermelo Fraenkel (ZF), qui représente

un ensemble d'axiomes consensuels servant de fondement à l'ensemble des mathématiques. Le système ZF est cependant très puissant, et nous n'avons aucune garantie de son absence de contradiction.

Les mathématiques à rebours visent à trouver les axiomes nécessaires et suffisants pour prouver les théorèmes des mathématiques de tous les jours. Il s'agit donc d'étudier des théorèmes existants, pour en trouver des preuves plus élémentaires, ou au contraire pour montrer l'optimalité de leur preuve. Mieux comprendre les hypothèses des théorèmes permet de mieux maîtriser leur « fragilité » face à une potentielle contradiction du système de preuves. Il s'agit donc d'une démarche méta-mathématique visant à répondre à la question « Quelle confiance peut-on avoir en nos mathématiques ? »

Au premier abord, cette démarche n'est pas liée à la calculabilité. Cependant, les mathématiques à rebours se réfèrent à une théorie de base, RCA_0 , capturant les *mathématiques calculables*, et qui représente une base de confiance plus en lien avec le monde concret, car ses objets étant calculables, ils peuvent être représentés par un algorithme, donc possèdent une description finitaire. Les mathématiques à rebours consistent donc, étant donné un théorème T , à chercher des axiomes A tels que RCA_0 prouve l'équivalence entre A et T . Par le choix de la théorie de base RCA_0 , les équivalences sont des procédés calculatoires faisant appel aux outils de la calculabilité.

Hypercalculabilité

Une des raisons du succès de la calculabilité en tant qu'outil d'analyse des mathématiques réside dans l'existence d'une solide intuition de la notion de calcul, permettant ainsi de guider la manipulation des concepts et de prouver des théorèmes sans s'embarrasser d'un lourd formalisme. L'hypercalculabilité vise à étendre la portée de ces outils à des modèles de calcul plus puissants, qui peuvent être vus comme des machines ayant la possibilité de poursuivre leur exécution pendant un temps de calcul infini (formellement en temps de calcul ordinal). Tout comme les notions de calculabilité classique peuvent être capturées par des formules logiques, il en va de même pour l'hypercalculabilité. Par exemple, là où les ensembles d'entiers que l'on peut énumérer (dans le désordre) par un programme informatique sont ceux qui peuvent être décrits par une formule dite Σ_1^0 de l'arithmétique, ceux qui sont énumérables par un programme informatique hypercalculable sont ceux qui peuvent être définis par une formule dite Π_1^1 de l'arithmétique.

Nous verrons que cet aspect des choses rapproche l'hypercalculabilité de la théorie descriptive des ensembles, une branche de la théorie des ensembles qui classe ces derniers selon le degré de difficulté à les décrire. L'hypercalculabilité peut être vue comme un pont entre la théorie descriptive des ensembles et la calculabilité classique.

4.2. Autres branches de la calculabilité

Afin de permettre à cet ouvrage de conserver une taille raisonnable, nous avons fait le choix de faire l'impasse sur deux branches importantes de la calculabilité, à savoir la théorie des structures calculables et l'étude des degrés d'énumération.

Théorie des structures calculables

Il s'agit d'une branche de la calculabilité qui étudie dans quelle mesure les propriétés algébriques d'une structure mathématique affectent leur complexité descriptive. Par structure, on entend des ensembles munis d'opérations, comme les groupes, les anneaux et les corps, mais également toute structure au sens de la théorie des modèles. Cette branche emprunte ses techniques à la fois à la théorie des modèles et à la calculabilité classique pour répondre à cette question.

Concrètement, cette théorie étudie des structures dénombrables et pose des questions de la forme « Étant donné une structure calculable $\mathcal{A}$, quels sont les degrés Turing possibles des structures isomorphes à $\mathcal{A}$? » ou bien « Étant donné deux structures calculables et isomorphes, de quelle puissance calculatoire a-t-on besoin pour calculer leur isomorphisme ? » Par exemple, les instances calculables des ordres denses sans extrémités sont toutes deux à deux calculatoirement isomorphes. On les appelle *calculatoirement catégoriques*.

Degrés d'énumération

La calculabilité classique place « le calculable » comme puissance calculatoire de référence. Mais certains problèmes s'expriment de manière naturelle sous forme d'ensembles non calculables, dont les éléments peuvent toutefois être énumérés dans le désordre par un processus calculable. On appelle ces ensembles *calculatoirement énumérables* (c. e.). En particulier, si E est un ensemble d'entiers c. e. et si $n \in E$, il est possible de s'en rendre compte en un temps fini, en lançant la procédure d'énumération et en attendant que n apparaisse. En revanche, si $n \notin E$, alors il ne sera pas possible en général de le savoir en un temps fini. Par exemple, l'ensemble des équations diophantiennes (des équations à coefficients entiers, comme $3x^3 - 2y^2 + x - 2 = 0$) qui admettent des solutions entières est calculatoirement énumérable, car il suffit de chercher exhaustivement des solutions, et d'énumérer l'équation si une telle solution existe.

Les degrés d'énumération placent « le calculatoirement énumérable » comme puissance de référence. On peut définir une *réduction d'énumération* $A \leq_e B$ ssi toute énumération des éléments de B calcule une énumération des éléments de A . Cette réduction est une pré-ordre partiel, qui induit une

notion de *degré d'énumération* : le degré d'énumération de A est l'ensemble $\deg_e(A)$ de tous les ensembles B tels que $A \leq_e B$ et $B \leq_e A$. L'étude des degrés d'énumération munis de l'ordre partiel $\leq_e$ constitue une branche active de recherche en calculabilité.

Chapitre 2

Infinis de Cantor

Si l'on devait donner une date à la naissance de la logique moderne, nous situerions sans hésiter celle-ci en 1872, date à laquelle Georg Cantor expose sa première démonstration du théorème 4.1 à venir, où il établit la non-dénombrabilité des nombres réels. Cantor isolera plus tard la quintessence de cette première preuve à travers son fameux *argument diagonal*, qui aura une place centrale en calculabilité.

Les travaux de Cantor sur l'infini marquent le début d'une théorie des ensembles « complexe », qui jouera un grand rôle dans la quête fondationnelle des mathématiques du début du XX^e siècle, dont nous parlerons en détail en première partie du chapitre 9, sur la fameuse « crise des fondements ». Cette crise débouchera sur le développement de la logique mathématique telle que nous la connaissons aujourd'hui, avec la théorie des ensembles moderne, dite ZFC, mais aussi avec le développement des premières théories du calcul, utilisées par Gödel pour montrer son fameux théorème d'incomplétude, que l'on peut considérer comme une déclinaison sophistiquée de l'argument diagonal de Cantor.



Georg Cantor, 1845–1918

De quoi s'agit-il exactement ? Cantor montre que les ensembles infinis n'ont pas tous la même « taille ». Il y a strictement plus de nombres réels que de nombres entiers, dans un sens que nous définirons précisément dans quelques lignes. Cantor se basera sur cette découverte pour développer une étude mathématique de l'infini. Il créera notamment les nombres transfinis, qui constitueront la colonne vertébrale des définitions mathématiques, et que nous aborderons dans le chapitre 27.

Cantor ne fut cependant pas le premier à remarquer que l'infini n'obéissait pas aux mêmes règles que le fini. En particulier, une caractéristique surprenante des ensembles infinis est que le tout n'est pas forcément plus grand que ses parties. Galilée en fait une exposition lumineuse dans son ouvrage « Discours concernant deux sciences nouvelles » [71] à travers un dialogue savoureux entre deux personnages, Salviati et Simplicio :

- Salviati. *J'estime que les épithètes comme « plus grand », « plus petit » et « égal » ne conviennent pas aux grandeurs infinies, dont il est impossible de dire que l'une est plus grande, plus petite ou égale à une autre. Mais voici pour le prouver un raisonnement qui me revient à l'esprit : vous savez parfaitement je suppose quels nombres sont carrés et quels nombres ne le sont pas.*
- Simplicio. *Je sais parfaitement qu'un nombre carré provient de la multiplication d'un autre nombre par lui-même ; ainsi quatre, neuf, etc. sont des nombres carrés résultant de la multiplication de deux, trois, etc. par eux-mêmes.*
- Salviati. *Fort bien, quant aux nombres qui ne proviennent pas de nombres multipliés par eux-mêmes, ce ne sont pas des carrés. Par conséquent, si je dis que les nombres pris dans leur totalité, en incluant les carrés et les non-carrés, sont plus nombreux que les carrés seuls, j'énoncerai, n'est-ce pas, une proposition vraie ?*
- Simplicio. *Très certainement.*
- Salviati. *Si je demande maintenant combien il y a de nombres carrés, on peut répondre sans se tromper qu'il y en a autant que de racines correspondantes, attendu que tout carré a sa racine et toute racine son carré, qu'un carré n'a pas plus d'une racine et une racine pas plus d'un carré.*
- Simplicio. *Exactement.*
- Salviati. *Mais si je demande combien il y a de racines, on ne peut nier qu'il y en a autant que de nombres, puisque tout nombre est la racine de quelque carré il faudrait admettre que les carrés sont aussi nombreux que tous les nombres pris ensemble.*

On observe à la lecture du dialogue de Galilée le piège du paradoxe se refermer sur Simplicio. Galilée utilise pour cela un concept qui sera repris par Cantor : deux ensembles A et B « ont le même nombre d'éléments » si l'on peut faire correspondre exactement les éléments de A et les éléments de B , autrement dit s'il existe une bijection entre les deux ensembles.

1. Équipotence et subpotence

Rappelons qu'une fonction $f : E \rightarrow F$ est *injective* si

$$\forall x, y \quad x \neq y \Rightarrow f(x) \neq f(y),$$

surjective si son image est l'ensemble F tout entier, et *bijjective* si elle est à la fois injective et surjective.

Définition 1.1. Deux ensembles E et F sont *équipotents* s'il existe une bijection entre eux. On écrira alors $|E| = |F|$. ◇

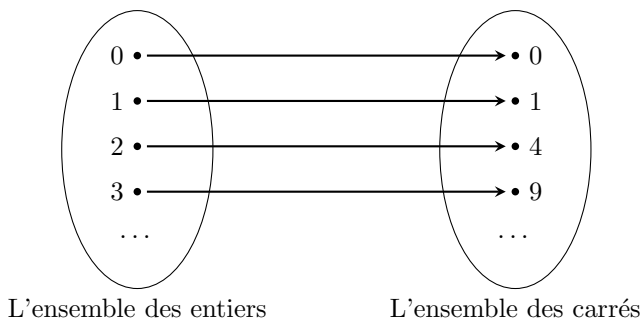


FIGURE 1.2 – L'argument de Galilée pour dire qu'il y a « autant » d'entiers que de carrés

Selon notre définition les entiers, et les carrés d'entiers ont donc la même cardinalité : il y a autant d'éléments dans les deux ensembles. Ce qui semble paradoxal, c'est que les carrés d'entiers forment une partie stricte de l'ensemble des entiers. Le paradoxe est résolu de la manière la plus simple qui soit : l'intuition que l'on a sur les ensembles finis, qui veut qu'une sous-partie stricte d'un ensemble contienne moins d'éléments n'est tout simplement plus vraie pour les ensembles infinis.

Remarque

La notation $|E| = |F|$ semble suggérer l'égalité entre deux objets $|E|$ et $|F|$, que l'on nomme respectivement *cardinalité* de E et *cardinalité* de F . Il est possible de donner une définition précise de $|E|$, en tant qu'objet mathématique. Nous nous contenterons pour le moment de définir la cardinalité comme la notion informelle de taille d'un ensemble, et si l'objet $|E|$ n'est pas clairement défini, l'énoncé $|E| = |F|$ l'est, et suffit à notre traitement de l'infini.

Notons que Galilée utilise son idée pour expliquer justement pourquoi il n'y a aucun sens à comparer la taille des ensembles infinis. C'est là qu'intervient tout le génie de Cantor, qui découvrira que contrairement à l'intuition de Galilée, il est possible de donner une notion formelle de taille aux ensembles infinis : il existe des infinis « plus gros » que d'autres.

Intuitivement, un ensemble F est au moins aussi gros qu'un ensemble E si l'on peut faire correspondre aux éléments de E des éléments distincts de F , autrement dit si l'on peut associer à chaque élément de E un « représentant » qui lui est propre dans F .

Définition 1.3. Un ensemble E est *subpotent* à un ensemble F s'il existe une injection de E dans F . On écrit alors $|E| \leq |F|$. Si E n'est pas subpotent à F , on écrit $|E| \not\leq |F|$. ◇

Il est facile de vérifier que cette relation est *transitive*, c'est-à-dire, que si $|E| \leq |F|$ et $|F| \leq |G|$, alors $|E| \leq |G|$. En effet, si les fonctions $f : E \rightarrow F$ et $g : F \rightarrow G$ sont deux injections, alors leur composition $g \circ f : E \rightarrow G$ est une injection témoignant de la relation $|E| \leq |G|$. Il est cependant beaucoup moins clair que si $|E| \leq |F|$ et $|F| \leq |E|$, alors $|E| = |F|$. En déroulant les définitions, la question revient à savoir si, lorsqu'il existe une injection de E dans F et une autre de F dans E , il existe une bijection entre E et F . Voyons tout de suite que c'est bien le cas : il s'agit du théorème de Cantor–Bernstein.

2. Théorème de Cantor–Bernstein

Le cœur de la preuve du théorème de Cantor–Bernstein tient dans le lemme suivant.

Lemme 2.1. Si $B \subseteq A$ sont des ensembles, et $f : A \rightarrow B$ est une fonction injective, alors il existe une bijection $h : A \rightarrow B$. ★

PREUVE. Le lecteur pourra s'aider de la figure 2.2.

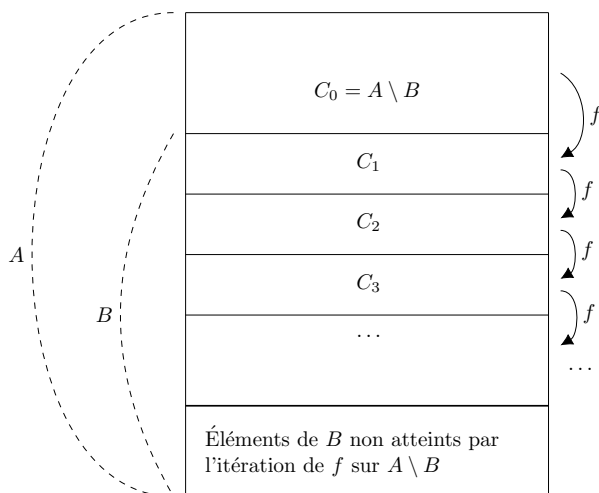


FIGURE 2.2 – Illustration de la preuve du lemme 2.1

Soit $C_0, C_1, C_2, \dots$ la suite définie par récurrence comme suit :

$$C_0 = A \setminus B \text{ et } C_{n+1} = f(C_n).$$

On note $C = \bigcup_n C_n$. Un simple raisonnement par récurrence sur n permet de montrer, à l'aide de l'injectivité de f , que les ensembles C_n pour $n \in \mathbb{N}$ sont deux à deux disjoints, bien que cela ne soit pas nécessaire dans la preuve. Soit la fonction $h : A \rightarrow B$ définie par :

$$h(x) = \begin{cases} f(x) & \text{si } x \in C \\ x & \text{si } x \notin C. \end{cases}$$

Montrons que h est injective. La fonction f étant injective, la fonction h restreinte à C est injective. La fonction h restreinte à $A \setminus C$ est la fonction identité, et est donc également injective. Enfin, l'image de $C = \bigcup_n C_n$ par h est

$$\bigcup_n f(C_n) = \bigcup_n C_{n+1} \subseteq C,$$

et l'image de $A \setminus C$ par h est $A \setminus C$. La fonction h est la réunion de deux fonctions injectives possédant des images disjointes, et est donc injective.

Montrons enfin que h est surjective. Soit $y \in B$; si $y \notin C$, alors $h(y) = y$ et donc y a un prédécesseur par h . Si $y \in C$, comme $y \notin C_0$, il existe un $n \in \mathbb{N}$ tel que $y \in C_{n+1}$. Par définition de $C_{n+1} = f(C_n)$, il existe un $x \in C_n$ tel que $h(x) = f(x) = y$. ■

Nous pouvons à présent montrer le théorème annoncé.

Théorème 2.3 (Cantor-Bernstein)

Si A et B sont des ensembles, et $f : A \rightarrow B$ et $g : B \rightarrow A$ sont des fonctions injectives, alors il existe une bijection entre A et B .

PREUVE. Soit $A' = g(B)$. L'application $g \circ f$ est une injection de A dans A' . Par le lemme 2.1, il existe donc une bijection $h : A \rightarrow A'$. La fonction g étant une bijection entre B et A' , la fonction $g^{-1} \circ h : A \rightarrow B$ est une bijection. ■

Corollaire 2.4

Deux ensembles mutuellement subpotents sont équipotents.

La question de savoir s'il existe des infinis de tailles distinctes, et en particulier avec l'un d'eux strictement plus gros que l'autre, revient donc à savoir si l'on peut trouver deux ensembles A, B pour lesquels $|A| \leq |B|$ mais $|B| \not\leq |A|$. Nous allons voir que c'est bien le cas.

3. Ensembles dénombrables

L'ensemble infini par excellence est bien entendu $\mathbb{N}$, l'ensemble des nombres entiers. On utilise dans ce cas un vocabulaire spécifique.

Définition 3.1. Un ensemble infini A est dit *dénombrable* s'il existe une bijection $f : \mathbb{N} \rightarrow A$, autrement dit si A et $\mathbb{N}$ sont équipotents. Un ensemble infini A est dit *indénombrable* s'il n'existe pas de bijection $f : \mathbb{N} \rightarrow A$. ◇

Remarque

Certains auteurs utilisent le terme « dénombrable » pour désigner un ensemble subpotent à $\mathbb{N}$, autrement dit un ensemble fini, ou en bijection avec $\mathbb{N}$. Ils parlent alors d'ensemble *infini dénombrable* pour désigner les ensembles dénombrables tels que nous les avons introduits.

Intuitivement, l'infini des entiers naturels est le plus petit infini, au sens où il est subpotent à tout ensemble infini.

Proposition 3.2. L'ensemble $\mathbb{N}$ est subpotent à tout ensemble infini. ★

PREUVE. Soit A un ensemble infini. Nous allons définir une fonction injective $f : \mathbb{N} \rightarrow A$ par récurrence sur $\mathbb{N}$. Soit $f(0)$ un élément quelconque de A . Supposons que les valeurs $f(0), \dots, f(n)$ soient définies. En particulier, $B = \{f(0), \dots, f(n)\} \subseteq A$ est un ensemble fini tandis que A est infini.

Il existe donc nécessairement un élément dans $A \setminus B$. Soit $f(n+1)$ cet élément. Par construction, f est injective. ■

Notons qu'en toute généralité, la preuve précédente utilise *l'axiome du choix*, dont nous reparlerons dans la section 9-4, et qui est nécessaire afin de choisir à chaque étape un élément de $A \setminus B$. Toutefois, dans la plupart des cas (comme dans le corollaire suivant), cet axiome n'est pas absolument nécessaire.

Corollaire 3.3

Tout sous-ensemble infini d'un ensemble dénombrable est dénombrable.

PREUVE. Soit A un ensemble dénombrable, et soit $B \subseteq A$ un sous-ensemble infini. L'ensemble A étant équipotent à $\mathbb{N}$, il est donc subpotent à $\mathbb{N}$. Comme $B \subseteq A$, il est subpotent à A , donc à $\mathbb{N}$. En outre, par la proposition 3.2, $\mathbb{N}$ est subpotent à B . Par le théorème de Cantor-Bernstein (théorème 2.3), B et $\mathbb{N}$ sont équipotents. ■

Exercice 3.4. (★) Soit A un ensemble dénombrable, et soit une bijection $f : \mathbb{N} \rightarrow A$. Soit enfin $B \subseteq A$ un sous-ensemble infini. Construire directement une bijection de $\mathbb{N}$ vers B qui ne s'appuie pas sur l'axiome du choix. On entend par là que la définition de la fonction doit reposer sur un algorithme explicite, et non sur une procédure abstraite permettant de choisir un élément dans un ensemble non vide, sans que l'on ne sache de quel élément il s'agit. ◇

Introduisons à présent une bijection que nous utiliserons régulièrement dans ce livre, qui est celle couramment utilisée pour attester de la dénombrabilité du produit $\mathbb{N} \times \mathbb{N}$.

Proposition 3.5. L'ensemble $\mathbb{N} \times \mathbb{N}$ est dénombrable. ★

PREUVE. Soit $\alpha : \mathbb{N} \rightarrow \mathbb{N}^2$ la fonction telle que les valeurs

$$\alpha(0) = (0, 0), \quad \alpha(1) = (1, 0), \quad \alpha(2) = (0, 1), \dots$$

énumèrent les couples d'entiers par diagonales successives, comme dans la figure 3.6.

La fonction est injective par construction, et toute paire apparaîtra à une étape de l'énumération. Ainsi, α est une bijection de $\mathbb{N}$ vers $\mathbb{N} \times \mathbb{N}$ témoignant de l'équipotence entre les deux ensembles. ■

Il est possible de donner une définition analytique de la fonction réciproque de α définie dans la proposition précédente. Il s'agira donc d'une bijection

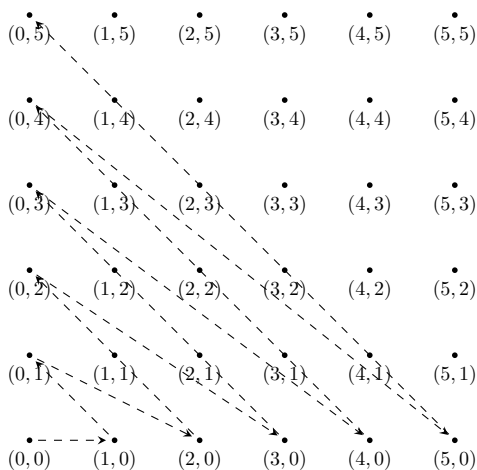


FIGURE 3.6 – Illustration de la preuve de la proposition 3.5

de $\mathbb{N}^2$ vers $\mathbb{N}$, que nous appellerons α_2 et que le lecteur pourra découvrir à travers l'exercice ci-après.

Exercice 3.7. (★) Soit $\alpha_2 : \mathbb{N}^2 \rightarrow \mathbb{N}$ définie par :

$$\begin{aligned}\alpha_2(x, y) &= y + \sum_{i=0}^{x+y} i \\ &= y + \frac{(x+y+1)(x+y)}{2}.\end{aligned}$$

1. Montrer que α_2 est bijective.
2. Montrer que $\alpha_2(a, b) \geq a$ et $\alpha_2(a, b) \geq b$.

◇

La bijection α_2 de l'exercice précédent sera très souvent utilisée dans les développements de ce livre, via la notation suivante.

Notation

On note $\langle n, m \rangle$ l'entier sur lequel est envoyé le couple (n, m) via la bijection α_2 .

Notons que si l'on a une bijection $\alpha_2 : \mathbb{N}^2 \rightarrow \mathbb{N}$, on peut définir une bijection $\alpha_3 : \mathbb{N}^3 \rightarrow \mathbb{N}$ en prenant simplement $\alpha_3(x, y, z) = \alpha_2(x, \alpha_2(y, z))$. On peut continuer ainsi pour définir des bijections de $\mathbb{N}^n$ vers $\mathbb{N}$ pour tout $n \in \mathbb{N}^*$, ce qui conduit à la notation suivante.

Notation

On note $\langle x_1, \dots, x_k \rangle$ l'entier sur lequel est envoyé le k -uplet $(x_1, \dots, x_k)$ via la bijection de $\mathbb{N}^k$ vers $\mathbb{N}$ décrite ci-dessus.

Profitions de nos bijections fraîchement introduites pour en déduire le corollaire suivant.

Corollaire 3.8

Le produit cartésien de deux ensembles dénombrables est dénombrable.

PREUVE. Soient A et B des ensembles dénombrables et soient $f : A \rightarrow \mathbb{N}$ et $g : B \rightarrow \mathbb{N}$ des bijections qui en témoignent. La fonction $h : A \times B \rightarrow \mathbb{N}$ définie par $h(x, y) = \langle f(x), g(y) \rangle$ est une bijection. ■

Terminons cette section par trois exercices, permettant de manipuler les concepts vus jusqu'ici. Nous attirons en particulier, l'attention sur le premier d'entre eux que l'on utilisera régulièrement dans les développements à venir.

Exercice 3.9. (★) Montrer que $\mathbb{Z}$ est un ensemble dénombrable. En déduire que $\mathbb{Z} \times \mathbb{Z}$ est un ensemble dénombrable et enfin que l'ensemble $\mathbb{Q}$ des nombres rationnels — qui s'écrivent sous la forme p/q pour $p, q \in \mathbb{Z}$ avec $q \neq 0$ — est lui aussi dénombrable. ◇

Exercice 3.10. (★) Soit $f : \mathbb{N} \rightarrow A$ une fonction surjective vers un ensemble A infini. Montrer que A est dénombrable. ◇

Exercice 3.11. (★) Soit $(B_n)_{n \in \mathbb{N}}$ une suite d'ensembles dénombrables avec des bijections respectives f_n avec $\mathbb{N}$. Montrer que $B = \bigcup_n B_n$ est un ensemble dénombrable.

Attention : si l'on ne dispose pas des bijections $(f_n)_{n \in \mathbb{N}}$, on peut toujours montrer que B est dénombrable, mais il faut utiliser l'*axiome du choix*, afin de choisir uniformément pour chaque B_n une de ses bijections avec $\mathbb{N}$. Nous en reparlerons dans la section 9-4. ◇

4. Argument diagonal de Cantor

Attelons-nous à présent au théorème annoncé au début de ce chapitre : il existe des infinis plus gros que d'autres, et en particulier un infini plus gros que celui des entiers. Le théorème suivant utilise le fameux argument diagonal de Cantor, qui sera repris à de très nombreuses occasions et sous des formes variées tout au long de ce livre.

R	$=$	0,	$9 - x_{00}$	$9 - x_{11}$	$9 - x_{22}$	$9 - x_{33}$	$9 - x_{44}$	$9 - x_{55}$	$9 - x_{66}$
$f(0)$	$=$	$N_0,$	x_{00}	x_{01}	x_{02}	x_{03}	x_{04}	x_{05}	x_{06}
$f(1)$	$=$	$N_1,$	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
$f(2)$	$=$	$N_2,$	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{16}
$f(3)$	$=$	$N_3,$	x_{30}	x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}
$f(4)$	$=$	$N_4,$	x_{40}	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}
$f(5)$	$=$	$N_5,$	x_{50}	x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}
$f(6)$	$=$	$N_6,$	x_{60}	x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}
...									

FIGURE 4.2 – *Illustration de l'argument diagonal de Cantor. On construit notre réel R à partir de la diagonale du tableau, la décimale $9 - x_{ii}$ étant nécessairement différente de x_{ii} . Notons que le nombre R illustré ici n'est pas exactement celui décrit dans la preuve, mais le résultat est le même : l'élément R n'est pas dans l'image de f .*

Théorème 4.1 (Cantor)

L'ensemble des nombres réels est indénombrable.

PREUVE. Le lecteur peut s'aider de la figure 4.2, qui illustre l'argument de la preuve. On raisonne par l'absurde. Supposons au contraire qu'il existe une bijection $f : \mathbb{N} \rightarrow \mathbb{R}$. Nous allons construire un nombre réel $R \in \mathbb{R}$ qui n'est pas dans l'image de f . On définit simplement R de la manière suivante : la partie entière de R est 0, et pour tout n , si la n -ième décimale de $f(n)$ est différente de 0, alors la n -ième décimale de R est égale à 0. À l'inverse, si la n -ième décimale de $f(n)$ est égale à 0, alors la n -ième décimale de R est égale à 1.

Il est clair que pour tout entier n , notre nombre réel R ne peut être égal à $f(n)$, car la n -ième décimale de R est différente de la n -ième décimale de $f(n)$. ■

Que nous dit le théorème précédent ? Qu'il y a « plus » de nombres réels que de nombres entiers. Ces ensembles de nombres sont tous les deux infinis, mais l'infini des réels est « plus grand » que celui des entiers. En effet, quand on essaye de faire correspondre un nombre entier à chaque nombre réel, on s'aperçoit qu'il y a toujours des réels « en trop », qui ne correspondent à aucun entier. On a donc $|\mathbb{R}| \not\leq |\mathbb{N}|$. Notons que l'on a en revanche $|\mathbb{N}| \leq |\mathbb{R}|$, via l'injection identité. On pourra dans ce cas écrire $|\mathbb{N}| < |\mathbb{R}|$, signifiant qu'il y a une injection de $\mathbb{N}$ dans $\mathbb{R}$, mais pas d'injection de $\mathbb{R}$ dans $\mathbb{N}$.

Y a-t-il un infini plus grand que celui des réels ? Cantor a également répondu à cette question, par un argument similaire : pour tout ensemble A , il existe un ensemble B tel que $|A| < |B|$.

Théorème 4.3 (Cantor)

Pour tout ensemble A , l'ensemble $\mathcal{P}(A)$ constitué des sous-ensembles de A est tel que $|A| < |\mathcal{P}(A)|$.

PREUVE. L'ensemble A est subpotent à $\mathcal{P}(A)$, en considérant l'injection qui à tout élément $x \in A$ associe le singleton $\{x\}$. Ainsi, $|A| \leq |\mathcal{P}(A)|$. Supposons à présent par l'absurde que $|\mathcal{P}(A)| = |A|$, c'est-à-dire supposons qu'il existe une bijection $f : A \rightarrow \mathcal{P}(A)$. Considérons l'ensemble

$$B = \{x \in A : x \notin f(x)\},$$

et soit y tel que $f(y) = B$. Alors, $y \in B$ ssi $y \notin f(y)$, autrement dit ssi $y \notin B$, ce qui est une contradiction. ■

On peut légitimement se demander si la cardinalité des ensembles est toujours comparable : étant donné deux ensembles A, B , existe-t-il toujours une injection de l'un dans l'autre ? La réponse à cette question dépend là encore de *l'axiome du choix*. Nous en reparlerons dans la section 9-4 ainsi que dans la section 27-4.1.

5. Réels non calculables

Le théorème de Cantor va nous fournir notre premier argument de l'existence de nombres réels incalculables. Nous utilisons pour la proposition et le corollaire suivants les notions pour le moment informelles de « programme informatique » et de « réel calculable ». Elles seront toutes les deux précisément définies plus loin dans ce livre ; considérons pour l'instant qu'un réel est calculable s'il existe un programme informatique qui énumère dans l'ordre la liste infinie de ses décimales.

Proposition 5.1. L'ensemble des programmes informatiques est dénombrable. ★

PREUVE. Un programme informatique (écrit dans quelque langage de programmation que ce soit) se présente sous la forme d'une suite finie de caractères, où chaque caractère appartient à un alphabet fini (disons l'ensemble des caractères de la table ascii). Montrons qu'il existe une bijection de $\mathbb{N}$ vers l'ensemble des suites finies de caractères ascii. Pour cela, nous définissons une liste infinie de toutes les suites finies de caractères ascii : on liste d'abord toutes les suites comportant exactement un caractère ascii,

puis toutes les suites comportant deux caractères ascii, puis toutes celles en comportant trois, etc. Il est clair que n'importe quelle suite finie de caractères ascii apparaît quelque part dans notre liste infinie.

Il suffit à présent de retirer de notre liste les suites finies ne correspondant pas à un programme valide. On définit alors l'élément $f(n)$ comme étant le n -ième élément de la liste résultant de cette opération. Il est clair que f est une bijection. En particulier, l'ensemble des programmes informatiques est dénombrable. ■

Corollaire 5.2

Il existe des nombres réels qui ne sont pas calculables.

PREUVE. Soit P l'ensemble des programmes informatiques. Supposons que tout réel est calculé par un élément de P . Soit p_1 le premier programme informatique calculant un réel. Ici « premier » veut dire premier selon l'ordre obtenu via la bijection entre les programmes informatiques et $\mathbb{N}$. Supposons que $p_1, \dots, p_n$ soient définis et calculent tous un réel différent. Soit p_{n+1} le premier programme informatique calculant un réel différent de ceux calculés par $p_1, \dots, p_n$. On définit par récurrence de cette manière la suite $(p_n)_{n \in \mathbb{N}}$. On obtient alors une bijection entre $\mathbb{R}$ et un sous-ensemble infini de P . Comme P est dénombrable, ce sous-ensemble infini l'est également, ce qui donne une bijection entre $\mathbb{N}$ et $\mathbb{R}$, et dès lors une contradiction. ■

La preuve du corollaire 5.2 est non constructive : on montre l'existence de nombres incalculables sans en donner d'exemple précis. Ce sera évidemment fait à maintes reprises dans la suite de cet ouvrage, à travers une étude détaillée de différents types de nombres incalculables. Avant de nous y atteler, mentionnons deux ou trois notions importantes concernant l'étude de l'infini menée par Cantor suite à sa découverte.

6. Espace de Cantor

Cantor a montré qu'il n'y avait pas de plus grand infini. Parmi tous les infinis possibles, le plus petit est « le dénombrable ». Un autre infini nous intéresse également, celui des nombres réels :

Définition 6.1. Un ensemble A est dit *avoir la puissance du continu* si $|A| = |\mathbb{R}|$. ◇

La puissance du continu caractérise donc l'infini des réels. Cantor conjectura qu'il n'y avait pas d'infini strictement compris entre $|\mathbb{N}|$ et $|\mathbb{R}|$, mais

sans réussir à le démontrer. Sa conjecture connue sous le nom *d'hypothèse du continu* sera considérée pendant près d'un siècle comme une des plus importantes questions mathématiques. Gödel montrera en 1938 [74] qu'il n'est pas possible de démontrer que l'hypothèse du continu est fausse, et Cohen mettra un point final à la question en 1963 [37] en montrant qu'il n'est pas possible non plus de montrer que l'hypothèse du continu est vraie : il s'agit d'une question indépendante du reste des mathématiques, que l'on peut considérer vraie ou fausse sans introduire de contradiction. Nous en discuterons plus en détail dans la section 9-4.

Parmi les ensembles ayant la puissance du continu, on prêtera une attention particulière à l'ensemble des suites infinies de 0 et de 1, qui constituera l'essentiel de notre terrain de jeu tout au long de cet ouvrage. Par suite infinie de 0 et de 1, on entend suite indexée par les entiers, c'est-à-dire des suites de la forme $x_0x_1x_2x_3\dots$ où chaque $x_i \in \{0, 1\}$.

Définition 6.2. On appelle *espace de Cantor* l'ensemble des suites infinies de 0 et de 1, on le note $2^{\mathbb{N}}$, et ses éléments seront dénotés via des lettres majuscules, en général A, B, C, X, Y, Z . ◇

L'espace de Cantor a la puissance du continu.

Proposition 6.3. On a : $|2^{\mathbb{N}}| = |[0, 1]| = |\mathbb{R}|$. ★

PREUVE. Montrons d'abord $|[0, 1]| = |\mathbb{R}|$. La fonction identité est une injection de $[0, 1]$ dans $\mathbb{R}$. On vérifie sans peine que la fonction

$$f(x) = \begin{cases} 1/2 + 1/(x+2) & \text{si } x \geq 0 \\ 1/2 - 1/(|x|+2) & \text{si } x < 0 \end{cases}$$

est une injection de $\mathbb{R}$ dans $[0, 1]$. Par le théorème de Cantor-Bernstein (voir le théorème 2.3), on a donc $|[0, 1]| = |\mathbb{R}|$.

Montrons à présent $|2^{\mathbb{N}}| = |[0, 1]|$. Pour définir une injection de $2^{\mathbb{N}}$ dans $[0, 1]$, il convient d'être prudent : certains nombres réels ont deux développements binaires possibles, ainsi $1.00000\dots = 0.11111\dots$ — où $0.11111\dots$ est le nombre décimal $0.99999\dots$ écrit en binaire. Il en va de même pour tout nombre réel dont le développement binaire se termine par une infinité de 0 consécutifs : celui-ci a alors un développement binaire équivalent qui se termine par une infinité de 1 consécutifs. On définira donc l'injection $f : 2^{\mathbb{N}} \rightarrow \mathbb{R}$ qui à X associe le nombre réel de $[0, 1]$ dont le développement *ternaire*, c'est-à-dire en base 3, est constitué des bits de X . L'usage de la représentation ternaire permet de contourner ces problèmes d'égalités et de rendre ainsi la fonction f injective. L'injection $g : [0, 1] \rightarrow 2^{\mathbb{N}}$ est définie en associant à tout réel $r \in [0, 1]$ son développement binaire R .

Lorsqu'il existe plusieurs représentations du même nombre réel, on choisira par convention celle se terminant par une infinité de 0. Par le théorème de Cantor-Bernstein, on a donc $|2^{\mathbb{N}}| = |[0, 1]|$. ■

Notation

Étant donné $X \in 2^{\mathbb{N}}$ et $n \in \mathbb{N}$, on note $X(n)$ le n -ième élément de la suite X que l'on appellera aussi le n -ième *bit* de X . Ainsi, si la suite X s'écrit $x_0x_1x_2\dots$, alors $X(0) = x_0$, $X(1) = x_1$, $X(2) = x_2$, $\dots$

Le fait que certains réels aient deux représentations binaires possibles font de $|2^{\mathbb{N}}|$ et $[0, 1]$ des espaces topologiquement différents. Par exemple, pour tous $x, y \in \mathbb{R}$ avec $x < y$, il existe un réel z strictement entre les deux. En revanche, si l'on munit $2^{\mathbb{N}}$ de l'ordre lexicographique $<_{lex}$ défini par

$$X <_{lex} Y \text{ si } X(n) < Y(n),$$

où n est le plus petit entier tel que $X(n) \neq Y(n)$, alors les suites infinies

$$X = 0111111\dots \text{ et } Y = 100000\dots$$

n'ont aucun élément strictement entre elles pour cet ordre.

Cette différence est anecdotique du point de vue de la complexité calculatoire des éléments, et il nous arrivera parfois de parler de réel ou de suites binaires infinies indistinctement. La différence a toutefois son importance pour le développement de la calculabilité dans d'autres espaces topologiques que $2^{\mathbb{N}}$. Nous en discuterons brièvement dans la section 22-4.2.

Voyons à présent qu'il existe en revanche une bijection très naturelle entre l'ensemble $\mathcal{P}(\mathbb{N})$ — l'ensemble des parties de $\mathbb{N}$ — et l'espace de Cantor. Ainsi, $\mathcal{P}(\mathbb{N})$ a la puissance du continu.

Proposition 6.4. On a : $|2^{\mathbb{N}}| = |\mathcal{P}(\mathbb{N})|$. ★

PREUVE. Soit $f : 2^{\mathbb{N}} \rightarrow \mathcal{P}(\mathbb{N})$ la fonction qui à $X \in 2^{\mathbb{N}}$ associe l'ensemble $Y = \{n \in \mathbb{N} : X(n) = 1\}$. La fonction f est clairement une bijection. ■

La bijection entre l'ensemble $\mathcal{P}(\mathbb{N})$ des parties de $\mathbb{N}$ et l'espace de Cantor $2^{\mathbb{N}}$ est tellement élémentaire que l'on peut considérer $\mathcal{P}(\mathbb{N})$ et $2^{\mathbb{N}}$ comme deux représentations du même concept mathématique. Dans la suite, nous parlerons indistinctement de sous-ensemble de $\mathbb{N}$ ou de suite infinie de 0 et de 1, et utiliserons la même notation $2^{\mathbb{N}}$ pour désigner l'ensemble de ces éléments. Ainsi, étant donné $X \in 2^{\mathbb{N}}$ vu comme une suite, on aura $X(n) = 1$ ssi n appartient à X vu comme un ensemble.

Digression

Le nom « espace de Cantor » vient sans doute du concept éponyme *d'ensemble triadique de Cantor*. On définit $A_0 = [0, 1]$, puis A_1 est A_0 moins son tiers central :

$$A_1 = [0, 1/3] \cup [2/3, 1].$$

Puis A_2 est A_1 moins le tiers central de chacun de ses intervalles :

$$A_2 = [0, 1/9] \cup [2/9, 1/3] \cup [2/3, 7/9] \cup [8/9, 1].$$

De manière générale, pour passer de A_n à A_{n+1} , on enlève le tiers central de chacun des intervalles de A_n . L'ensemble triadique de Cantor est le résultat à la limite, de l'application de cette opération, c'est-à-dire l'ensemble $\bigcap_{n \in \mathbb{N}} A_n$.

L'espace de Cantor $2^{\mathbb{N}}$ défini plus haut, est topologiquement équivalent à l'ensemble triadique de Cantor. En particulier, chaque point de $\bigcap_{n \in \mathbb{N}} A_n$ peut être décrit comme une suite de 0 et de 1 de la manière suivante : le n -ième bit d'un point correspond à déterminer s'il est à droite ou à gauche du n -ième tiers que l'on enlève de l'intervalle courant. On peut également voir l'espace de Cantor comme l'ensemble des réels de $[0, 1]$ dont la représentation ternaire évite le chiffre 1.

Première partie

Calculabilité classique

Chapitre 3

Fondements de la calculabilité

Notre objectif est de mener une étude mathématique de la calculabilité. Pour ce faire, il est d'usage de définir mathématiquement ce que l'on entend par *fonction calculable*. Cette quête d'une définition formelle capturant ce concept épistémologique a constitué la genèse de la calculabilité, et abouti à ce que l'on appelle de nos jours la *thèse de Church-Turing*. Cette thèse énonce que tout processus calculable peut être exécuté avec une *machine de Turing*, un modèle de calcul imaginé par Alan Turing en 1936 et qui peut être considéré comme un précurseur des ordinateurs modernes. D'autres approches que celle des machines de Turing permettent de capturer la notion de fonction calculable, parmi lesquelles nous citerons les fonctions générales récursives et le λ -calcul. Ces différents modèles seront présentés plus en détail dans l'interlude sur la thèse de Church-Turing (voir le chapitre 6), et nous adopterons pour ce chapitre une approche moins formelle.

1. Fonctions calculables

En calculabilité, le formalisme des machines de Turing tend à servir de modèle de référence, non seulement pour des raisons historiques — Turing fut le premier à avoir su convaincre la communauté que son modèle capturerait tous les processus calculables — mais également parce que ce modèle met en relief la notion d'étape atomique de calcul, ce qui ouvre la porte à la théorie de la complexité. Il est de coutume de débiter l'étude de la calculabilité par celle de ses modèles de calcul, et de prouver leur équivalence, afin de se convaincre de la robustesse de la notion de fonction calculable

et du bien-fondé des définitions. Il s'agit d'un développement assez long et fastidieux. Aussi est-il facile d'être rebuté par cette étape à l'issue de laquelle on croit trop facilement — à tort — que la calculabilité se résume à de complexes et rébarbatives techniques de codage.

Nous avons donc fait le choix de rompre avec la tradition, et reporter la définition et l'étude mathématique des modèles de calcul au chapitre 6, afin de faciliter le premier contact avec la calculabilité, et d'accéder plus directement à ses concepts fondamentaux. L'avènement des ordinateurs et la démocratisation de l'enseignement de la programmation ont ancré durablement la notion d'algorithme dans la culture scientifique. Nous adopterons donc la définition informelle suivante.

Définition 1.1. Une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ est *calculable* si elle peut être définie par un algorithme, ou autrement dit programmée dans un langage de programmation moderne. 

Il s'ensuit de notre choix pédagogique que les preuves de nos premiers théorèmes feront largement appel à l'intuition des propriétés que l'on attend d'une fonction calculable. Il s'agit cependant de théorèmes à part entière, dans le sens où il est possible de les prouver à partir des définitions formelles du chapitre 6.

Entiers en informatique

Nous nous intéressons essentiellement aux fonctions de $\mathbb{N}$ dans $\mathbb{N}$. Dans la plupart des langages de programmation standard, les entiers sont bornés. Pour notre définition théorique, il est important de prendre en compte *tous les entiers*.

Algorithmes. Afin de s'accorder sur ce que l'on entend par un langage de programmation moderne ou algorithme, listons-en les principaux aspects, qui seront repris formellement dans la section 6-3 pour notre définition des programmes structurés sur le modèle des machines à registres.

- (1) Le langage doit pouvoir manipuler les entiers, à l'aide de constantes, de variables entières, et des opérations arithmétiques usuelles, à savoir l'addition, la soustraction, la multiplication, et la division entière. Le lecteur pourra y ajouter d'autres types, comme les chaînes de caractères ou les nombres à virgule flottante, sans que cela ne change la puissance de calcul.
- (2) Le langage doit pouvoir manipuler des expressions booléennes, et effectuer des opérations de comparaison sur les entiers.
- (3) Le langage doit contenir les structures de contrôle usuelles, à savoir des instructions conditionnelles de type « `if ... then ... else ...` » et

des boucles « **for** » et « **while** », qui se répètent tant qu'une certaine condition est vraie.

- (4) Nous supposons que la mémoire de la machine n'est pas bornée, et qu'elle peut en utiliser autant que nécessaire (notamment pour lire son entrée, qui peut être un entier arbitrairement grand).

Types de données

Le point (1) insiste sur le fait que l'ajout d'autres types de données que le type entier est superflu. C'est vrai à condition bien sûr de considérer que nos entiers peuvent être arbitrairement grands (pour encoder par exemple de grandes chaînes de caractères), ce qui sera par convention toujours le cas. Le lecteur pourra trouver un exemple d'encodage de tableaux par des entiers dans la proposition 6-3.26.

Mémoire non bornée

On peut être surpris par le point (4) ci-dessus, qui autorise une mémoire non bornée. Insistons sur le fait que l'on ne s'autorise pas une mémoire infinie pour autant : un calcul qui se termine n'a effectué qu'un nombre fini d'opérations et n'a donc pu utiliser qu'une quantité finie de mémoire. Simplement, on ne s'occupe pas en calculabilité de la complexité en espace des algorithmes.

Exemples. Avant de commencer l'étude formelle des fonctions calculables et de leurs propriétés, listons quelques exemples de fonctions calculables, afin de commencer à nous former une intuition.

- (1) Les opérations arithmétiques usuelles sont calculables. En particulier, l'addition, la multiplication, la soustraction et la division entière sont calculables.
- (2) La fonction qui à n associe le n -ième nombre premier est calculable, de même que la décomposition d'un nombre en ses facteurs premiers.
- (3) La fonction qui, prenant en paramètre une liste de positions de villes, renvoie un des chemins les plus courts passant par toutes ces villes une seule fois, est calculable¹.

Inversement, il existe comme nous allons le voir de nombreuses fonctions non calculables. Cependant, tandis qu'il suffit de donner un algorithme pour montrer qu'une fonction est calculable, montrer qu'une fonction n'est pas calculable demande souvent un raisonnement plus élaboré, car il ne suffit pas que la fonction n'ait pas d'algorithme connu ; il faut montrer qu'il est théoriquement impossible de la programmer. Chacun des énoncés suivants est donc un théorème à part entière.

1. Il s'agit du problème bien connu du *voyageur de commerce*.

- (1) La fonction qui prend en entrée un programme informatique (codé par un entier ou une chaîne de caractères), et décide si son exécution va un jour s'arrêter, n'est pas calculable (voir le théorème 7.8).
- (2) La fonction qui prend en entrée une formule dans le langage de l'arithmétique (par exemple, $\ll \forall x \forall y \forall z x^3 + y^3 \neq z^3 \gg$), et renvoie 1 s'il existe une démonstration mathématique de cette formule dans le système axiomatique de l'arithmétique, et 0 sinon, n'est pas calculable (voir le théorème 9-3.9).
- (3) La fonction qui prend en entrée une équation diophantienne, i.e. une équation à coefficients entiers — par exemple $3x^2 + 2xy + 4y^2 = 0$ — et décide si cette équation admet une solution entière, n'est pas calculable (voir le théorème 12-1.2).

Fonctions partielles. Comme expliqué dans l'encadrement « Entiers en informatique » ci-dessus, on se restreindra en général aux programmes prenant en paramètre un entier, et retournant un autre entier si le calcul se termine. Il est essentiel de noter que les fonctions engendrées par les programmes informatiques sont *partielles*, au sens où le calcul peut ne jamais s'arrêter sur certaines de leurs entrées. Cette partialité est due aux structures de contrôle de type « **while** » dont la condition de terminaison peut n'être jamais satisfaite, comme le montre l'exemple suivant, qui calcule — de la pire manière possible — la racine carrée d'un entier :

```
function Racine(n) {
  r = 0;
  while(r*r ≠ n) {
    r = r+1;
  }
  return r;
}
```

Si n n'est pas le carré d'un nombre entier, la boucle **while** va s'exécuter à l'infini, et le programme ne renverra jamais de valeur. Lorsque le programme ne s'arrête pas sur une entrée n , on considère que la fonction de $\mathbb{N}$ dans $\mathbb{N}$ qui lui est associée n'est pas définie sur n . Le domaine de définition de la fonction partielle associée à un programme est donc l'ensemble des entrées sur lesquelles il s'arrête. Les fonctions définies par des programmes sont appelées *fonctions partielles calculables*. Lorsque le programme s'arrête sur toutes ses entrées, la fonction engendrée est alors appelée *fonction totale calculable*, ou tout simplement *fonction calculable*.

Notation

Si f et g sont deux fonctions partielles, on notera $f(x) = g(x)$ pour signifier soit que f et g sont toutes les deux définies et renvoient la même valeur sur x , soit que ni f ni g ne sont définies sur x .

Codes. Dans notre étude mathématique, nous représenterons les programmes informatiques par des entiers, en supposant une fonction de codage fixée. Par *programmes informatiques*, il faut comprendre ici une suite finie de caractères — supposée avoir du sens dans un langage de programmation choisi au préalable. Concrètement, on dira qu'un entier e *code* pour un programme P si e est l'entier codé par la représentation binaire de la chaîne de caractères correspondant au programme P . En particulier, ce codage est injectif et intuitivement calculable. Notons que l'on pourrait imaginer beaucoup d'autres codages possibles. Nous en verrons un autre de manière détaillée dans la preuve du théorème 6-3.27. En attendant, celui-ci conviendra parfaitement.

Notation

On note $\Phi_e : \mathbb{N} \rightarrow \mathbb{N}$ la fonction partielle définie par le programme informatique de code e .

On supposera que tout entier e code pour un programme valide. En pratique, il existe dans tous les langages de programmation des chaînes de caractères correspondant à des programmes mal formés. Certains entiers e codent pour des suites de caractères inintelligibles. Si c'est le cas, on peut alors s'en rendre compte (cela correspond à avoir une erreur de syntaxe quand on essaye de compiler un programme) et considérer que e correspond alors à un programme qui ne s'arrête pas.

Notation

Soient $\Phi_e : \mathbb{N} \rightarrow \mathbb{N}$ une fonction partielle calculable et $x \in \mathbb{N}$ un entier. On écrira $\Phi_e(x) \downarrow$ si le programme codé par e s'arrête sur l'entrée x (nécessairement après un nombre fini d'étape de calcul) et renvoie un résultat entier. Dans le cas inverse, on écrira $\Phi_e(x) \uparrow$.

Le domaine de définition de la fonction partielle calculable $\Phi_e : \mathbb{N} \rightarrow \mathbb{N}$ est donc $\{x \in \mathbb{N} : \Phi_e(x) \downarrow\}$. Si $\Phi_e(x) \downarrow$, on écrira parfois $\Phi_e(x) \downarrow = y$ pour signifier que le programme de code e s'arrête sur l'entrée x et renvoie l'entier y . À l'inverse, on écrira parfois $\Phi_e(x) \uparrow \neq y$ pour signifier l'opposé, c'est-à-dire $\Phi_e(x) \uparrow \vee \Phi_e(x) \downarrow \neq y$.

Temps de calcul. Tout langage de programmation vient avec une notion d'étape d'exécution et de temps de calcul. Une étape d'exécution est une opération atomique du langage, indécomposable en sous-étapes. Elle correspond à une instruction élémentaire du langage. La notion d'étape d'exécution induit celle de temps de calcul, qui se définit par le nombre d'étapes d'exécution réalisées depuis le lancement du calcul. Lorsqu'un programme s'arrête sur une entrée, son temps de calcul est fini.

Notation

Soient $\Phi_e : \mathbb{N} \rightarrow \mathbb{N}$ une fonction partielle calculable et $x, t \in \mathbb{N}$ deux entiers. On écrira $\Phi_e(x)[t] \downarrow$ si le programme codé par e s'arrête *avant* t étapes de calcul. Dans le cas inverse, on écrira $\Phi_e(x)[t] \uparrow$.

Notons que $\Phi_e(x) \downarrow$ ssi il existe un temps de calcul t tel que $\Phi_e(x)[t] \downarrow$. En outre, si $\Phi_e(x)[t] \downarrow$, alors $\Phi_e(x)[s] \downarrow$ pour tout $s \geq t$.

Remarque

On sera amené à manipuler des fonctions calculables à plusieurs paramètres. Pour un entier $n \in \mathbb{N}^*$ fixé, on désignera comme ci-dessus par $\Phi_e : \mathbb{N}^n \rightarrow \mathbb{N}$ la fonction partielle à n paramètres entiers codée par e et l'on écrira $\Phi_e(x_1, \dots, x_n) \downarrow = y$ si le programme de code e s'arrête sur les entrées $x_1, \dots, x_n$ et renvoie l'entier y .

2. Ensembles calculables

L'adjectif « calculable » s'applique naturellement aux fonctions ; en effet, comme décrit plus haut, tout entier e code pour un programme auquel correspond une fonction partielle calculable. Le plus souvent cependant, quand on parle de fonctions calculables, on considérera qu'il s'agit de fonctions totales.

Définition 2.1. Soit $n \in \mathbb{N}^*$. Une fonction $f : \mathbb{N}^n \rightarrow \mathbb{N}$ est *calculable* s'il existe un code de programme e tel que, pour tout $(x_1, \dots, x_n)$, on a $\Phi_e(x_1, \dots, x_n) \downarrow = f(x_1, \dots, x_n)$. ◇

La calculabilité peut également être utilisée pour mesurer la complexité descriptive des objets mathématiques dénombrables, et en particulier celle des ensembles d'entiers. Intuitivement, un ensemble d'entiers $E \subseteq \mathbb{N}$ est calculable s'il peut être décrit par un processus calculable. Un ensemble étant totalement spécifié par ses éléments, il est calculable si sa fonction caractéristique est calculable.

Définition 2.2. Soit $n \in \mathbb{N}^*$. Un ensemble $A \subseteq \mathbb{N}^n$ est calculable s'il existe un code de programme e tel que, pour tous $x_1, \dots, x_n$, on a

- ▷ $\Phi_e(x_1, \dots, x_n) \downarrow = 1$ ssi $(x_1, \dots, x_n) \in A$.
- ▷ $\Phi_e(x_1, \dots, x_n) \downarrow = 0$ ssi $(x_1, \dots, x_n) \notin A$.

◇

On appellera parfois *prédicats* les sous-ensembles de $\mathbb{N}^n$, en utilisant alors la notation $A(x_1, \dots, x_n)$ pour signifier $(x_1, \dots, x_n) \in A$. Le terme de prédicat vient de la vision de $A \subseteq \mathbb{N}^n$ non comme un ensemble, mais comme une propriété des n -uplets d'entiers. Ainsi, $A(x_1, \dots, x_n)$ signifie que le n -uplet $(x_1, \dots, x_n)$ a la propriété A . À l'inverse, $\neg A(x_1, \dots, x_n)$ signifie que le n -uplet $(x_1, \dots, x_n)$ n'a pas la propriété A .

Exercice 2.3. Montrer que la bijection de couplage définie dans la proposition 2-3.5 et l'exercice 2-3.7, qui à $(a, b) \in \mathbb{N}^2$ associe $\langle a, b \rangle \in \mathbb{N}$, est calculable. Montrer que les fonctions inverses π_0, π_1 telles que $a = \pi_0(\langle a, b \rangle)$ et $b = \pi_1(\langle a, b \rangle)$ sont elles aussi calculables. ◇

Exercice 2.4. Soit $A \subseteq \mathbb{N}^2$ un prédicat calculable. Montrer que les ensembles :

- (1) $\{(x, y) \in \mathbb{N}^2 : \forall z < y (x, z) \in A\}$
- (2) $\{(x, y) \in \mathbb{N}^2 : \exists z < y (x, z) \in A\}$

sont eux aussi calculables. ◇

Ensemble récursif

Historiquement, les ensembles calculables étaient appelés *récursifs* en raison du paradigme de calcul des fonctions générales récursives dans lequel les définitions par récurrence jouent un rôle prépondérant (voir le chapitre 6). Peu à peu, avec l'amélioration de notre compréhension de la notion de calcul, la terminologie du domaine a évolué. On peut cependant régulièrement voir les termes de *théorie de la récursion* et d'*ensemble récursif* pour parler de calculabilité et d'ensemble calculable.

3. Programme universel

Le modèle de calcul imaginé par Turing en 1936 consiste à définir une machine pour chaque fonction calculable. Si l'on compare une machine de Turing à un dispositif physique, cela consiste, pour chaque tâche que l'on veut accomplir, à créer un robot exécutant cette tâche spécifique.

Nous allons maintenant énoncer un premier théorème fondamental de la calculabilité et prouvé par Turing dans son article original : l'existence

d'une machine de Turing *universelle*, capable de simuler toutes les autres machines de Turing. Ce travail est parfois considéré comme précurseur de l'architecture de von Neumann², dont l'un des aspects est le stockage des programmes dans la mémoire. En informatique moderne, ce théorème peut être vu comme énonçant l'existence d'un programme informatique *universel*, permettant de simuler tous les autres programmes informatiques.

Théorème 3.1

Soit $n \in \mathbb{N}^*$. Il existe un code e de programme informatique pour lequel $\Phi_e : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ est tel, que pour tous $x_1, \dots, x_n$, on a

- ▷ $\Phi_e(a, x_1, \dots, x_n) \uparrow$ ssi $\Phi_a(x_1, \dots, x_n) \uparrow$;
- ▷ $\Phi_e(a, x_1, \dots, x_n) \downarrow = y$ ssi $\Phi_a(x_1, \dots, x_n) \downarrow = y$.

En pratique, un programme universel existe très concrètement dans beaucoup de langages. Par exemple, en Java, il s'agit simplement de la machine virtuelle qui compile et exécute n'importe quel programme écrit en Java. Pour les langages n'utilisant pas de machine virtuelle, un programme universel sera simplement la donnée d'un interpréteur qui décompose le programme qu'il reçoit en une série d'instructions qu'il exécute pas à pas. Si un tel programme est bien entendu complexe à concevoir, il ne devrait faire aucun doute pour le programmeur qu'il s'agit de quelque chose de tout à fait possible : il s'agit simplement d'une machine virtuelle. Le lecteur pourra consulter la preuve du théorème 6-3.27, qui démontre l'existence d'un programme universel pour le modèle de calcul spécifique des machines à registre.

L'existence d'un tel programme nous permet de définir des fonctions qui font des manipulations sur des codes avant de les exécuter, ou exécutent dynamiquement des codes passés en paramètre. Par exemple,

$$f(x, y) = \Phi_{x+1}(y + 2)$$

est une définition valide, car elle est équivalente à $f(x, y) = \Phi_e(x + 1, y + 2)$, où Φ_e est le programme universel.

4. Théorème SMN

Le théorème SMN est notre premier théorème sur la manipulation des codes de programmes informatiques. Il ne s'agit pas de quelque chose de conceptuellement bien difficile. Soit $\Phi_e : \mathbb{N}^{m+n} \rightarrow \mathbb{N}$ la fonction de code e . Alors, étant donné $x_1, \dots, x_m$, on peut transformer de manière calculable

2. Qui est à peu de choses près toujours celle utilisée de nos jours.

notre code e en un code a tel que le calcul $\Phi_a(y_1, \dots, y_n)$ donne le même résultat que le calcul $\Phi_e(x_1, \dots, x_m, y_1, \dots, y_n)$. Le point important est que la transformation de e en a est calculable en fonction de e et de $x_1, \dots, x_m$.

Théorème 4.1 (Théorème SMN)

Pour tous $n, m \in \mathbb{N}^*$, il existe une fonction totale calculable

$$S_n^m : \mathbb{N}^{m+1} \rightarrow \mathbb{N},$$

telle que pour tous $e, x_1, \dots, x_m, y_1, \dots, y_n$,

$$\Phi_{S_n^m(e, x_1, \dots, x_m)}(y_1, \dots, y_n) = \Phi_e(x_1, \dots, x_m, y_1, \dots, y_n).$$

PREUVE. Décrivons la fonction S_n^m . Étant donné $e, x_1, \dots, x_m$, décoder e pour obtenir le programme P_e . Modifier calculatoirement le programme P_e pour lui ajouter des instructions assignant « en dur » les valeurs $x_1, \dots, x_m$ aux variables correspondantes, puis calculer le code du nouveau programme. Par exemple, si le programme $\Phi_e : \mathbb{N}^4 \rightarrow \mathbb{N}$ est

```
function MonProgramme(x1, x2, x3, x4) {
    // CODE
}
```

Le programme $\Phi_{S_2^2(e, 5, 3)}$ correspond à la chaîne

```
function MonProgramme2(x3, x4) {
    x1 = 5;
    x2 = 3;
    // CODE
}
```

■

On utilisera dans la suite le théorème SMN sans faire appel à lui explicitement, avec des phrases comme « pour tout x , soit e_x le code du programme qui prend y en entrée et fait ... [quelque chose qui dépend de x et y] », étant entendu alors que le processus pour obtenir e à partir de x est calculable. On dit aussi dans ce cas que le processus est *uniforme* en x .

Codage acceptable

Le théorème d'existence d'un programme universel et le théorème SMN ne sont pas valides pour n'importe quelle fonction de codage. Rappelons ici celle que nous utilisons : chaque programme P est codé par l'entier correspondant à la représentation binaire de la chaîne de caractère qui contient P .

Rogers [183] a prouvé que toute fonction de codage satisfaisant le théorème de programme universel et le théorème SMN était le résultat d’une permutation calculable de ce codage canonique. Il existe cependant d’autres codages des fonctions partielles calculables ne satisfaisant pas ces théorèmes. En particulier, Friedberg [64] a défini un codage calculable de toutes les fonctions partielles calculables sans répétition. Cela n’est bien entendu pas le cas pour notre codage, pour lequel une même fonction partielle a une infinité de codes différents. C’est ce que nous nous apprêtons à voir avec le lemme 5.1.

5. Lemme de remplissage

Le lemme de remplissage est utile de temps à autre et indique que pour tout programme informatique e , il existe une infinité de programmes équivalents, dont on peut de surcroît calculer le code à partir de e : il suffit de rajouter des instructions qui ne servent à rien.

Lemme 5.1 (Lemme de remplissage)

Il existe une fonction calculable totale $h : \mathbb{N}^2 \rightarrow \mathbb{N}$ telle que pour tous $e, n \in \mathbb{N}$, on a $h(e, n) \geq n$, et $\Phi_{h(e, n)} = \Phi_e$. ★

PREUVE

Étant donné le code e et un entier n , décoder e pour obtenir le programme P_e . Ajouter n instructions inutiles à P_e , puis coder le nouveau programme en un entier i .

Par exemple, si le langage possède une instruction **skip** qui n’effectue rien, alors il suffit d’ajouter au programme une suite d’instructions **skip** comme suit :

```
function MonProgramme(x) {
    // CODE
    skip;
    skip;
    ...
}
```

■

Notons bien que $\Phi_{h(e, n)}$ et Φ_e sont égales en tant que fonctions mathématiques, mais ont des codes informatiques différents.

6. Théorème du point fixe de Kleene

Le théorème du point fixe de Kleene, appelé parfois *recursion theorem* en anglais, est bien plus subtil que le théorème SMN. Stephen Cole Kleene est considéré avec Kurt Gödel, Alan Turing, Emil Post et Alonzo Church, son professeur, comme un des fondateurs de la calculabilité. Il formalise avec Post la notion de *degré d'insolubilité*, que l'on appellera plus tard *degré Turing* et qui sera définie précisément dans le chapitre 4. Parmi ses travaux les plus remarquables, figurent la définition et l'étude des ensembles hyperarithmétiques et des ordinaux calculables [114], que nous verrons dans la partie IV et pour lesquels nous aurons besoin de créer des programmes informatiques *ayant accès à leur propre code*.

La notion peut sembler douteuse : comment peut-on utiliser dans la définition d'un objet A l'objet A lui-même ? Les auto-références mènent souvent à des paradoxes. Nous allons cependant voir que dans le cas de programmes informatiques, l'accès à son propre code est tout à fait valide, et même bien utile dans certains cas. Nous verrons un exemple remarquable d'utilisation du théorème du point fixe dans la preuve du théorème 19-1.7. Voyons plus précisément de quoi il s'agit. Le théorème stipule que pour toute fonction qui modifie des programmes, il existe un programme dont le comportement n'est pas modifié par la fonction.



Stephen Cole Kleene, 1909–1994

Théorème 6.2

Pour toute fonction totale calculable $f : \mathbb{N} \rightarrow \mathbb{N}$, il existe $e \in \mathbb{N}$ tel que pour tout n ,

$$\Phi_{f(e)}(n) = \Phi_e(n).$$

Avant de passer à la preuve, voyons un peu en quoi cette mystérieuse affirmation permet de créer des programmes ayant accès à leur propre code. Supposons qu'un programme M utilise une variable `var` ayant été initialisée à une certaine valeur au début de son exécution. On peut alors définir une fonction totale calculable f qui prend un entier n en paramètre, et renvoie le code du programme M , qui commence son exécution avec `var` initialisée à n . D'après le théorème du point fixe, il y a une valeur e telle que les programmes de code e et $f(e)$ ont le même comportement. Donc, e

est un code de programme équivalent à celui du programme M s'exécutant avec la variable `var` initialisée à e : il y a une version de M qui peut accéder à son propre code. Voici, plus formellement, ce que nous venons d'énoncer.

Corollaire 6.3

Pour toute fonction partielle calculable $g : \mathbb{N}^2 \rightarrow \mathbb{N}$, il existe $e \in \mathbb{N}$ tel que pour tout n , on a

$$\Phi_e(n) = g(e, n)$$

PREUVE. Soit i tel que $\Phi_i(x, n) = g(x, n)$ pour tous x, n . Par le théorème SMN (voir le théorème 4.1), pour tous x, n , on a $\Phi_{S_2^1(i, x)}(n) = \Phi_i(x, n)$. Soit f la fonction définie par $f(x) = S_2^1(i, x)$. Par le théorème du point fixe (voir le théorème 6.2), il existe e tel que, pour tout n , $\Phi_{f(e)}(n) = \Phi_e(n)$. En particulier, $\Phi_e(n) = \Phi_{f(e)}(n) = \Phi_i(e, n) = g(e, n)$. ■

La preuve du théorème 6.2, bien que concise, est un peu obscure ; c'est pourquoi nous fournissons au préalable un morceau de code dont l'objectif est de donner l'intuition au programmeur de la manière dont on peut écrire un programme ayant accès à son propre code. L'exemple est ici donné dans le langage Javascript.

Dans ce qui suit, les deux points de suspension doivent chacun contenir le même code, peu importe lequel. En Javascript, les « backquotes » délimitent une chaîne de caractères sur plusieurs lignes. La fonction `replace` remplacera la première occurrence de « # » par le contenu de la variable `v`.

```
function fct() {
  let v=`
function fct() {
  let v='#'
  v = v.replace('#', v)
  ... //mon code
}`
  v = v.replace('#', v)
  ... //mon code
}
```

L'exécution de ce programme se fera avec la variable `v` ayant pour contenu le programme lui-même, à ceci près que les « backquotes » se transforment alors en simple « quotes ». Il est bien entendu possible de faire en sorte que la variable `v` contienne *exactement* le programme, mais cela rendrait l'exemple beaucoup moins compréhensible. Ayant pour objectif de donner une intuition et non une preuve, nous avons conservé les choses ainsi. Pas-

sons justement à présent à la preuve formelle de notre théorème, dans le cadre des notations et principes que nous avons définis jusque-là.

PREUVE DU THÉORÈME 6.2. Soit a le code d'une machine à un paramètre, qui sur l'entrée n renvoie le code d'une machine à un paramètre m , qui effectue les opérations suivantes.

- (1) Elle lance le calcul de $f(\Phi_n(n))$.
- (2) Si l'on a $f(\Phi_n(n)) \downarrow$, alors elle renvoie le résultat du calcul de la machine de code $f(\Phi_n(n))$ sur l'entrée m (et sinon ne s'arrête pas).

Formellement, a est tel que, pour tous $n, m \in \mathbb{N}$,

$$\Phi_{\Phi_a(n)}(m) = \Phi_{f(\Phi_n(n))}(m).$$

On fait ici un léger abus de notation : si $\Phi_n(n) \uparrow$, alors $m \mapsto \Phi_{f(\Phi_n(n))}(m)$ dénote la fonction nulle part définie. Notez que Φ_a est une fonction totale : pour tout n , dans le calcul de $\Phi_a(n)$, on ne cherche pas à faire les étapes (1) et (2), mais seulement à calculer le code d'une machine qui les fait. La démonstration qu'un tel code a existe est donnée par le théorème SMN, voici comment. La fonction $(n, m) \mapsto \Phi_{f(\Phi_n(n))}(m)$ est calculable (éventuellement partielle), et il y a donc un code b tel que

$$\Phi_b(n, m) = \Phi_{f(\Phi_n(n))}(m).$$

D'après le théorème SMN, il y a une fonction totale calculable s telle que $\Phi_{s(b, n)}(m) = \Phi_{f(\Phi_n(n))}(m)$. Comme s est totale calculable, il y a un code a tel que $\Phi_a(n) = s(b, n)$.

Le point fixe sera alors $\Phi_a(a)$. En effet, on a

$$\forall m, \Phi_{\Phi_a(a)}(m) = \Phi_{f(\Phi_a(a))}(m).$$

Cela conclut la preuve. ■

Théorème de point fixe et paradoxe

Comme expliqué plus haut, le théorème du point fixe de Kleene permet de concevoir des programmes *auto-référents*, c'est-à-dire des programmes qui peuvent lire leur code au cours de l'exécution, et adapter leur comportement en fonction. La capacité à faire de l'auto-référence est souvent source de paradoxes.

Le paradoxe du barbier, par exemple, raconte l'histoire d'un barbier philanthrope qui décida de raser tous les gens qui ne se rasaient pas eux-mêmes. Dût-il se raser lui-même ? Paradoxe... En mathématiques, le paradoxe de Russell suit le même schéma : soit E l'ensemble de tous les ensembles qui n'appartiennent pas à eux-mêmes. Par exemple, $\mathbb{N}$ n'est pas un entier, donc $\mathbb{N} \notin \mathbb{N}$, donc $\mathbb{N} \in E$. La question problématique est alors « est-ce que $E \in E$? »

Pourquoi le théorème du point fixe de Kleene n'engendre-t-il pas de paradoxe ? Si l'on essaye de suivre le même schéma que les deux paradoxes précédents, on va définir une fonction Φ_e qui connaît son code e , et donc peut décider, pour toute entrée n , d'exécuter $\Phi_e(n)$ et de renvoyer une valeur différente de celle renvoyée par $\Phi_e(n)$. On aurait donc $\Phi_e(n) \neq \Phi_e(n)$. La solution vient de la partialité des fonctions : en effet, Φ_e ne sera tout simplement pas définie en n et s'exécutera à l'infini.

Le lecteur désireux d'explorer les possibilités du théorème du point fixe pourra s'atteler à l'exercice suivant, dont l'objet est de démontrer le théorème de Rice, qui sera abordé plus en détail dans la section 5-6.

Exercice 6.4. (*) Soit $A \subseteq \mathbb{N}$ tel que $A \neq \mathbb{N}$, $A \neq \emptyset$ et tel que A est calculable.

1. Montrer qu'il existe une fonction calculable f telle que l'on a $x \in A$ ssi $f(x) \notin A$.
2. En utilisant le théorème du point fixe, en déduire qu'il existe i, j tels que $\Phi_i = \Phi_j$, avec $i \in A$ et $j \notin A$.
3. En déduire qu'il n'existe pas de prédicats calculables A tels que $e \in A$ ssi e est le code d'un programme qui fait la multiplication par 2.
4. Généraliser la question précédente pour montrer le *théorème de Rice* : pour tout « comportement non trivial », il n'est pas possible de calculer l'ensemble des codes de programmes ayant ce comportement. Formellement : soit un prédicat $P \subseteq \mathbb{N}$ avec $P \neq \mathbb{N}$ et $P \neq \emptyset$ tel que, pour tous e_1, e_2 pour lesquels $\Phi_{e_1} = \Phi_{e_2}$, on a $e_1 \in P \leftrightarrow e_2 \in P$. Alors, P n'est pas calculable. $\diamond$

7. Ensembles calculatoirement énumérables

La calculabilité place le « calculable » comme puissance calculatoire de référence. Il s'agit de la plus faible notion calculatoire que ce paradigme permet d'identifier. Comme nous l'avons vu, un ensemble E est calculable s'il existe une procédure qui, étant donné un élément, indique si cet élément appartient à E ou non. La procédure doit toujours s'arrêter et donner une réponse correcte.

Il existe cependant un certain nombre de problèmes mathématiques qui peuvent s'exprimer naturellement sous forme d'ensembles dont les éléments sont énumérables par une procédure calculable, mais dans le désordre. Ces ensembles sont appelés *calculatoirement énumérables*.

Par exemple, soit E l'ensemble des théorèmes mathématiques. Il n'existe pas de procédure qui, étant donné une formule mathématique, renvoie vrai ou faux selon que cette formule soit prouvable ou non. Cependant, il est possible d'énumérer les théorèmes, en listant toutes les chaînes de caractères possibles, testant s'il s'agit d'une preuve valide, et si c'est le cas, en énumérant la conclusion de la preuve (nous en discuterons plus en détail dans le chapitre 9).

Nous allons maintenant définir formellement la notion d'ensemble calculatoirement énumérable sous une forme qui peut sembler éloignée de la définition informelle que nous venons de donner. Nous verrons à travers la proposition 7.2 que ces définitions coïncident.

Définition 7.1. Un ensemble $A \subseteq \mathbb{N}$ est *calculatoirement énumérable* (c. e.) s'il existe un code de programme e tel que $n \in A \leftrightarrow \Phi_e(n) \downarrow$, pour tout $n \in \mathbb{N}$. ◇

Notons que les ensembles calculatoirement énumérables étaient historiquement appelés *récurivement énumérables*. Il est encore fréquent de voir des articles utilisant l'ancienne terminologie, et en particulier l'abréviation r. e. au lieu de c. e.

On peut voir la machine de code e comme un processus qui énumère A : pour chaque entier n , on cherche un entier t tel que $\Phi_e(n)$ s'arrête en t étapes de calcul. Si un tel t est trouvé, on énumère alors n dans notre ensemble. Sachant qu'il existe une infinité d'entiers, il convient de fixer un ordre d'exécution pour ne réaliser qu'un nombre fini de calculs à chaque étape. L'idée est de décomposer étape par étape comme suit.

1. Tester si $\Phi(0)[0] \downarrow$.
2. Tester si $\Phi(0)[1] \downarrow$ ou $\Phi(1)[1] \downarrow$.
3. Tester si $\Phi(0)[2] \downarrow$ ou $\Phi(1)[2] \downarrow$ ou $\Phi(2)[2] \downarrow$.
4. ...

La première fois que l'on trouve une étape t telle que $\Phi(n)[t] \downarrow$ pour un certain entier n , on énumère ce dernier. Une telle énumération peut se voir comme une fonction calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ où $f(n)$ est le n -ième élément énuméré dans A . Cette idée est reprise dans la preuve de la proposition suivante.

Proposition 7.2. Un ensemble infini A est calculatoirement énumérable ssi il existe une fonction totale calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ injective telle que

$$f(\mathbb{N}) = A. \quad \star$$

PREUVE. Soit A un ensemble calculatoirement énumérable infini, et soit e tel que $\Phi_e(n) \downarrow$ ssi $n \in A$. On définit la fonction calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ comme suit.

- ▷ Pour calculer $f(0)$, on cherche le plus petit t tel que $\Phi_e(s)[t] \downarrow$ pour un certain $s \leq t$, et l'on renvoie alors le plus petit tel entier $s \leq t$.
- ▷ Pour calculer $f(n+1)$, on lance d'abord le calcul de $f(i)$ pour tout $i \leq n$, puis on cherche le plus petit t tel que $\Phi_e(s)[t] \downarrow$ pour un certain $s \leq t$ tel que s est différent de chaque $f(i)$ pour $i \leq n$, et l'on renvoie alors le plus petit tel entier $s \leq t$.

Le processus pour déterminer $f(n)$ est bien calculable, et comme notre ensemble calculatoirement énumérable est infini, la fonction f s'arrêtera sur toutes ses valeurs. Il est alors clair que $f(\mathbb{N}) = A$.

Supposons à présent que $f(\mathbb{N}) = A$ pour une fonction totale calculable f . On définit la fonction g qui sur n cherche le plus petit t tel que $f(t) = n$, et ensuite s'arrête (et sinon cherche indéfiniment sans jamais s'arrêter). On a bien $g(n) \downarrow$ ssi $\exists t \ f(t) = n$. ■

Il devrait être clair pour le lecteur qu'un ensemble calculable est calculatoirement énumérable.

Exercice 7.3. Montrer que tout ensemble calculable est calculatoirement énumérable. ◇

Nous allons voir que l'inverse n'est pas forcément vrai : il y a des ensembles calculatoirement énumérables qui ne sont pas calculables. La proposition suivante donne plus précisément la connexion entre ces deux concepts.

Proposition 7.4. Un ensemble A est calculable ssi A et $\mathbb{N} \setminus A$ sont tous les deux calculatoirement énumérables. ★

PREUVE. Soit A un ensemble calculable. D'après l'exercice 7.3, il est calculatoirement énumérable. Par ailleurs, l'ensemble $\mathbb{N} \setminus A$ est lui aussi calculable, et par conséquent calculatoirement énumérable.

Supposons à présent que l'on ait deux codes e_1, e_2 tels que $\Phi_{e_1}(n) \downarrow$ ssi $n \in A$ et $\Phi_{e_2}(n) \downarrow$ ssi $n \in \mathbb{N} \setminus A$. On définit la fonction calculable $f(n)$ qui cherche le plus petit t tel que $\Phi_{e_1}(n)[t] \downarrow$ ou tel que $\Phi_{e_2}(n)[t] \downarrow$, puis renvoie 1 dans le premier cas et 0 dans le deuxième. Il est clair que la fonction f calcule l'ensemble A . ■

Si un ensemble calculatoirement énumérable n'est pas en général calculable, il peut être approximé par une suite croissante d'ensembles uniformément calculables, au sens suivant. On dit qu'une suite d'ensembles $A_0, A_1, \dots$ est *uniformément calculable* s'il existe une fonction $f : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ calculable et telle que, pour tous x, n , on a $f(x, n) = 1$ ssi $x \in A_n$.

Définition 7.5. Une *approximation c. e.* d'un ensemble A est une suite uniformément calculable d'ensembles $A_0, A_1, \dots$ telle que $A_n \subseteq A_{n+1}$ pour tout n , et $\bigcup_n A_n = A$. $\diamond$

Proposition 7.6. Un ensemble A est calculatoirement énumérable si, et seulement si, il possède une approximation c. e. $\star$

PREUVE. Supposons que A soit c. e. Par définition, il existe un code de programme e tel que $x \in A \leftrightarrow \Phi_e(x) \downarrow$, pour tout $x \in \mathbb{N}$. Soit $A_0, A_1, \dots$ la suite uniformément calculable d'ensembles définie par $A_n = \{x : \Phi_e(x)[n] \downarrow\}$. Il est clair que $A_n \subseteq A_{n+1}$, car si la machine Φ_e s'arrête sur une entrée x avant n étapes, elle s'arrêtera avant $n+1$ étapes.

Supposons maintenant que l'ensemble A possède une approximation c. e., à savoir $A_0, A_1, \dots$. Soit le programme Φ_e qui, pour une entrée x , calcule $A_0(x)$, puis $A_1(x)$, puis $A_2(x)$, et ainsi de suite, jusqu'à ce que $A_n(x) = 1$ pour un n et s'arrête à ce moment-là. Si $A_n(x) = 0$ pour tout n , alors $\Phi_e(x) \uparrow$, sinon $\Phi_e(x) \downarrow$. Par construction, $\{x : \Phi_e(x) \downarrow\} = \bigcup_n A_n = A$. $\blacksquare$

Notation

Étant donné un ensemble c. e. A , on notera $A[0], A[1], \dots$ une approximation c. e. de A fixée. En particulier, $A[s]$ est l'*approximation de A au temps s* . Par convention, $A[s]$ est un ensemble fini avec $\max A[s] < s$ si $A[s] \neq \emptyset$. On utilisera parfois aussi la notation A_s à la place de $A[s]$.

Comme nous l'avons dit, il existe des ensembles calculatoirement énumérables qui ne sont pas calculables. L'exemple canonique est connu sous le nom de « problème de l'arrêt ».

Définition 7.7. On appelle le *problème de l'arrêt* ou plus simplement *l'arrêt*, que l'on note $\emptyset'$, l'ensemble :

$$\emptyset' = \{n \in \mathbb{N} : \Phi_n(n) \downarrow\}.$$

$\diamond$

Alan Turing démontre en 1936 que l'arrêt n'est pas calculable, en utilisant un argument diagonal, comme celui introduit par Cantor pour démontrer la non-dénombrabilité des réels. Avant d'en donner une preuve mathématique, nous en donnons l'intuition via un peu de code utilisant la syntaxe Java.

Supposons par l'absurde que nous ayons à notre disposition une fonction `boolean Halt(String p, String v)` qui renvoie `true` si la méthode se trouvant dans la chaîne `p` s'arrête quand elle est exécutée avec le paramètre `v`, et renvoie `false` sinon. Comme d'habitude, si `p` contient une

chaîne de caractère ne correspondant pas à une fonction valide ou ne correspondant pas à une fonction prenant un paramètre de type `String`, alors `Halt` renvoie `false`. Considérons le programme suivant.

```
function Diagonale(x) {
  if (Halt(x, x)) {
    while (true); //boucle infinie
  } else {
    return; //fin
  }
}
```

Que renvoie l'exécution du programme `Diagonale` avec comme paramètre une chaîne de caractère `x` correspondant au programme `Diagonale` lui-même ? On voit que l'on arrive à un paradoxe :

- ▷ si `Halt(x,x)` renvoie `true`, alors `Diagonale` ne s'arrête pas sur `Diagonale` comme paramètre ;
- ▷ dans le cas inverse, `Diagonale` s'arrête sur `Diagonale` comme paramètre.

Ainsi, la méthode `Halt` ne tient pas ses promesses.

Théorème 7.8

L'arrêt est un ensemble calculatoirement énumérable qui n'est pas calculable.

PREUVE. La fonction partielle $f : n \mapsto \Phi_n(n)$ est clairement calculable, et l'on a $f(n) \downarrow$ ssi $\Phi_n(n) \downarrow$. Aussi a-t-on, par définition, $\emptyset' = \{n : f(n) \downarrow\}$. Donc, $\emptyset'$ est calculatoirement énumérable.

Supposons maintenant par l'absurde que $\emptyset'$ soit un ensemble calculable. En particulier, d'après la proposition 7.4, l'ensemble $\mathbb{N} \setminus \emptyset'$ est calculatoirement énumérable, et il existe un code e tel que $n \in \mathbb{N} \setminus \emptyset'$ ssi $\Phi_e(n) \downarrow$. Alors, pour tout n ,

$$\Phi_e(n) \downarrow \leftrightarrow n \in \mathbb{N} \setminus \emptyset' \leftrightarrow \Phi_n(n) \uparrow.$$

En particulier, pour $n = e$,

$$\Phi_e(e) \downarrow \leftrightarrow e \in \mathbb{N} \setminus \emptyset' \leftrightarrow \Phi_e(e) \uparrow,$$

ce qui est une contradiction. L'arrêt n'est donc pas calculable, et en particulier le complémentaire de l'arrêt n'est pas calculatoirement énumérable. ■

Nous invitons le lecteur à considérer les exercices suivants, qui permettent de réfléchir un peu sur les ensembles calculatoirement énumérables.

Exercice 7.9. (★) Un ensemble infini est *calculatoirement énumérable dans l'ordre* s'il existe une fonction totale $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $f(\mathbb{N}) = A$ et telle que $f(n) < f(n+1)$. Montrer que si un ensemble infini A est calculatoirement énumérable dans l'ordre, alors A est calculable. $\diamond$

Exercice 7.10. (★) Montrer que tout ensemble calculatoirement énumérable infini contient un sous-ensemble calculable infini. $\diamond$

Exercice 7.11. (★★) Deux ensembles A et B sont *calculatoirement inséparables* si $A \cap B = \emptyset$ et si aucun ensemble calculable C ne permet de séparer A et B ; autrement dit, aucun ensemble calculable C n'est tel que $A \subseteq C$ et $C \cap B = \emptyset$. Montrer qu'il existe deux ensembles c.e. calculatoirement inséparables. $\diamond$

Exercice 7.12. (★★) Montrer qu'étant donné $A, B \subseteq \mathbb{N}$ deux ensembles calculables, l'ensemble $D_{A,B} = \{x - y : x \geq y \text{ et } x \in A \text{ et } y \in B\}$ n'est pas nécessairement calculable.

Indication.— Montrer que pour tout ensemble c.e. $C \subseteq \mathbb{N}$, il existe deux ensembles calculables $A, B \subseteq \mathbb{N}$ tels que $x \in C$ ssi $2^x \in D_{A,B}$. $\diamond$

Certaines personnes — une petite minorité, rassurez-vous — auront peut-être intégré avec une très grande facilité tout ce qui a été vu jusqu'ici, au point sans doute de s'ennuyer un peu. C'est à celles-là que s'adresse l'exercice suivant, qui devrait les occuper un petit moment...

Exercice 7.13. (★★★) (Friedberg [64]). Un ensemble c.e. X est *maximal* si $\mathbb{N} \setminus X$ est infini et si tout ensemble c.e. $Y \supseteq X$ est tel que $Y \setminus X$ est fini ou tel que $\mathbb{N} \setminus Y$ est fini. Montrer qu'il existe un ensemble c.e. maximal.

Indication.— On note W_e l'ensemble c.e. donné par $\{n \in \mathbb{N} : \Phi_e(n) \downarrow\}$. On pourra commencer par trouver un processus uniforme qui sur chaque $e \in \mathbb{N}$ associe un code d tel que $\mathbb{N} \setminus W_d$ est infini, et tel que si $W_d \subseteq W_e$; alors, soit W_e est fini, soit $\mathbb{N} \setminus W_e$ est fini. $\diamond$

Chapitre 4

Degrés Turing

Un ensemble non calculable peut être vu comme un problème insoluble : il n'existe pas d'algorithme permettant de décider si un entier appartient ou non à cet ensemble. Un des objectifs de la calculabilité est d'étudier et de comprendre l'univers des problèmes insolubles, via différentes comparaisons et classifications. On introduit pour cela différents outils. Le plus important d'entre eux est l'objet de ce chapitre et trouve sa genèse dans « Systems of logic based on ordinals » [226], le fameux article d'Alan Turing présentant ses travaux de thèse, et sera formalisé et étudié plus tard par Post [181] puis Post et Kleene [118] : étant donné un ensemble non calculable A , on imagine pouvoir l'utiliser comme « oracle » afin d'augmenter la puissance de calcul de nos machines. On dira alors que deux ensembles sont dans le même *degré d'insolubilité* ou dans le même *degré Turing*, si chacun peut se calculer avec un algorithme utilisant l'autre comme « oracle ».

1. Les chaînes finies

Avant d'entrer dans le vif du sujet, nous devons introduire un peu de vocabulaire et de notation sur les chaînes binaires. Formellement, une chaîne binaire est une fonction partielle de $\mathbb{N}$ vers $\{0, 1\}$ dont le domaine de définition est un segment initial de $\mathbb{N}$. De manière plus informelle, il s'agit d'une suite finie de 0 et de 1.

Définition 1.1. On note $2^{<\mathbb{N}}$ l'ensemble des suites finies de 0 et de 1. Les variables σ, τ, ρ seront normalement utilisées pour dénoter des éléments de $2^{<\mathbb{N}}$, que l'on appellera généralement des *chaînes*.

Ces suites seront manipulées via les symboles suivants :

- ▷ ϵ : la chaîne vide, de taille 0
- ▷ i^n pour $i \in \{0, 1\}$: une suite de n répétitions du bit i
- ▷ $\sigma\tau$ ou $\sigma \frown \tau$: la concaténation de σ et τ
- ▷ $\sigma \preceq \tau$: la chaîne σ est un préfixe de τ , c'est-à-dire $\exists \rho$ tel que $\sigma\rho = \tau$
- ▷ $\sigma \prec \tau$: la chaîne σ est un préfixe strict de τ , c'est-à-dire $\exists \rho \neq \epsilon$ tel que $\sigma\rho = \tau$
- ▷ $|\sigma|$: la taille de σ
- ▷ $\sigma(n)$ pour $n < |\sigma|$: la valeur du n -ième bit de σ , en commençant à 0
- ▷ On dira que deux chaînes σ, τ sont *incompatibles* si l'on n'a ni $\sigma \preceq \tau$ ni $\tau \preceq \sigma$ (on écrira aussi $\sigma \not\preceq \tau$ et $\tau \not\preceq \sigma$). Si à l'inverse $\sigma \preceq \tau$ ou $\tau \preceq \sigma$, les deux chaînes σ et τ sont compatibles. $\diamond$

On identifiera parfois un bit $i \in \{0, 1\}$ avec la chaîne de longueur 1 dont l'unique bit est i . Ainsi, on notera σi ou $\sigma \frown i$ la concaténation d'une chaîne σ et d'un bit i . Suivant les définitions précédentes, $|\epsilon| = 0$, et pour toute chaîne σ non vide, le premier et dernier bit sont respectivement $\sigma(0)$ et $\sigma(|\sigma| - 1)$. Les chaînes et les suites infinies peuvent se combiner.

Définition 1.2. On adoptera les notations suivantes pour $\sigma \in 2^{<\mathbb{N}}$ et $X \in 2^{\mathbb{N}}$:

- ▷ σX : la concaténation de σ et X
- ▷ $\sigma \prec X$: la chaîne σ est un préfixe de X , c'est-à-dire $\exists Y \in 2^{\mathbb{N}} \sigma Y = X$
- ▷ $X \upharpoonright_n$ pour $n \geq 0$: le préfixe de X de taille n . $\diamond$

2. Calcul avec oracle

Intuitivement, un calcul avec oracle, par exemple $A \subseteq \mathbb{N}$, est simple à comprendre : il s'agit d'un calcul qui peut à tout moment utiliser une instruction qui « pose une question » à l'oracle, de la forme : « est-ce que n appartient à A ? » Cette instruction peut s'apparenter à un appel de fonction, qui renvoie toujours la bonne réponse.

Si l'oracle utilisé n'est pas calculable, on peut donc écrire des programmes informatiques qui calculent, à l'aide de cet oracle, des objets normalement non calculables, à commencer par l'oracle lui-même. Notre objectif est à présent d'étudier les ensembles d'entiers du point de vue de la puissance de calcul qu'ils fournissent quand ils sont utilisés comme oracles.

Définition 2.1. Une *fonctionnelle* Turing ou simplement une *fonctionnelle* est une fonction calculable par un algorithme ayant la possibilité d'utiliser, en plus de son jeu d'instruction habituel, une instruction permettant de savoir si un entier n appartient à l'oracle. $\diamond$

Une fonctionnelle prend donc un ou plusieurs paramètres entiers, comme pour les fonctions calculables, et un ou plusieurs paramètres d'oracles, ici des ensembles d'entiers (on parle aussi parfois de paramètres « réels » quand on les voit comme le développement binaire de nombres réels). On appellera aussi *paramètres du premier ordre* les paramètres entiers et *paramètres du second ordre* les paramètres qui sont des ensembles d'entiers.

Notation

On note $\Phi_e(A, n)$ ou $\Phi_e^A(n)$ le résultat du calcul de la fonctionnelle Φ_e avec l'oracle A et sur l'entrée n . On écrira de la même manière $\Phi_e(A, n) \downarrow$, $\Phi_e(A, n) \uparrow$, $\Phi_e(A, n)[t] \downarrow$, $\Phi_e(A, n)[t] \uparrow$ pour signifier que la fonctionnelle Φ_e respectivement s'arrête, ne s'arrête pas, s'arrête en temps inférieur à t , ne s'arrête pas en temps inférieur à t , avec l'oracle A et sur l'entrée n .

Afin d'avoir une vision uniforme, on suppose à présent que l'on ne travaille qu'avec des fonctionnelles. Il est facile de voir que les fonctions calculables sont exactement celles qui le sont par des fonctionnelles utilisant l'ensemble vide comme oracle (ou n'importe quel autre ensemble calculable).

Définition 2.2. Un ensemble A est dit *X-calculable* ou calculable relativement à X s'il est calculable par une fonctionnelle Turing utilisant X comme oracle. Un ensemble A est dit *calculatoirement énumérable relativement à X* si c'est le domaine de définition d'une fonctionnelle Turing utilisant X comme oracle. $\diamond$

La notion d'oracle se généralise aux fonctions. Avec un oracle $f : \mathbb{N} \rightarrow \mathbb{N}$, un calcul consiste à ajouter la fonction f aux primitives du langage de programmation. Ainsi, le programme pourra à tout moment interroger f sur des entrées pour en connaître les résultats. On peut par conséquent parler d'ensemble *f-calculable* s'il est calculable par une fonctionnelle Turing utilisant f comme oracle. De manière équivalente, un ensemble est *f-calculable* s'il est G_f -calculable, où $G_f \in 2^{\mathbb{N}}$ est un encodage du graphe de la fonction f par un élément de $2^{\mathbb{N}}$, par exemple avec $\langle n, m \rangle \in G_f$ ssi $f(n) = m$.

Notation

On écrira aussi *X-c.e.* pour signifier calculatoirement énumérable relativement à X .

Exemple 2.3. Supposons par exemple que l'on dispose d'un oracle X inclus dans $\mathbb{N}$ tel que la fonction f qui à n associe le n -ième élément de X croisse suffisamment vite pour borner le temps d'arrêt des programmes informatiques. Formellement : $\Phi_e(e) \downarrow$ implique $\Phi_e(e)[f(e)] \downarrow$ pour tout $e \in \mathbb{N}$. Alors, il est aisé de créer une fonctionnelle Turing permettant de calculer $\emptyset'$ à partir de X : pour savoir si $e \in \emptyset'$, il suffit de parcourir X jusqu'à trouver son e -ième élément $f(e)$, puis de calculer $\Phi_e(e)$ durant $f(e)$ étapes de calcul. Si $\Phi_e(e)[f(e)] \downarrow$, alors $e \in \emptyset'$. Sinon, $e \notin \emptyset'$.

3. Relativisation des preuves

La plupart des arguments de calculabilité s'appliquent aux machines à oracle en remplaçant « machine » par « machine avec un oracle X ». Cette opération purement syntaxique que l'on appelle *relativisation* (à un oracle) donne donc gratuitement un schéma de résultats similaires, paramétrés par un oracle X . Prenons l'exemple de l'indécidabilité du problème de l'arrêt, en remplaçant la notion de calcul par celle de calcul avec oracle X .

Théorème 3.1

Pour tout oracle $\mathbf{X}$, l'ensemble $Y = \{n : \Phi_n^{\mathbf{X}}(n) \downarrow\}$ n'est pas $\mathbf{X}$ -calculable.

PREUVE. La fonction partielle $f : n \mapsto \Phi_n^{\mathbf{X}}(n)$ est clairement $\mathbf{X}$ -calculable, et l'on a $f(n) \downarrow$ ssi $\Phi_n^{\mathbf{X}}(n) \downarrow$. Aussi a-t-on, par définition, $Y = \{n \in \mathbb{N} : f(n) \downarrow\}$. Donc, Y est $\mathbf{X}$ -calculatoirement énumérable.

Supposons maintenant par l'absurde que l'ensemble Y est $\mathbf{X}$ -calculable. En particulier, d'après la proposition 3-7.4 relativisée à $\mathbf{X}$, l'ensemble $\mathbb{N} \setminus Y$ est $\mathbf{X}$ -calculatoirement énumérable, et il existe un code e tel que $n \in \mathbb{N} \setminus Y$ ssi $\Phi_e^{\mathbf{X}}(n) \downarrow$. On a alors pour tout n

$$\Phi_e^{\mathbf{X}}(n) \downarrow \leftrightarrow n \in \mathbb{N} \setminus Y \leftrightarrow \Phi_n^{\mathbf{X}}(n) \uparrow.$$

En particulier, pour $n = e$, on a

$$\Phi_e^{\mathbf{X}}(e) \downarrow \leftrightarrow e \in \mathbb{N} \setminus Y \leftrightarrow \Phi_e^{\mathbf{X}}(e) \uparrow,$$

ce qui est une contradiction. Donc, Y n'est pas $\mathbf{X}$ -calculable, et en particulier le complémentaire de Y n'est pas $\mathbf{X}$ -calculatoirement énumérable. ■

Il convient cependant d'être précautionneux lors de la relativisation d'un argument. En effet, certaines définitions masquent des appels à des machines, et leur définition doit alors être également relativisée. Par exemple, si l'on

définit le *problème de l'arrêt* comme l'ensemble $\{n : \Phi_n(n) \downarrow\}$, la relativisation de l'énoncé « Le problème de l'arrêt n'est pas calculable » n'est pas « Pour tout X , le problème de l'arrêt n'est pas X -calculable », mais « Pour tout X , le problème de l'arrêt relativisé à X n'est pas X -calculable », où le « problème de l'arrêt relativisé à X » est en fait défini comme l'ensemble $\{n : \Phi_n^X(n) \downarrow\}$.

De manière générale, la relativisation d'un théorème s'obtient en ajoutant un oracle X à chaque machine, et en remplaçant *calculable* par *X -calculable*. C'est par exemple le cas pour une relativisation naïve du théorème SMN.

Théorème 3.2 (Théorème SMN relativisé - version 1)

Pour tout oracle X , et pour tous entiers n et m non nuls, il existe une fonction **X -calculable** totale $S_n^m : \mathbb{N}^{m+1} \rightarrow \mathbb{N}$ telle que pour tout e ,

$$\Phi_{S_n^m(e, x_1, \dots, x_m)}^X(y_1, \dots, y_n) = \Phi_e^X(x_1, \dots, x_m, y_1, \dots, y_n).$$

Une analyse de la preuve du théorème SMN révèle cependant que la fonction S_n^m ne dépend pas de l'oracle X , car elle effectue des manipulations purement syntaxiques sur la machine. Il est donc possible de formuler une version relativisée plus forte du théorème SMN, où la fonction S_n^m est calculable, bien que la version précédente soit également valide.

Théorème 3.3 (Théorème SMN relativisé - version 2)

Pour tout oracle X , pour tous $n, m \in \mathbb{N}^*$ il existe une fonction **calculable** totale $S_n^m : \mathbb{N}^{m+1} \rightarrow \mathbb{N}$ telle que pour tout e ,

$$\Phi_{S_n^m(e, x_1, \dots, x_m)}^X(y_1, \dots, y_n) = \Phi_e^X(x_1, \dots, x_m, y_1, \dots, y_n).$$

Digression

La relativisation des arguments est un phénomène empirique qui d'une certaine manière reflète notre compréhension partielle de la notion de calcul. Toutefois, et notamment en théorie de la complexité, les preuves ne se relativisent pas nécessairement. L'exemple emblématique est la question de la séparation des classes de complexité P et NP. Chacune de ces classes peut se relativiser à un oracle. Baker, Gill, et Solovay [9] ont montré qu'il existait des oracles X pour lesquels $P^X = NP^X$ et d'autres pour lesquels $P^X \neq NP^X$. Il est alors nécessaire pour résoudre la question P vs NP d'utiliser un argument qui ne se relativise pas. Il s'agit en calculabilité de quelque chose de tout à fait inhabituel, mais pas nécessairement en complexité où nous avons d'autres exemples de problèmes résolus qui

ne se relativisent pas. On sait par exemple que les classes IP et PSPACE coïncident [196], tout en étant capable de produire des oracles X pour lesquels $\text{IP}^X \neq \text{PSPACE}^X$ [60].

4. Propriété de l'usage

Étant donné un calcul $\Phi_e(A, n)$ sur un oracle A , il est clair que si l'on suppose que $\Phi_e(A, n) \downarrow$, alors la fonctionnelle Φ_e n'a utilisé qu'une partie finie de l'oracle A pour renvoyer le résultat : cela découle simplement du fait qu'un calcul s'effectue toujours en un nombre fini d'étapes. Cela nous amène à définir la notion de calcul avec des oracles finis.

Notation

Étant donné une suite finie $\sigma \in 2^{<\mathbb{N}}$, on note $\Phi_e(\sigma, n) \downarrow$ ou $\Phi_e^\sigma(n) \downarrow$ pour signifier que le calcul s'arrête sur l'entrée n et avec σ comme morceau d'oracle, la fonctionnelle n'ayant alors besoin pour son calcul que de poser à l'oracle des questions auxquelles σ peut répondre, c'est-à-dire des questions d'appartenance de i à l'oracle pour des entiers i strictement inférieurs à $|\sigma|$.

Dans la notation précédente, si le calcul a besoin d'accéder à des valeurs de l'oracle qui dépassent la taille de σ , alors on note $\Phi_e(\sigma, n) \uparrow$ ou $\Phi_e^\sigma(n) \uparrow$. La partie finie de l'oracle A interrogée jusqu'à ce que le calcul $\Phi_e(A, n)$ termine s'appelle l'*usage* du calcul.

Proposition 4.1 (Propriété de l'usage). Soient Φ_e une fonctionnelle Turing, X un oracle et $n \in \mathbb{N}$ une entrée. Alors, $\Phi_e(X, n) \downarrow$ si, et seulement si, il existe un préfixe fini $\sigma \prec X$ tel que $\Phi_e(\sigma, n) \downarrow$. ★

Nous verrons dans la section 8-2 que la propriété de l'usage correspond au concept de continuité sur l'espace de Cantor. Malgré sa simplicité conceptuelle, la propriété de l'usage joue un rôle primordial en calculabilité. Nous en ferons par exemple une utilisation forte dans la section 8 sur la méthode des extensions finies.

Définition 4.2. Étant donné une fonctionnelle Φ et un oracle X , on note $\text{use}_\Phi^X : \mathbb{N} \rightarrow \mathbb{N}$ la fonction partielle X -calculable qui sur n renvoie la taille minimale du préfixe de X nécessaire à l'arrêt du calcul de $\Phi(X, n)$. ♦

Remarque

D'après la propriété de l'usage, toute fonctionnelle Turing Γ peut être représentée par l'ensemble c.e. W des triplets (σ, x, y) tels que

si $(\sigma, x, y) \in W$, alors $\Gamma^\sigma(x) \downarrow = y$. Une telle énumération satisfait la propriété de cohérence suivante : pour tous $(\sigma, x, y) \in W$ et $(\tau, x, y') \in W$ tels que $\sigma \prec \tau$, on a $y = y'$.

Exercice 4.3. Montrer que si Y est X -calculable et Z est Y -calculable, alors Z est X -calculable. $\diamond$

5. Degrés Turing

Les machines à oracle induisent une notion de calculabilité relative. Informellement, un ensemble Y est X -calculable si X est au moins aussi puissant que Y , au sens où tout ce qui est calculable par Y l'est également par X . Cela donne lieu à la *réduction Turing*.

Définition 5.1 (Réduction Turing). Pour tous ensembles $X, Y \subseteq \mathbb{N}$, on écrit $X \leq_T Y$ — et l'on dit que X est *Turing réductible* à Y — si X est Y -calculable. On écrit $X <_T Y$ si $X \leq_T Y$ mais $Y \not\leq_T X$. $\diamond$

La réduction Turing forme un *pré-ordre* sur les ensembles d'entiers, c'est-à-dire que cette relation est réflexive et transitive. En effet, si Y est X -calculable et Z est Y -calculable, Z est X -calculable. Ce n'est cependant pas un ordre partiel sur les ensembles, car la réduction Turing n'est pas antisymétrique. Par exemple, les ensembles des nombres pairs et des nombres impairs sont trivialement mutuellement calculables puisqu'ils sont tous deux calculables, mais ils ne sont pas égaux en tant qu'ensembles d'entiers.

Il est cependant possible de transformer ce pré-ordre en un ordre partiel en identifiant tous les ensembles mutuellement calculables. Cela donne la notion de *degré Turing* qui représente une puissance calculatoire plus robuste que la notion d'ensemble pour la réduction Turing.

Définition 5.2. On écrit $X \equiv_T Y$ si $X \leq_T Y$ et $Y \leq_T X$. On dira alors que X et Y sont *Turing-équivalents*. On appelle *degrés Turing* les classes d'équivalences de la relation $\equiv_T$. Le degré Turing d'un ensemble X est l'ensemble $\deg_T(X) = \{Y : Y \equiv_T X\}$. $\diamond$

Par construction, si $X \leq_T Y$, alors tout élément dans le degré Turing de Y calcule n'importe quel élément dans le degré Turing de X . La réduction de Turing induit donc un ordre partiel sur les degrés Turing, que l'on notera tout simplement $\leq$.

Notation

On notera $(\mathcal{D}, \leq)$ l'ensemble des degrés Turing $\mathcal{D}$ partiellement ordonné par $\leq$. On utilisera en général les lettres $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \dots$ pour désigner ses éléments. On écrira parfois $X \leq_T \mathbf{d}$ pour $\deg_T(X) \leq \mathbf{d}$.

Exercice 5.3. Montrer que si $X \equiv_T Y$ et $A \equiv_T B$, on a alors $X \leq_T A$ si, et seulement si, $Y \leq_T B$. ◇

Deux ensembles sont donc Turing-équivalents s'ils ont la même puissance calculatoire, et la relation $\leq$ permet de comparer non plus des ensembles, mais des degrés de puissance calculatoire. En particulier, les degrés Turing sont stables par variations finies, au sens où si $X \in \mathbf{d}$ et $Y =^* X$, alors $Y \in \mathbf{d}$. Ici, $Y =^* X$ signifie que X et Y ne diffèrent que sur un nombre fini de bits.

Exercice 5.4. Montrer que les degrés Turing sont stables par variation finie. ◇

Une grande partie de la calculabilité classique consiste à comprendre la structure $(\mathcal{D}, \leq)$. L'ordre $\leq$ est-il total sur $\mathcal{D}$? Est-il bien fondé? Si c'est un ordre partiel, quelle est la taille maximale d'un ensemble ne contenant que des éléments deux à deux incomparables? Nous verrons que la structure des degrés Turing est d'une grande richesse, mais aussi d'une grande complexité. Commençons par quelques observations immédiates.

Tout d'abord, les degrés Turing possèdent un élément minimal, à savoir le degré des ensembles calculables. On le notera $\mathbf{0}$. Nous avons ensuite la proposition suivante.

Proposition 5.5. Tout degré Turing est dénombrable. ★

PREUVE. Soit $\mathbf{d}$ un degré Turing et soit $X \in \mathbf{d}$. En particulier,

$$\mathbf{d} = \deg_T(X) = \{Y : X \equiv_T Y\} \subseteq \{\{n : \Phi_e^X(n) \downarrow = 1\} : e \in \mathbb{N}\},$$

donc $\mathbf{d}$ est fini ou dénombrable. Par ailleurs, $\mathbf{d}$ est stable par variations finies, donc est infini. Ainsi, $\mathbf{d}$ est dénombrable. ■

La collection des sous-ensembles de $\mathbb{N}$ étant indénombrable, et chacun appartenant à un degré Turing, il s'ensuit de la proposition précédente que les degrés Turing sont en quantité indénombrable. Supposons en effet le contraire. Alors, d'après l'exercice 2-3.11, la réunion de tous les degrés Turing serait un ensemble dénombrable, en tant que réunion dénombrable d'ensembles dénombrables. Comme cette réunion est égale à $2^{\mathbb{N}}$, on a là une contradiction. Remarquons que l'exercice 2-3.11 utilise l'axiome du choix (dont nous parlerons en détail dans la section 9-4). Celui-ci n'est toutefois pas nécessaire : nous verrons plusieurs constructions effectives

de fonctions $f : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ (voir l'exercice 8-5.4 ou l'exercice 8-5.3) telles que $f(X)$ et $f(Y)$ sont dans des degrés Turing différents pour tous $X \neq Y$, ce qui fait de f une injection de $2^{\mathbb{N}}$ dans les degrés Turing et montre en particulier $|2^{\mathbb{N}}| \leq |\mathcal{D}|^1$.

Définition 5.6. La *jointure effective* de deux ensembles A et B est l'ensemble $A \oplus B = \{2n : n \in A\} \cup \{2n + 1 : n \in B\}$ (notons que la réunion entre les deux ensembles est disjointe). $\diamond$

La jointure effective de deux ensembles est une façon d'encoder l'information de chaque ensemble de manière à pouvoir la décoder calculatoirement. Il existe bien entendu de nombreuses manières d'encoder deux ensembles en un seul, la jointure effective étant la plus directe et efficace, au sens suivant.

Proposition 5.7. Soient A et B deux ensembles. Alors, $\deg_T(A \oplus B)$ est la borne supérieure des degrés $\deg_T(A)$ et $\deg_T(B)$, c'est-à-dire que tout degré au-dessus de $\deg_T(A)$ et de $\deg_T(B)$ est aussi au-dessus de $\deg_T(A \oplus B)$. Ainsi, toute paire de degrés Turing $\mathbf{c}$ et $\mathbf{d}$ admet une borne supérieure que l'on notera $\mathbf{c} \cup \mathbf{d}$. $\star$

PREUVE. Il est clair que la jointure $A \oplus B$ permet de calculer A et B . Ainsi, $\deg_T(A \oplus B)$ est un majorant de $\deg_T(A)$ et $\deg_T(B)$. Soit $\deg_T(C)$ un majorant de $\deg_T(A)$ et $\deg_T(B)$. En particulier, $C \geq_T A$ et $C \geq_T B$. Soient i et j des codes tels que $\Phi_i(C, n) = A(n)$ et $\Phi_j(C, n) = B(n)$ pour tout n . La fonction $f : \mathbb{N} \rightarrow \{0, 1\}$ définie par

$$f(n) = \begin{cases} \Phi_i(C, n/2) & \text{si } n \text{ est pair} \\ \Phi_j(C, (n-1)/2) & \text{sinon} \end{cases}$$

est C -calculable, et l'on a $f(n) = 1$ ssi $n \in A \oplus B$ pour tout entier n . Donc, $C \geq_T A \oplus B$. $\blacksquare$

Notation

Nous avons fait usage dans la preuve de la proposition précédente du prédicat $\ll \Phi_i(C, n) \downarrow = A(n) \gg$ pour tout n . Il nous arrivera quelques fois d'utiliser la notation $\Phi_i(C) = A$, plus courte.

Attention, la borne supérieure $\mathbf{c} \cup \mathbf{d}$ de $\mathbf{c}$ et $\mathbf{d}$ n'a bien entendu rien à voir avec la réunion ensembliste des degrés $\mathbf{c}$ et $\mathbf{d}$. De manière générale, les lettres en caractère gras $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \dots$ seront utilisées pour parler des degrés en tant qu'objets abstraits au sein d'un ordre partiel, le détail des ensembles d'entiers constituant chaque degré n'étant alors pas pertinent.

1. L'inégalité $|\mathcal{D}| \leq |2^{\mathbb{N}}|$ semble évidente, mais elle utilise l'axiome du choix, afin de sélectionner uniformément un élément dans chaque degré Turing. Nous en reparlerons brièvement dans la section 12-2.3.

Nous jonglerons désormais entre les ensembles d'entiers et les degrés Turing. Chaque degré Turing pouvant être représenté par un ensemble d'entiers (l'un de ses membres), une opération sur les degrés Turing se fera normalement via une opération sur l'un de ses représentants, de manière à ce que le résultat attendu soit indépendant du choix d'un tel représentant. La jointure effective constitue un premier exemple illustrant notre propos.

Exercice 5.8. Montrer que si l'on suppose que $X \leq_T Y$ et $A \leq_T B$, alors

$$X \oplus A \leq_T Y \oplus B. \quad \diamond$$

L'exercice précédent montre que la jointure effective induit une opération sur les degrés Turing. Nous allons étudier dans la section suivante une nouvelle opération sur les degrés jouant un rôle essentiel en calculabilité : le *saut Turing*.

6. Saut Turing

Le saut Turing est une opération fondamentale en calculabilité, et se définit comme la relativisation du problème de l'arrêt.

Définition 6.1. Étant donné un ensemble X , on définit

$$X' = \{n : \Phi_n^X(n) \downarrow\}.$$

Le *saut Turing* est l'opérateur $X \mapsto X'$. $\diamond$

On peut par exemple à présent définir l'arrêt relativement à l'arrêt : l'ensemble des codes de programmes informatiques qui s'arrêtent sur leur propre entrée, mais en utilisant l'arrêt en tant qu'oracle. On le note $\emptyset''$. Il n'est pas très difficile de montrer que le saut Turing induit une opération sur degrés Turing, et nous en laissons la preuve en exercice.

Exercice 6.2. (*) Montrer que si $X \leq_T Y$, alors $X' \leq_T Y'$. $\diamond$

Nous verrons un renforcement du résultat de l'exercice précédent avec l'exercice 5-5.7. On notera donc $\mathbf{d}'$ le saut Turing d'un degré Turing $\mathbf{d}$. En particulier, $\mathbf{0}'$ est le degré Turing du problème de l'arrêt.

Proposition 6.3. On a $X <_T X'$ pour tout $X \in 2^{\mathbb{N}}$. Ainsi a-t-on $\mathbf{d} < \mathbf{d}'$ pour tout degré Turing $\mathbf{d}$. $\star$

PREUVE. Montrons d'abord que X' calcule X . Soit $f : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ la fonction partielle X -calculable définie par

$$f(e, n) = \begin{cases} 1 & \text{si } e \in X \\ \uparrow & \text{sinon.} \end{cases}$$

Par le théorème SMN relativisé à X (voir le théorème 3.3), il existe une fonction totale calculable $g : \mathbb{N} \rightarrow \mathbb{N}$ telle que pour tous entiers e et n , on a $\Phi_{g(e)}^X(n) = f(e, n)$. Ainsi, pour tout e :

- ▷ si $e \in X$, alors $\Phi_{g(e)}^X$ est la fonction constante 1, et donc $g(e) \in X'$;
- ▷ si $e \notin X$, alors $\Phi_{g(e)}^X$ est la fonction nulle part définie, et $g(e) \notin X'$.

En particulier, X' peut calculer X : pour savoir si $n \in X$, il suffit de regarder si $g(n) \in X'$.

La preuve que $X \not\geq_T X'$ est une relativisation du fait que $\emptyset'$ n'est pas calculable, ce qui a été démontré préalablement avec le théorème 3.1. ■

Il existe donc une hiérarchie strictement croissante de degrés Turing :

$$0 < 0' < 0'' < \dots$$

Dans la preuve précédente, notons que la fonctionnelle Turing utilisée pour calculer X à partir de X' est *la même* pour tout oracle X . C'est quelque chose qui sera utilisé de temps à autre, par exemple dans le chapitre 26. Nous donnons l'occasion, ci-après, au lecteur non convaincu d'y réfléchir.

Exercice 6.4. Montrer qu'il existe une fonctionnelle Φ_e telle que :

- ▷ $\Phi_e(X', n) = X(n)$ pour tout $X \in 2^{\mathbb{N}}$ et pour tout $n \in \mathbb{N}$.
- ▷ $\Phi_e(Y, n) \downarrow$ pour tout $Y \in 2^{\mathbb{N}}$ et pour tout $n \in \mathbb{N}$. ◇

Notons enfin que le saut Turing n'est pas un opérateur injectif, comme nous le verrons par la suite à travers les ensembles low et high. Nous utiliserons de temps à autre la notion suivante de Turing-complétude.

Définition 6.5. Un ensemble A est dit *Turing-complet* ou (simplement) *complet* si $A \geq_T \emptyset'$. Un degré Turing *complet* est un degré $\mathbf{d} \geq 0'$. Un ensemble ou degré qui n'est pas complet est *incomplet*. ◇

7. Calculabilité à la limite

Nous allons maintenant étudier certaines propriétés des ensembles calculables par le problème de l'arrêt. Ces ensembles admettent notamment une caractérisation très naturelle en termes d'approximations.

Définition 7.1. Une fonction $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ est *stable* si pour tout $x \in \mathbb{N}$, $\lim_y f(x, y)$ existe. Un ensemble A est *calculable à la limite* s'il existe une fonction stable calculable $f : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ telle que pour tout x

$$\lim_y f(x, y) = 1 \Leftrightarrow x \in A. \quad \diamond$$

Lemme 7.2 (Lemme de limite de Shoenfield). Un ensemble $A \subseteq \mathbb{N}$ est $\emptyset'$ -calculable si, et seulement si, il est calculable à la limite. ★

PREUVE. On pourra se reporter aux figures 7.4 et 7.3 pour s'aider dans la compréhension de la preuve.

$\Rightarrow$. Supposons que $A \leq_T \emptyset'$ via une fonctionnelle Φ_e . Soit $\emptyset'_0 \subseteq \emptyset'_1 \subseteq \dots$ une approximation c. e. de $\emptyset'$, et soit $f : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ la fonction qui pour une entrée (x, s) regarde si $\Phi_e(\emptyset'_s, x)[s] \downarrow$. Si tel est le cas, $f(x, s) = \Phi_e(\emptyset'_s, x)[s]$. Sinon, $f(x, s)$ se voit attribuer une valeur arbitraire.

Montrons que f est stable et que sa limite est A . Soit $x \in \mathbb{N}$; par la propriété de l'usage, comme $\Phi_e(\emptyset'_s, x) \downarrow$, ce calcul est effectué en utilisant les n premiers bits de l'oracle pour un certain entier n . Soit alors s tel que $\emptyset'_s \upharpoonright_n = \emptyset' \upharpoonright_n$ et tel que $\Phi_e(\emptyset'_s, x) \downarrow$ s'arrête après s étapes de calcul (il suffit de prendre s suffisamment grand). Alors, pour tout $t \geq s$,

$$\Phi_e(\emptyset'_t, x)[t] \downarrow = \Phi_e(\emptyset'_s, x),$$

auquel cas $\lim_t f(x, t) = \Phi_e(\emptyset', x) = A(x)$.

$\Leftarrow$. Supposons maintenant que A est calculable à la limite, par une fonction stable calculable $f : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$. Alors,

$$A = \{x : \exists y \forall z \geq y \ f(x, z) = 1\} \quad \text{et} \quad \overline{A} = \{x : \exists y \forall z \geq y \ f(x, z) = 0\}.$$

Nous allons définir une procédure $\emptyset'$ -calculable pour déterminer si $x \in A$ ou $x \in \overline{A}$. Soient $u, v : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ deux fonctions totales calculables telles que pour tous x, y, n ,

$$\begin{aligned} \Phi_{u(x,y)}(n) &= \begin{cases} 1 & \text{si } \exists z \geq y \ f(x, z) \neq 1 \\ \uparrow & \text{sinon,} \end{cases} \\ \Phi_{v(x,y)}(n) &= \begin{cases} 1 & \text{si } \exists z \geq y \ f(x, z) \neq 0 \\ \uparrow & \text{sinon.} \end{cases} \end{aligned}$$

En particulier,

$$u(x, y) \notin \emptyset' \text{ ssi } \forall z \geq y \ f(x, z) = 1 \quad \text{et} \quad v(x, y) \notin \emptyset' \text{ ssi } \forall z \geq y \ f(x, z) = 0.$$

Ainsi, $A = \{x : \exists y \ u(x, y) \notin \emptyset'\}$ et $\overline{A} = \{x : \exists y \ v(x, y) \notin \emptyset'\}$.

Étant donné un entier x , pour savoir si $x \in A$ ou $x \in \overline{A}$, il suffit de chercher le plus petit y tel que $u(x, y) \notin \emptyset'$ ou $v(x, y) \notin \emptyset'$. On finira nécessairement par trouver un tel entier y . Si $u(x, y) \notin \emptyset'$, alors $x \in A$. Si $v(x, y) \notin \emptyset'$, alors $x \notin A$. La procédure est $\emptyset'$ -calculable, et l'on a ainsi $A \leq_T \emptyset'$. ■

Approximation du bit numéro x	Codes de machines
0	► $u(x, 0)$ et $v(x, 0)$
1	► $u(x, 1)$ et $v(x, 1)$
0	► $u(x, 2)$ et $v(x, 2)$
0	► $u(x, 3)$ et $v(x, 3)$
1	► $u(x, 4)$ et $v(x, 4)$
0	► $u(x, 5)$ et $v(x, 5)$
0	► $u(x, 6)$ et $v(x, 6)$
...	

FIGURE 7.3 – *Illustration de la preuve de $\Leftarrow$ du lemme de Shoenfield : étant donné l'approximation d'un bit, on crée à l'étape y le code $u(x, y)$ du programme qui s'arrête partout si une valeur différente de 1 est prise par le bit x à une étape plus grande que y , et le code $v(x, y)$ du programme qui s'arrête partout si une valeur différente de 0 est prise par le bit x à une étape plus grande que y .*

Exercice 7.5. Montrer que si Y est X -c. e. et Z est Y -c. e., alors Z n'est pas nécessairement X -c. e. ◇

Tout ensemble $\emptyset'$ -calculable A se voit donc associer une fonction stable f dont la limite est A . On présente souvent cette fonction sous la forme d'une succession d'ensembles *uniformément calculables* $A_0, A_1, \dots$ définie par $A_y = \{x : f(x, y) = 1\}$ pour tout y .

La calculabilité uniforme

Nous avons introduit le concept de suite $A_0, A_1, \dots$ *uniformément calculable* : chaque élément A_n de la suite est calculable par la même fonction, paramétrée par un paramètre supplémentaire n qui indique que l'on calcule le n -ième élément de la suite.

Insistons sur le fait qu'une suite d'ensembles calculables $(X_i)_{i \in \mathbb{N}}$ n'est pas nécessairement uniformément calculable : il n'existe pas forcément d'algorithme permettant de calculer X_i en fonction de i . À titre d'exemple trivial, chaque X_i peut simplement être une suite infinie de 0, à l'exception du bit en position i qui est égal au i -ième bit de $\emptyset'$. Chaque X_i est un ensemble fini et est donc de ce fait calculable. En revanche, un algorithme permettant de calculer *uniformément* X_i en fonction de i permettrait de calculer l'arrêt, et ne peut donc exister.

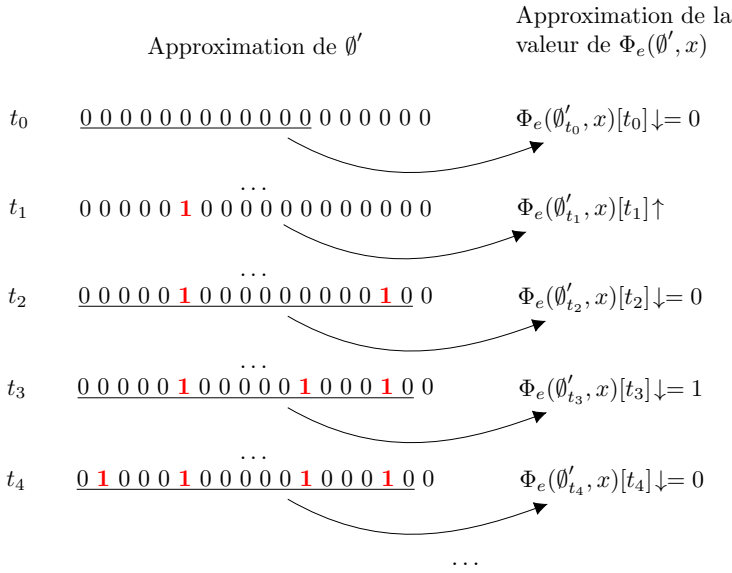


FIGURE 7.4 – Illustration de la preuve de $\Rightarrow$ du lemme de Shoenfield. La première colonne représente des approximations successives de $\emptyset'$. La partie soulignée représente l'usage du calcul $\Phi_e(\emptyset', x)[t_n]$ lorsqu'il s'arrête. À mesure que l'on énumère le i -ième élément dans l'arrêt à l'étape de calcul t_i , on lance le calcul de Φ_e avec l'approximation courante de l'arrêt comme oracle, et pour t_i étapes de calcul. Par la propriété de l'usage, le processus converge nécessairement.

Définition 7.6. Soit $A \leq_T \emptyset'$ un ensemble. Une approximation Δ_2^0 de A est une suite uniformément calculable d'ensembles $A_0, A_1, \dots$ telle que pour tout x , $\lim_y A_y(x)$ existe et vaut $A(x)$. ◇

Les approximations Δ_2^0 ne sont pas canoniques. Il est toujours possible par exemple d'« accélérer » une approximation Δ_2^0 $A_0, A_1, \dots$ en considérant la suite $A_{g(0)}, A_{g(1)}, \dots$ pour une fonction calculable et strictement croissante $g : \mathbb{N} \rightarrow \mathbb{N}$.

Remarque

L'appellation « approximation Δ_2^0 » sera justifiée dans le chapitre 5, où l'on donnera une nouvelle caractérisation des ensembles $\emptyset'$ -calculables comme ceux définissables par un prédicat Δ_2^0 .

Les approximations Δ_2^0 permettent de définir deux fonctions importantes, à savoir le modulus et la fonction de calcul. Ces fonctions expriment la com-

plexité calculatoire de l'ensemble A sous forme de rapidité de croissance : toute fonction croissant plus vite que le modulus de A ou que sa fonction de calcul permet de recalculer A .

Définition 7.7. Soit $A_0, A_1, A_2, \dots$ une approximation Δ_2^0 d'un ensemble A .

1. Le *modulus* de l'approximation Δ_2^0 est la fonction $\mu_A : \mathbb{N} \rightarrow \mathbb{N}$ qui à x associe le plus petit entier n tel que la suite $A_n \upharpoonright_x, A_{n+1} \upharpoonright_x, \dots$ soit constante.
2. La *fonction de calcul* de l'approximation Δ_2^0 est la fonction $c_A : \mathbb{N} \rightarrow \mathbb{N}$ qui à x associe le plus petit entier $n \geq x$ tel que $A_n \upharpoonright_x = A \upharpoonright_x$. $\diamond$

Contrairement aux approximations c.e qui, lorsqu'elles font apparaître un élément dans A , ne le retirent plus jamais, une approximation Δ_2^0 de A a le droit de « changer d'avis » un nombre arbitrairement grand (mais fini) de fois sur l'appartenance d'un élément x à A . En particulier, la fonction de calcul croît en général plus lentement que le modulus, car un ensemble A_n peut coïncider avec l'ensemble A sur un long segment initial sans pour autant avoir atteint son seuil de stabilité sur ce segment. La fonction de calcul, contrairement au modulus en général, est calculable par l'ensemble A .

Il est facile de vérifier que toute fonction dominant le modulus d'une approximation Δ_2^0 d'un ensemble calcule cet ensemble.

Exercice 7.8. Soit $A_0, A_1, \dots$ une approximation Δ_2^0 d'un ensemble A , et soit μ_A son modulus. Montrer que toute fonction dominant μ_A calcule A . Une fonction f domine une fonction g si $f(n) \geq g(n)$ pour tout $n \in \mathbb{N}$. $\diamond$

Il n'est cependant pas clair que cela soit également le cas de la fonction de calcul. C'est pourtant ce que montre la proposition suivante.

Proposition 7.9 (Martin et Miller [157]). Soit $A_0, A_1, \dots$ une approximation Δ_2^0 d'un ensemble A . Soit c_A sa fonction de calcul. Toute fonction dominant c_A calcule A . $\star$

PREUVE. Soit f une fonction dominant c_A . Soit $M(x)$ le plus grand $y \leq x$ tel que pour tout $x \leq t \leq f(x)$, $A_t \upharpoonright_y = A_{f(x)} \upharpoonright_y$. La fonction M est totale f -calculable. De plus, M tend vers $+\infty$, car l'approximation de A étant Δ_2^0 , elle va se stabiliser sur des segments initiaux de plus en plus grands. Enfin, comme $x \leq c_A(x) \leq f(x)$, alors si $M(x) = y$, $A_x \upharpoonright_y = A_{c_A(x)} \upharpoonright_y = A \upharpoonright_y$. Ainsi, pour décider si $n \in A$, il suffit de trouver un entier x tel que $M(x) > n$, puis tester si $n \in A_x$. Cette procédure est f -calculable. $\blacksquare$

Notons qu'il est important de demander que $c_A(x) \geq x$ dans la définition de fonction de calcul. La proposition précédente devient fausse lorsque l'on omet cette précision.

Remarque

Les notions de modulus et de fonction de calcul ne sont pas caractéristiques d'un ensemble $\emptyset'$ -calculable, mais d'une approximation Δ_2^0 d'un ensemble.

Un même ensemble $\emptyset'$ -calculable possède une infinité d'approximations Δ_2^0 , chacune ayant son modulus et sa fonction de calcul.

8. Méthode des extensions finies

Nous allons maintenant présenter une méthode relativement simple et pourtant très puissante, permettant de créer des ensembles satisfaisant des propriétés calculatoires « sur mesure ». Il s'agit de la *méthode des extensions finies*.

En calculabilité, on appelle *propriété de faiblesse* une propriété close par le bas dans les degrés Turing, c'est-à-dire que si X a une propriété de faiblesse et si $Y \leq_T X$, alors Y a aussi cette propriété de faiblesse. À l'inverse, une *propriété de force* est une propriété close par le haut dans les degrés Turing, c'est-à-dire que si X a une propriété de force et si $X \leq_T Y$, alors Y a aussi cette propriété de force.

La méthode des extensions finies est particulièrement adaptée lorsque l'on se pose la question de l'existence d'ensembles satisfaisant simultanément des propriétés de force et de faiblesse. Il faut alors créer un ensemble sur mesure, ni trop fort, ni trop faible du point de vue calculatoire. Nous allons illustrer cette méthode en prouvant deux propositions.

Proposition 8.1 (Kleene et Post, 1954). Il existe deux ensembles A et B incomparables par la réduction de Turing. ★

PREUVE. Nous allons construire simultanément deux ensembles A et B , vus comme des suites binaires infinies — souvenons-nous de la correspondance entre les deux.

L'ensemble A doit satisfaire une propriété de force (A n'est pas calculable par B) et une propriété de faiblesse (A ne calcule pas B). L'ensemble B doit de son côté satisfaire les propriétés duales de force et de faiblesse.

Contrats. Les propriétés calculatoires, qu'elles soient de force ou de faiblesse, sont en général des schémas de propriétés, au sens où elles se déclinent en une infinité de propriétés plus élémentaires et plus faciles à satisfaire de manière indépendante. Par exemple, la propriété « $A \not\leq_T B$ » correspond à la collection de propriétés « $\Phi_e^A \neq B$ » pour tout code de fonctionnelle e , où « $\Phi_e^A \neq B$ » signifie que soit Φ_e^A est une fonction partielle, soit $\Phi_e^A(x) \downarrow \neq B(x)$ pour un $x \in \mathbb{N}$. On appelle ces propriétés

élémentaires des *contrats* (ou *requirements* en anglais). La première étape d'une construction par méthode des extensions finies consiste à identifier les contrats. Nous en avons donc de deux sortes $(\mathcal{R}_e)_{e \in \mathbb{N}}$ et $(\mathcal{S}_e)_{e \in \mathbb{N}}$:

$$\begin{aligned} \mathcal{R}_e & : \quad \exists x \Phi_e^A(x) \uparrow \vee \exists x \Phi_e^A(x) \downarrow \neq B(x) \\ \mathcal{S}_e & : \quad \exists x \Phi_e^B(x) \uparrow \vee \exists x \Phi_e^B(x) \downarrow \neq A(x). \end{aligned}$$

Si tous les contrats $\mathcal{R}_e$ sont satisfaits, alors $A \not\geq_T B$, tandis que si tous les contrats $\mathcal{S}_e$ le sont, $B \not\geq_T A$. Étant donné que l'on ne s'intéresse qu'aux fonctionnelles calculant des éléments de $2^{\mathbb{N}}$, on considérera — comme souvent dans ce genre de cas — que $\Phi_e(X, n) \uparrow$ si jamais $\Phi_e(X, n) \downarrow \notin \{0, 1\}$.

Satisfaction d'un contrat. Supposons que l'on ne veuille satisfaire qu'un seul contrat, disons $\mathcal{R}_e$. Deux cas se présentent.

- ▷ Cas 1. Il existe un ensemble X tel que $\Phi_e^X(0) \downarrow = i$ pour un $i \in \{0, 1\}$ donné. On peut alors fixer $A = X$ et B peut être n'importe quel ensemble tel que $B(0) \neq i$. On aura alors satisfait le contrat $\mathcal{R}_e$ en s'assurant que $\Phi_e^A(0) \downarrow \neq B(0)$.
- ▷ Cas 2. Quel que soit l'ensemble X , on a $\Phi_e^X(0) \uparrow$. Ce cas-ci est encore plus simple, A et B peuvent être n'importe quels ensembles.

L'argument précédent a entièrement spécifié les ensembles A et B pour satisfaire un contrat $\mathcal{R}_e$, sans laisser de liberté aux autres contrats pour être satisfaits. Nous allons donc tenter d'être plus économes pour laisser de la place à la satisfaction des autres contrats. Pour cela, nous allons faire en sorte de ne spécifier qu'un segment initial fini de A et B pour satisfaire un contrat donné.

Construction. Les ensembles A et B vont être construits par approximations finies, sous forme de deux suites de chaînes binaires finies, représentant des préfixes de plus en plus longs de A et B .

$$\sigma_0 \preceq \sigma_1 \preceq \dots \quad \text{et} \quad \tau_0 \preceq \tau_1 \preceq \dots$$

On ne demande pas nécessairement à ce que $\sigma_n \prec \sigma_{n+1}$: la suite de chaînes peut stagner pendant un certain temps. En revanche, pour tout n , on demande que pour $m > n$ suffisamment grand on ait $\sigma_n \prec \sigma_m$. De cette manière, les chaînes $\sigma_0 \preceq \sigma_1 \preceq \dots$ convergent petit à petit vers une unique suite infinie A et les chaînes $\tau_0 \preceq \tau_1 \preceq \dots$ convergent petit à petit vers une unique suite infinie B . Formellement, on définit A et B via de nouvelles notations : étant donné une chaîne binaire σ , on notera $[\sigma]$ l'ensemble des suites binaires infinies ayant σ comme préfixe. On aura donc $A \in [\sigma_n]$ et $B \in [\tau_n]$ pour tout $n \in \mathbb{N}$. En s'assurant qu'il existe des chaînes de longueur arbitrairement grande, on fait en sorte que $\bigcap_n [\sigma_n]$ et $\bigcap_n [\tau_n]$ contiennent chacun exactement un élément : A et B respectivement.

Les approximations finies de A et B vont être définies par étapes, de manière à satisfaire successivement chaque contrat. Comme à une étape donnée, seuls des préfixes finis de A et B sont connus, il va donc falloir satisfaire un contrat quel que soit ce qui viendra après ces préfixes dans la suite de la construction. Une paire de chaînes σ_n et τ_n *force* un contrat $\mathcal{R}_e$ (ou un contrat $\mathcal{S}_e$) si la propriété du contrat est satisfaite pour tous $A \succeq \sigma_n$ et $B \succeq \tau_n$. On doit donc s'assurer que le contrat est satisfait pour tous les éléments de $[\sigma_n]$ et de $[\tau_n]$. Notons au passage que si σ_n et τ_n forcent un contrat, alors pour tout $\sigma \succeq \sigma_n$ et $\tau \succeq \tau_n$ on a $[\sigma] \subseteq [\sigma_n]$ et $[\tau] \subseteq [\tau_n]$, et donc σ et τ forcent encore le contrat.

Les différents contrats vont être entrelacés afin que chacun reçoive l'attention à une étape de la construction. Lors d'une étape paire $n = 2e$, on définira σ_{n+1} et τ_{n+1} de manière à forcer $\mathcal{R}_e$, tandis que lors d'une étape impaire, $n = 2e + 1$, on forcera $\mathcal{S}_e$. Les contrats seront donc satisfaits selon l'ordre

$$\mathcal{R}_0, \mathcal{S}_0, \mathcal{R}_1, \mathcal{S}_1, \mathcal{R}_2, \mathcal{S}_2, \dots$$

Satisfaction d'un contrat. Nous allons maintenant à nouveau satisfaire le contrat $\mathcal{R}_e$ en ne spécifiant cette fois-ci qu'un préfixe fini des oracles A et B . Le cas des contrats $\mathcal{S}_e$ est symétrique. Supposons que des segments initiaux σ_n et τ_n ont déjà été spécifiés pour A et B (on commence au départ la construction avec $\sigma_0 = \tau_0 = \epsilon$ où ϵ est le mot vide). Autrement dit, les suites binaires infinies finales A et B devront respecter $\sigma_n \preceq A$ et $\tau_n \preceq B$. Soit $x = |\tau_n|$. En particulier, x est la première position sur laquelle B n'est pas encore spécifié. Toutes les valeurs de B aux positions précédant x sont déjà fixées par τ_n . Les deux cas suivants se présentent.

- ▷ Cas 1. Il existe un ensemble $X \succeq \sigma_n$, tel que $\Phi_e^X(x) \downarrow = i$ pour un i dans $\{0, 1\}$ donné. Dans ce cas, par la propriété de l'usage, ce calcul fait appel à un nombre fini de bits de l'oracle, et il existe donc un segment initial $\sigma_{n+1} \preceq X$ tel que $\Phi_e^{\sigma_{n+1}}(x) \downarrow = i$, ou autrement dit tel que $\Phi_e^Y(x) \downarrow = i$ pour tout ensemble $Y \succeq \sigma_{n+1}$. On peut choisir le segment initial σ_{n+1} de manière à ce qu'il soit au moins aussi long que σ_n , ce qui assure $\sigma_{n+1} \succeq \sigma_n$. Sachant que l'ensemble A aura σ_{n+1} pour segment initial, on s'est assuré que $\Phi_e^A(x) \downarrow = i$. Soit τ_{n+1} la chaîne obtenue à partir de τ_n en lui ajoutant le bit $1 - i$. Autrement dit, $|\tau_{n+1}| = |\tau_n| + 1$, $\tau_{n+1} \succeq \tau_n$ et $\tau_{n+1}(x) = 1 - i$. On s'est alors assuré que pour tout $B \succeq \tau_{n+1}$, $B(x) = 1 - i$. Ainsi, les chaînes σ_{n+1} et τ_{n+1} étendent respectivement σ_n et τ_n , et forcent le contrat $\mathcal{R}_e$ en s'assurant que $\Phi_e^A(x) \downarrow \neq B(x)$ pour tous $A \succeq \sigma_n$ et $B \succeq \tau_n$.
- ▷ Cas 2. Pour tout ensemble $X \succeq \sigma_n$, on a $\Phi_e^X(x) \uparrow$. Dans ce cas, σ_n et τ_n forcent déjà le contrat $\mathcal{R}_e$ en s'assurant $\Phi_e^A(x) \uparrow$ pour un certain x . Il suffit donc de prendre $\sigma_{n+1} = \sigma_n$ et $\tau_{n+1} = \tau_n$.

On s'assurera que les longueurs des éléments des suites $(\sigma_n)_{n \in \mathbb{N}}$ et $(\tau_n)_{n \in \mathbb{N}}$ soient de plus en plus grandes, en ajoutant un bit arbitraire au bout de chaque chaîne à la fin de chaque étape, avant de passer à l'étape suivante. Comme expliqué précédemment, le fait de forcer un contrat est clos par extension de chaînes, et cela n'altérera donc pas la validité de la construction. La preuve de la proposition 8.1 est terminée. ■

Une nouvelle notation a été introduite dans la preuve précédente ; nous en officialisons ci-après l'utilisation.

Notation

Étant donné $\sigma \in 2^{<\mathbb{N}}$, on note $[\sigma]$ l'ensemble des $X \in 2^{\mathbb{N}}$ tels que $\sigma \prec X$.

Nous allons donner une autre illustration de la méthode des extensions finies en prouvant une proposition plus forte, impliquant la proposition 8.1. On fixe cette fois-ci un ensemble A , arbitraire en dehors du fait qu'on le suppose non calculable. On construit alors un ensemble B que A ne calcule pas, et qui ne calcule pas A . Il s'agit d'une construction *a priori* plus difficile, car on ne contrôle plus qu'un seul des deux ensembles. Notons par ailleurs qu'il sera nécessaire pour cette preuve d'utiliser le fait que A est non calculable : en effet, dans le cas inverse, tout ensemble B que l'on construit permettrait de calculer A .

Proposition 8.2. Pour tout ensemble non calculable A , il existe un ensemble B tel que $B \not\leq_T A$ et $A \not\leq_T B$. ★

PREUVE. Contrairement à la proposition 8.1, l'ensemble A est déjà fixé. Nous allons construire uniquement l'ensemble B par la méthode des approximations finies. L'ensemble B doit encore satisfaire une propriété de force ($B \not\leq_T A$) et une propriété de faiblesse ($A \not\leq_T B$). Les contrats sont donc identiques à ceux de la proposition 8.1, à savoir

$$\begin{aligned} \mathcal{R}_e & : \exists x \Phi_e^A(x) \uparrow \vee \exists x \Phi_e^A(x) \downarrow \neq B(x) \\ \mathcal{S}_e & : \exists x \Phi_e^B(x) \uparrow \vee \exists x \Phi_e^B(x) \downarrow \neq A(x). \end{aligned}$$

Cependant, les ensembles A et B ne jouant plus un rôle symétrique, les contrats $\mathcal{R}_e$ et $\mathcal{S}_e$ vont être satisfaits chacun à leur manière.

Construction. L'ensemble B va être construit par approximations successives

$$\tau_0 \preceq \tau_1 \preceq \tau_2 \preceq \dots$$

pour définir B comme unique élément de l'ensemble $\bigcap_n [\tau_n]$. Une chaîne τ_n force un contrat $\mathcal{R}_e$ (ou un contrat $\mathcal{S}_e$) si la propriété est satisfaite pour tout $X \in [\tau_n]$. À chaque étape de la construction, un contrat va être forcé,

en les entrelaçant comme précédemment :

$$\mathcal{R}_0, \mathcal{S}_0, \mathcal{R}_1, \mathcal{S}_1, \mathcal{R}_2, \mathcal{S}_2, \dots$$

Satisfaction d'un contrat $\mathcal{R}_e$. À l'étape n , supposons que la chaîne τ_n est définie. Nous voulons trouver une extension $\tau_{n+1} \succeq \tau_n$ forçant le contrat $\mathcal{R}_e$. La satisfaction de ce type de contrat est très similaire à celle de la proposition 8.1. Soit $x = |\tau_n|$. Deux cas se présentent.

- ▷ Cas 1. On a $\Phi_e^A(x) \downarrow = i$ pour un $i \in \{0, 1\}$. Il suffit alors de définir τ_{n+1} comme l'unique chaîne de longueur $|\tau_n| + 1$ étendant τ_n telle que

$$\tau_{n+1}(x) = 1 - i.$$

Comme $B \in [\tau_{n+1}]$, $B(x) = 1 - i$, donc $\Phi_e^A(x) \downarrow \neq B(x)$.

- ▷ Cas 2. On a $\Phi_e^A(x) \uparrow$. Le contrat $\mathcal{R}_e$ est alors trivialement satisfait, car Φ_e^A est une fonction partielle. Il suffit donc de prendre $\tau_{n+1} = \tau_n$.

On notera que l'on n'a fait aucune supposition sur l'ensemble A pour satisfaire les contrats $\mathcal{R}_e$. L'hypothèse selon laquelle A n'est pas calculable sera exploitée pour satisfaire les contrats $\mathcal{S}_e$.

Satisfaction d'un contrat $\mathcal{S}_e$. La difficulté de la satisfaction d'un contrat tel que $\mathcal{S}_e$ provient du fait que l'on n'a pas de contrôle sur l'ensemble A qui est entièrement spécifié. Plus précisément, lors de la satisfaction d'un contrat $\mathcal{R}_e$, on fixe une entrée x qui n'est pas encore spécifiée pour B , de manière à choisir sa valeur dans le cas 1 pour la rendre différente de la valeur de $\Phi_e^A(x)$. Ce n'est pas possible dans le cas du contrat $\mathcal{S}_e$, toutes les valeurs de A étant fixées. Il va donc falloir exploiter le fait que l'ensemble A n'est pas calculable. Notons en revanche que la satisfaction des contrats $\mathcal{R}_e$ laissait une certaine liberté, notamment dans le choix de x . En effet, seul un nombre fini d'entrées est spécifié à une étape donnée pour B , et donc la quasi-totalité des entrées peut être choisie pour x . Nous allons exploiter cette liberté de choix pour satisfaire les contrats $\mathcal{S}_e$. Trois cas se présentent.

- ▷ Cas 1. Il existe une entrée x et un ensemble $X \succeq \tau_n$ tels que

$$\Phi_e^X(x) \downarrow \neq A(x).$$

Par la propriété de l'usage, il existe alors une chaîne finie $\tau_{n+1} \succeq \tau_n$ telle que $\Phi_e^X(x) \downarrow \neq A(x)$ pour tout $X \in [\tau_{n+1}]$. La chaîne τ_{n+1} force donc le contrat $\mathcal{S}_e$.

- ▷ Cas 2. Il existe une entrée x telle que pour tous les ensembles $X \succeq \tau_n$, on a $\Phi_e^X(x) \uparrow$. Dans ce cas, la chaîne τ_n force déjà le contrat $\mathcal{S}_e$ en s'assurant que $\Phi_e^B(x) \uparrow$.
- ▷ Cas 3. Aucun des deux cas précédents ne se présente. Nous allons montrer alors qu'il est possible de calculer l'ensemble A , et donc d'en déduire une contradiction.

Voici la procédure pour calculer la valeur de $A(x)$: chercher une chaîne finie $\tau \succeq \tau_n$ telle que $\Phi_e^\tau(x) \downarrow$ et renvoyer le résultat de ce calcul. Nous prétendons les deux faits suivants :

- (1) il existe une telle chaîne, et donc que la recherche se terminera ;
- (2) quelle que soit τ telle que $\Phi_e^\tau(x) \downarrow$, alors $\Phi_e^\tau(x) \downarrow = A(x)$.

Pour montrer (1), remarquons que la négation du cas 2 signifie que pour tout x (et en particulier pour ce x considéré), il existe un ensemble $X \succeq \tau_n$ tel que $\Phi_e^X(x) \downarrow$. Par la propriété de l'usage, il existe alors un segment initial $\tau \succeq \tau_n$ de X tel que $\Phi_e^\tau(x) \downarrow$.

Montrons (2). Si $\Phi_e^\tau(x) \downarrow \neq A(x)$, alors le cas 1 serait vrai en prenant n'importe quel $X \succeq \tau$. Il s'ensuit que $\Phi_e^\tau(x) \downarrow$ implique $\Phi_e^\tau(x) = A(x)$. Nous avons donc décrit une procédure calculable pour déterminer la valeur de $A(x)$ quel que soit x , contredisant l'hypothèse selon laquelle A n'est pas calculable.

Cela conclut la preuve de la proposition 8.2. ■

Avant de conclure cette section dédiée à la méthode des extensions finies, mentionnons que de manière générale, cette méthode n'est pas *effective*, au sens où aucune contrainte de calculabilité n'est imposée à la suite des chaînes finies en construction. Il est cependant possible de faire une analyse fine de l'argument pour déterminer la puissance calculatoire nécessaire pour trouver un τ_{n+1} , étant donné τ_n . On obtient alors des bornes supérieures sur la complexité de l'ensemble construit.

9. Degrés low

Comme nous l'avons vu, le saut Turing est invariant par degré Turing. Ainsi, pour tout ensemble calculable X , $X' \equiv_T \emptyset'$. Il est naturel de se demander si seuls les ensembles calculables ont un saut Turing équivalent à $\emptyset'$, et plus généralement si le saut Turing est une fonction injective sur les degrés Turing. La proposition suivante montre que ce n'est pas le cas.

Proposition 9.1. Il existe un ensemble non calculable A tel que

$$A' \equiv_T \emptyset'.$$

★

PREUVE. L'ensemble A va être construit à l'aide d'une version effective de la méthode des extensions finies (voir la section 8), en s'assurant que l'intégralité de la construction est calculable en $\emptyset'$.

Contrats. L'ensemble A doit satisfaire une propriété de force (A non calculable) et une propriété de faiblesse ($A' \leq_T \emptyset'$).

La propriété de force se décline en une infinité de contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$:

$$\mathcal{R}_e : \exists x \Phi_e(x) \uparrow \vee \exists x \Phi_e(x) \downarrow \neq A(x).$$

La propriété de faiblesse est d'un type nouveau. Pour la satisfaire, nous allons faire en sorte de « contrôler » le saut Turing de A au fur et à mesure de la construction, tout en s'assurant que toute la construction elle-même est calculable en $\emptyset'$. En s'aidant de ce fait, on aura une fonction $\emptyset'$ -calculable f pour laquelle on devra satisfaire une infinité de contrats $(\mathcal{S}_e)_{e \in \mathbb{N}}$:

$$\mathcal{S}_e : \Phi_e^A(e) \downarrow \rightarrow f(e) = 1 \text{ et } \Phi_e^A(e) \uparrow \rightarrow f(e) = 0.$$

Informellement, le contrat $\mathcal{S}_e$ est satisfait si, à un moment fini de la construction, nous savons si $\Phi_e^A(e) \downarrow$ ou $\Phi_e^A(e) \uparrow$, quelle que soit la suite de la construction.

Construction. L'ensemble A va être construit par approximations successives

$$\sigma_0 \preceq \sigma_1 \preceq \sigma_2 \preceq \dots$$

pour définir A comme unique élément de $\bigcap_n [\sigma_n]$. Une chaîne σ_n *force* un contrat $\mathcal{R}_e$ ou $\mathcal{S}_e$ si la propriété est satisfaite pour tout $B \in [\sigma_n]$. À chaque étape de la construction, un contrat va être forcé, en les entrelaçant comme précédemment :

$$\mathcal{R}_0, \mathcal{S}_0, \mathcal{R}_1, \mathcal{S}_1, \mathcal{R}_2, \mathcal{S}_2, \dots$$

Nous devons de plus nous assurer que la construction est calculable en $\emptyset'$. Ainsi, pour connaître la valeur de $A'(e)$, il suffira d'exécuter à l'aide de $\emptyset'$ la construction jusqu'à l'étape $2e$ satisfaisant $\mathcal{S}_e$, et de renvoyer le résultat. Cette procédure $\emptyset'$ -calculable assure que $A' \leq_T \emptyset'$. Nous allons donc montrer comment satisfaire chaque type de contrat indépendamment, tout en analysant la complexité calculatoire de chaque étape pour s'assurer que σ_{n+1} peut être obtenu à partir de σ_n à l'aide de l'oracle $\emptyset'$.

Satisfaction d'un contrat $\mathcal{R}_e$. Supposons que σ_n est déjà défini. Nous voulons trouver $\sigma_{n+1} \succeq \sigma_n$ forçant le contrat $\mathcal{R}_e$. Soit $x = |\sigma_n|$; autrement dit, x est la première valeur de A qui n'est pas encore spécifiée. Deux cas se présentent.

- ▷ Cas 1. On a $\Phi_e(x) \uparrow$, auquel cas $\mathcal{R}_e$ est trivialement satisfait, et l'on peut définir $\sigma_{n+1} = \sigma_n$
- ▷ Cas 2. On a $\Phi_e(x) \downarrow$, auquel cas la chaîne σ_{n+1} obtenue à partir de σ_n en lui ajoutant le bit $1 - \Phi_e(x)$ force $\mathcal{R}_e$.

En fonction du cas, nous allons donc définir une extension σ_{n+1} différente. Il faut s'assurer que cette extension peut être obtenue calculatoirement à l'aide de l'oracle $\emptyset'$. La distinction entre les deux cas n'est pas calculable en soi, car elle demande de décider si $\Phi_e(x)$ s'arrête ou non. Nous pouvons cependant utiliser $\emptyset'$ pour répondre à cette question comme suit : soit Φ_i

la fonction partielle calculable définie pour tout n par $\Phi_i(n) = \Phi_e(x)$. Le code i peut être construit calculatoirement, car il s'agit d'une simple manipulation de machine. Il s'ensuit que l'on est dans le premier cas ssi $i \notin \emptyset'$. Dans le premier cas, $\sigma_{n+1} = \sigma_n$ est trivialement calculable, tandis que dans le second cas, il suffit d'exécuter $\Phi_e(x)$ pour récupérer la valeur renvoyée, et obtenir ainsi σ_{n+1} . Nous pouvons donc trouver une extension σ_{n+1} forçant le contrat $\mathcal{R}_e$ à l'aide de l'oracle $\emptyset'$.

Satisfaction d'un contrat $\mathcal{S}_e$. Supposons que σ_n est déjà défini. Nous voulons trouver $\sigma_{n+1} \succeq \sigma_n$ tel que le comportement de $\Phi_e^A(e)$ est déjà défini par σ_{n+1} , autrement dit $\Phi_e^X(e) \downarrow$ pour tout $X \in [\sigma_{n+1}]$ ou $\Phi_e^X(e) \uparrow$ pour tout $X \in [\sigma_{n+1}]$. Deux cas se présentent encore.

- ▷ Cas 1. Il existe une chaîne $\tau \succeq \sigma_n$ telle que $\Phi_e^\tau(e) \downarrow$. Dans ce cas, en prenant $\sigma_{n+1} = \tau$, nous assurons que $\Phi_e^A(e) \downarrow$, car $A \in [\sigma_{n+1}]$.
- ▷ Cas 2. Pour toute chaîne $\tau \succeq \sigma_n$, $\Phi_e^\tau(e) \uparrow$. Dans ce cas, par la propriété de l'usage, quel que soit l'oracle $A \in [\sigma_n]$, $\Phi_e^A(e) \uparrow$. En définissant $\sigma_{n+1} = \sigma_n$, nous forçons donc $\Phi_e^A(e) \uparrow$.

Ainsi, dans chacun des cas, nous avons forcé le comportement de $\Phi_e^A(e)$ avec un préfixe fini de l'oracle.

Ici encore, il s'agit de trouver σ_{n+1} à partir de σ_n calculatoirement à l'aide de l'oracle $\emptyset'$. Comme pour le contrat $\mathcal{R}_e$, nous définissons une fonction partielle calculable Φ_i qui, pour chacune de ses entrées, cherche une chaîne $\tau \succeq \sigma_n$ telle que $\Phi_e^\tau(e)[|\tau|] \downarrow$, et s'arrête s'il en trouve une. Ainsi, $i \in \emptyset'$ si, et seulement si, nous sommes dans le premier cas. Une fois le cas déterminé, la chaîne σ_{n+1} peut être trouvée calculatoirement. Cela conclut la preuve de la proposition 9.1. ■

Parmi les premières notions de faiblesse introduites et étudiées en calculabilité par Cooper et Soare de manière indépendante, se trouve la hiérarchie des ensembles low_n , dont nous donnons ici le premier niveau.

Définition 9.2. Un ensemble $A \subseteq \mathbb{N}$ est *low* si $A' \leq_T \emptyset'$. ◇

Informellement, un ensemble est low s'il est indistinguable d'un ensemble calculable du point de vue du saut Turing. Nous avons vu que si $X \leq_T Y$, alors $X' \leq_T Y'$.

Ainsi, comme $\emptyset \leq_T A$ pour tout ensemble A , $\emptyset' \leq_T A'$. Il s'ensuit qu'un ensemble est low ssi $A' \equiv_T \emptyset'$. Nous verrons plus loin d'autres exemples d'ensembles low non calculables, notamment des ensembles calculatoirement énumérables (voir le chapitre 13).

10. Degrés high

Si l'on considère un ensemble $A \leq_T \emptyset'$, quelles sont les puissances extrêmes que peut prendre son saut Turing A' ?

Rappelons que si $X \leq_T Y$, alors $X' \leq_T Y'$. En particulier, $A' \geq_T \emptyset'$. D'un autre côté, $A' \leq_T \emptyset''$ car $A \leq_T \emptyset'$. On a donc

$$\emptyset' \leq_T A' \leq_T \emptyset'' \text{ si } A \leq_T \emptyset'.$$

Notons que dans le cas où $A \not\leq_T \emptyset'$, le saut Turing de A peut être arbitrairement complexe.

Les ensembles dont le saut Turing est égal à la borne minimale, à savoir $\emptyset'$, sont les ensembles low. Nous avons vu qu'il existait des ensembles low non calculables. Nous allons maintenant nous pencher sur les ensembles dont le saut Turing calcule la borne maximale $\emptyset''$.

Définition 10.1. Un ensemble $A \subseteq \mathbb{N}$ est *high* si $\emptyset'' \leq_T A'$. ◇

Remarque

La notion d'ensemble high a été historiquement définie uniquement pour les ensembles $A \leq_T \emptyset'$. Elle a depuis été étendue à tous les ensembles, mais il est important de garder cette différence historique en tête lorsqu'on lit les articles fondateurs de la calculabilité.

Réfléchissons à présent aux ensembles high. À l'inverse des low, leur saut Turing a plus de puissance de calcul qu'attendu : il permet de calculer le double saut. Le problème de l'arrêt lui-même est évidemment un exemple trivial d'ensemble high. Tout comme pour les ensembles low, la notion d'ensemble high a son intérêt pour les exemples non triviaux : les ensembles high qui ne calculent pas $\emptyset'$. Il est en fait possible de montrer que pour tout ensemble C non calculable il existe un ensemble high qui ne calcule pas C .

Dans la preuve suivante, nous utilisons pour la première fois des oracles qui sont non pas des ensembles d'entiers, mais des fonctions $f : \mathbb{N} \rightarrow \mathbb{N}$. Une telle fonction peut être représentée par la suite binaire infinie X_f telle que $X_f(\langle n, m \rangle) = 1$ ssi $f(n) = m$. Il est clair que X_f permet de calculer f , et que tout ensemble Y permettant de calculer f peut aussi calculer X_f : l'ensemble X_f est une représentation minimum de la fonction f dans les degrés Turing. D'autres représentations équivalentes en termes de degré Turing sont possibles, par exemple avec $X_f = 1^{f(0)}01^{f(1)}01^{f(2)}0\dots$ (On commence par les $f(0)$ premiers bits à 1, suivis d'un 0, puis on continue avec les $f(1)$ bits suivants à 1, suivit d'un 0, etc.).

Proposition 10.2. Pour tout ensemble non calculable C , il existe un ensemble high A tel que $A \not\geq_T C$. ★

PREUVE. Elle est assez similaire à celle de la proposition 8.2 avec la méthode des extensions finies.

Contrats. L'ensemble A doit satisfaire une propriété de force ($A' \geq_T \emptyset''$) et une propriété de faiblesse ($C \not\leq_T A$). Les contrats pour la propriété de faiblesse sont standard, à savoir

$$\mathcal{S}_e : \exists x \Phi_e^A(x) \uparrow \vee \exists x \Phi_e^A(x) \downarrow \neq C(x).$$

Le cas de la propriété de force est différent. Tout d'abord, il ne s'agit pas de contrôler ce que l'ensemble A calcule, mais de contrôler ce que son saut Turing calcule. Ensuite, cette propriété de force ne s'exprime pas sous une forme négative (ne pas se faire calculer par un autre ensemble) mais sous une forme positive (calculer un objet compliqué). Les formulations négatives se prouvent souvent par diagonalisation, tandis que les formulations positives sont plus constructives. Nous n'allons donc pas exprimer la propriété de force sous forme de contrats, mais l'imposer structurellement dans la nature de l'objet que l'on construit.

Dans les utilisations précédentes de la méthode des extensions finies, nous avons construit un ensemble en créant une suite infinie de chaînes binaires formant des approximations finies de l'ensemble.

Cette fois-ci, nous allons construire une fonction stable $f : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ dont la limite est $\emptyset''$. Comme expliqué dans le paragraphe précédant la preuve, rappelons que cela peut se ramener à l'utilisation d'un élément de $2^{\mathbb{N}}$ en considérant l'ensemble X_f défini par $\langle n, m \rangle \in X_f$ ssi $f(n) = m$. Par la relativisation du lemme de limite de Shoenfield à f (voir le lemme 7.2), un ensemble B est f -calculable à la limite ssi $B \leq_T f'$. En particulier, pour $B = \emptyset''$, si $\emptyset''$ est f -calculable à la limite, alors $\emptyset'' \leq_T f'$, autrement dit f est high. L'ensemble A est donc n'importe quel ensemble dans le degré Turing de f .

Construction. La fonction f va être construite à partir d'approximations finies successives de plus en plus précises. Ces approximations, au lieu d'être des chaînes binaires, vont être des couples (g, m) , où

- ▷ $g \subseteq \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ est une fonction partielle à deux paramètres dont le domaine est fini, représentant un morceau de la fonction f que l'on est en train de construire.
- ▷ m est un entier signifiant que désormais, lorsque l'on étendra le domaine de g avec une nouvelle entrée (x, y) , si $x < m$ alors $g(x, y) = \emptyset''(x)$.

Autrement dit, la limite des m premières « colonnes » de la fonction f est déjà atteinte et a la bonne valeur. Nous appellerons ces couples des *conditions*, car elles conditionnent une partie du comportement de la fonction f .

De la même manière que la relation de suffixe $\sigma \preceq \tau$ pour les chaînes binaires signifie que τ est une approximation plus précise de la suite que l'on construit, nous allons définir une relation d'extension sur les conditions $(g, m) \preceq (h, n)$ pour signifier que la condition (h, n) est plus précise, ou plus contraignante, que la condition (g, m) .

Étant donné une fonction partielle $h \subseteq \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$, on notera $\text{dom } h$ son domaine de définition. On dit donc que la condition (h, n) *étend* (g, m) (noté $(h, n) \succeq (g, m)$) si $n \geq m$ et si, en outre,

(P1) $g \subseteq h$, c'est-à-dire $\text{dom } g \subseteq \text{dom } h$ et pour tout $(x, y) \in \text{dom } g$,

$$g(x, y) = h(x, y);$$

(P2) pour tout $(x, y) \in \text{dom } h \setminus \text{dom } g$, si $x < m$, alors $h(x, y) = \emptyset''(x)$.

La propriété (P1) signifie que les fonctions finies doivent être compatibles, et plus précisément que la fonction h doit étendre la fonction g , tandis que la propriété (P2) formalise l'idée selon laquelle le second paramètre d'une condition fixe les colonnes de la fonction en les stabilisant. La figure 10.3 illustre ce qui vient d'être expliqué.

Arrêtons-nous un instant pour nous familiariser avec ce nouvel objet mathématique et apprendre à le manipuler. Tout d'abord, pour toute condition (g, m) et tout n , (g, n) est également une condition. Si de plus $n \geq m$, alors (g, n) est une extension. L'entier m n'impose aucune contrainte en soi sur la fonction finie g pour former une condition (g, m) . En revanche, m impose des restrictions sur les extensions de (g, m) . Plus précisément, si $n \geq m$, alors l'ensemble des extensions de (g, n) est un sous-ensemble des extensions de (g, m) . Enfin, $(\emptyset, 0)$ est une condition valide, où $\emptyset$ est la fonction définie nulle part.

Du point de vue de la calculabilité, l'ensemble des conditions est un ensemble calculable. En revanche, la relation d'extension entre deux conditions n'est pas calculable à cause de la propriété (P2) qui fait intervenir $\emptyset''$. Cependant, si l'on fixe m , alors la relation d'extension entre des conditions ayant m pour seconde composante est calculable, car cela ne fait intervenir qu'un segment fini $\emptyset'' \upharpoonright_m$ de l'ensemble $\emptyset''$. Il suffit de « coder en dur » ce segment initial dans le programme. Cette observation sera exploitée pour satisfaire les contrats $\mathcal{S}_e$.

De la même manière que l'on note $[\sigma]$ l'ensemble des suites binaires infinies ayant pour segment initial σ , on notera $[g, m]$ l'ensemble des fonctions totales $f : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ telles que $g \subseteq f$ et telles que pour

y				$m = 3$			
	$f(0, y)$	$f(1, y)$	$f(2, y)$	$f(3, y)$	$f(4, y)$	$f(5, y)$	$f(6, y)$
0	0	1	0	0	0	0	0
1	0	0	0	0	1	0	0
2	1	1	1	0	1	1	0
3	0	0	1	0	0	1	1
4	1	0	0	1	1	0	1
5	1	0	0	0	0	1	1
6	1	0	0	0	1	1	1
7	1	0	0	1	0	0	0
				$m = 3$			
				$f(3, y)$	$f(4, y)$	$f(5, y)$	$f(6, y)$
				0	1	1	0
				$\emptyset''(3)$	$\emptyset''(4)$	$\emptyset''(5)$	$\emptyset''(6)$

FIGURE 10.3 – La partie en gris foncé représente une condition avec $m = 3$. La partie en gris clair représente une extension de cette condition : chaque i -ième colonne pour $i < m$ est fixée pour toute extension à une valeur qui doit correspondre au i -ième bit du double arrêt.

tout $(x, y) \notin \text{dom } g$ avec $x < m$ on a $f(x, y) = \emptyset''(x)$. Ainsi, $[g, m]$ est la collection de l'ensemble des fonctions candidates que l'on peut obtenir en complétant l'approximation partielle (g, m) . Notons que $[g, m]$ ne contient pas que des fonctions stables.

Nous allons donc construire f par approximations successives sous forme de conditions

$$(g_0, m_0) \preceq (g_1, m_1) \preceq (g_2, m_2) \preceq \dots$$

pour définir $f = \bigcup_t g_t$. Autrement dit, $\text{dom } f = \bigcup_t \text{dom } g_t$, et pour tout couple $(x, y) \in \text{dom } f$, $f(x, y) = g_t(x, y)$ pour un t tel que $(x, y) \in \text{dom } g_t$. La fonction f est bien définie grâce à la propriété de compatibilité (P1). Si l'on s'assure que les entiers m_t deviennent arbitrairement grands, il est facile de vérifier par la propriété (P2) que la fonction f résultante est stable, et a pour limite $\emptyset''$. Notons en particulier que $f \in \bigcap_t [g_t, m_t]$.

Une condition (g, m) force un contrat $\mathcal{S}_e$ si la propriété est satisfaite pour toute $f \in [g, m]$. Ici, nous avons remplacé les occurrences de A par f dans le contrat $\mathcal{S}_e$. À chaque étape de la construction, un contrat va être forcé.

Satisfaction d'un contrat $\mathcal{S}_e$. L'argument est similaire à celui de la proposition 8.2, mais en manipulant des conditions et non des chaînes binaires. Soit (g_t, m_t) une condition. Les trois cas suivants se présentent.

- ▷ Cas 1. Il existe une entrée x et une fonction f dans $[g_t, m_t]$ telles que

$$\Phi_e^f(x) \downarrow \neq C(x).$$

Dans ce cas, par la propriété de l'usage, il existe une extension (g_{t+1}, m_t) de (g_t, m_t) telle que $\Phi_e^f(x) \downarrow \neq C(x)$ pour toute f dans $[g_{t+1}, m_t]$. Notons que $m_{t+1} = m_t$. La condition (g_{t+1}, m_t) force donc le contrat $\mathcal{S}_e$.

- ▷ Cas 2. Il existe une entrée x telle que pour toutes les fonctions f dans $[g_t, m_t]$, $\Phi_e^f(x) \uparrow$. Dans ce cas, la condition (g_t, m_t) force déjà le contrat $\mathcal{S}_e$ en s'assurant que $\Phi_e^f(x) \uparrow$.
- ▷ Cas 3. Aucun des deux cas précédents ne se présente. Nous allons montrer alors qu'il est possible de calculer l'ensemble C , et donc d'en déduire une contradiction. Voici la procédure pour calculer la valeur de $C(x)$: chercher une condition (h, m_t) étendant (g_t, m_t) avec la même seconde composante m_t , telle que $\Phi_e^h(x) \downarrow$. Nous prétendons les deux faits suivants :

(1) il existe une telle extension, et donc que la recherche se terminera ;

(2) quelle que soit $(h, m_t) \succeq (g_t, m_t)$, $\Phi_e^h(x) \downarrow \rightarrow \Phi_e^h(x) = C(x)$.

Pour montrer (1), remarquons que la négation du cas 2 signifie que pour tout x (et en particulier pour ce x considéré), il existe une fonction $f \in [g_t, m_t]$ telle que $\Phi_e^f(x) \downarrow$. Par la propriété de l'usage, il existe alors une condition $(h, m_t) \succeq (g_t, m_t)$ telle que $\Phi_e^h(x) \downarrow$.

Montrons (2). Si $\Phi_e^h(x) \downarrow \neq C(x)$, alors le cas 1 serait vrai en prenant n'importe quel $f \in [h, m_t] \subseteq [g_t, m_t]$. Il s'ensuit que $\Phi_e^h(x) \downarrow = C(x)$. Enfin, remarquons que nous n'avons considéré que des conditions avec le même m_t . Comme expliqué plus haut, pour m_t fixé, la relation d'extension est calculable.

Nous avons donc décrit une procédure calculable pour déterminer la valeur de $C(x)$ quel que soit x , contredisant l'hypothèse selon laquelle C n'est pas calculable.

Il suffit alors de satisfaire chaque contrat $\mathcal{S}_e$ en étendant petit à petit nos conditions, tout en faisant croître « artificiellement » leurs secondes composantes vers $+\infty$ afin de s'assurer que la solution finale est bien une fonction stable dont la limite est $\emptyset''$. Cela conclut la preuve de la proposition 10.2. ■

Corollaire 10.4

Il existe un ensemble A à la fois high et Turing-incomplet. Autrement dit, $A' \geq_T \emptyset''$ et $A \not\geq_T \emptyset'$.

PREUVE. Immédiat par la proposition 10.2, en prenant $C = \emptyset'$. ■

Remarque

L'appellation « condition » est un nom générique emprunté à la théorie du forcing, et qui sera amené à désigner des objets mathématiques très différents tout au long de cet ouvrage, bien que correspondant tous à l'idée d'une approximation d'un objet que l'on construit. Cette notion sera développée dans le chapitre 11.

Notons que l'argument de la preuve précédente ne dépend pas spécifiquement de $\emptyset''$ et l'on aurait pu construire pour tout B un ensemble A tel que $A' \geq_T B$ et $A \not\geq_T C$. Nous verrons dans le chapitre 13 qu'il existe des ensembles c. e. low non calculables. Sacks [187] a également montré l'existence d'ensembles c. e. incomplets de degrés high.

Il y a une différence de taille entre les ensembles low et high : les premiers sont tous calculables en $\emptyset'$, et ils sont par conséquent en quantité dénombrable. Les deuxièmes peuvent en revanche être arbitrairement complexes, et l'on montre facilement qu'ils sont en quantité indénombrable. Nous verrons en revanche avec le corollaire 19-3.9 et la proposition 10-3.38 qu'il y a « peu » d'ensembles high, du point de vue de la théorie de la mesure et de la théorie des catégories de Baire.

Terminons ce chapitre par quelques exercices.

Exercice 10.5. (*) Adapter la preuve de la proposition 10.2 pour montrer que pour tout ensemble B et tout ensemble non calculable C , il existe un ensemble A tel que $A' \geq_T B$ et $A \not\geq_T C$. ◇

Exercice 10.6. ()** Adapter la preuve de la proposition 10.2 pour montrer qu'il existe un ensemble high incomplet et $\emptyset'$ -calculable. ◇

Exercice 10.7. ()** Montrer par la méthode des extensions finies qu'il existe une suite d'ensembles $(A_n)_{n \in \mathbb{N}}$ deux à deux incomparables pour la réduction Turing, c'est-à-dire tels que

$$A_n \not\leq_T A_m \quad \text{et} \quad A_m \not\leq_T A_n,$$

pour tous $n \neq m \in \mathbb{N}$. ◇

Chapitre 5

Hiérarchie arithmétique

On introduit une hiérarchie de complexité sur les ensembles d'entiers. Les ensembles calculatoirement énumérables sont dits Σ_1^0 et leurs complémentaires sont dits Π_1^0 . Les intersections dénombrables d'ensembles Σ_1^0 sont dits Π_2^0 et les réunions dénombrables d'ensembles Π_1^0 sont dits Σ_2^0 , et ainsi de suite.

On appelle cette construction *hiérarchie arithmétique*, car les ensembles qu'elle contient sont exactement ceux qui sont définissables par une formule du premier ordre dans le langage de l'arithmétique de Peano. Nous reparlerons de cette équivalence dans la section 9-3. Il y a une correspondance directe entre la définition d'un ensemble par réunions/intersections, et le fait de pouvoir le définir par une formule comportant des quantificateurs $\exists/\forall$. Ainsi, une réunion correspond à un quantificateur $\exists$ et une intersection à un quantificateur $\forall$.

Exemple 1. L'ensemble des codes e pour les fonctions calculables totales peut s'écrire

$$\{e \in \mathbb{N} : \forall n \exists t \Phi_e(n)[t] \downarrow\} = \bigcap_n \bigcup_t \{e \in \mathbb{N} : \Phi_e(n)[t] \downarrow\}.$$

Il s'agit d'un ensemble Π_2^0 , c'est-à-dire d'une intersection dénombrable d'ensembles Σ_1^0 , chacun d'entre eux étant par ailleurs une réunion dénombrable d'ensembles calculables.

1. Propriétés élémentaires

Commençons par une définition formelle de la hiérarchie arithmétique.

Définition 1.1. Soient $m, n \geq 1$.

1. Un ensemble $A \subseteq \mathbb{N}^m$ est dit Σ_n^0 s'il existe un ensemble calculable R inclus dans $\mathbb{N}^{n+m}$ tel que

$$A = \{(y_1, \dots, y_m) : \overbrace{\exists x_1 \forall x_2 \dots Q x_n}^{n \text{ quantificateurs}} (x_1, \dots, x_n, y_1, \dots, y_m) \in R\},$$

où Q vaut $\exists$ si n est impair, et $\forall$ si n est pair.

2. Un ensemble $A \subseteq \mathbb{N}^m$ est dit Π_n^0 s'il existe un ensemble calculable R inclus dans $\mathbb{N}^{n+m}$ tel que

$$A = \{(y_1, \dots, y_m) : \overbrace{\forall x_1 \exists x_2 \dots Q x_n}^{n \text{ quantificateurs}} (x_1, \dots, x_n, y_1, \dots, y_m) \in R\},$$

où Q vaut $\forall$ si n est impair, et $\exists$ si n est pair. $\diamond$

Attention dans la définition précédente, c'est l'alternance entre les quantificateurs $\exists$ et $\forall$ qui compte.

Ainsi, par exemple, l'ensemble

$$\{y : \exists x_1 \forall x_2 R(y, x_1, x_2)\}$$

pour R calculable est un ensemble Σ_2^0 , mais l'ensemble

$$\{y : \exists x_1 \exists x_2 \exists x_3 R(y, x_1, x_2, x_3)\}$$

est, malgré ses trois quantificateurs, un ensemble Σ_1^0 : on peut facilement éliminer les répétitions de quantificateurs du même type. Considérons par exemple une formule de la forme

$$\exists x_1 \exists x_2 \forall y_1 \forall y_2 R(x_1, x_2, y_1, y_2).$$

Cette dernière pourra simplement se récrire sous la forme $\exists x \forall y R'(x, y)$ où R' sera une version de R modifiée, qui considérera x et y respectivement comme des paires $\langle x_1, x_2 \rangle$ et $\langle y_1, y_2 \rangle$. Le prédicat R' utilisera alors les projections π_1 et π_2 réalisant les fonctions inverses de la bijection calculable de couplage $\langle, \rangle : \mathbb{N}^2 \rightarrow \mathbb{N}$, afin de récupérer x_1, x_2, y_1, y_2 . Le lecteur peut revenir à l'exercice 3-2.3 pour se convaincre que la bijection $\langle, \rangle$ et que ses deux fonctions inverses sont calculables. Un abus de notation consistera par exemple à écrire $\exists \langle x_1, x_2 \rangle \forall \langle y_1, y_2 \rangle R(x_1, x_2, y_1, y_2)$, signifiant que le prédicat R se charge de récupérer x_1, x_2 à partir de $\langle x_1, x_2 \rangle$, et de même pour y_1, y_2 .

Prédicats

On parlera parfois de prédicats Σ_n^0 pour désigner une formule de la forme $\exists x_1 \forall x_2 \dots Qx_n R(x_1, x_2, \dots, x_n)$, où R est un prédicat calculable.

Notez qu'en utilisant le fait que le complémentaire d'un ensemble calculable est calculable, on voit facilement que les ensembles Σ_n^0 sont les complémentaires des ensembles Π_n^0 . Il est par ailleurs tout à fait possible pour un ensemble d'être à la fois Σ_n^0 et Π_n^0 . On introduit pour cela la notion suivante.

Définition 1.2. Soit $m \geq 1$. Un ensemble $A \subseteq \mathbb{N}^m$ est dit Δ_n^0 pour $n > 0$ s'il est à la fois Σ_n^0 et Π_n^0 . ◇

La hiérarchie arithmétique établit un premier niveau de distinction entre les ensembles arithmétiquement définissables. Intuitivement, un ensemble Σ_{n+1}^0 est strictement plus complexe qu'un ensemble Σ_n^0 ou même Π_n^0 . En effet, chacun des deux derniers peut s'écrire sous une forme Σ_{n+1}^0 en rajoutant simplement des quantificateurs inutilisés.

Exemple 1.3. L'ensemble Σ_2^0 donné par $\{y : \exists x_1 \forall x_2 R(y, x_1, x_2)\}$ peut également s'écrire sous une forme $\Pi_3^0 : \{y : \forall z \exists x_1 \forall x_2 R(y, x_1, x_2)\}$ ou sous une forme $\Sigma_3^0 : \{y : \exists x_1 \forall x_2 \exists z R(y, x_1, x_2)\}$.

En revanche, il n'est pas toujours possible d'utiliser moins de quantificateurs. Nous allons montrer avec le corollaire 5.6 qu'il existe pour tout $n > 0$ des ensembles Δ_{n+1}^0 qui ne peuvent être décrits de manière Σ_n^0 ou Π_n^0 : la hiérarchie arithmétique est stricte.

Donnons avant de continuer quelques exemples d'ensembles de la hiérarchie arithmétique.

Exemple 1.5. $\triangleright$ Tout ensemble $A \subseteq \mathbb{N}$ calculable est Δ_1^0 (nous en verrons une preuve avec la proposition 3.4). Il suffit de rajouter une quantification existentielle ou universelle inutile au prédicat calculable A pour pouvoir l'exprimer respectivement comme ensemble Σ_1^0 ou Π_1^0 .

- $\triangleright$ Tout ensemble $A \subseteq \mathbb{N}$ calculatoirement énumérable est Σ_1^0 (nous le verrons précisément avec la proposition 3.3). En effet, un tel A est décrit comme $\{x : \exists t \Phi_e(x)[t] \downarrow\}$ pour un certain e .
- $\triangleright$ Nous avons vu avec l'exemple 1 que l'ensemble des codes de fonctions totales est Π_2^0 . Son complémentaire, à savoir l'ensemble des codes de fonctions partielles, est donc $\Sigma_2^0 : \{e : \exists n \forall t \Phi_e(n)[t] \uparrow\}$

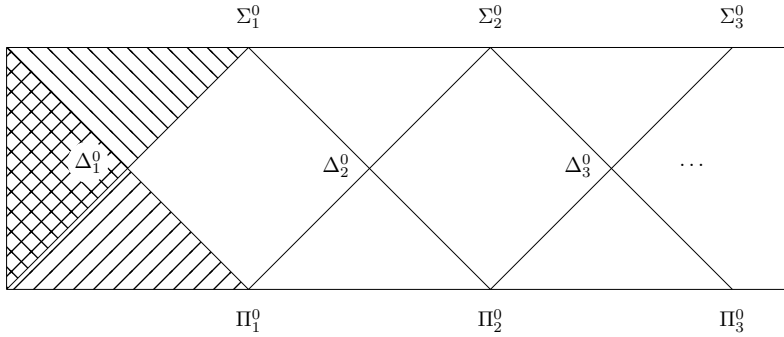


FIGURE 1.4 – Représentation de la hiérarchie arithmétique. Le triangle supérieur gauche hachuré représente les Σ_1^0 . Le triangle inférieur gauche hachuré représente les Π_1^0 . L'intersection entre les deux — le triangle doublement hachuré — représente les Δ_1^0 . Le reste du diagramme continue de la même manière.

▷ L'ensemble des codes de fonctions qui sont totales, sauf pour un nombre fini d'éléments, est Σ_3^0 :

$$\{e : \exists n \forall m \exists t \text{ tel que } m < n \text{ ou } \Phi_e(m)[t] \downarrow\}.$$

Nous verrons un peu plus loin que les exemples précédents sont optimaux. Par exemple, l'ensemble des codes de fonctions qui sont totales sauf pour un nombre fini d'éléments ne peut pas s'exprimer de manière Π_3^0 : il s'agit d'un ensemble Σ_3^0 strict.

Nous montrons à présent une série de trois propositions sur la stabilité des ensembles Σ_m^0 et Π_m^0 par différentes opérations. Par exemple, la stabilité par réunion finie signifie qu'une réunion finie d'ensembles Σ_m^0 est encore un ensemble Σ_m^0 . Les propositions seront démontrées uniquement en considérant des sous-ensembles de $\mathbb{N}$, mais se généralisent sans problème aux sous-ensembles de $\mathbb{N}^n$ pour n arbitraire.

Proposition 1.6. Les ensembles Σ_m^0 (resp. Π_m^0) sont stables par réunions finies et intersections finies. ★

PREUVE. Soit $m > 0$. Soient

$$\begin{aligned} A_0 &= \{n : \exists x_1 \forall x_2 \dots Qx_m R_0(n, x_1, \dots, x_m)\} \\ A_1 &= \{n : \exists y_1 \forall y_2 \dots Qy_m R_1(n, y_1, \dots, y_m)\} \end{aligned}$$

deux ensembles Σ_m^0 , où Q est le symbole $\exists$ si m est impair et $\forall$ si m est pair. On laisse au lecteur le soin de montrer, par inclusion dans un sens

puis dans l'autre, que :

$$A_0 \cap A_1 = \left\{ n : \begin{array}{l} \exists \langle x_1, y_1 \rangle \forall \langle x_2, y_2 \rangle \dots Q \langle x_m, y_m \rangle \\ \text{tels que } R_0(n, x_1, \dots, x_m) \wedge R_1(n, y_1, \dots, y_m) \end{array} \right\}.$$

Les ensembles Σ_m^0 sont donc stables par intersections finies. On a par ailleurs de la même manière :

$$A_0 \cup A_1 = \left\{ n : \begin{array}{l} \exists \langle x_1, y_1 \rangle \forall \langle x_2, y_2 \rangle \dots Q \langle x_m, y_m \rangle \\ \text{tels que } R_0(n, x_1, \dots, x_m) \vee R_1(n, y_1, \dots, y_m) \end{array} \right\}.$$

Les ensembles Σ_m^0 sont donc stables par réunions finies. Par passage au complémentaire, les ensembles Π_m^0 sont eux aussi stables par réunions et intersections finies. ■

Souvenons-nous du concept de suite uniformément calculable, introduit à l'occasion des développements qui suivent le lemme 4-7.2. Le concept d'uniformité passe à la hiérarchie arithmétique de la manière suivante : dans la prochaine proposition, une réunion dénombrable $(A_i)_{i \in \mathbb{N}}$ d'ensembles *uniformément* Σ_m^0 est donc une réunion telle que chaque ensemble A_i admet la même description Σ_m^0 , mais avec comme paramètre i . Formellement, si

$$A_i = \{ n : \exists x_1 \forall x_2 \dots Q x_m R_i(n, x_1, \dots, x_m) \},$$

il faut qu'il existe un processus calculable permettant à partir de i de renvoyer un code pour le prédicat R_i . De manière équivalente, on peut considérer que A_i est décrit comme

$$A_i = \{ n : \exists x_1 \forall x_2 \dots Q x_m R(i, n, x_1, \dots, x_m) \},$$

où R est un prédicat calculable prenant i comme paramètre supplémentaire.

Proposition 1.7. Les ensembles Σ_m^0 (resp. Π_m^0) sont stables par réunions dénombrables uniformes (resp. intersections dénombrables uniformes). ★

PREUVE. Soit $m > 0$, et soit $(A_i)_{i \in \mathbb{N}}$ une suite d'ensembles uniformément Σ_m^0 :

$$A_i = \{ n : \exists x_1 \forall x_2 \dots Q x_m R(i, n, x_1, \dots, x_m) \},$$

où Q est le symbole $\exists$ si m est impair et $\forall$ si m est pair. On montre aisément par inclusion dans un sens puis dans l'autre que

$$\bigcup_i A_i = \{ n : \exists \langle i, x_1 \rangle \forall x_2 \dots Q x_m R(i, n, x_1, \dots, x_m) \}.$$

Les ensembles Σ_m^0 sont donc stables par réunions dénombrables uniformes. Par passage au complémentaire, les ensembles Π_m^0 sont eux aussi stables par intersections dénombrables uniformes. ■

Notons que la stabilité des ensembles Σ_n^0 par réunion dénombrable uniforme équivaut à la stabilité par quantification existentielle (resp. quantification universelle pour les ensembles Π_n^0).

Exercice 1.8. Montrer que les ensembles Σ_n^0 ne sont pas stables par réunions dénombrables non uniformes. $\diamond$

On montre à présent la stabilité par quantification bornée uniforme. Une quantification bornée de la forme $\forall x < m$ peut être vue comme une réunion finie de m ensembles paramétrés par x . La proposition suivante n'est toutefois pas identique à la proposition 1.6 : on veut ici l'uniformité en fonction de la borne qui peut elle-même être une variable, ou dépendre d'autres variables sur lesquelles on quantifie. Grosso modo, ce que dit la proposition suivante est que l'ensemble

$$\{n : \exists x \forall y < x \exists t R(n, x, y, t)\},$$

où R est un prédicat calculable, peut se récrire comme

$$\{n : \exists x \exists t \forall y < x \exists s < t R(n, x, y, s)\}.$$

Le prédicat $\forall y < x \exists s < t R(n, x, y, s)$ étant calculable, l'ensemble est Σ_1^0 .

Proposition 1.9. Les ensembles Σ_m^0 et Π_m^0 sont stables par quantifications bornées uniformes. $\star$

PREUVE. On montre que l'on peut toujours déplacer la quantification bornée vers la droite, jusqu'à ce que celle-ci se retrouve à côté du prédicat calculable.

Soit $A = \{(n, k) : \forall y < k \exists x R(n, y, x)\}$, où R est un ensemble calculable ou Π_m^0 pour $m > 0$. Alors, on a aussi

$$A = \{(n, k) : \exists x \forall y < k \exists z < x R(n, y, z)\}.$$

L'égalité vient du fait que si pour tout $y < k$ un certain x_k témoigne qu'une formule est vraie, comme le nombre de témoins est fini, ceux-ci sont bornés dans $\mathbb{N}$. La variable x qui faisait auparavant office de témoin est à présent utilisée comme borne sur tous les témoins possibles. On a de la même manière par passage au complémentaire

$$\{(n, k) : \exists y < k \forall x R(n, y, x)\} = \{(n, k) : \forall x \exists y < k \forall z < x R(n, y, z)\},$$

où R est un ensemble calculable ou Σ_m^0 .

Si l'on a à présent $A = \{(n, k) : \exists y < k \exists x R(n, y, x)\}$, où R est un ensemble calculable ou Π_m^0 , alors on a aussi $A = \{(n, k) : \exists x \exists y < k R(n, y, x)\}$ de manière immédiate. On a de la même manière par passage au complémentaire $\{(n, k) : \forall y < k \forall x R(n, y, x)\} = \{(n, k) : \forall x \forall y < k R(n, y, x)\}$, où R est un ensemble calculable ou Σ_m^0 .

Par récurrence, on déplace ainsi les quantifications bornées vers la droite jusqu'à ce que celles-ci soient toutes collées au prédicat calculable. On utilise enfin le fait que les prédicats calculables sont stables par quantifications bornées (voir l'exercice 3-2.4) pour conclure. ■

2. Hiérarchie arithmétique et calculabilité

Voyons à présent à quoi correspondent les premiers niveaux de la hiérarchie arithmétique.

Proposition 2.1. Un ensemble $A \subseteq \mathbb{N}$ est Σ_1^0 ssi il est calculatoirement énumérable. ★

PREUVE. Supposons que l'ensemble A soit calculatoirement énumérable. Alors, $A = \{n : \exists t \Phi_e(n)[t] \downarrow\}$ pour un certain e . Le prédicat $\Phi_e(n)[t] \downarrow$ est calculable. Donc, A est Σ_1^0 . Supposons à présent que A soit Σ_1^0 . Soit $A = \{n : \exists t R(n, t)\}$, où R est un prédicat calculable. Alors, on définit facilement la machine de code e qui sur l'entrée n cherche le plus petit t tel que $R(n, t)$ et s'arrête, ou continue sa recherche indéfiniment sinon. On a alors $\Phi_e(n) \downarrow$ ssi $\exists t R(n, t)$. ■

Proposition 2.2. Un ensemble $A \subseteq \mathbb{N}$ est Δ_1^0 ssi il est calculable. ★

PREUVE. Un ensemble A est Δ_1^0 ssi A et $\mathbb{N} \setminus A$ sont Σ_1^0 ssi A et $\mathbb{N} \setminus A$ sont calculatoirement énumérables (d'après la proposition 2.1), ou encore ssi A est calculable (d'après la proposition 3-7.4). ■

Nous avons vu qu'il y a des ensembles calculatoirement énumérables qui ne sont pas calculables, et en particulier dont le complémentaire n'est pas calculatoirement énumérable. Cela implique que certains ensembles sont Σ_1^0 mais pas Δ_1^0 , et en particulier pas Π_1^0 . Nous allons voir dans les prochaines sections que la hiérarchie est stricte partout : pour tout n , il existe des ensembles Σ_n^0 qui ne sont pas Π_n^0 , et il existe des ensembles Δ_{n+1}^0 qui ne sont ni Σ_n^0 , ni Π_n^0 .

Intuitivement, le nombre de quantificateurs correspond au « nombre de fois qu'il faudrait compter jusqu'à l'infini » pour déterminer l'appartenance d'un élément à l'ensemble. Ainsi, pour un ensemble Σ_1^0 s'écrivant comme $\{n : \exists t R(t, n)\}$, où R est un prédicat calculable, il faudrait tester la valeur de vérité de $R(t, n)$ pour tous les entiers t , afin d'en trouver un qui témoigne de l'appartenance de n à l'ensemble, ou bien afin d'être sûr que n n'y appartient pas.

Pour un ensemble Π_2^0 de la forme

$$\{n : \forall t_1 \exists t_2 R(t_1, t_2, n)\},$$

il faudrait une première procédure qui teste tous les entiers t_1 , et qui pour chacun de ces tests, examine la valeur de vérité de $R(t_1, t_2, n)$ pour tous les entiers t_2 . Cela correspondrait en quelque sorte à deux boucles imbriquées se déroulant chacune sur l'ensemble de tous les entiers.

3. Relativisation à un oracle

La hiérarchie arithmétique permet de définir des ensembles de plus en plus complexes, et dans un sens de moins en moins calculables. Toutefois, la classe des ensembles arithmétiquement définissables reste dénombrable. Il reste donc « beaucoup » d'ensembles, pour ainsi dire la majorité, qui ne peuvent être ni calculés, ni même définis par une formule de l'arithmétique. Il est évidemment difficile d'en parler, et toute tentative de les cerner plus précisément les rendraient définissables dans un certain langage, élargissant simplement un peu plus la classe immanquablement dénombrable des ensembles dont on arrive à dire quelque chose, laissant de côté la majorité des autres ensembles, cachés, inaccessibles.

Nous contournerons le problème tout au long des chapitres à venir, essentiellement en étudiant des « groupes d'ensembles » plutôt que chaque ensemble individuellement. Nous verrons en particulier dans les chapitres à venir de nombreuses propriétés calculatoires partagées par certains ensembles, en général une quantité indénombrable d'entre eux. Nous mènerons ensuite dans la partie II une étude des ensembles typiques du point de vue de la théorie de la mesure, c'est-à-dire des ensembles que l'on obtient avec probabilité 1 si l'on sélectionne leurs bits au hasard.

Pour le moment, nous nous contentons de relativiser la hiérarchie arithmétique à un oracle : étant donné un ensemble X quelconque, on considère les ensembles Σ_n^0 , Π_n^0 et Δ_n^0 que l'on peut définir relativement à la connaissance de X . On se base pour cela sur les calculs avec oracle du théorème 4-2.2, et l'on itère comme dans la définition 1.1.

Définition 3.1. Soient $m, n \geq 1$. Soit $X \subseteq \mathbb{N}$.

1. Un ensemble $A \subseteq \mathbb{N}^m$ est dit $\Sigma_n^0(X)$ s'il existe un ensemble X -calculable $R \subseteq \mathbb{N}^{n+m}$ tel que

$$A = \{(y_1, \dots, y_m) : \overbrace{\exists x_1 \forall x_2 \dots Q x_n}^{n \text{ quantificateurs}} (x_1, \dots, x_n, y_1, \dots, y_m) \in R\},$$

où Q vaut $\exists$ si n est impair, et $\forall$ si n est pair.

2. Un ensemble $A \subseteq \mathbb{N}^m$ est dit $\Pi_n^0(X)$ s'il existe un ensemble X -calculable $R \subseteq \mathbb{N}^{n+m}$ tel que

$$A = \{(y_1, \dots, y_m) : \overbrace{\forall x_1 \exists x_2 \dots Q x_n}^{n \text{ quantificateurs}} (x_1, \dots, x_n, y_1, \dots, y_m) \in R\},$$

où Q vaut $\forall$ si n est impair, et $\exists$ si n est pair. $\diamond$

Définition 3.2. Soit $m \geq 1$. Soit $X \subseteq \mathbb{N}$. Un ensemble $A \subseteq \mathbb{N}^m$ est dit $\Delta_n^0(X)$ pour $n > 0$ s'il est à la fois $\Sigma_n^0(X)$ et $\Pi_n^0(X)$. $\diamond$

Les propositions suivantes se montrent comme leurs équivalents respectifs de la section précédente.

Proposition 3.3. Un ensemble $A \subseteq \mathbb{N}$ est $\Sigma_1^0(X)$ ssi il est X -c.e. $\star$

Proposition 3.4. Un ensemble $A \subseteq \mathbb{N}$ est $\Delta_1^0(X)$ ssi il est X -calculable. $\star$

Notons qu'un oracle pourra fournir de la puissance de calcul supplémentaire, permettant à certains ensembles normalement non Σ_n^0 , de devenir $\Sigma_n^0(X)$. On peut même par exemple définir un oracle X tel que tous les ensembles arithmétiques, c'est-à-dire Σ_n^0 pour un certain n , deviennent $\Delta_1^0(X)$. Toutefois, quelle que soit la puissance de calcul de X , les ensembles arithmétiques en X restent en quantité dénombrable.

4. Degrés many-one

La classification de la hiérarchie arithmétique en termes d'ensembles Σ_n^0 et Π_n^0 n'est pas une notion de degrés Turing, car un ensemble Σ_n^0 peut être Turing équivalent à un ensemble qui ne l'est pas. De fait, tout ensemble Σ_n^0 A est Turing équivalent à son complémentaire $\Pi_n^0 \bar{A}$. En ce sens, la réduction Turing est « grossière », car elle ne distingue pas un ensemble de son complémentaire.

Nous allons introduire une notion plus fine que la réduction Turing, au sens où elle implique cette dernière. Il s'agit de la réduction many-one, qui comme nous le verrons préserve la hiérarchie arithmétique. Dans les faits, beaucoup de preuves de réduction Turing que nous avons vues sont des réductions many-one.

Définition 4.1. Soient deux ensembles A, B . On dit que A est *many-one réductible* à B , et l'on écrit $A \leq_m B$ s'il existe une fonction totale calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $n \in A \leftrightarrow f(n) \in B$. Si $A \leq_m B$ et $B \leq_m A$, on écrit $A \equiv_m B$. On écrit $A <_m B$ si $A \leq_m B$ mais $B \not\leq_m A$. On appelle *degrés many-one* les classes d'équivalence de la relation $\equiv_m$. $\diamond$

Quand on a $A \leq_m B$, la connaissance de B est suffisante pour calculer A , et l'on a en particulier $A \leq_T B$: soit f la fonction totale calculable telle que $n \in A \leftrightarrow f(n) \in B$. Alors, pour savoir si $n \in A$, on utilise f pour « poser une question » à l'ensemble B : est-ce que $f(n) \in B$? Si la réponse est oui, alors $n \in A$. Sinon, $n \notin A$. La réduction many-one est très restrictive : si $A \leq_m B$, alors pour connaître un bit de A on n'a le droit de poser qu'une seule question à l'oracle B . Comme si cela ne suffisait pas, la réponse à la question détermine directement l'appartenance du bit à A sans qu'il ne soit possible d'inverser cette décision. L'importance de cette réduction, très restrictive, vient du fait qu'elle préserve la hiérarchie arithmétique.

Proposition 4.2. Soit $A \subseteq \mathbb{N}$ un ensemble $\Sigma_m^0(X)$ (resp. $\Pi_m^0(X)$) pour un certain $X \subseteq \mathbb{N}$, et soit $B \leq_m A$. Alors, B est $\Sigma_m^0(X)$ (resp. $\Pi_m^0(X)$). ★

PREUVE. Soit A un ensemble $\Sigma_m^0(X)$. Alors,

$$A = \{n : \exists x_1 \forall x_2 \dots Qx_m R(n, x_1, \dots, x_m)\},$$

où R est un ensemble X -calculable. Soit f la fonction totale calculable telle que $n \in B$ ssi $f(n) \in A$. On a donc

$$B = \{n : \exists x_1 \forall x_2 \dots Qx_m R(f(n), x_1, \dots, x_m)\},$$

ce qui est bien une description $\Sigma_m^0(X)$ de B .

On montre de la même manière que si A est $\Pi_m^0(X)$ et $B \leq_m A$, alors B est $\Pi_m^0(X)$. ■

Parmi les ensembles Σ_1^0 , l'arrêt des programmes informatiques joue un rôle particulier : il est le plus « puissant » des ensembles Σ_1^0 , dans le sens où tout ensemble Σ_1^0 est many-one réductible à l'arrêt.

Proposition 4.3. Un ensemble A est Σ_1^0 si, et seulement si, $A \leq_m \emptyset'$. ★

PREUVE. Supposons $A \leq_m \Sigma_1^0$. Alors, il existe e tel que $n \in A$ ssi $\Phi_e(n) \downarrow$. En utilisant le théorème SMN, on définit une fonction calculable $s : \mathbb{N} \rightarrow \mathbb{N}$ telle que pour tout m on ait $\Phi_e(n) = \Phi_{s(n)}(m)$. On aura en particulier $n \in A$ ssi $\Phi_e(n) \downarrow$, ou ce qui revient au même ssi $\Phi_{s(n)}(s(n)) \downarrow$, ou encore ssi $s(n) \in \emptyset'$. Donc, $A \leq_m \emptyset'$.

Supposons à présent $A \leq_m \emptyset'$. Alors, comme $\emptyset'$ est Σ_1^0 , l'ensemble A est lui aussi Σ_1^0 , d'après la proposition 4.2. ■

On dit aussi que $\emptyset'$ est un ensemble Σ_1^0 -complet, comme nous le verrons avec la définition 5.2.

5. Théorème de Post

Nous allons voir qu'il y a une correspondance précise entre les itérations du saut Turing et la hiérarchie arithmétique.

Définition 5.1. Étant donné un ensemble $X \subseteq \mathbb{N}$, on définit récursivement sur $n \geq 0$:

1. $X^{(0)} = X$
2. $X^{(n+1)} = X^{(n)'}.$

◇

Ainsi, $\emptyset^{(1)} = \emptyset'$, $\emptyset^{(2)} = \emptyset''$, etc. Nous allons à présent montrer que, pour tout n , l'ensemble $\emptyset^{(n)}$ est Σ_n^0 -complet : il s'agit d'un ensemble Σ_n^0 , qui a également une puissance calculatoire maximale parmi les ensembles Σ_n^0 , dans le sens où tout ensemble Σ_n^0 est many-one réductible à $\emptyset^{(n)}$.

Définition 5.2. Un ensemble $A \subseteq \mathbb{N}$ est $\Sigma_n^0(X)$ -complet (resp. $\Pi_n^0(X)$ -complet) s'il est $\Sigma_n^0(X)$ (resp. $\Pi_n^0(X)$) et si, pour tout ensemble $\Sigma_n^0(X)$ (resp. $\Pi_n^0(X)$) B , on a $B \leq_m A$.

◇

Proposition 5.3. Soit $n > 0$. L'ensemble $\emptyset^{(n)}$ est Σ_n^0 -complet. De la même manière, pour tout ensemble $X \subseteq \mathbb{N}$, l'ensemble $X^{(n)}$ est $\Sigma_n^0(X)$ -complet.★

PREUVE. Montrons par récurrence sur n que pour tout $n > 0$ l'ensemble $\emptyset^{(n)}$ est Σ_n^0 . Par définition, l'ensemble $\emptyset^{(1)}$ est Σ_1^0 . Supposons que l'ensemble $\emptyset^{(n)}$ soit Σ_n^0 . Alors, l'ensemble $\emptyset^{(n+1)}$ est défini comme :

$$\left\{ m : \exists t \in \mathbb{N} \exists \sigma \in 2^{<\mathbb{N}} \left(\begin{array}{c} (\forall s < |\sigma| \quad (\sigma(s) = 1 \text{ et } s \in \emptyset^{(n)}) \\ \text{ou} \quad (\sigma(s) = 0 \text{ et } s \notin \emptyset^{(n)}) \end{array} \right) \right\}.$$

et $\Phi_m(\sigma, m)[t] \downarrow$

La description de $\emptyset^{(n+1)}$ se fait donc avec des quantificateurs existentiels, suivi d'un prédicat utilisant $s \in \emptyset^{(n)}$, ce qui est par récurrence Σ_n^0 , et utilisant $s \notin \emptyset^{(n)}$, ce qui est par récurrence Π_n^0 , et donc Σ_{n+1}^0 . En utilisant les propriétés de stabilité des propositions 1.9 et 1.6, le prédicat qui suit les quantificateurs existentiels $\exists t \in \mathbb{N} \exists \sigma \in 2^{<\mathbb{N}}$ est en particulier Σ_{n+1}^0 uniformément en m, σ et t . En utilisant à présent la stabilité par réunion dénombrable uniforme de la proposition 1.7, l'ensemble $\emptyset^{(n+1)}$ est donc Σ_{n+1}^0 .

Montrons à présent par récurrence sur n que tout ensemble Σ_n^0 est many-one réductible à $\emptyset^{(n)}$.

C'est le cas par la proposition 4.3 pour $n = 1$. Supposons que ce soit le cas pour un certain n . Soit $A = \{x : \exists y R(x, y)\}$, où R est un ensemble Π_n^0 .

Par récurrence, il existe une fonction totale calculable $g : \mathbb{N} \rightarrow \mathbb{N}$ telle que $(x, y) \in R$ ssi $g(\langle x, y \rangle) \notin \emptyset^{(n)}$. On définit la fonction totale calculable f telle que $\Phi_{f(x)}^{\emptyset^{(n)}}$ s'arrête sur toute entrée si $\exists y g(\langle x, y \rangle) \notin \emptyset^{(n)}$, et ne s'arrête sur aucune entrée sinon. On a donc $f(x) \in \emptyset^{(n+1)}$ ssi $\Phi_{f(x)}(\emptyset^{(n)}, f(x)) \downarrow$, autrement dit ssi $\exists y g(\langle x, y \rangle) \notin \emptyset^{(n)}$, ou encore ssi $\exists y R(x, y)$, ou enfin ssi $x \in A$. L'ensemble A est donc many-one réductible à $\emptyset^{(n+1)}$.

La relativisation à un oracle X est similaire et ne présente pas de difficulté particulière. ■

Corollaire 5.4

Un ensemble A est $\Sigma_n^0(X)$ ssi $A \leq_m X^{(n)}$.

PREUVE. Par la proposition précédente et par la définition de la $\Sigma_n^0(X)$ -complétude. ■

Nous arrivons finalement au théorème de Post.

Théorème 5.5 (Théorème de Post)

Soient A un ensemble et $n \geq 0$.

- (1) A est Σ_{n+1}^0 ssi A est $\Sigma_1^0(\emptyset^{(n)})$, ou encore ssi A est $\emptyset^{(n)}$ -c. e.
- (2) A est Δ_{n+1}^0 ssi A est $\Delta_1^0(\emptyset^{(n)})$, ou encore ssi $A \leq_T \emptyset^{(n)}$

PREUVE. Montrons (1). Par le corollaire 5.4, A est Σ_{n+1}^0 ssi $A \leq_m \emptyset^{(n+1)}$. En relativisant la proposition 4.3 à $\emptyset^{(n)}$, on obtient $A \leq_m \emptyset^{(n)'} (qui est égal à $\emptyset^{(n+1)})$ ssi A est $\Sigma_1^0(\emptyset^{(n)})$. Enfin, d'après la proposition 3.3, A est $\Sigma_1^0(\emptyset^{(n)})$ ssi A est $\emptyset^{(n)}$ -c. e.$

Montrons (2). Par définition, A est Δ_{n+1}^0 ssi A et $\bar{A}$ sont tous les deux Σ_{n+1}^0 . Par le point précédent, c'est équivalent à dire que A et $\bar{A}$ sont $\emptyset^{(n)}$ -c. e. En relativisant la proposition 3-7.4 à $\emptyset^{(n)}$, A et $\bar{A}$ sont $\emptyset^{(n)}$ -c. e. ssi $A \leq_T \emptyset^{(n)}$. Enfin, d'après la proposition 3.4, on a $A \leq_T \emptyset^{(n)}$ ssi A est $\Delta_1^0(\emptyset^{(n)})$. ■

Corollaire 5.6

La hiérarchie arithmétique est stricte. En d'autres termes,

- ▷ pour tout $n > 0$, il existe un ensemble Σ_n^0 qui n'est pas Π_n^0 et un ensemble Π_n^0 qui n'est pas Σ_n^0 ;
- ▷ pour tout $n > 0$, il existe un ensemble Δ_{n+1}^0 qui n'est ni Σ_n^0 ni Π_n^0 .

PREUVE. D'après la proposition 5.3, l'ensemble $\emptyset^{(n)}$ est Σ_n^0 . Il ne peut pas être Π_n^0 auquel cas il serait Δ_n^0 , et donc calculable en $\emptyset^{(n-1)}$ d'après le théorème 5.5, contredisant le fait que X ne calcule jamais son saut Turing. On montre de la même manière que $\mathbb{N} \setminus \emptyset^{(n)}$ est Π_n^0 , mais pas Σ_n^0 .

On peut enfin construire pour tout n l'ensemble Δ_{n+1}^0 suivant : l'ensemble X tel que $X(2m) = \emptyset^{(n)}(m)$ et tel que $X(2m+1) = (\mathbb{N} \setminus \emptyset^{(n)})(m)$. Il est clair que X est $\emptyset^{(n)}$ -calculable, et donc Δ_{n+1}^0 d'après le théorème 5.5. Enfin, si l'on suppose par l'absurde que X est Σ_n^0 (resp. Π_n^0), cela permet de donner une description Σ_n^0 de $\mathbb{N} \setminus \emptyset^{(n)}$ en ne gardant que les bits impairs de X (resp. une description Π_n^0 de $\emptyset^{(n)}$ en ne gardant que les bits pairs de X), ce qui est en contradiction avec le fait que $\emptyset^{(n)}$ n'est pas Π_n^0 (resp. avec le fait que $\mathbb{N} \setminus \emptyset^{(n)}$ n'est pas Σ_n^0). ■

Exercice 5.7. (*) Montrer que, pour tous $X, Y \in 2^{\mathbb{N}}$, on a $X \equiv_T Y$ ssi $X' \equiv_m Y'$. ◇

6. Théorème de Rice

Le théorème de Rice dit en substance qu'aucune propriété sémantique des programmes n'est décidable. Par exemple, comme vu dans l'exercice 3-6.4, il est impossible de décider si un programme informatique effectue une multiplication par 2 ou pas. Nous entendons *propriétés sémantiques* par opposition à *propriétés syntaxiques*. Ces dernières sont sensibles aux variations de code, par exemple « ce programme possède trois variables distinctes » ou « ce programme contient deux boucles `for` ». Les propriétés sémantiques ne parlent pas directement des programmes, mais des fonctions qu'ils représentent. Par exemple, les propriétés « cete fonction est définie sur l'entrée 42 » ou « cette fonction ne renvoie que des valeurs paires » sont des propriétés sémantiques. Elles ne dépendent pas des détails d'implémentation des fonctions concernées.

Définition 6.1. Un *ensemble sémantique de codes* est un ensemble A inclus dans $\mathbb{N}$, tel que pour tous $x, y \in \mathbb{N}$,

$$\text{si } x \in A \text{ et } \Phi_x = \Phi_y, \text{ alors } y \in A.$$

◇

Parmi les ensembles sémantiques de codes, on notera l'ensemble vide $\emptyset$ qui correspond à une propriété jamais satisfaite, et l'ensemble $\mathbb{N}$ représentant une propriété toujours vraie. Un ensemble sémantique de codes est *non trivial* si $A \neq \emptyset$ et $A \neq \mathbb{N}$.

Théorème 6.2

Si A est un ensemble sémantique de codes non trivial, alors soit $\emptyset' \leq_m A$ soit $\emptyset' \leq_m \bar{A}$.

PREUVE. Soit Φ_{e_0} la fonction nulle part définie. Supposons que $e_0 \in \bar{A}$, l'autre cas étant traité par symétrie. Comme A est non trivial, A est non vide. Fixons un code $e_1 \in A$. En particulier, $\Phi_{e_0} \neq \Phi_{e_1}$. Par le théorème SMN (voir le théorème 3-4.1), il existe une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ totale et calculable telle que

$$\Phi_{f(x)}(y) = \begin{cases} \Phi_{e_1}(y) & \text{si } x \in \emptyset' \\ \uparrow & \text{si } x \notin \emptyset'. \end{cases}$$

Montrons que la fonction f est une réduction many-one de $\emptyset'$ à A . Si $x \in \emptyset'$, alors $\Phi_{f(x)} = \Phi_{e_1}$, et ainsi $f(x) \in A$. Inversement, si $x \notin \emptyset'$, $\Phi_{f(x)} = \Phi_{e_0}$, donc $f(x) \in \bar{A}$. ■

Le théorème de Rice affirme qu'aucune propriété non triviale sur les fonctions partielles calculables n'est décidable. Bien que les applications de ce théorème soient relativement limitées en calculabilité, le théorème de Rice revêt une grande importance pour la compréhension qu'il apporte sur la nature de la calculabilité. En particulier, l'indécidabilité du problème de l'arrêt n'est pas un phénomène isolé, car il est partagé par toutes les propriétés sur les fonctions partielles calculables.

Corollaire 6.3 (Théorème de Rice)

Soit $\mathcal{C}$ une classe de fonctions partielles $\mathbb{N} \rightarrow \mathbb{N}$ calculables. Alors, l'ensemble $A = \{x : \Phi_x \in \mathcal{C}\}$ est incalculable sauf si $\mathcal{C} = \emptyset$ ou $\mathcal{C}$ est la classe de toutes les fonctions partielles calculables.

PREUVE. L'ensemble A est un ensemble sémantique de codes. Si $\mathcal{C} = \emptyset$ ou $\mathcal{C}$ est la classe de toutes les fonctions partielles calculables, alors $A = \emptyset$ ou $A = \mathbb{N}$ et, dans les deux cas, A est calculable. Si $\mathcal{C}$ n'est ni vide ni la classe de toutes les fonctions partielles calculables, alors A est non trivial, et par le théorème 6.2 soit $\emptyset' \leq_m A$, soit $\emptyset' \leq_m \bar{A}$. Dans les deux cas, $\emptyset' \leq_T A$, et A est non calculable. ■

7. Codes arithmétiques

Les ensembles d'entiers sont de manière générale des objets infinis, et ne peuvent donc pas être représentés par des entiers naturels sans que certains d'entre eux ne soient omis de cette représentation. C'est l'objet de l'argument diagonal de Cantor (voir la section 2-4). Certains ensembles d'entiers peuvent cependant être décrits de manière finitaire, à commencer par les ensembles calculables.

Définition 7.1. Un *code* Δ_1^0 d'un ensemble calculable A est un entier e tel que $\Phi_e = A$ (c'est-à-dire $\forall n \ \Phi_e(n) \downarrow = A(n)$). $\diamond$

Notons qu'un ensemble est calculable ssi il a un code Δ_1^0 . Par le lemme 3-5.1 de remplissage, tout ensemble calculable est représenté par une infinité de codes Δ_1^0 . D'après le théorème de Rice, l'ensemble des codes Δ_1^0 d'un ensemble calculable fixé n'est pas décidable. Il n'existe même pas de procédure pour décider si un entier e est un code Δ_1^0 d'un ensemble. En effet, cela reviendrait à être capable d'énumérer de manière calculable tous les ensembles calculables, et nous ferait tomber sous la coupe de l'argument diagonal de Cantor.

Exercice 7.2. (*) Soit TOT l'ensemble des codes Δ_1^0 ; autrement dit,

$$\text{TOT} = \{e : \Phi_e \text{ est total}\}$$

(en supposant sans perte de généralité que $\Phi_e(n)$ renvoie 0 ou 1 pour tout n). Prouver que TOT est Π_2^0 -complet, c.-à-d. que $\mathbb{N} \setminus \emptyset'' \leq_m \text{TOT}$. $\diamond$

Les codes Δ_1^0 permettent de donner une nouvelle définition d'une suite d'ensembles uniformément calculable. Rappelons qu'une suite $X_0, X_1, \dots$ d'ensembles est uniformément calculable si la fonction $f : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ définie par $f(x, s) = 1$ ssi $x \in X_s$ est totale calculable.

Proposition 7.3. Une suite $X_0, X_1, \dots$ d'ensembles est uniformément calculable si, et seulement si, il existe une suite calculable $e_0, e_1, \dots$ tel que e_s est un code Δ_1^0 de X_s pour tout s . $\star$

PREUVE. $\Rightarrow$. Supposons que $X_0, X_1, \dots$ soit uniformément calculable à travers la fonction $f : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$. Par le théorème SMN, il existe une fonction totale calculable $h : \mathbb{N} \rightarrow \mathbb{N}$ telle que pour tous e, x , on ait l'égalité $\Phi_{h(e)}(x) = f(x, e)$. Il s'ensuit que la suite $h(0), h(1), \dots$ est une suite calculable telle que $h(s)$ est un code Δ_1^0 de X_s .

$\Leftarrow$. Supposons maintenant qu'il existe une suite calculable $e_0, e_1, \dots$ de codes Δ_1^0 . Alors, vu l'existence d'une machine universelle, la fonction f définie sur $\mathbb{N} \times \mathbb{N}$ par $f(x, s) = \Phi_{e_s}(x) \in \mathbb{N}$ est totale calculable, et montre que la suite d'ensembles $X_0, X_1, \dots$ est uniformément calculable. $\blacksquare$

Les ensembles calculatoirement énumérables sont également représentables par un système de code. On utilisera pour cela souvent la notation ci-après.

Notation

On note W_e l'ensemble $\{n \in \mathbb{N} : \Phi_e(n) \downarrow\}$, lequel est calculatoirement énumérable. La notation se relativise à un oracle X , et W_e^X ou $W_e(X)$ dénotera ainsi l'ensemble $\{n \in \mathbb{N} : \Phi_e(X, n) \downarrow\}$.

Définition 7.4. Un *code* Σ_1^0 d'un ensemble c.e. A est un entier e tel que $W_e = A$. ◇

Toujours par le théorème de Rice, l'ensemble des codes Σ_1^0 d'un ensemble c.e. fixé n'est pas calculable. Cependant, contrairement aux codes Δ_1^0 , *tout entier* est un code Σ_1^0 . Les ensembles calculables étant *a fortiori* calculatoirement énumérables, ils possèdent à la fois des codes Δ_1^0 et Σ_1^0 . La représentation par des codes Δ_1^0 est cependant calculatoirement plus informative, au sens où elle donne accès à plus d'informations sur l'ensemble qu'elle représente. En particulier, la fonction partielle $f : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ telle que si e est un code Δ_1^0 d'un ensemble A , est définie pour $x \in \mathbb{N}$ par $f(e, x) \downarrow = 1$ si $x \in A$ et $f(e, x) \downarrow = 0$ si $x \notin A$, est calculable¹, tandis que la fonction équivalente pour les codes Σ_1^0 ne l'est pas.

Exercice 7.5. Montrer qu'il n'existe pas de fonction $f : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ qui soit totale, calculable et telle que $f(e, x) = 1$ si $x \in W_e$ et $f(e, x) = 0$ si $x \notin W_e$. ◇

De manière générale, on peut représenter tout ensemble de la hiérarchie arithmétique par des entiers à l'aide du théorème de Post.

Définition 7.6. Un *code* Σ_{n+1}^0 (resp. Δ_{n+1}^0) d'un ensemble A est un entier e tel que $W_e(\emptyset^{(n)}) = A$ (resp. $\Phi_e(\emptyset^{(n)}) = A$). ◇

Remarque

Étant donné un ensemble A , on aura tendance à privilégier le type de code qui apporte le plus d'information calculatoire sur A . Ainsi, si A est calculable, il est préférable de manipuler un code Δ_1^0 plutôt que Σ_1^0 .

Cette recommandation s'applique également à des concepts de calculabilité plus élaborés, comme les ensembles low. Pour rappel, un ensemble A est low si $A' \leq_T \emptyset'$. Comme $A \leq_T A' \leq_T \emptyset'$, les ensembles low sont en particulier Δ_2^0 , et peuvent donc être représentés par un code Δ_2^0 , c'est-à-dire un entier e tel que $\Phi_e(\emptyset') = A$. Cependant, cette représentation perd de l'information spécifique aux ensembles low (voir l'exercice 7.11), comme la capacité de $\emptyset'$ de décider A' . Il est donc préférable de représenter l'ensemble A par un code e tel que $\Phi_e(\emptyset') = A'$.

Définition 7.7. Un *code de lowness* d'un ensemble A est un entier e tel que $\Phi_e(\emptyset') = A'$. ◇

1. Si e n'est pas un code Δ_1^0 , la fonction f agit tout de même comme si c'était le cas, et l'on aura éventuellement alors $f(e, x) \uparrow$.

Enfin, dans le cas des ensembles finis, il est possible de stocker plus d'information que le simple fait d'être calculable. En effet, étant donné une suite $e_0, e_1, \dots$ de codes Δ_1^0 d'ensembles finis, il n'est pas possible de calculer uniformément le cardinal de ces ensembles (voir l'exercice 7.10). On préférera donc la notion de code canonique qui contient notamment l'information de la taille de l'ensemble.

Définition 7.8. Le *code canonique* d'un ensemble fini F est l'entier naturel $\sum_{i \in F} 2^i$. ◇

On vérifie sans peine via l'exercice suivant qu'un code canonique contient toute l'information d'un ensemble fini, y compris la possibilité de connaître son dernier élément.

Exercice 7.9. Soit $D_0, D_1, \dots$ la suite des ensembles finis telle que D_n est de code canonique n .

1. Montrer que la fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ définie par $f(n) = |D_n|$ est calculable.
2. Montrer que la fonction $g : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ définie par $g(n, x) = 1$ ssi $x \in D_n$ est calculable. ◇

Les deux exercices suivants permettent de montrer que les informations que nous donnent les codes canoniques d'ensembles finis ou les codes de lowness ne peuvent être obtenues via des codes calculables ou Δ_2^0 .

Exercice 7.10. (★★) Supposer par l'absurde qu'il existe une fonction partielle calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que si e est un code Δ_1^0 d'un ensemble fini, alors $f(e)$ renvoie la taille de cet ensemble. En utilisant le théorème du point fixe, montrer qu'il existe a un code Δ_1^0 d'un ensemble fini tel que la taille de cet ensemble est différente de $f(a)$. ◇

Exercice 7.11. (★★) Supposer par l'absurde qu'il existe une fonction partielle calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que si e est un code Δ_2^0 d'un ensemble low X , alors $f(e)$ renvoie un code Δ_2^0 pour X' . En utilisant le théorème du point fixe, montrer qu'il existe a un code Δ_2^0 d'un ensemble calculable X tel que $f(a)$ est le code Δ_2^0 d'un ensemble différent de X' . ◇

Chapitre 6

La thèse de Church-Turing

1. L'Entscheidungsproblem et la quête du Graal

En l'an 1928, dans une période troublée par de profondes remises en questions sur les fondements des mathématiques, David Hilbert et Wilhelm Ackermann posèrent la question de l'existence d'un algorithme permettant de décider de la validité de n'importe quel énoncé mathématique. Ici, le mot *algorithme* est à prendre dans un sens large, pour désigner un ensemble d'étapes de calcul élémentaires que peut réaliser un mathématicien. Ce défi lancé aux logiciens et passé à la postérité sous le nom d'*Entscheidungsproblem* — problème de décision — marque le début d'une longue quête fondationnelle sur la formalisation de la notion d'algorithme.

Les théorèmes d'incomplétude de Gödel¹ prouvés en 1931, énonçant l'existence d'énoncés fondamentalement indécidables dans toute théorie raisonnable permettant de formaliser l'arithmétique, furent particulièrement malvenus dans une période qui cherchait à initier une nouvelle vague d'optimisme et de confiance dans les mathématiques. Ils eurent aussi comme conséquence de faire pencher la balance vers l'existence d'une solution négative à l'Entscheidungsproblem.

Une réponse positive à l'Entscheidungsproblem aurait été un algorithme ou une série d'étapes déterministes, permettant de démontrer tout énoncé mathématique ou sa négation. Une réponse négative consiste quant à elle en la preuve qu'une telle méthode systématique ne peut exister. Cette direction pose un tout autre problème, à savoir définir formellement le concept

1. Voir le chapitre 9

d'algorithme, ou de manière à peu près équivalente, de trouver une formalisation robuste et consensuelle de la notion de fonction calculable.

Il convient de préciser que le premier ordinateur, l'ENIAC, a été construit en 1940, soit une décennie après la formulation de l'Entscheidungsproblem. La notion de fonction effectivement calculable ne fait donc pas référence à ce qui est calculable par un ordinateur, mais par l'esprit humain. Il s'agissait de trouver un procédé systématique, ou algorithme au sens informel du terme, permettant à un mathématicien de répondre à toute question mathématique.

Plusieurs définitions furent proposées au cours des années qui suivirent, chacune conjecturée de manière plus ou moins convaincante comme capturant exactement la notion épistémologique de fonction effectivement calculable. Ces définitions furent assez rapidement prouvées équivalentes, mais peinèrent à convaincre la communauté scientifique de leur capacité à capturer toutes les fonctions effectivement calculables. Ce n'est qu'en 1936 qu'Alan Turing amena un consensus en présentant son modèle de calcul, la *machine de Turing*, avec une démonstration éclatante de son équivalence avec les autres modèles, en particulier avec celui utilisé par Gödel pour montrer son fameux théorème d'incomplétude, apportant ainsi une réponse définitive à l'Entscheidungsproblem. Le lecteur intéressé par l'histoire de la calculabilité trouvera un excellent chapitre dédié au sujet dans l'ouvrage *Turing computability : Theory and Applications* de Robert Soare.

Bien que les différents modèles de calcul aient été depuis prouvés équivalents, nous allons détailler les principaux parmi ceux qui marquèrent cette histoire, chacun présentant son propre intérêt, en mettant l'accent sur un aspect différent de la notion de fonction calculable. Dans ce qui suit, nous ne considérerons que des fonctions partielles, de $\mathbb{N}^n$ vers $\mathbb{N}$ pour $n \geq 1$. On écrira $g(\bar{x}) \downarrow = y$ pour signifier que g est définie sur $\bar{x} \in \mathbb{N}^n$ et vaut y ; la notation $\bar{x}$ étant un raccourci pour $x_1, \dots, x_n$.

Les fonctions générales récursives. Les fonctions récursives furent introduites sous une forme restreinte par Gödel dans le cadre de ses théorèmes d'incomplétude en 1931. Cette classe de fonctions a été généralisée par Gödel et Herbrand en 1934 pour obtenir les fonctions générales récursives, capturant d'après la thèse de Church-Turing — thèse détaillée dans la section suivante — l'intégralité des fonctions calculables.

L'idée sous-tendant la définition des fonctions générales récursives est très simple : partir de quelques fonctions élémentaires, dont la calculabilité ne fait pas place au doute, puis les combiner pour obtenir de nouvelles fonctions plus complexes, toujours calculables. Quelles sont les combinaisons valides permettant de créer de nouvelles fonctions ? Si deux fonctions sont

calculables, leur composition devrait naturellement être calculable : il suffit d'exécuter en série les étapes de calcul de chacune des fonctions. De manière un peu moins évidente, les fonctions définies par récursion à partir de fonctions calculables sont encore calculables. En effet, si l'on définit une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ par $f(0) = v$ pour un entier v , et $f(n+1) = g(n, f(n))$ pour une fonction calculable $g : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, pour obtenir la valeur de $f(n)$, il suffit de calculer successivement $f(1), f(2), \dots, f(n)$ à l'aide des valeurs précédentes et en suivant les étapes de calcul de la fonction g . Enfin, si une fonction $g : \mathbb{N} \rightarrow \mathbb{N}$ est calculable, la recherche de la plus petite entrée n telle que $g(n) = 0$ est également calculable, via une boucle qui incrémente n jusqu'à avoir $g(n) = 0$ (cette recherche pouvant ne jamais aboutir).

Définition 1.1. La classe $\mathcal{C}$ des *fonctions générales récursives* est la plus petite classe de fonctions partielles contenant les fonctions de base suivantes :

- (a) La fonction successeur : $\text{succ}(x) = x + 1$
 - (b) Les fonctions constantes : $c_m^n(x_1, \dots, x_n) = m$ pour tous $n, m \geq 0$
 - (c) Les projections : $p_i^n(x_1, \dots, x_n) = x_i$ pour tout n et tout $i \in 1, \dots, n$
- et qui est close par les opérations suivantes.
- (i) Composition : si $g_1, \dots, g_m \in \mathcal{C}$ sont des fonctions de n variables et $h \in \mathcal{C}$ est une fonction de m variables, alors la fonction

$$f(\bar{x}) = h(g_1(\bar{x}), \dots, g_m(\bar{x}))$$

est dans $\mathcal{C}$.

- (ii) Récursion primitive : si $g, h \in \mathcal{C}$ sont des fonctions partielles de respectivement n et $n+2$ variables, la fonction f définie par $f(\bar{x}, 0) = g(\bar{x})$ et $f(\bar{x}, m+1) = h(\bar{x}, m, f(\bar{x}, m))$ est alors une fonction partielle de $n+1$ variables dans $\mathcal{C}$.
- (iii) Minimisation : si $g \in \mathcal{C}$ est une fonction partielle de $n+1$ variables, alors la fonction partielle f qui à $\bar{x}$ associe le plus petit entier m , s'il existe, tel que $g(\bar{x}, i) \downarrow$ pour tout $i \leq m$ et tel que $g(\bar{x}, m) = 0$, est dans $\mathcal{C}$. Si m n'existe pas, alors f n'est pas définie sur $\bar{x}$.

La classe des *fonctions primitives récursives* est la plus petite classe de fonctions totales contenant les fonctions de (a),(b),(c) et close par les schémas (i) et (ii) de composition et récursion primitive. $\diamond$

Les fonctions générales récursives présentent l'avantage de mettre en valeur les propriétés de clôture de la notion de fonction calculable, à commencer par la clôture par composition. Notons que les fonctions primitives récursives sont totales, contrairement aux fonctions générales récursives, pour lesquelles le schéma de minimisation introduit une possibilité de partialité. Intuitivement, son implémentation en programmation consisterait

en une boucle **while** cherchant exhaustivement un entier m satisfaisant la propriété. Si cet entier n'existe pas, l'exécution du programme ne sortira jamais de la boucle, et le programme ne s'arrêtera pas. Nous étudierons ce modèle de manière détaillée dans la section 3 en nous efforçant de montrer qu'il correspond bien à la notion de fonction effectivement calculable. Mentionnons avant cela deux autres modèles de calcul.

Le λ -calcul. Défini par Church dans les années 30, le λ -calcul est un formalisme minimaliste servant de fondement théorique aux langages de programmation. Contrairement aux fonctions générales récursives, qui manipulent deux types d'objets, à savoir les entiers et les fonctions sur les entiers, le λ -calcul n'a qu'un seul objet primitif : les λ -fonctions. Celles-ci ne prennent pas en paramètre des entiers, mais des λ -fonctions. Il faudra donc recourir au codage des entiers par des λ -fonctions pour définir une notion de fonction calculable sur les entiers.

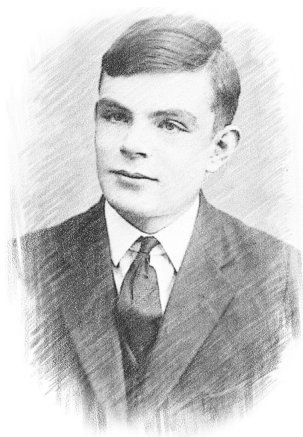
Les λ -fonctions sont définies par un langage d'expressions, comme c'est souvent le cas en mathématiques. Par exemple, $x, y \mapsto x + y$ est une expression définissant la somme de deux entiers. Cependant, en λ -calcul, les paramètres étant eux-mêmes des λ -fonctions, les seules opérations valides sont celles de manipulation de fonctions, à savoir, l'ajout d'un paramètre à une fonction, et l'application d'une fonction à ses paramètres. Par exemple, $f \mapsto (g \mapsto f(g))$ définit la fonction qui prend en paramètre une fonction f , et renvoie la fonction qui prend en paramètre une fonction g , et renvoie le résultat de l'application de f à g .

L'aspect minimaliste du λ -calcul facilite les raisonnements sur le formalisme en lui-même, mais demande beaucoup plus de travail pour définir des fonctions non triviales sur les entiers. Il est notamment plus difficile de se convaincre que toutes les fonctions calculables peuvent-être représentées par des λ -fonctions. Comme exprimé précédemment, il est nécessaire de fixer une convention pour représenter les entiers. Il paraît assez naturel d'identifier l'entier n avec la λ -fonction prenant en entrée une λ -fonction f et retournant sa n -ième itération. Par exemple, l'entier 0 est représenté par la fonction qui prend en entrée une fonction f , et renvoie sa 0-ième itération, autrement dit renvoie la fonction identité. Dans le formalisme du λ -calcul, elle s'écrit $f \mapsto (x \mapsto x)$. De la même manière, l'entier 2 correspond à la fonction $f \mapsto (x \mapsto f(f(x)))$. Appelons $\bar{n}$ la λ -fonction représentant l'entier n ; une fonction $g : \mathbb{N} \rightarrow \mathbb{N}$ est λ -définissable s'il existe une λ -fonction h qui à $\bar{n}$ associe $\overline{g(n)}$.

La syntaxe du λ -calcul est assez absconse au premier abord, et demande un peu de manipulation pour se familiariser avec les concepts. De nos jours, de nombreuses variantes et enrichissements sont étudiés afin de fournir un fondement théorique aux langages de programmation fonctionnelle. Il s'agit d'un domaine de recherche très actif.

Les machines de Turing. Si le λ -calcul fournit une base théorique aux langages de programmation, les machines de Turing peuvent être vues comme précurseurs de l'ordinateur moderne.

Conçue en 1936 par Alan Turing, cette machine est inspirée de la machine à écrire de son père. Elle comporte un *ruban*, qui peut être vu comme une succession de *cases* indexées par des entiers. On peut faire l'analogie avec des bits auxquels on accède dans la mémoire d'un ordinateur via un système d'adressage. La machine dispose aussi d'une tête de lecture/écriture qui peut se déplacer d'une case à l'autre, puis en lire ou en modifier le contenu. La machine va s'exécuter en fonction d'une entrée n . Au début du calcul, la tête de lecture se trouve au début du ruban, dont les n premières cases sont initialisées à 1, tandis que les cases restantes valent 0.



Alan Turing, 1912–1954

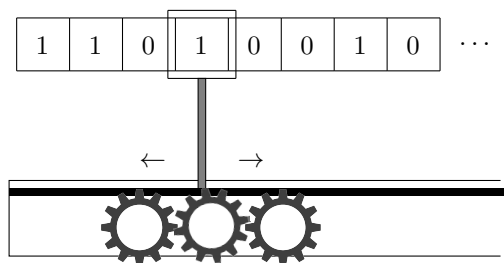


FIGURE 1.3 – Représentation d'une machine de Turing, avec sa tête de lecture/écriture se déplaçant sur les cases du ruban

La machine possède également un nombre fini d'*états*, incluant un état de départ dans lequel elle se trouve au début du calcul, ainsi qu'un état d'arrêt qui indique la fin du calcul lorsque la machine l'atteint. Les déplacements de la tête de lecture, le remplacement de la valeur de la case courante, ainsi que les changements d'état, sont soumis à un ensemble d'instructions représentées comme suit : étant donné l'état courant de la machine et la valeur de la case où se trouve la tête de lecture, une règle va décider de l'état suivant de la machine, de changer éventuellement la valeur de la case, et

de déplacer la tête d'une case à droite ou à gauche. Une machine de Turing est donc un automate élaboré conçu pour réaliser une tâche spécifique. Dans son article fondateur, Turing a démontré l'existence d'une machine de Turing *universelle* : une machine permettant de simuler le calcul de n'importe quelle autre machine de Turing.

Contrairement aux fonctions générales récursives ou au λ -calcul, le modèle de Turing constitue un processus mécanique très concret. On peut de fait réellement construire une machine de Turing universelle. Ce modèle a notamment l'avantage de rendre explicite la notion d'étape atomique de calcul ainsi que de quantifier l'espace mémoire utilisé. Pour ces raisons, les machines de Turing sont prises comme modèle de référence pour définir la théorie de la complexité algorithmique.

S'abstraire des modèles. À l'instar d'une citation de Michael Fellows², la calculabilité n'est pas plus l'étude des modèles de calcul que l'astronomie n'est la science des télescopes. La calculabilité s'émancipe très rapidement des détails d'implémentation des modèles de calcul, pour manipuler les programmes de manière abstraite. Il est malgré tout instructif de voir en détail au moins un modèle de calcul, en particulier pour se forger une intuition quand celle-ci fait défaut, et c'est ce que nous ferons très bientôt dans la section 3.

2. Thèse de Church-Turing

En 1934, face au succès du λ -calcul, Church soumit l'idée à Gödel selon laquelle les fonctions λ -définissables captureraient la notion de fonction effectivement calculable, laissant Gödel dubitatif. Avec le développement des fonctions générales récursives de Herbrand et Gödel la même année et la preuve de leur équivalence avec le λ -calcul, Church formula publiquement sa thèse, connue sous le nom de *thèse de Church*, affirmant que les fonctions générales récursives coïncidaient avec les fonctions effectivement calculables. Son argumentation ne convainc pas Gödel, bien qu'il fût l'un des instigateurs des fonctions générales récursives. Avec son modèle de machine éponyme, Alan Turing fit finalement consensus en 1936, en démontrant ce dernier équivalent au modèle du λ -calcul et à celui des fonctions générales récursives, ce qui conduisit à ce que l'on appelle aujourd'hui la thèse de Church-Turing.

2. « L'informatique n'est pas plus l'étude des ordinateurs, que l'astronomie n'est celle des télescopes. » [59]

Thèse 2.1 (Church-Turing).

Les fonctions effectivement calculables sont celles calculables par une machine de Turing, ou de manière équivalente les fonctions générales récursives ou encore les fonctions λ -définissables. ■

Si Church fut le premier à formuler la thèse selon laquelle les fonctions λ -définissables correspondaient aux fonctions effectivement calculables, on attribue généralement la paternité de la calculabilité à Turing. La thèse de Church-Turing n'étant pas un énoncé mathématique, il n'est pas possible de la prouver formellement. Elle peut néanmoins être validée par ce qui se rapproche le plus d'une preuve au sens social du terme, c'est-à-dire par un argument engendrant un consensus de la communauté scientifique. C'est ce à quoi Turing est parvenu par la démonstration suivante.

La démonstration de Turing. Pour justifier sa thèse, Turing a employé trois types d'arguments :

- (1) une description du processus par lequel un mathématicien effectue un calcul et sa formalisation par une machine de Turing,
- (2) la preuve de l'équivalence des machines de Turing avec les modèles de calculs existants,
- (3) le développement explicite de grandes classes de fonctions calculables par les machines de Turing. Voici les grandes lignes du premier argument de Turing :

Considérons un mathématicien, M. Dupont, effectuant un calcul. M. Dupont possède un crayon et une quantité potentiellement illimitée de papier. Étant donné la précision finie de son crayon, chaque feuille ne peut contenir qu'un nombre fini de symboles. Par simplification, et sans perte de généralité, on peut considérer que chaque feuille est une case d'un ruban infini, ne contenant qu'un seul symbole appartenant à un alphabet fini suffisamment grand. Le processus de calcul est le suivant : alors qu'il se trouve dans un état mental e_0 , M. Dupont se situe face à la feuille courante, dans son champ de vision. Il lit les notes, avant de les corriger, en effaçant et changeant le symbole écrit sur la feuille. Cette lecture va faire évoluer ses pensées, et il se retrouvera dans un état mental e_1 . Il va potentiellement tourner la page, ou revenir en arrière pour relire des notes précédentes, jusqu'à arriver à la fin de son calcul. M. Dupont considérera alors son calcul terminé, et se retrouvera dans l'état mental correspondant que l'on appellera *état final*.

Du bon usage de la thèse de Church-Turing. Il convient de se faire une idée claire de ce que dit la thèse de Church-Turing, de ses limites et de son usage. La thèse de Church-Turing n'est ni un théorème, ni une conjecture. Elle ne peut être formellement prouvée ou réfutée, par le simple fait qu'elle

n'est pas un énoncé mathématique, mais plutôt un pont entre un concept mathématique et une notion épistémologique. Cette thèse n'est pas pour autant une affirmation arbitraire, car elle est étayée par un raisonnement qui peut être soumis à la critique.

Si la thèse de Church-Turing peut être remise en question, voire un jour majoritairement rejetée, le développement de la calculabilité n'en repose pas moins sur des bases solides, indépendantes de cette correspondance. L'équivalence entre les fonctions calculables par les machines de Turing, les fonctions générales récursives, les fonctions λ -définissables, et les fonctions programmables en C, Java ou Python, est bien un théorème qui ne dépend pas de la thèse de Church-Turing. S'il est fréquent en calculabilité de décrire de manière informelle un algorithme pour ensuite en déduire l'existence d'une machine de Turing l'implémentant, ce procédé n'est pas à strictement parler un appel à la thèse de Church-Turing. Il s'agit plutôt d'une preuve informelle permettant de convaincre l'interlocuteur que, si nécessaire, il serait aisé de programmer cet algorithme dans un quelconque langage de programmation.

En outre, si la thèse de Church-Turing venait à être invalidée, l'édifice conceptuel construit autour de la calculabilité resterait, et continuerait vraisemblablement à être étudié. Il existe une hiérarchie de langages formels et de modèles de calcul, appelée *hiérarchie de Chomsky*. On y trouve par exemple les *langages rationnels*, correspondant aux langages reconnus par une classe de machines appelées *automates finis*. Bien que ces modèles soient pour la plupart moins expressifs que les machines de Turing, ils n'en sont pas moins un sujet très actif de recherche de nos jours. Si l'on trouve un jour de nouveaux paradigmes de calcul, la notion existante de fonction calculable n'en demeurera pas moins une classe de fonctions très robuste, et continuera à être étudiée au même titre que les langages rationnels ou tout autre niveau de la hiérarchie de Chomsky.

Certains phénomènes naturels sont étudiés dans l'espoir de résoudre des problèmes non calculables. Ces notions de calculabilité sont réunies sous le nom d'*hypercalcul* (nous en verrons une approche formelle dans la partie IV). Il n'existe pas à ce jour de perspective de réalisation de tels calculs. La découverte de nouveaux phénomènes de calcul dans la nature n'invaliderait cependant probablement pas la thèse de Church-Turing, car ils auraient peu de chances de satisfaire la définition de fonction effectivement calculable d'après Rosser [184], c'est-à-dire « une méthode dont chaque étape est précisément prédéterminée et qui produira de manière certaine une réponse en un nombre fini d'étapes. »

3. Étude détaillée des fonctions récursives

L'objectif de cette section est de convaincre le lecteur que les fonctions générales récursives coïncident avec les fonctions effectivement calculables telles que définies informellement dans les sections 3-2 et 3-1, comme étant « les fonctions que l'on peut programmer ». Les intérêts d'une telle étude sont multiples. Elle permet en premier lieu d'avoir une définition mathématique précise de ce qu'est une fonction calculable : il s'agit sans perte de généralité des fonctions générales récursives. Ensuite, l'étude que nous donnons fournira du même coup une preuve mathématique de l'existence d'une machine universelle telle que définie dans le théorème 3-3.1, concept utilisé tout au long de cet ouvrage. Pour finir, notre étude isolera la notion de fonction primitive récursive comme une sous-classe stricte des fonctions calculables ; outre une certaine importance épistémologique (voir le théorème 3.22), cette sous-classe présente un indéniable intérêt en logique mathématique. Nous en verrons un exemple d'utilisation dans l'étude des mathématiques à rebours, à la section 23-7.

3.1. Machines à registres et diagrammes de programmation

Pour nous convaincre que les fonctions générales récursives coïncident avec les fonctions calculables, nous partons d'un modèle de calcul proche des langages de programmation modernes : *les programmes structurés*, exécutés par des *machines à registres*. Nos développements reprennent ici dans les grandes lignes le cours de calculabilité d'A. Durand et P. Rozière [51] du Master de logique mathématique de l'université Paris Diderot.

Les machines à registres. La littérature scientifique en a décliné plusieurs versions, sous le nom de « random access machine » (Melzak [155], Lambek [135], Shepherdson and Sturgis [197], Peter [174], Elgot et Robinson [55]). Il s'agit de machines dites à « random access memory » ou RAM, nom générique aujourd'hui pour désigner la mémoire vive des ordinateurs. Le terme « random access » (accès aléatoire) doit être compris par opposition à « sequential access » (accès séquentiel), et fait référence au fait que l'on peut accéder à chaque case mémoire directement à partir de son adresse, contrairement par exemple au modèle des machines de Turing, pour lesquelles une tête de lecture doit se déplacer case après case dans la mémoire pour arriver à l'endroit désiré.

La mémoire d'une machine à registre sera toutefois plus élémentaire qu'une mémoire RAM moderne : il s'agit simplement d'un nombre fini de *registres* $R_0, R_1, \dots, R_k$ pour $k \in \mathbb{N}$ arbitraire. Chaque registre peut contenir un entier quelconque positif ou nul. Notons que les entiers peuvent être arbitrairement grands, et donc que chaque registre représente un espace mémoire « non borné ».

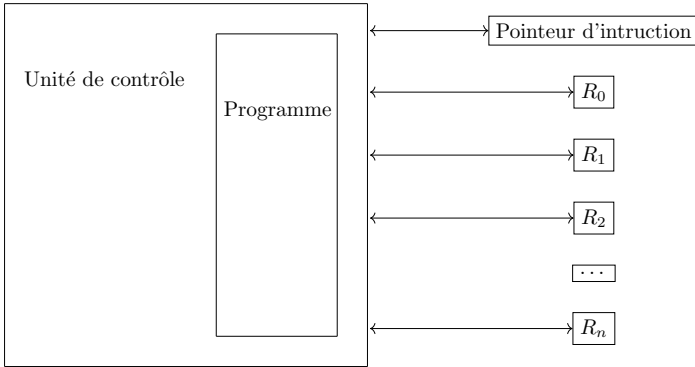


FIGURE 3.1 – Le modèle des machines à registres

Les programmes structurés. Une machine à registre va exécuter un programme qui consiste en une liste finie d'instructions permettant de faire des calculs. Il existe de nombreuses variantes possibles quant au jeu d'instruction que l'on s'autorise. Nous en présentons un, volontairement proche de celui d'un langage de programmation impérative moderne.

Définition 3.2. Un *programme structuré* peut contenir les instructions suivantes.

1. ▷ Incrémentation d'un registre : $\ll R_i := R_i + 1 \gg$.
 ▷ Décrémentent d'un registre : $\ll R_i := R_i - 1 \gg$.
 ▷ Assignment d'un registre : $\ll R_i := x \gg, x \in \mathbb{N}$, ou $\ll R_i := R_j \gg$.

Ces instructions respectivement incrémentent la valeur de R_i de 1, la décrémentent de 1 (sauf si la valeur est de 0, auquel cas rien ne se passe), mettent la valeur de R_i à x ou la mettent à celle de R_j .

2. L'instruction conditionnelle : $\ll \text{if } R_i = 0 \text{ then } S \text{ else } S' \gg$, où les suites finies $S = (S_0, \dots, S_n)$ et $S' = (S'_0, \dots, S'_m)$ sont des suites d'instructions structurées.

Chaque instruction de S est exécutée séquentiellement si R_i est égal à 0. Sinon, chaque instruction de S' est exécutée séquentiellement.

3. L'instruction de boucle for : $\ll \text{for } i = 1 \text{ to } R_i \text{ do } S \gg$, où la suite finie $S = (S_0, \dots, S_n)$ est une suite d'instructions structurées.

Soit N le nombre présent dans le registre R_i au moment où le programme commence cette instruction. Chaque instruction de S est exécutée séquentiellement, le tout N fois. Notez que si la valeur de R_i change pendant l'exécution des instructions de S , cela ne change pas le nombre de fois que la boucle s'effectue.

4. L'instruction de boucle `while` : « `while  $R_i \neq 0$  do  $S$` » où la suite finie $S = (S_0, \dots, S_n)$ est une suite d'instructions structurées.

Chaque instruction de S est exécutée séquentiellement tant que le registre R_i est différent de 0.

L'exécution d'un programme structuré s'arrête une fois la dernière instruction exécutée. ◇

Notons qu'un programme structuré n'a qu'un nombre fini d'instructions et ne peut donc utiliser qu'un nombre fini de registres.

Boucle `for` vs Boucle `while`

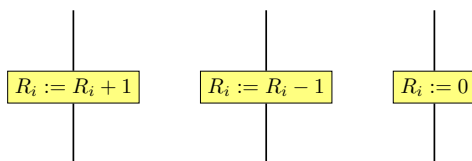
Le lecteur remarquera peut-être que l'instruction de boucle `for` est redondante dans la mesure où elle peut toujours être remplacée par une instruction de boucle `while`. Nous verrons que l'inverse n'est pas vrai. En particulier, le nombre de fois qu'une boucle `for` s'exécute est nécessairement fini, ce qui n'est pas le cas des boucles `while`, au sein desquelles un calcul peut se retrouver bloqué dans ce que l'on appelle classiquement en programmation *une boucle infinie*. Nous verrons que l'instruction de boucle `while` est indispensable, et que certaines fonctions calculables (et totales) ne peuvent se programmer en n'utilisant que des boucles `for`.

Les diagrammes de programmation. La programmation structurée trouve sa genèse dans les travaux de Goldstine et von Neumann [75], qui manifestent, dès 1946, un souci non plus de capturer mathématiquement la notion de calcul, mais celui de développer un système de programmation. Ils développent une manière de présenter des algorithmes à base de *diagrammes de programmation*.

Nous en donnons ici la présentation simplifiée que l'on trouve dans l'ouvrage de référence de Piergiorgio Odifreddi [169].

Définition 3.3. Un diagramme de programmation est obtenu en connectant entre elles des briques de base des deux types suivants.

▷ Des instructions d'assignements :



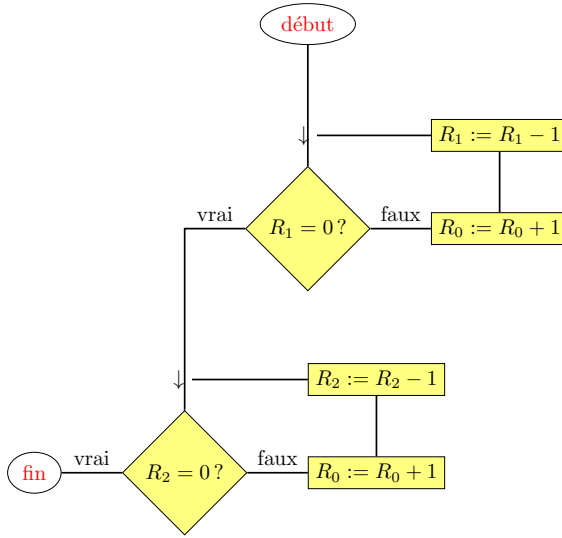
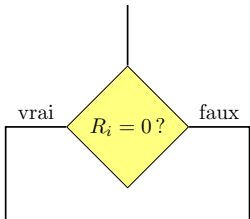


FIGURE 3.4 – Un diagramme de programmation pour la fonction d'addition de deux nombres. On suppose que le calcul démarre avec $R_0 = 0$, $R_1 = n_1$ et $R_2 = n_2$. À la fin de l'exécution, on aura R_0 égal à $n_1 + n_2$.

▷ Une instruction conditionnelle :



Un diagramme de programmation possède une entrée et une ou plusieurs sorties. Le calcul s'effectue de manière linéaire en exécutant chaque bloc depuis l'entrée vers d'une de ses sorties. ◇

À titre d'exemple, la figure 3.4 est un diagramme de programmation correspondant à la fonction d'addition.

Les diagrammes de programmation peuvent être programmés sur des machines à registre avec un jeu d'instructions comportant des sauts conditionnels et inconditionnels (ces derniers étant simplement appelés « sauts »), c'est-à-dire des instructions de type *goto* en anglais, qui permettent de déterminer quelle est la prochaine instruction du programme qui sera exécutée. Il s'agit encore aujourd'hui du mécanisme à l'œuvre dans les différents

langages assembleurs des micro-processeurs.

Définition 3.5. Un programme *goto* est une suite numérotée d'instructions $I_0, I_1, \dots, I_n$ où chaque instruction est de l'un des types suivants :

1. ▷ Incrémentation d'un registre : $\ll R_i := R_i + 1 \gg$.
 ▷ Décrémentent d'un registre : $\ll R_i := R_i - 1 \gg$.
 ▷ Assignement d'un registre à 0 : $\ll R_i := 0 \gg$.
2. L'instruction de saut conditionnel : $\ll \text{if } R_i = 0 \text{ goto } n_1 \text{ else goto } n_2 \gg$.
 Si R_i est égal à 0, l'instruction numéro n_1 est la prochaine à être exécutée, sinon c'est l'instruction numéro n_2 .
3. Le saut incondionnel : $\ll \text{goto } m \gg$.

L'instruction numéro m est la prochaine à être exécutée.

L'exécution d'un programme *goto* s'arrête quand il n'y a plus de prochaine instruction à exécuter (ce qui en particulier peut arriver après une instruction de type $\ll \text{goto } m \gg$ dans un programme comportant moins de m instructions.) ◇

Remarque

Notons que le saut incondionnel $\ll \text{goto } m \gg$ peut être remplacé par un saut conditionnel $\ll \text{if } R_i = 0 \text{ goto } m \text{ else goto } m \gg$. Un programme *goto* peut donc être exprimé sans l'instruction 3.

Il est clair que les diagrammes de programmation sont interchangeables avec les programmes de type *goto*, et nous utiliserons un formalisme ou l'autre en fonction de la situation.

Fonctions calculables. Les notations $\Phi_e(x_1, \dots, x_n) \downarrow = y$ et $\Phi_e(x_1, \dots, x_n) \uparrow$ utilisées tout au long du livre s'appliquent naturellement aux programmes exécutés par des machines à registres, une fois les conventions de passage des paramètres et de récupération du résultat fixées.

Définition 3.6. Étant donné P un programme de type structuré ou bien *goto*, on écrit $P(x_1, \dots, x_k)$ pour désigner l'exécution de P avec les registres $R_1, \dots, R_k$ initialisés à $x_1, \dots, x_k$, et tous les autres registres initialisés à 0. On écrit $P(x_1, \dots, x_k) \downarrow = x$ pour signifier qu'une telle exécution s'arrête avec la valeur x dans le registre R_0 , et $P(x_1, \dots, x_k) \uparrow$ pour signifier que l'exécution ne s'arrête pas. ◇

On peut à présent utiliser notre modèle de calcul pour donner une définition mathématique précise de fonction calculable. Officialisons auparavant les notations $\text{dom } f$ et $\text{Im } f$, qui désignent respectivement le domaine et l'image d'une fonction f .

Notation

Étant donné une fonction (éventuellement partielle) $f : A \rightarrow B$, on note $\text{dom } f$ le domaine de définition de f , et

$$\text{Im } f = \{X \in B : \exists Y \in \text{dom } f \ f(Y) = X\}$$

son image.

Définition 3.7. Une fonction (éventuellement partielle) $f : \mathbb{N}^n \rightarrow \mathbb{N}$ est *calculable par programme structuré* (resp. *par programme goto*) s'il existe un programme structuré (resp. un programme goto) P_f , tel que

$$P_f(x_1, \dots, x_n) \downarrow = f(x_1, \dots, x_n)$$

pour tous éléments $x_1, \dots, x_n \in \text{dom } f$ et tel que $P_f(x_1, \dots, x_n) \uparrow$ pour tous éléments $x_1, \dots, x_n \notin \text{dom } f$. ◇

Justification du modèle de calcul. Le programmeur ne sera peut-être pas convaincu par le fait que le modèle des machines à registres avec programmes structurés (ou goto) permet effectivement de programmer toutes les fonctions qu'il pourrait écrire dans son langage favori.

Un aspect en particulier peut susciter l'inquiétude : un langage de programmation moderne permet l'utilisation de tableaux, et même de tableaux dynamiques, dont la taille peut augmenter au fur et à mesure des besoins. Nous verrons par exemple dans la définition 3.24 que la fonction dite d'Ackermann se calcule naturellement à l'aide d'une structure de pile, et qu'il n'est pas du tout clair que l'on puisse s'en passer. Fort heureusement, nous montrerons avec la proposition 3.26 qu'il est parfaitement possible de simuler les manipulations de tableaux dynamiques au sein des machines à registres, en codant ces derniers dans les registres, qui, rappelons-le peuvent contenir des entiers arbitrairement grands.

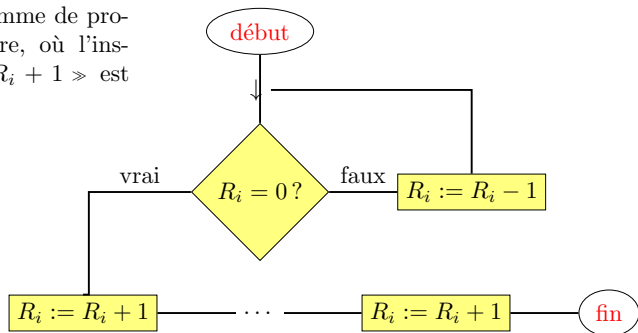
Simulation des programmes structurés par les programmes goto.

Nous commençons par montrer que les programmes goto, malgré leur simplicité, sont suffisants pour calculer tout ce que peuvent calculer les programmes structurés.

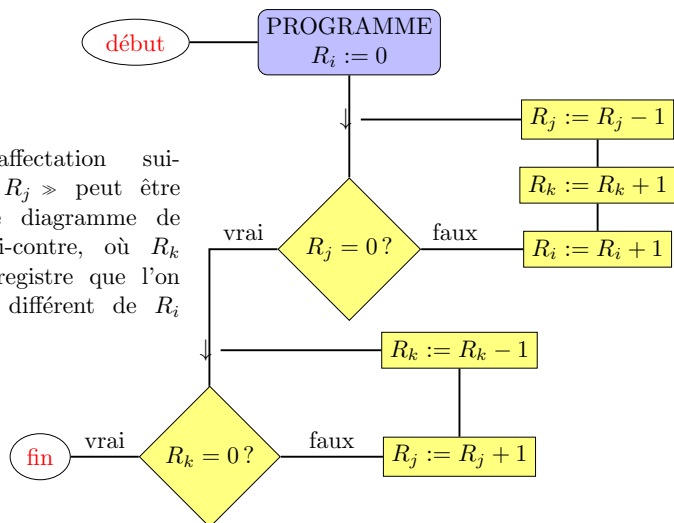
Proposition 3.8. Soit $n \in \mathbb{N}^*$, et soit $f : \mathbb{N}^n \rightarrow \mathbb{N}$ une fonction calculable par un programme structuré. Alors, la fonction f est calculable par un programme goto. ★

PREUVE. Il suffit de montrer que chaque instruction d'un programme structuré peut être remplacée par un diagramme de programmation.

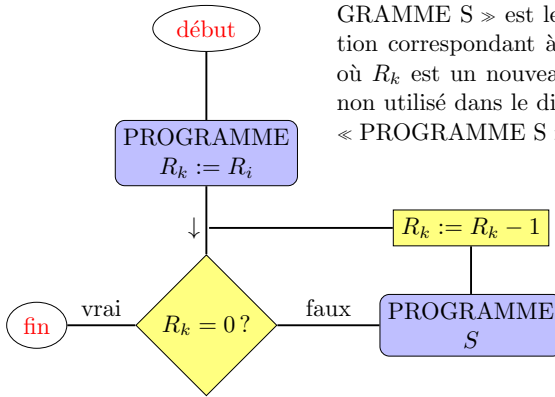
L'instruction d'affectation suivante « $R_i := n$ » peut être remplacée par le diagramme de programmation ci-contre, où l'instruction « $R_i := R_i + 1$ » est répétée n fois.



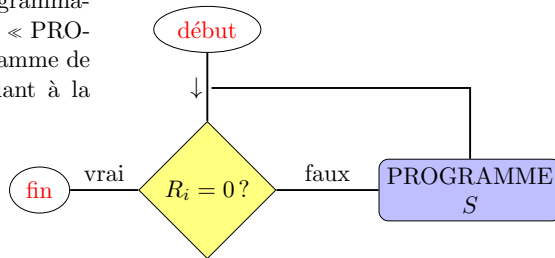
L'instruction d'affectation suivante « $R_i := R_j$ » peut être remplacée par le diagramme de programmation ci-contre, où R_k est un nouveau registre que l'on suppose à 0, et différent de R_i et R_j .



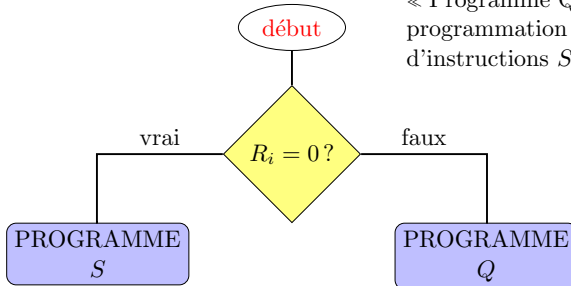
L'instruction de boucle « for $i = 1$ to R_i do S » peut être remplacée par le diagramme de programmation ci-contre, où la boîte « PROGRAMME S » est le diagramme de programmation correspondant à la liste d'instructions S , et où R_k est un nouveau registre différent de R_i et non utilisé dans le diagramme de programmation « PROGRAMME S ».



L'instruction de boucle « while $R_i \neq 0$ do S » peut être remplacée par le diagramme de programmation ci-contre, où la boîte « PROGRAMME S » est le diagramme de programmation correspondant à la liste d'instructions S .



L'instruction d'affectation « if $R_i = 0$ then S else Q » peut être remplacée par le diagramme de programmation ci-contre, où les deux boîtes « Programme S » et « Programme Q » sont les diagrammes de programmation correspondant aux suites d'instructions S et Q .



■

3.2. Les fonctions récursives sont calculables.

Les programmes structurés suivent les concepts de la programmation dite *impérative* : des instructions modifiant l'état de la machine sont exécutées les une après les autres. Les fonctions générales récursives suivent quant à elles le paradigme de la programmation dite *fonctionnelle* : un programme est une composition de fonctions mathématiques et un calcul l'évaluation de ces fonctions. Un avantage de la programmation fonctionnelle souvent mis en avant est l'absence d'*effets de bord* : le résultat d'une fonction dépend de ses paramètres et uniquement de ses paramètres (l'état de la machine sur laquelle la fonction s'exécute n'a pas d'incidence). On peut par exemple brancher la sortie d'une fonction sur l'entrée d'une autre sans s'attendre à de mauvaises surprises. À titre de comparaison, la combinaison de programmes structurés P_f, P_g calculant des fonctions $f, g : \mathbb{N} \rightarrow \mathbb{N}$ en un programme calculant la fonction $x \mapsto f(g(x))$ demande un peu de travail, afin justement d'éviter les effets de bord.

Définition 3.9. Un programme de type structuré ou bien de type goto est *propre* s'il termine son calcul avec tous ses registres — à l'exception de R_0 — dans le même état qu'au début du calcul. $\diamond$

Exercice 3.10. Montrer que pour tout programme structuré M , il existe un programme structuré N propre tel que $M(\bar{x}) \downarrow = y \leftrightarrow N(\bar{x}) \downarrow = y$. $\diamond$

Théorème 3.11 (Wang [229], Peter [175], Ershov [57])

*Toute fonction générale récursive (éventuellement partielle) est calculable par un programme structuré. Toute fonction primitive récursive est calculable par un programme structuré sans boucle **while**.*

PREUVE. On laisse au lecteur le soin de montrer que les fonctions constantes, les fonctions de projection et la fonction successeur sont toutes calculables par un programme structuré. Il reste à traiter les cas suivants.

Schéma de composition. Soit

$$f(x_1, \dots, x_m) = g(h_1(x_1, \dots, x_m), \dots, h_k(x_1, \dots, x_m)),$$

pour des fonctions $g : \mathbb{N}^k \rightarrow \mathbb{N}$ et $h_1, \dots, h_k : \mathbb{N}^m \rightarrow \mathbb{N}$ calculables par des programmes structurés $G, H_1, \dots, H_m$. Par l'exercice 3.10, on peut supposer que $G, H_1, \dots, H_m$ sont propres. Supposons que chacun de ces programmes utilise au plus les registres $R_0, \dots, R_z$ pour $z > m$. Le programme F pour calculer f est donné par la suite d'instructions suivantes.

 Programme F

 $\langle \text{Instructions de } H_1 \rangle$
 $R_{z+1} := R_0$
 $R_0 := 0$
 $\langle \text{Instructions de } H_2 \rangle$
 $R_{z+2} := R_0$
 $R_0 := 0$
 $\dots$
 $\langle \text{Instructions de } H_k \rangle$
 $R_{z+k} := R_0$
 $R_0 := 0$
 $R_1 := R_{z+1}$
 $\dots$
 $R_k := R_{z+k}$
 $\langle \text{Instructions de } G \rangle$

Notons que si $G, H_1, \dots, H_m$ n'utilisent pas de boucle **while**, alors le programme pour calculer F n'en utilise pas non plus.

Schéma de récursion primitive. Soient

$$\begin{aligned} f(x_1, \dots, x_n, 0) &= g(x_1, \dots, x_n) \\ f(x_1, \dots, x_n, y+1) &= h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y)) \end{aligned}$$

pour g et h calculables respectivement par des programmes structurés propres G et H , utilisant au plus les registres $R_0, \dots, R_z$ pour $z > n+2$. Le programme F suivant permet de calculer f .

 Programme F

 $R_{z+1} := R_{n+1}$
 $\langle \text{Instructions de } G \rangle$
 $R_{n+1} := 0$
 $R_{n+2} := R_0$
for $i = 1$ **to** R_{z+1} **do**
 $R_0 := 0$
 $\langle \text{Instructions de } H \rangle$
 $R_{n+1} := R_{n+1} + 1$
 $R_{n+2} := R_0$
end

Notons une fois de plus que si G, H n'utilisent pas de boucles **while**, alors le programme pour calculer F n'en utilise pas non plus.

Schéma de minimisation. Soit

$$f(x_1, \dots, x_n) = \min\{x \in \mathbb{N} : \forall i \leq x \ (g(x_1, \dots, x_n, i) \downarrow \wedge g(x_1, \dots, x_n, x) = 0)\},$$

pour g calculable par un programme propre G , utilisant au plus les registres $R_0, \dots, R_z$ pour $z > n + 1$. Le programme suivant permet de calculer f .

Programme F

$R_{z+1} := 1$

$R_{n+1} := 0$

while $R_{z+1} \neq 0$ **do**

$R_0 := 0$

 ⟨Instructions de G ⟩

$R_{z+1} := R_0$

$R_{n+1} := R_{n+1} + 1$

end

$R_{n+1} := R_{n+1} - 1$

$R_0 := R_{n+1}$

Cela conclut la preuve. ■

3.3. Étude des fonctions primitives récursives

Nous passons à présent à la partie la plus difficile de ce chapitre. Afin de montrer que les fonctions calculables par programmes structurés sont des fonctions générales récursives, nous avons besoin d'un certain nombre d'outils, notamment sur les fonctions primitives récursives. Nous alternons différentes propositions et exercices permettant de voir qu'il s'agit d'une large classe de fonctions. Rappelons les notations de la définition 1.1 pour les fonctions primitives récursives de bases.

Notation

On note $p_i^n : \mathbb{N}^n \rightarrow \mathbb{N}$ la projection telle que $p_i^n(x_1, \dots, x_n) = x_i$. On note $c_i^n : \mathbb{N}^n \rightarrow \mathbb{N}$ la fonction constante telle que $c_i^n(x_1, \dots, x_n) = i$. On note $\text{succ} : \mathbb{N} \rightarrow \mathbb{N}$ la fonction successeur, définie par $\text{succ}(n) = n + 1$.

Exercice 3.12. (★) Montrer que l'addition, la multiplication et la fonction exponentielle sont primitives récursives. ◇

Exercice 3.13. (★) Montrer que les fonctions prédécesseur et soustraction sont primitives récursives (comme nos fonctions sont à valeur dans $\mathbb{N}$, on utilisera 0 à la place du résultat si celui-ci est négatif). ◇

Exercice 3.14. (★) Montrer que la fonction $\text{sg} : \mathbb{N} \rightarrow \mathbb{N}$ qui à 0 associe 0 et qui à tous les autres entiers associe 1 est primitive réursive. Montrer que la fonction $\overline{\text{sg}} : \mathbb{N} \rightarrow \mathbb{N}$ qui à 0 associe 1 et qui à tous les autres entiers associe 0 est primitive réursive. $\diamond$

Définition 3.15. Un prédicat $P \subseteq \mathbb{N}^n$ est primitif récursif (resp. général récursif) s'il existe une fonction primitive réursive (resp. générale réursive) $f : \mathbb{N}^n \rightarrow \{0, 1\}$ telle que $(x_1, \dots, x_n) \in P \leftrightarrow f(x_1, \dots, x_n) = 1$ pour tous $x_1, \dots, x_n \in \mathbb{N}$. $\diamond$

Exemple 3.16. Les prédicats de comparaison $\leq, <, \geq, >, =, \neq$ sont primitifs récursifs via les fonctions suivantes :

$$\begin{array}{llll} a \leq b & = & \text{sg}(\text{succ}(b) - a) & a > b & = & b < a \\ a < b & = & \text{sg}(b - a) & a = b & = & (a \leq b) \times (b \leq a) \\ a \geq b & = & b \leq a & a \neq b & = & \overline{\text{sg}}(a = b). \end{array}$$

Formellement, les projections sont utilisées si nécessaire, par exemple pour inverser l'ordre des paramètres dans la définition de $\geq$ à partir de celle de $\leq$.

Proposition 3.17. Les fonctions primitives récursives sont closes par définition par cas sur un prédicat primitif récursif : si g et h sont deux fonctions primitives récursives de $\mathbb{N}^p$ dans $\mathbb{N}$, et si P est un prédicat primitif récursif sur $\mathbb{N}^p$, alors la fonction

$$\begin{aligned} f(x_1, \dots, x_p) &= g(x_1, \dots, x_p) \quad \text{si } P(x_1, \dots, x_p) \\ &= h(x_1, \dots, x_p) \quad \text{sinon} \end{aligned}$$

est primitive réursive. $\star$

PREUVE. Comme P est un prédicat primitif récursif, il existe une fonction $d : \mathbb{N}^n \rightarrow \mathbb{N}$ telle que

$$\begin{aligned} d(x_1, \dots, x_p) &= 1 \quad \text{si } P(x_1, \dots, x_p) \\ &= 0 \quad \text{sinon.} \end{aligned}$$

On définit donc :

$$\begin{aligned} f(x_1, \dots, x_p) &= g(x_1, \dots, x_p) \times \text{sg}(d(x_1, \dots, x_p)) \\ &\quad + h(x_1, \dots, x_p) \times \overline{\text{sg}}(d(x_1, \dots, x_p)). \quad \blacksquare \end{aligned}$$

La preuve de la proposition suivante fournit un exemple d'application de la définition par cas.

Proposition 3.18. La division entière est primitive récursive. ★

PREUVE. On utilise le schéma de récursion primitive couplé au schéma de définition par cas. On définit $\text{div}(a, b) = \text{div}(a, b, a)$, où :

$$\begin{aligned} \text{div}(a, b, 0) &= 0 \\ \text{div}(a, b, n+1) &= \text{succ}(n) && \text{si } \text{succ}(n) \times b = a \\ &= \text{div}(a, b, n) && \text{sinon} \end{aligned} \quad \blacksquare$$

Exercice 3.19. (★) Montrer que les prédicats primitifs récursifs sont clos par conjonction, disjonction, négation, quantification existentielle bornée et quantification universelle bornée. ◇

Exercice 3.20. (★) Montrer que les fonctions primitives récursives sont closes par minimisation bornée : si $f : \mathbb{N}^{p+1} \rightarrow \mathbb{N}$ est primitive récursive, alors la fonction $g : \mathbb{N}^{p+1} \rightarrow \mathbb{N}$ qui à $x_1, \dots, x_p, n$ associe le plus petit $t \leq n$ tel que $f(x_1, \dots, x_p, t) = 0$ (et 0 si un tel $t \leq n$ n'existe pas) est primitive récursive. ◇

Souvenons-nous des bijections de Cantor comme définies dans la section 2-3. Le codage des paires $\langle x_1, x_2 \rangle$ est une notation pour l'application de la bijection $\alpha_2 : \mathbb{N}^2 \rightarrow \mathbb{N}$ définie par $\alpha_2(x, y) = y + \frac{(x+y+1)(x+y)}{2}$. Le codage $\langle x_1, \dots, x_k \rangle$ des n -uplets est une notation pour l'application des bijections $\alpha_k : \mathbb{N}^k \rightarrow \mathbb{N}$ définies inductivement par

$$\alpha_{k+1}(x_1, x_2, \dots, x_{k+1}) = \alpha_2(x_1, \alpha_k(x_2, \dots, x_{k+1})).$$

Proposition 3.21. Pour tout n , la bijection

$$x_1, \dots, x_n \mapsto \langle x_1, \dots, x_n \rangle$$

est primitive récursive. Pour tout n et tout $i \leq n$, la fonction

$$\langle x_1, \dots, x_n \rangle \mapsto x_i$$

est primitive récursive. ★

PREUVE. L'addition, la multiplication et la division entière étant primitives récursives, la fonction $(x, y) \mapsto y + \frac{(x+y+1)(x+y)}{2}$ est elle aussi primitive récursive par le schéma de composition. Il en va donc de même pour tout n pour les fonctions $x_1, \dots, x_n \mapsto \langle x_1, \dots, x_n \rangle$ qui sont définies inductivement par composition.

La fonction qui à $\langle x_1, x_2 \rangle$ associe x_1 est primitive récursive, fait qui s'établit en utilisant la minimisation bornée et la clôture des prédicats primitifs

récur­sifs par quantification existentielle bornée :

$$f(n) = \min\{x \leq n : \exists y \leq n \langle x, y \rangle = n\}.$$

On laisse au lecteur le soin de broder sur cette idée pour montrer que toutes les projections sont ainsi primitives récur­sives. ■

Nous avons à ce stade tous les éléments nécessaires pour montrer un premier théorème important. Nous avons vu avec le théorème 3.11 que les fonctions primitives récur­sives peuvent être programmées par des programmes structurés n'utilisant pas de boucle **while**. La réciproque est vraie.

Théorème 3.22

*Toute fonction calculable par un programme structuré sans boucles **while** est primitive récur­sive.*

PREUVE. Pour $k \in \mathbb{N}$ donné, et pour un programme structuré P sans boucles **while** et utilisant au plus les registres $R_0, \dots, R_k$, on définit la fonction $f_P : \mathbb{N} \rightarrow \mathbb{N}$ par $f_P(\langle x_0, \dots, x_k \rangle) = \langle v_0, \dots, v_k \rangle$ où v_i est la valeur du registre R_i en fin d'exécution du programme P , quand son exécution commence avec ses registres initialisés aux valeurs $x_0, \dots, x_k$.

Montrons que pour tout programme structuré P sans boucles **while**, la fonction f_P correspondante est primitive récur­sive. Il est clair que c'est le cas pour le programme vide. Soit Q le programme dont l'unique instruction est $R_i := R_i + 1$. Alors, la fonction f_Q est donnée par

$$f_Q(\langle x_0, \dots, x_k \rangle) = \langle x_0, \dots, x_i + 1, \dots, x_k \rangle.$$

On laisse au lecteur le soin de trouver la fonction primitive récur­sive correspondant aux instructions $R_i := R_i - 1$, $R_i := c$ pour $c \in \mathbb{N}$ et $R_i := R_j$.

Supposons à présent que la proposition est vraie pour des programmes P, P' , via des fonctions $f_P, f_{P'}$, et soit Q le programme « if $R_j = 0$ then P else P' ». La fonction f_Q est donc donnée par

$$\begin{aligned} f_Q(\langle x_0, \dots, x_k \rangle) &= f_P(\langle x_0, \dots, x_k \rangle) && \text{si } x_j = 0 \\ &= f_{P'}(\langle x_0, \dots, x_k \rangle) && \text{sinon.} \end{aligned}$$

Supposons à présent que la proposition est vraie pour des programmes P , via des fonctions f_P , et soit Q de forme suivante : « for $i = 1$ to R_j do P ». La fonction f_Q est donnée par :

$$f_Q(\langle x_0, \dots, x_k \rangle) = g(\langle x_0, \dots, x_k \rangle, x_j)$$

où

$$\begin{aligned} g(\langle x_0, \dots, x_k \rangle, 0) &= \langle x_0, \dots, x_k \rangle \\ g(\langle x_0, \dots, x_k \rangle, z + 1) &= f_P(g(\langle x_0, \dots, x_k \rangle, z)). \end{aligned}$$

Supposons à présent la proposition vraie pour des programmes P, P' , via des fonctions $f_P, f_{P'}$, et soit Q composé des instructions de P suivies de celles de P' . Alors, $f_Q = f_{P'}(f_P(\langle x_0, \dots, x_k \rangle))$.

En utilisant chacun des cas décrits, on montre par récurrence que l'état des registres de tout programme structuré sans boucles **while** est une fonction primitive récursive. Il suffit alors de récupérer la valeur du registre R_0 . ■

3.4. Étude de la fonction d'Ackermann

Il existe plusieurs manières de voir que les fonctions calculables ne sont pas toutes primitives récursives. La suivante est la plus naturelle pour l'expert en calculabilité pour lequel les diagonalisations effectives n'ont plus de secret.

Exercice 3.23. (★) Montrer qu'il existe un ensemble calculable $A \subseteq \mathbb{N}$ tel que $n \mapsto \Phi_e(n)$ est une fonction primitive récursive pour tout $e \in A$ et tel que toute fonction primitive récursive a un code dans A . En déduire qu'il existe une fonction totale calculable qui n'est pas primitive récursive. ◇

Un exemple couramment donné de fonction non primitive récursive calculable est la fonction d'Ackermann.

Définition 3.24 (Fonction d'Ackermann [3]). On définit les diverses fonctions $A_n : \mathbb{N} \rightarrow \mathbb{N}$ par récurrence sur $n \in \mathbb{N}$ de la manière suivante :

- ▷ A_0 est la fonction $x \mapsto 2^x$;
- ▷ $A_{n+1}(x)$ est l'application x fois de la fonction A_n sur 1 :

$$A_n(A_n(\dots(A_n(1)))).$$

Formellement,

$$\begin{aligned} A_0(x) &= 2^x \\ A_{n+1}(0) &= 1 \\ A_{n+1}(x) &= A_n(A_{n+1}(x-1)). \end{aligned}$$

La fonction d'Ackermann est la fonction $n \mapsto A_n(n)$. ◇

La fonction d'Ackermann a une croissance extrêmement rapide :

$$A(0) = 1, \quad A(1) = 2, \quad A(2) = 16,$$

et $A(3)$ est déjà égal à 65 536 itérations de la fonction $x \mapsto 2^x$ (en commençant sur 0), c'est-à-dire :

$$A(3) = 2^{(2^{(\dots^{(2^0)}))}), \text{ où la puissance est itérée 65 536 fois.}$$

Malgré sa très forte croissance, la fonction d'Ackermann est calculable : pour calculer $A_n(n)$, on peut utiliser une pile contenant soit des fonc-

tions A_n (en pratique une représentation de ces fonctions), soit des entiers. Par exemple, si l'on empile A_n , A_{n+1} et ensuite k , cela correspond au calcul $A_n(A_{n+1}(k))$. Ainsi le sommet de la pile est-il toujours un entier, et l'élément qui suit (s'il existe) est toujours une fonction. Aussi, pour calculer $A_n(n)$, on procède comme suit.

1. On empile A_n , puis on empile n .
2. Tant que la pile contient plus d'un élément :
 - (a) on dépile l'entier k , puis on dépile la fonction A_m ;
 - (b) si $m = 0$, on empile 2^k ;
 - (c) sinon, si $k = 0$ on empile 1 ;
 - (d) sinon, on empile A_{m-1} , puis A_m et enfin $k - 1$.

L'algorithme s'arrêtera quand la pile ne contiendra plus qu'un élément, le résultat du calcul de $A_n(n)$.

Nous laissons en exercice la preuve que la fonction d'Ackermann croît plus vite que toute fonction primitive récursive, et n'est donc pas elle-même primitive récursive.

Exercice 3.25 (Cori et Lascar[42]). (★)

- (1) Montrer que $A_n(x) > x$ pour tous n, x .
- (2) Montrer que la fonction A_n est strictement croissante pour tout n .
- (3) Montrer que la fonction $n \mapsto A_n(x)$ est croissante pour tout x .
- (4) Montrer que $A_{n+1}(x + y) \geq A_n^{(y)}(x)$ pour tous n, x, y , où $f^{(m)}(x)$ dénote $f(f(\dots f(x) \dots))$ où l'application de f est itérée m fois.
- (5) Montrer que pour tout n , on a $A_{n+2}(x) > A_n^{(x+1)}(x + 1)$ pour presque tout x .
- (6) Soit $n > 0$, et soit $f : \mathbb{N}^n \rightarrow \mathbb{N}$. On note $P(f)$ le prédicat

$$\exists k \forall x_1, \dots, x_n A_k(\max(x_1, \dots, x_n)) > f(x_1, \dots, x_n).$$

Montrer que $P(f)$ est vrai pour toute fonction primitive récursive f .

En déduire que la fonction d'Ackermann n'est pas primitive récursive. $\diamond$

Les boucles de type **while** sont donc indispensables pour calculer certaines fonctions, et avec elles, surviennent la possibilité d'écrire des programmes qui ne s'arrêtent pas.

3.5. Les fonctions calculables sont récursives

Nous nous apprêtons à présent à montrer que le schéma de minimisation des fonctions récursives permet de calculer n'importe quelle fonction programmable avec ou sans boucles **while**. Pour ce faire, nous commençons

par voir comment simuler des structures de listes de taille arbitraire par des entiers. Cela semble en effet pour le moment être un manque de notre modèle de machine à registres et de programmes structurés. Nous avons par exemple besoin d'une pile pour calculer la fonction d'Ackermann via l'algorithme mentionné ci-dessus.

Proposition 3.26. Il existe une bijection $[] : \bigcup_{n \in \mathbb{N}} \mathbb{N}^n \rightarrow \mathbb{N}$ (où l'on note $[x_1, \dots, x_n]$ l'entier correspondant au n -uplet $(x_1, \dots, x_n)$), telle que les opérations suivantes sont primitives récursives.

1. La fonction $::$ d'ajout en tête de liste définie par

$$a :: [x_1, \dots, x_n] = [a, x_1, \dots, x_n].$$

2. Les fonctions hd et tl de tête et queue de liste, définies par

$$\begin{aligned} \text{hd}([]) &= 0 & \text{tl}([]) &= 0 \\ \text{hd}(x :: l) &= x & \text{tl}(x :: l) &= l. \end{aligned}$$

3. La fonction $| |$ de taille d'une liste définie par $|[x_1, \dots, x_n]| = n$.

4. Les fonctions get, set telles que

$$\begin{aligned} \text{get}([x_0, \dots, x_{n-1}], i) &= x_i & \text{si } i < n \\ \text{get}([x_0, \dots, x_{n-1}], i) &= 0 & \text{sinon} \end{aligned}$$

et

$$\begin{aligned} \text{set}([x_0, \dots, x_{n-1}], a, i) &= [x_0, \dots, x_{i-1}, a, x_{i+1}, \dots, x_{n-1}] & \text{si } i < n \\ \text{set}([x_0, \dots, x_{n-1}], a, i) &= [x_0, \dots, x_{n-1}] & \text{sinon.} \end{aligned}$$

★

PREUVE. La bijection $[] : \bigcup_{n \in \mathbb{N}} \mathbb{N}^n$ est définie par récurrence en commençant par la liste vide $[] = 0$ et en appliquant l'opération d'ajout en tête de liste définie par $a :: l = 1 + \alpha_2(a, l)$, où $\alpha_2 : \mathbb{N}^2 \rightarrow \mathbb{N}$ est la bijection de l'exercice 2-3.7. Il est clair que la fonction d'ajout en tête est primitive récursive, de même que les fonctions hd et tl obtenues grâce aux fonctions inverses de $(x_1, x_2) \mapsto \langle x_1, x_2 \rangle$. Montrons que le codage des listes obtenu ainsi est bien une bijection.

Montrons par récurrence que deux listes de tailles différentes ne peuvent pas être codées par le même élément. Le code de la liste vide est 0 et le code d'une liste non vide est de la forme $1 + \alpha_2(a, l) \neq 0$. Donc, le code de la liste vide est toujours différent du code d'une liste non vide.

Supposons à présent que toutes les listes de taille n ont un code différent de celui des listes de taille $m > n$. Montrons que toutes les listes de taille $n + 1$ ont un code différent de celui des listes de taille $m > n + 1$. Les codes des listes de taille $n + 1$ sont de la forme $1 + \alpha_2(a, l_1)$ pour l_1 le code d'une liste de taille n . Les codes des listes de taille $m > n + 1$ sont de

la forme $1 + \alpha_2(b, l_2)$ pour l_2 le code d'une liste de taille $m > n$. Par hypothèse de récurrence, on a forcément $l_1 \neq l_2$ et comme α_2 est injectif, on a forcément $1 + \alpha_2(a, l_1) \neq 1 + \alpha_2(b, l_2)$. Donc, les codes des listes de taille $n + 1$ sont différents des codes des listes de taille $m > n + 1$. Par récurrence, on en déduit que les codes de listes de tailles différentes sont différents.

Montrons à présent par récurrence sur k que si $(a_1, \dots, a_k) \neq (b_1, \dots, b_k)$, on a alors $[a_1, \dots, a_k] \neq [b_1, \dots, b_k]$.

Pour $k = 1$, on a $a_1 \neq b_1$ implique $1 + \alpha_2(a_1, 0) \neq 1 + \alpha_2(b_1, 0)$, car α_2 est injective. On a donc $[a_1] \neq [b_1]$. Supposons que ce soit le cas pour k , et montrons que c'est le cas pour $k + 1$. Supposons $(a_1, \dots, a_{k+1}) \neq (b_1, \dots, b_{k+1})$. Si $a_1 \neq b_1$, on a alors $1 + \alpha_2(a_1, [a_2, \dots, a_{k+1}]) \neq 1 + \alpha_2(b_1, [b_2, \dots, b_{k+1}])$, car α_2 est injective. Si $(a_2, \dots, a_{k+1}) \neq (b_2, \dots, b_{k+1})$, alors par hypothèse de récurrence on a $[a_2, \dots, a_{k+1}] \neq [b_2, \dots, b_{k+1}]$, et donc

$$1 + \alpha_2(a_1, [a_2, \dots, a_{k+1}]) \neq 1 + \alpha_2(b_1, [b_2, \dots, b_{k+1}]),$$

car α_2 est injective. Par récurrence, pour tout k on a donc

$$(a_1, \dots, a_k) \neq (b_1, \dots, b_k) \text{ implique } [a_1, \dots, a_k] \neq [b_1, \dots, b_k].$$

La fonction $[\]$ est ainsi injective.

Montrons à présent que $[\]$ est surjective. Supposons par l'absurde que ce n'est pas le cas. Dans ce cas, il existe un plus petit n tel que n n'est le code d'aucune liste. Notons que l'on a forcément $n > 0$, car 0 est le code de la liste vide. Aussi, comme α_2 est surjective, il existe (a, b) tel que $\alpha_2(a, b) = n - 1$, et donc tel que $1 + \alpha_2(a, b) = n$. On a par ailleurs nécessairement $b \leq n - 1 < n$. Aussi, par minimalité de n , il doit exister une liste l dont b est le code. Donc, n est le code de la liste $a :: l$, ce qui contredit notre hypothèse sur n .

Afin de montrer que les fonctions $| \cdot |$, get et set sont primitives récursives, on donne une définition primitive récursive de la fonction $\text{tl}(l, n)$ qui ampute une liste l de ses n premiers éléments :

$$\begin{aligned} \text{tl}(l, 0) &= l \\ \text{tl}(l, n + 1) &= \text{tl}(\text{tl}(l, n)). \end{aligned}$$

La taille est alors définie via le schéma de minimisation bornée par l'entier $|l| = \min\{n \leq l : \text{tl}(l, n) = []\}$. La fonction get a la définition primitive récursive suivante : $\text{get}(l, n) = \text{hd}(\text{tl}(l, n))$. La fonction set est quant à elle définie en deux fois, en ajoutant d'abord un paramètre supplémentaire :

$$\begin{aligned} \text{set}(l, a, i) &= \text{set}(l, a, i, i) \\ \text{set}(l, a, n, 0) &= a :: \text{tl}(l, \text{succ}(n)) \\ \text{set}(l, a, n, i + 1) &= \text{get}(l, n - \text{succ}(i)) :: \text{set}(l, a, n, i) \end{aligned} \quad \blacksquare$$

Nous avons à présent tous les éléments nécessaires pour montrer que les fonctions calculables par des programmes structurés sont récursives.

Théorème 3.27

Toute fonction calculable par un programme goto (et donc aussi par un programme structuré) est une fonction générale récursive.

Le reste de la section est consacré à la preuve, pour laquelle nous avons besoin de fixer un codage des programmes goto et des machines à registre.

Codage des programmes goto. On code les instructions des programmes goto comme suit :

- ▷ « $R_i = R_i + 1$ » est codé par $\langle 0, i \rangle$
- ▷ « $R_i = R_i - 1$ » est codé par $\langle 1, i \rangle$
- ▷ « $R_i = 0$ » est codé par $\langle 2, i \rangle$
- ▷ « if $R_i = 0$ goto n_1 else n_2 » est codé par $\langle 3, \langle i, \langle n_1, n_2 \rangle \rangle \rangle$
- ▷ « goto n » est codé par $\langle 4, n \rangle$.

Pour des raisons d'uniformité, il sera utile d'avoir une borne sur l'indice maximal des registres utilisés. Le code d'un programme goto est simplement donné par le code : $\langle k, [I_1, \dots, I_n] \rangle$, où k est tel que le programme utilise au plus les registres $R_0, \dots, R_k$ et où I_e est le code de la e -ième instruction pour $1 \leq e \leq n$.

Codage des machines à registres. On fixe à présent un codage de l'état d'une machine à k registres. Cet état est donné par la valeur des registres ainsi que par l'indice de la prochaine instruction à exécuter. Pour un nombre de registres k donné, ce code est $\langle m, [R_0, \dots, R_k] \rangle$, où m est le numéro d'instruction et R_i est la valeur du registre numéro i pour $0 \leq i \leq k$.

Fonction d'initialisation. Fixons maintenant une fonction d'initialisation init , qui étant donné un code $e = \langle k, I \rangle$ d'un programme (où I est une liste d'instructions), des valeurs $x_1, \dots, x_n$ (pour $n \leq k$), donne le code représentant l'état de la machine à k registres, au début du calcul.

$$\text{init}(\langle k, I \rangle, x_1, \dots, x_n) = \langle 0, 0 :: x_1 :: \dots :: x_n :: \text{aux}(k - n) \rangle$$

avec aux définie par

$$\begin{aligned} \text{aux}(0) &= [] \\ \text{aux}(k + 1) &= 0 :: \text{aux}(k). \end{aligned}$$

Il est clair que la fonction init est primitive récursive.

Fonctions de transition 1. On définit une fonction de transition tr_1 , qui étant donné le code $e = \langle k, I \rangle$ d'un programme et le code $p = \langle m, R \rangle$ de l'état d'une machine, renvoie le numéro de la prochaine instruction à exécuter à l'étape de calcul suivante. On va utiliser pour cela une fonction $\text{cur} : \mathbb{N} \rightarrow \mathbb{N}$ qui, étant donné le code $e = \langle k, I \rangle$ d'un programme et le code $p = \langle m, R \rangle$ de l'état d'une machine, permet d'obtenir l'instruction courante de la machine :

$$\text{cur}(\langle k, I \rangle, \langle m, R \rangle) = \text{get}(I, m).$$

En utilisant la fonction cur , les fonctions π_1, π_2 telles que $n = \langle \pi_1(n), \pi_2(n) \rangle$ on définit $\text{tr}_1(e, p)$ comme étant :

$$\begin{array}{ll} \pi_1(p) + 1 & \text{si } \pi_1(\text{cur}(e, p)) \leq 2 \\ \pi_1(\pi_2(\pi_2(\text{cur}(e, p)))) & \text{si } \pi_1(\text{cur}(e, p)) = 3 \text{ et} \\ & \text{get}(\pi_2(p), \pi_1(\pi_2(\text{cur}(e, p)))) = 0 \\ \pi_2(\pi_2(\pi_2(\text{cur}(e, p)))) & \text{si } \pi_1(\text{cur}(e, p)) = 3 \text{ et} \\ & \text{get}(\pi_2(p), \pi_1(\pi_2(\text{cur}(e, p)))) \neq 0 \\ \pi_2(\text{cur}(e, p)) & \text{si } \pi_1(\text{cur}(e, p)) = 4. \end{array}$$

Il est clair que tr_1 est primitive récursive.

Fonctions de transition 2. On définit à présent une fonction de transition tr_2 qui étant donné le code e d'un programme et le code p de l'état d'une machine, permet d'obtenir l'état des registres de la machine à l'étape de calcul suivante.

Pour cela l'on utilisera deux fonctions primitives récursives $\text{inc} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ et $\text{dec} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, telles que

$$\begin{aligned} \text{inc}([x_0, \dots, x_n], i) &= [x_0, \dots, x_i + 1, \dots, x_n] \\ \text{dec}([x_0, \dots, x_n], i) &= [x_0, \dots, \max(0, x_i - 1), \dots, x_n] \end{aligned}$$

définies de la manière suivante.

$$\begin{aligned} \text{inc}(l, i) &= \text{set}(l, \text{succ}(\text{get}(l, i)), i) \\ \text{dec}(l, i) &= \text{set}(l, \text{pred}(\text{get}(l, i)), i). \end{aligned}$$

On peut à présent définir $\text{tr}_2(e, p)$ comme étant :

$$\begin{array}{ll} \text{inc}(\pi_2(p), \pi_2(\text{cur}(e, p))) & \text{si } \pi_1(\text{cur}(e, p)) = 0 \\ \text{dec}(\pi_2(p), \pi_2(\text{cur}(e, p))) & \text{si } \pi_1(\text{cur}(e, p)) = 1 \\ \text{set}(\pi_2(p), 0, \text{cur}(e, p)) & \text{si } \pi_1(\text{cur}(e, p)) = 2 \\ \pi_2(p) & \text{sinon.} \end{array}$$

Il est clair que tr_2 est primitive récursive.

Fin de la preuve. On définit à présent la fonction $\text{tr} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ de transition d'un état à un autre :

$$\text{tr}(e, p) = \langle \text{tr}_1(e, p), \text{tr}_2(e, p) \rangle.$$

On définit ensuite la fonction primitive récursive

$$\text{st} : \mathbb{N} \times \mathbb{N}^n \times \mathbb{N} \rightarrow \mathbb{N}$$

telle que $\text{st}(e, x_1, \dots, x_n, t)$ renvoie l'état de la machine qui exécute le programme e , avec les registres $R_1, \dots, R_n$, initialisés à $x_1, \dots, x_n$ respectivement, après t étapes de calcul.

$$\begin{aligned} \text{st}(e, x_1, \dots, x_n, 0) &= \text{init}(e, x_1, \dots, x_n) \\ \text{st}(e, x_1, \dots, x_n, t+1) &= \text{tr}(e, \text{st}(e, x_1, \dots, x_n, t)). \end{aligned}$$

On arrive enfin à l'étape pour laquelle on a besoin de schéma de minimisation, laissant la possibilité à une fonction de ne pas être définie sur certaines entrées. La fonction récursive $\text{time} : \mathbb{N} \times \mathbb{N}^n \rightarrow \mathbb{N}$ donne le plus petit temps de calcul nécessaire pour que la machine s'arrête, c'est-à-dire arrive à un numéro d'instruction plus grand que le nombre d'instructions du programme. La fonction sera définie si, et seulement si, la machine s'arrête pour le programme et les entrées correspondantes.

$$\text{time}(e, x_1, \dots, x_n) = \min\{t \in \mathbb{N} : \pi_1(\text{st}(e, x_1, \dots, x_n, t)) \geq |\pi_2(e)|\}.$$

Finalement, voici la fonction récursive qui correspond au calcul de la machine de code e . On lance la fonction de transition pour le nombre d'étapes nécessaires avant que la machine ne s'arrête, et l'on renvoie la valeur du registre R_0 :

$$f(x_1, \dots, x_n) = \text{hd}(\pi_2(\text{st}(e, x_1, \dots, x_n, \text{time}(e, x_1, \dots, x_n)))).$$

Cela conclut la démonstration.

3.6. Conséquences

D'après la preuve précédente, étant donné notre codage d'un programme par un entier e , son exécution pour t étapes de calcul est une fonction primitive récursive et donc elle-même calculable par un programme structuré. La recherche de ce plus petit temps de calcul peut se faire à l'aide d'une boucle **while**. Remarquons de plus que la preuve est uniforme : la même fonction primitive récursive s'adapte en fonction de tout code e d'un programme.

Cela permet d'obtenir le théorème 3-3.1 de l'existence d'un programme universel, utilisé tout au long du livre, via la notation Φ_e pour le programme de code e .

Théorème (3-3.1)

Soit $n \in \mathbb{N}^*$. Il existe un code e de programme informatique pour lequel $\Phi_e : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ est tel que, pour tous $x_1, \dots, x_n$, on a

- ▷ $\Phi_e(a, x_1, \dots, x_n) \uparrow$ ssi $\Phi_a(x_1, \dots, x_n) \uparrow$;
- ▷ $\Phi_e(a, x_1, \dots, x_n) \downarrow = y$ ssi $\Phi_a(x_1, \dots, x_n) \downarrow = y$.

Le code e du théorème ci-dessus est un code de la fonction

$$(x_1, \dots, x_n) \mapsto \text{hd}(\pi_2(\text{st}(a, x_1, \dots, x_n, \text{time}(a, x_1, \dots, x_n))))$$

donnée à la fin de la section précédente. La fonction

$$(a, x_1, \dots, x_n) \mapsto \text{time}(a, x_1, \dots, x_n)$$

qui cherche le plus petit temps de calcul tel que le programme s'arrête est la seule qui utilise le schéma de minimisation. Notons que cela permet aussi de donner une définition mathématique précise aux notations

$$\Phi_a(x_1, \dots, x_n)[t] \downarrow \quad \text{et} \quad \Phi_a(x_1, \dots, x_n)[t] \uparrow.$$

Elles correspondent respectivement aux prédicats primitifs récursifs :

$$\begin{aligned} & \exists s \leq t \ \pi_1(\text{st}(e, x_1, \dots, x_n, s)) \geq |\pi_2(e)| \\ \text{et} \quad & \forall s \leq t \ \pi_1(\text{st}(e, x_1, \dots, x_n, s)) < |\pi_2(e)|. \end{aligned}$$

Chapitre 7

Immunité et croissance de fonction

La calculabilité étudie la puissance calculatoire des ensembles d'entiers, modulo la réduction Turing. Dans ce chapitre, nous allons étudier en particulier deux grandes familles de propriétés calculatoires, à savoir la capacité à calculer des ensembles difficiles à décrire (ensemble immune, hyperimmune, effectivement immune), et la capacité à calculer des fonctions à croissance rapide (fonction hyperimmune, fonction dominante). Prenons quelques exemples.

Exemple 1. D'après l'exercice 3-7.10, tout ensemble infini c. e. contient un sous-ensemble infini calculable. Quelle puissance calculatoire faut-il pour obtenir un ensemble infini ne possédant pas de sous-ensemble infini calculable ? Nous étudierons cela dans la section 1 sous le concept d'ensemble immune.

Exemple 2. D'après le théorème 3-6.2 du point fixe de Kleene, pour toute fonction totale calculable $f : \mathbb{N} \rightarrow \mathbb{N}$, il existe un code e tel que $\Phi_{f(e)} = \Phi_e$. Quelle est la puissance calculatoire d'une fonction sans point fixe ? Cette notion sera étudiée dans la section 2.

Exemple 3. Toute fonction calculable est trivialement dominée par une fonction calculable. Quelle puissance calculatoire faut-il pour calculer une fonction qui n'est dominée par aucune fonction calculable ? Ce sera le sujet de la section 4.

Il est difficile de se faire une intuition sur la puissance calculatoire *a priori* de propriétés formulées de manière aussi diverses, et en particulier de les comparer. Les propriétés tirées des trois exemples précédents admettent cependant des caractérisations qui rendront cette comparaison plus aisée. De manière générale, l'existence de nombreuses caractérisations d'une même puissance calculatoire avec des formulations très diverses est un gage de robustesse de la notion. C'est en particulier le cas des propriétés étudiées dans ce chapitre.

1. Ensembles immunes

La première famille de propriétés calculatoires sur les ensembles relève de la capacité à approximer les éléments d'un ensemble. Un ensemble est calculable s'il est possible de déterminer calculatoirement quels éléments lui appartiennent ou non. Au niveau suivant, un ensemble est calculatoirement énumérable s'il existe une procédure calculable pour lister tous les éléments qui lui appartiennent, mais potentiellement dans le désordre, ce qui fait qu'il n'est généralement pas possible d'être certain qu'un élément n'appartient pas à l'ensemble. Nous allons maintenant étudier la puissance calculatoire des ensembles infinis pour lesquels il n'est même pas possible d'énumérer de manière calculable une quantité infinie de ses éléments.

Définition 1.1. Un ensemble infini $A \subseteq \mathbb{N}$ est *immune* s'il ne contient pas de sous-ensemble infini c. e. ◇

Comme nous l'avons vu, tout ensemble infini c. e. contient un sous-ensemble infini calculable. Ainsi, de manière équivalente, un ensemble infini est immune si, et seulement si, il ne contient pas de sous-ensemble infini calculable. En particulier, tout ensemble immune A est nécessairement non calculable, car A serait alors son propre sous-ensemble infini calculable contredisant son immunité.

L'immunité est une notion d'ensemble, mais non pas de degré. En effet, si A est un ensemble immune, l'ensemble $A \oplus \mathbb{N} = \{2n : n \in A\} \sqcup \{2n+1 : n \in \mathbb{N}\}$ est de même degré Turing que A , mais $A \oplus \mathbb{N}$ possède le sous-ensemble infini calculable $\{2n+1 : n \in \mathbb{N}\}$. Inversement, tout degré Turing non calculable contient un ensemble immune, comme le montre la proposition suivante.

Proposition 1.2. Tout ensemble non calculable est Turing équivalent à un ensemble immune. ★

PREUVE. Soit A un ensemble non calculable, et soit $B = \{\sigma \in 2^{<\mathbb{N}} : \sigma \prec A\}$ l'ensemble des segments initiaux de A . En particulier, $A \equiv_T B$. Il est par ailleurs évident que tout sous-ensemble infini de B permet de calculer des

segments initiaux arbitrairement grands de A , et donc A (ainsi d'ailleurs que B). Comme A n'est pas calculable, B ne possède pas de sous-ensemble infini calculable. ■

La notion d'ensemble immune possède deux renforcements orthogonaux, à savoir les ensembles effectivement immunes, et les ensembles hyperimmunes. Ces deux notions sont des propriétés calculatoires fondamentales en calculabilité, et nous verrons pour chacune d'entre elles plusieurs définitions équivalentes.

Rappelons que W_e désigne l'ensemble c. e. de code $e : \{n : \Phi_e(n) \downarrow\}$.

Définition 1.3. Un ensemble infini $A \subseteq \mathbb{N}$ est *effectivement immune* s'il existe une fonction totale calculable $h : \mathbb{N} \rightarrow \mathbb{N}$ telle que pour tout code e , si $|W_e| \geq h(e)$ alors $W_e \not\subseteq A$. ◇

Intuitivement, un ensemble infini A est effectivement immune si non seulement les ensembles infinis finissent par se tromper et énumérer un élément en-dehors de A , mais plus encore, cette erreur doit arriver après suffisamment peu d'éléments énumérés, en fonction du code de l'énumération. En particulier, tout ensemble effectivement immune est immune.

Nous verrons que le concept d'immunité effective est particulièrement digne d'intérêt du point de vue des degrés Turing. La puissance de calcul correspondant à la capacité de calculer un ensemble effectivement immune possède de nombreuses caractérisations qui seront étudiées dans la section 2.

Rappelons que le *code canonique* d'un ensemble fini F est l'entier

$$n = \sum_{i \in F} 2^i.$$

Soit $D_0, D_1, \dots$ la collection des ensembles finis telle que D_n est canoniquement codé par n , pour tout $n \in \mathbb{N}$.

Définition 1.4. Un *tableau* est une collection d'ensembles finis mutuellement disjoints $F_0, F_1, \dots$. Un tableau $F_0, F_1, \dots$ est c. e. s'il existe une fonction calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que pour tout n , $F_n = D_{f(n)}$. Un ensemble infini A est *hyperimmune* si pour tout tableau c. e. $F_0, F_1, \dots$, il existe un entier n tel que $F_n \cap A = \emptyset$. ◇

Cette définition plus complexe, formalise l'idée selon laquelle non seulement il n'est pas possible de lister calculatoirement une infinité d'éléments de A , mais plus encore il n'est même pas possible de lister une infinité de « blocs » finis d'éléments disjoints deux à deux, tels que chaque bloc contient au moins un élément de A .

Tout comme pour le concept d'immunité effective, c'est l'extension de l'hyperimmunité aux degrés Turing qui nous intéressera surtout dans la suite. La puissance de calcul correspondant à la capacité de calculer un ensemble hyperimmune possède de nombreuses caractérisations qui seront étudiées dans la section 4.

Exercice 1.5. Montrer que tout ensemble hyperimmune est immune. $\diamond$

2. Fonctions DNC

Nous voyons à présent un exemple de degré Turing remarquable, dont l'étude remonte sans doute aux travaux d'Arslanov [8], qui étudia les degrés Turing permettant d'échapper au fameux théorème du point fixe de Kleene : étant donné une fonction calculable f , il existe e tel que $\Phi_e = \Phi_{f(e)}$. Quelle puissance est nécessaire pour calculer une fonction f pour laquelle ce n'est pas le cas ? Ces travaux ont été étendus par Jockusch, Lerman, Soare, et Solovay qui ont trouvé une caractérisation équivalente, qui constitue aujourd'hui la définition moderne de degré DNC.

Définition 2.1. Une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ est *diagonalement non calculable* (DNC) si $f(n) \neq \Phi_n(n)$ pour tout n . $\diamond$

Insistons sur le fait que la fonction f doit être totale dans la définition ci-dessus. Notons que si $\Phi_n(n) \uparrow$, il n'y a pas de restriction sur la valeur de $f(n)$. Un degré Turing est DNC s'il contient une fonction DNC.

Exercice 2.2. (*) Montrer que les degrés DNC sont clos par le haut, c'est-à-dire que si un ensemble calcule une fonction DNC, son degré est lui-même DNC. $\diamond$

La notion de degré DNC présente de nombreuses applications dans les liens entre calculabilité et aléatoire algorithmique, tout comme en mathématiques à rebours. Nous verrons en particulier avec le corollaire 18-4.3 que les DNC sont nombreux du point de vue de la mesure, mais d'après la proposition 10-3.36 peu nombreux du point de vue des catégories de Baire (nous verrons en particulier l'existence de degrés non DNC et non calculables).

Observons tout d'abord qu'il n'existe pas de fonction DNC calculable, par un simple argument diagonal, ce qui est à l'origine de l'appellation « diagonalement non calculable ». La proposition suivante montre en revanche que l'on peut calculer une fonction DNC à l'aide du problème de l'arrêt.

Proposition 2.3. Le degré Turing $0'$ du problème de l'arrêt est un degré DNC. $\star$

PREUVE. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ la fonction $\emptyset'$ -calculable, qui pour une entrée e , renvoie $1 - \Phi_e(e)$ si $e \in \emptyset'$, et 0 sinon. Cette fonction est DNC, car elle est totale, et lorsque $\Phi_e(e) \downarrow$, elle renvoie une valeur différente. Comme $\emptyset'$ calcule une fonction DNC, et que les degrés DNC sont clos par le haut, $\mathbf{0}'$ est DNC. ■

Nous verrons dans la section 3 que $\mathbf{0}'$ est le seul degré à la fois DNC et c.e. Il est clair que les degrés DNC sont en quantité indénombrable, car c'est également le cas des degrés au-dessus de $\mathbf{0}'$. Nous verrons avec la proposition 10-3.36 que les degrés non DNC sont eux aussi en quantité indénombrable.

Pour le moment, nous nous attachons à montrer que la notion de degré DNC est naturelle, dans le sens où elle possède de nombreuses caractérisations via des formulations très différentes (et nous en verrons d'autres encore dans la partie II sur l'aléatoire algorithmique).

Définition 2.4. Une fonction f est *libre de point fixe* si $\Phi_n \neq \Phi_{f(n)}$ pour tout n . ◇

Théorème 2.5 (Jockusch, Lerman, Soare, et Solovay [101])

Soit $X \in 2^{\mathbb{N}}$. Alors, les énoncés suivants sont équivalents.

- (1) X calcule une fonction diagonalement non calculable.
- (2) X calcule une fonction libre de point fixe.

PREUVE. Montrons (1) $\Rightarrow$ (2). Soit $g \leq_T X$ telle que $g(n) \neq \Phi_n(n)$ pour tout n . Alors également, $f \leq_T X$ telle que $f(n)$ est un code pour une fonction calculable définie uniquement sur l'entrée n , et qui à n associe $g(n)$. On a en particulier $\Phi_{f(n)}(n) \downarrow = g(n)$. Supposons qu'il existe n tel que $\Phi_{f(n)}(m) = \Phi_n(m)$ pour tout m . Alors, $\Phi_n(n) \downarrow = \Phi_{f(n)}(n) \downarrow = g(n)$, ce qui contredit la définition de g . Donc, f est une fonction libre de point fixe.

Montrons (2) $\Rightarrow$ (1). Soit $f \leq_T X$, une fonction libre de point fixe. Alors, à partir de X , on calcule la fonction $g : \mathbb{N} \rightarrow \mathbb{N}$ qui sur l'entier n crée le code e_n de la fonction $m \mapsto \Phi_{\Phi_n(n)}(m)$, et renvoie $f(e_n)$. On utilise ici le même abus de notation que dans la preuve du point fixe de Kleene : si $\Phi_n(n) \uparrow$, alors $m \mapsto \Phi_{\Phi_n(n)}(m)$ désigne la fonction nulle part définie. Notons que g est totale, car elle ne cherche pas à faire le calcul $\Phi_n(n)$. Supposons que g ne soit pas DNC, c'est-à-dire qu'il existe n tel que $g(n) = \Phi_n(n)$. Par définition de g , $f(e_n) = \Phi_n(n)$. En particulier, $\Phi_{f(e_n)} = \Phi_{\Phi_n(n)} = \Phi_{e_n}$. La fonction f n'est donc pas libre de point fixe, ce qui contredit la définition de f . Donc, g est une fonction DNC. ■

Voyons à présent l'équivalence entre degré DNC et degré effectivement immune. La troisième équivalence du théorème ci-dessous est plus technique, mais présente un grand intérêt et sera réutilisée par la suite. Il s'agit en fait d'un renforcement de la notion d'être DNC : considérons la suite $(A_n)_{n \in \mathbb{N}}$ d'ensembles c. e. définie par

$$A_n = \begin{cases} \{\Phi_n(n)\} & \text{si } \Phi_n(n) \downarrow \\ \emptyset & \text{sinon.} \end{cases}$$

Être de degré DNC est la capacité de calculer une fonction f telle que pour tout n entier, $f(n) \notin A_n$. La troisième équivalence étend ce résultat à l'énumération uniforme $(W_n)_{n \in \mathbb{N}}$ de tous les ensembles c. e. dont on sait borner le nombre d'éléments par un entier n .

Théorème 2.6

Soit $X \in 2^{\mathbb{N}}$. Les énoncés suivants sont équivalents.

- (1) X calcule une fonction diagonalement non calculable.
- (2) X calcule un ensemble effectivement immune.
- (3) X calcule une fonction $h : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ telle que pour $e, n \in \mathbb{N}$,

$$|W_e| \leq n \Rightarrow h(e, n) \notin W_e.$$

PREUVE. (1) $\Rightarrow$ (3). Soit $f \leq_T X$ une fonction DNC. Nous décrivons un processus uniforme en e, n afin de calculer une valeur qui, sous l'hypothèse $|W_e| \leq n$, n'est pas dans W_e . Pour chaque $0 \leq i < n$, on calcule le code $u(e, i)$ de la fonction partielle calculable qui, pour toute entrée, cherche le i -ième élément k dans l'énumération de W_e , s'il existe. Si un tel élément k est trouvé, la fonction l'interprète comme le n -uplet $\langle k_0, \dots, k_{n-1} \rangle$, et renvoie k_i . Sinon, la fonction ne s'arrête pas. Notons que $\Phi_{u(e, i)}$ est soit une fonction constante, soit la fonction définie nulle part.

Montrons que la fonction X -calculable

$$h(e, n) = \langle f(u(e, 0)), \dots, f(u(e, n-1)) \rangle$$

satisfait (3). Raisonnons par l'absurde, et supposons que $h(e, n) \in W_e$ avec $|W_e| \leq n$. Disons que $h(e, n)$ est le i -ième élément de W_e dans l'ordre d'énumération. Alors, $\Phi_{u(e, i)}$ est la fonction constante qui trouve

$$k = h(e, n) = \langle f(u(e, 0)), \dots, f(u(e, n-1)) \rangle$$

et renvoie le i -ième élément du tuple, c'est-à-dire $f(u(e, i))$. En particulier, $\Phi_{u(e, i)}(u(e, i)) \downarrow = f(u(e, i))$, ce qui contredit l'hypothèse selon laquelle f est DNC.

(3) $\Rightarrow$ (2). Soit $h \leq_T X$ une fonction satisfaisant (3). Soit $D_0, D_1, \dots$ une énumération effective de tous les ensembles finis tels que n est le code

canonique de D_n . Soit $g : \mathbb{N} \rightarrow \mathbb{N}$ une fonction partielle calculable telle que si $|W_e| \geq e + 1$; alors, $D_{g(e)} \subseteq W_e$ et $|D_{g(e)}| = e + 1$. Nous allons définir une suite X -calculable infinie croissante d'entiers $x_0 < x_1 < \dots$ telle que pour tout s ,

$$\forall e \leq s, (|W_e| > e \Rightarrow D_{g(e)} \not\subseteq \{x_i : i \leq s\}). \quad (\star)$$

Il s'ensuit que $H = \{x_n : n \in \mathbb{N}\}$ est effectivement immune, car si $W_e \subseteq H$, alors $|W_e| \leq e$. Le fait que l'on veuille $x_i < x_{i+1}$ est simplement pour que l'ensemble H soit X -calculable. Supposons maintenant que l'on ait déjà défini $x_0 < \dots < x_s$ satisfaisant $(\star)$. Soit

$$W_{v(s)} = \{y : y \leq x_s\} \cup \bigcup_{e \leq s+1 \text{ t.q. } g(e) \downarrow} D_{g(e)}.$$

Le but est d'utiliser notre fonction h pour trouver un élément qui n'est pas dans $W_{v(s)}$. Soit x_{s+1} un tel élément — nous verrons après comment l'obtenir. Notons d'abord que x_{s+1} est bien strictement plus grand que x_s . Notons ensuite que pour tout entier $e \leq s + 1$ tel que $|W_e| > e$ et tel que $D_{g(e)} \not\subseteq \{x_i : i \leq s\}$, alors également $D_{g(e)} \not\subseteq \{x_i : i \leq s + 1\}$. Notons finalement que pour $e \leq s$ tel que $|W_e| > e$, on a bien $D_{g(e)} \not\subseteq \{x_i : i \leq s\}$ par hypothèse, et pour $e = s + 1$, si $|W_e| > e$, on a $D_{g(e)} \not\subseteq \{x_i : i \leq s\}$ nécessairement, car $D_{g(e)}$ possède alors $s + 2$ éléments. Dans tous les cas, on aura bien la condition $(\star)$ pour la suite $(x_i)_{i \leq s+1}$. Montrons à présent comment trouver x_{s+1} à l'aide de la fonction h . L'ensemble $W_{v(s)}$ admet au plus $x_s + 1$ plus $1 + 2 + 3 + \dots + s + 2$ éléments, ce qui donne par la somme des termes d'une suite arithmétique $t_s = x_s + 1 + (s + 2)(s + 3)/2$. On définit alors $x_{s+1} = h(v(s), t_s)$. Par définition de h , $x_{s+1} \notin W_{v(s)}$. En particulier, $x_{s+1} > x_s$, et $(\star)$ est satisfait pour $s + 1$.

(2) $\Rightarrow$ (1) : Soit $H \leq_T X$ un ensemble effectivement immune. On considère $g : \mathbb{N} \rightarrow \mathbb{N}$ une fonction totale calculable telle que si $W_e \subseteq H$, alors $|W_e| < g(e)$. Nous allons montrer que X calcule une fonction libre de point fixe. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ la fonction X -calculable telle que $W_{f(e)}$ est l'ensemble des $g(e)$ premiers éléments de H . Notons que f est X -calculable, mais que l'ensemble $W_{f(e)}$ est un ensemble c. e. qui, en particulier, n'a pas besoin de X pour son énumération : on peut voir les éléments à énumérer comme étant « codés en dur » dans $f(e)$. Montrons que f est libre de point fixe. Soit $e \in \mathbb{N}$; si $W_{f(e)} = W_e$, alors $W_e \subseteq H$, mais $|W_e| = |W_{f(e)}| = g(e)$, ce qui contredit le choix de H et g . Comme X calcule une fonction libre de point fixe, alors par le théorème 2.5, X calcule une fonction DNC. ■

Nous nous arrêtons là pour le moment. Nous verrons d'autres caractérisations de la notion de degrés DNC en rapport avec l'aléatoire algorithmique. Mentionnons pour finir une hiérarchie qui découle naturellement de la définition des degrés DNC.

Définition 2.7. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ une fonction telle que

$$2 \leq f(n) \leq f(n+1).$$

Un ensemble $X \subseteq \mathbb{N}$ est de degré DNC_f si X calcule une fonction g telle que $g(n) < f(n)$ et $g(n) \neq \Phi_n(n)$ pour tout n . $\diamond$

Un examen de la définition indique que plus la fonction f croît lentement, plus il semble difficile pour un ensemble d'être DNC_f . C'est effectivement le cas, comme on peut le voir avec le théorème suivant, dû à Ambos-Spies, Kjos-Hanssen, Lempp et Slaman [6].

Théorème 2.8

Soit f une fonction telle que $2 \leq f(n) \leq f(n+1)$. Il existe une fonction g telle que $2 \leq g(n) \leq g(n+1)$ et avec $f < g$ pour laquelle $\text{DNC}_f \subsetneq \text{DNC}_g$, c'est-à-dire qu'il existe un ensemble X qui calcule une fonction DNC_g mais qui ne calcule aucune fonction DNC_f .

Comme annoncé plus tôt, nous verrons de nombreux exemples de degrés non DNC et non calculables, mais il est informatif de s'assurer de leur existence par une construction directe.

Exercice 2.9. () Montrer par la méthode des extensions finies qu'il existe des degrés Δ_2^0 non DNC et non calculables.** $\diamond$

3. Critère de complétude d'Arslanov

Le critère de complétude d'Arslanov traduit d'une certaine manière une incompatibilité entre les degrés c. e. et DNC.

Degré c. e.

Un degré Turing est dit calculatoirement énumérable ou c. e. s'il contient un ensemble calculatoirement énumérable. Nous verrons dans le chapitre 13 que $\mathbf{0}'$ est loin d'être le seul degré c. e.

Il est aisé de calculer une fonction DNC à l'aide du problème de l'arrêt, comme le montre la proposition 2.3. En revanche, le critère de complétude d'Arslanov prouve que $\mathbf{0}'$ est le seul degré à la fois c. e. et DNC.

Théorème 3.1 (Critère de complétude d'Arslanov [8])

Soit $A \in 2^{\mathbb{N}}$ un ensemble c. e. Alors, A est Turing complet ssi A calcule une fonction DNC.

PREUVE. Par la proposition 2.3, si l'ensemble A est Turing complet, il calcule une fonction DNC. Montrons la réciproque. Soit $(A_s)_{s \in \mathbb{N}}$ une approximation c.e. de A , c'est-à-dire avec $\lim_{s \rightarrow \infty} A_s = A$, et $A_s \subseteq A_{s+1}$. Notons que si $A_s \upharpoonright_n = A \upharpoonright_n$, alors aussi pour tout $t > s$ on aura $A_t \upharpoonright_n = A \upharpoonright_n$.

Soit Φ une fonctionnelle totale sur l'oracle A et telle que $\Phi(A, n) \neq \Phi_n(n)$ pour tout n . Uniformément en n , on calcule le code a_n de la fonction partielle qui sur toute entrée k agit comme suit : cherche le plus petit t tel que $\emptyset'[t](n) = 1$, puis si cela arrive et si $\Phi(A_t \upharpoonright_m, a_n)[t]$ s'arrête pour un certain $m \leq t$, alors renvoie la valeur de $\Phi(A_t \upharpoonright_m, a_n)[t]$, et sinon diverge. Notons que le code a_n est défini en fonction de a_n lui-même, ce qui est rendu possible grâce au théorème du point fixe.

Nous prétendons que pour tout n , si jamais $\emptyset'(n) = 1$, alors le plus petit t tel que $\Phi(A_t \upharpoonright_m, a_n)[t] \downarrow$ pour $m \leq t$ tel que $A_t \upharpoonright_m = A \upharpoonright_m$, est strictement plus grand que le plus petit t tel que $\emptyset'[t](n) = 1$. Supposons que ce ne soit pas le cas, c'est-à-dire il existe n pour lequel $\emptyset'(n) = 1$ et pour lequel étant donné t le plus petit entier tel que $\emptyset'[t](n) = 1$, on a également $\Phi(A_t \upharpoonright_m, a_n)[t] \downarrow$ pour $m \leq t$ tel que $A_t \upharpoonright_m = A \upharpoonright_m$. Dans ce cas, par la procédure décrite plus haut, on aura $\Phi_{a_n}(a_n) = \Phi(A_t \upharpoonright_m, a_n)$, et donc $\Phi_{a_n}(a_n) = \Phi(A \upharpoonright_m, a_n)$, ce qui contredit le fait que Φ calcule une fonction DNC sur l'oracle A .

L'oracle A peut donc calculer $\emptyset'$ en cherchant simplement le plus petit temps de calcul t tel que $A_t \upharpoonright_m = A \upharpoonright_m$ et $\Phi(A_t \upharpoonright_m, a_n)[t] \downarrow$ pour $m \leq t$, et puis en regardant si $\emptyset'[t](n) = 1$. Si c'est le cas, alors $\emptyset'(n) = 1$. Sinon $\emptyset'(n) = 0$. ■

Le critère de complétude d'Arslanov connaît plusieurs extensions, notamment par Jockusch et al. [101]. Nous donnons l'une d'entre elles dans un exercice qui permettra de manipuler les principes de la preuve précédente.

Exercice 3.2. (★★) (*Bienvenu et al.*[16]). Une fonction g est DNC relativement à C , noté $\text{DNC}(C)$, si $g(n) \neq \Phi_n(C, n)$ pour tout n . Soit $X \in 2^{\mathbb{N}}$ de degré DNC et soit C un ensemble c.e. Montrer que soit $X \oplus C \geq_T \emptyset'$, soit X est de degré DNC relativement à C .

Indication. – Soit g une fonction DNC. Définir des codes $a_{n,m}$ tels que

$$\Phi_{a_{n,m}}(a_{n,m}) = \Phi_m(C_s, m),$$

pour s le plus petit tel que $n \in \emptyset'[s]$. Montrer que soit il existe n tel que la fonction $m \mapsto g(a_{n,m})$ est DNC relativement à C , soit $g \oplus C$ calcule $\emptyset'$. ◊

Notons que l'exercice précédent implique le critère de complétude d'Arslanov, car si X et C sont de même degré, soit $C \geq_T \emptyset'$, soit C est de degré DNC relativement à C , ce qui est impossible.

4. Fonctions hyperimmunes

Nous abordons maintenant une seconde famille de propriétés calculatoires, basées sur la capacité à calculer des fonctions à croissance rapide. Les fonctions exponentielles, ou même de tours d'exponentielles sont calculables, et n'apportent donc pas de puissance de calcul supplémentaire. Nous parlons ici de fonctions dont la vitesse de croissance rend difficile toute représentation mentale.

Nous avons déjà vu dans la section 4-7 que la capacité à croître plus vite que certaines fonctions permettait de calculer les ensembles Δ_2^0 . Nous allons maintenant étudier la puissance de calcul liée aux fonctions qui ne sont dominées par aucune fonction calculable. L'étude de ces fonctions a été initiée par Martin et Miller [157].

Définition 4.1. Une fonction g domine une fonction f si $g(x) \geq f(x)$ pour tout $x \in \mathbb{N}$. Une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ est *hyperimmune* si elle n'est dominée par aucune fonction calculable. $\diamond$

En particulier, les fonctions calculables étant stables par modifications finies, une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ est hyperimmune si pour toute fonction calculable $g : \mathbb{N} \rightarrow \mathbb{N}$, $f(x) > g(x)$ pour une infinité de valeurs x .

Exercice 4.2. Montrer qu'une fonction f est hyperimmune si, et seulement si, il existe pour toute fonction totale calculable g une infinité d'entiers x tels que $f(x) > g(x)$. $\diamond$

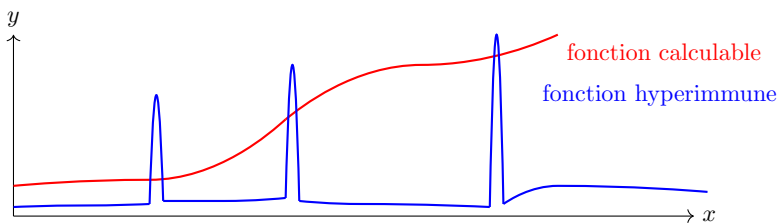


FIGURE 4.3 – Illustration d'une fonction hyperimmune : elle ne croît pas nécessairement très vite, mais pour toute fonction calculable f elle se situe infiniment souvent au-dessus de f .

Tout comme les degrés DNC, les degrés Turing des fonctions hyperimmunes sont clos par le haut. Un degré Turing est *hyperimmune* s'il contient une fonction hyperimmune, ou de manière équivalente si un de ses éléments calcule une fonction hyperimmune. L'appellation « fonction hyperimmune » provient de la correspondance suivante avec les ensembles hyperimmunes.

Proposition 4.4. Un ensemble infini X est hyperimmune si, et seulement si, la fonction $p_X : \mathbb{N} \rightarrow \mathbb{N}$ qui à n associe le n -ième élément de X est hyperimmune. ★

PREUVE. $\Rightarrow$. Soit X un ensemble hyperimmune et soit $g : \mathbb{N} \rightarrow \mathbb{N}$ une fonction totale calculable. Montrons que p_X n'est pas dominée par g . Soit h la fonction donnée par $h(x) = g(x) + x + 1$, et soit $(F_n)_{n \in \mathbb{N}}$ le tableau c. e. que l'on définit par $F_n = [h^{(n)}(0), h^{(n+1)}(0)[$, où $h^{(n)}$ est la n -ième itération de h , avec $h^{(0)}$ la fonction identité. Par hyperimmunité de X , il existe n tel que $F_n \cap X = \emptyset$. Cela signifie que $p_X(h^{(n)}(0)) \geq h^{(n+1)}(0) = h(h^{(n)}(0))$. En prenant $x = h^{(n)}(0)$, nous avons $p_X(x) \geq h(x) = g(x) + x + 1$, donc p_X n'est pas dominée par g .

$\Leftarrow$. Supposons que p_X soit une fonction hyperimmune. Raisonnons par l'absurde, en supposant que l'ensemble X ne soit pas hyperimmune. Autrement dit, il existe un tableau c. e. $(F_n)_{n \in \mathbb{N}}$ tel que $F_n \cap X \neq \emptyset$ pour tout n . Alors, la fonction $g : \mathbb{N} \rightarrow \mathbb{N}$ définie par $g(n) = \max \bigcup_{i \leq n} F_i$ est totale calculable, et domine p_X , contredisant l'hyperimmunité de p_X . ■

Il s'ensuit qu'un degré est hyperimmune si, et seulement si, il contient un ensemble hyperimmune.

Exercice 4.5. Montrer que les degrés hyperimmunes sont clos par le haut. Autrement dit, si X calcule une fonction hyperimmune, alors le degré Turing de X contient une fonction hyperimmune. ◇

Exercice 4.6. Montrer par la méthode des extensions finies qu'il existe un ensemble de degré hyperimmune. ◇

L'existence de degrés non calculables et non hyperimmunes n'est pour le moment pas claire. Nous verrons dans la section suivante que de tels degrés existent bien, même si nous comprendrons plus tard que ce n'est pas donné, dans le sens ou « beaucoup » d'ensembles sont de degrés hyperimmune (voir la proposition 10-3.35 et le théorème 19-3.4). De fait, il va falloir travailler pour en exhiber un qui ne le soit pas. Nous voyons en particulier dès à présent que la construction d'un degré non calculable et non hyperimmune ne peut pas se faire à l'aide de $\emptyset'$. En particulier, des constructions par extensions finies comme vues dans la section 4-8, qui sont toutes effectives en $\emptyset'$, ne pourront pas fonctionner.

Proposition 4.7 (Martin et Miller [157]). Tout ensemble Δ_2^0 non calculable est hyperimmune. ★

PREUVE. Soit A un ensemble Δ_2^0 non calculable, et soit $A_0, A_1, \dots$ une approximation Δ_2^0 de A . Rappelons que la *fonction de calcul* (Définition 4

-7.7) est définie comme la fonction $c_A : \mathbb{N} \rightarrow \mathbb{N}$ qui à x associe le plus petit entier $n \geq x$ tel que $A_n \upharpoonright_x = A \upharpoonright_x$. En particulier, $c_A \leq_T A$. D'après la proposition 4-7.9, toute fonction dominant c_A calcule A . Comme A n'est pas calculable, c_A n'est dominée par aucune fonction calculable, autrement dit c_A est hyperimmune. ■

Nous passons à présent à l'existence de degrés non calculables et non hyperimmunes, que l'on dira *calculatoirement dominés*.

5. Degrés calculatoirement dominés

Par clôture des degrés hyperimmunes par le haut, un degré Turing n'est pas hyperimmune si toute fonction f qu'il calcule est dominée par une fonction calculable g (qui dépend de f).

Définition 5.1. Un ensemble X est *calculatoirement dominé* si pour toute fonction $f \leq_T X$ il existe une fonction calculable g dominant f . Un degré Turing $\mathbf{d}$ est calculatoirement dominé^a si tout $X \in \mathbf{d}$ est calculatoirement dominé. ◇

a. En anglais, les terminologies « computably dominated » et « hyperimmune-free » coexistent.

Notons que le fait d'être calculatoirement dominé est une propriété de faiblesse et est donc close par le bas dans les degrés Turing. De fait, la propriété inverse — être de degré hyperimmune — est une propriété de force.

L'exemple le plus simple de degré calculatoirement dominé est le degré Turing des ensembles calculables. Le but de cette section est de démontrer l'existence de degrés calculatoirement dominés différents de $\mathbf{0}$. Nous verrons en fait un peu plus tard (théorème 8-5.1) que les degrés calculatoirement dominés sont en quantité indénombrable : il est possible de construire une injection $f : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ telle que pour tout X , l'ensemble $f(X)$ est de degré calculatoirement dominé, et même telle que $f(X)$ et $f(Y)$ soient dans des degrés Turing différents pour $X \neq Y$ (exercice 8-5.3). Le concept au cœur de la construction qui va suivre est celui de *f-arbre*.

Définition 5.2. Un *f-arbre* est une fonction totale $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ telle que pour tous $\sigma, \tau \in 2^{<\mathbb{N}}$, $\sigma \preceq \tau$ si, et seulement si, $T(\sigma) \preceq T(\tau)$. ◇

Soit T un *f-arbre*. Notons que l'image de T ($\text{Im } T$) n'est pas close par préfixe en général. Notons également que pour tout $\sigma \in 2^{<\mathbb{N}}$, $T(\sigma 0)$ et $T(\sigma 1)$ sont deux chaînes incompatibles étendant $T(\sigma)$. Seule l'image d'un *f-arbre* T est importante. La structure arborescente du domaine de T induit de manière canonique celle de $\text{Im } T$. Le lecteur peut se reporter à la figure 5.4 pour une représentation graphique d'un *f-arbre*.

Définition 5.3. Soit T un f -arbre. On appelle *nœuds* les éléments de $\text{Im } T$. Un *chemin* de T est une suite $P \in 2^{\mathbb{N}}$ dont une infinité de segments initiaux appartiennent à $\text{Im } T$. On notera $[T]$ l'ensemble des chemins de T . $\diamond$

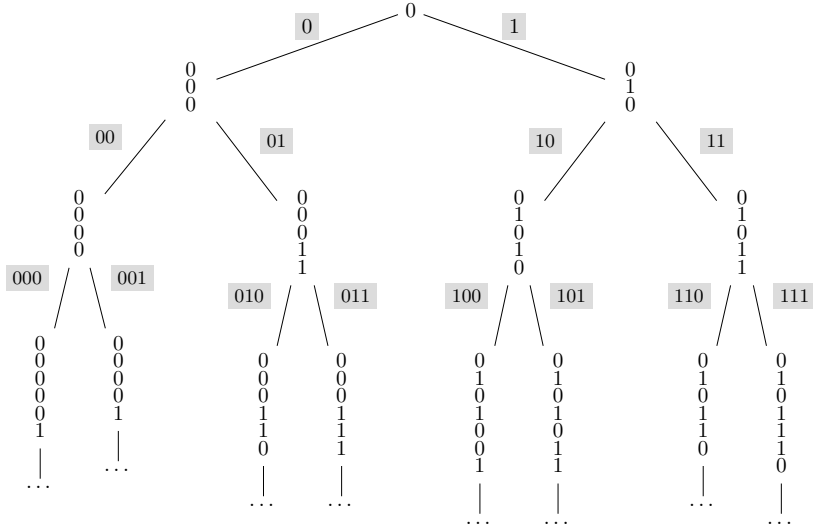


FIGURE 5.4 – Illustration d'un f -arbre T , dont le domaine est représenté par les chaînes en gris clair : $T(\epsilon) = 0$, $T(0) = 000$, $T(1) = 010$, ...

La figure 5.4, ci-dessus, illustre le fait qu'un f -arbre T induit une injection $f_T : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$, calculable à l'aide du f -arbre : pour $X \in 2^{\mathbb{N}}$, la suite $f_T(X)$ se calcule petit à petit comme étant $T(X \upharpoonright_1) \prec T(X \upharpoonright_2) \prec \dots$. De plus, par les propriétés d'un f -arbre, si $X \neq Y$, alors $f_T(X) \neq f_T(Y)$.

Définition 5.5. Un *sous- f -arbre* d'un f -arbre T est un f -arbre S tel que

$$\text{Im } S \subseteq \text{Im } T.$$

$\diamond$

Il s'ensuit que si S est un sous- f -arbre de T , alors $[S] \subseteq [T]$. Nous avons à présent les ingrédients nécessaires pour passer à la preuve du théorème annoncé.

Théorème 5.6 (Martin et Miller [157])

Il existe un degré $\mathbf{d} > \mathbf{0}$ calculatoirement dominé.

PREUVE. Nous voulons construire un ensemble A de degré calculatoirement dominé en respectant les contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$ et $(\mathcal{S}_e)_{e \in \mathbb{N}}$ suivants :

$$\mathcal{R}_e : W_e \neq A \quad \mathcal{S}_e : \Phi_e^A \text{ total} \Rightarrow \Phi_e^A \text{ dominé par une fonction calculable.}$$

Nous allons construire une suite infinie de f-arbres calculables $T_0, T_1, T_2, \dots$ tels que pour tout $e \in \mathbb{N}$,

- (1) T_{e+1} est un sous-f-arbre de T_e ;
- (2) $|T_e(\epsilon)| \geq e$;
- (3) Pour tout chemin $P \in [T_{2e+1}]$, le contrat $\mathcal{R}_e$ est satisfait ;
- (4) Pour tout chemin $P \in [T_{2e+2}]$, le contrat $\mathcal{S}_e$ est satisfait.

Satisfaction d'un contrat $\mathcal{R}_e$. Donnons-nous un f-arbre calculable T . Comme $\sigma_0 = T(0)$ et $\sigma_1 = T(1)$ sont des chaînes incompatibles, il existe $i \in \mathbb{N}$ tel que $\sigma_0(i) \neq \sigma_1(i)$. Ainsi, soit $\sigma_0(i) \neq W_e(i)$, soit $\sigma_1(i) \neq W_e(i)$. Supposons que l'on soit dans le premier cas, l'autre étant symétrique. Alors, le sous-f-arbre S de T défini par $S(\rho) = T(0\rho)$ assure que pour tous les chemins $P \in [S]$, $\sigma_0 \leq P$, donc $P \neq W_e$. De plus, S est calculable en T , donc calculable.

Satisfaction d'un contrat $\mathcal{S}_e$. Soit T un f-arbre calculable. Nous allons construire un sous-f-arbre calculable S tel que soit Φ_e^P est partiel pour tout $P \in [S]$, soit Φ_e^P est total et dominé par une même fonction calculable pour tout $P \in [S]$. Notons qu'en plus de satisfaire le contrat $\mathcal{S}_e$, la fonctionnelle Φ_e^P sera soit partielle, soit totale quel que soit le chemin P . Les deux cas suivants se présentent.

Cas 1. Il existe un nœud $\sigma \in \text{Im } T$ et une entrée $x \in \mathbb{N}$ tels que $\Phi^\tau(x) \uparrow$ pour tout $\tau \in \text{Im } T$ tel que $\tau \succeq \sigma$. Soit $\rho \in 2^{<\mathbb{N}}$ tel que $T(\rho) = \sigma$. Par suite, le sous-f-arbre S de T défini par $S(\mu) = T(\rho\mu)$ assure que pour tous les chemins $P \in [S]$, des segments initiaux τ de P arbitrairement longs satisferont $\Phi^\tau(x) \uparrow$. Par la propriété de l'usage, il s'ensuit que $\Phi^P(x) \uparrow$.

Cas 2. Pour tout nœud $\sigma \in \text{Im } T$ et toute entrée $x \in \mathbb{N}$, il existe $\tau \in \text{Im } T$ tel que $\sigma \preceq \tau$ et $\Phi^\tau(x) \downarrow$. Nous allons définir S , un sous-f-arbre calculable de T tel que pour tout $\rho \in 2^{<\mathbb{N}}$, $\Phi_e^{S(\rho)}(|\rho|) \downarrow$. On calcule $S(\epsilon)$ comme étant le premier nœud de $\text{Im } T$ que l'on trouve tel que $\Phi_e^{S(\epsilon)}(0) \downarrow$. Supposons que l'on a calculé $S(\rho)$. Soit μ tel que $S(\rho) = T(\mu)$. Nous calculons $S(\rho 0)$ et $S(\rho 1)$ comme suit : pour chaque entier $i < 2$, $S(\rho i)$ est le premier nœud $\tau \in \text{Im } T$ que l'on trouve, avec $\tau \succeq T(\mu i)$ et tel que $\Phi^\tau(|\rho| + 1) \downarrow$. L'algorithme qui recherche les nœuds $S(\rho 0)$ et $S(\rho 1)$ aboutira toujours, par l'hypothèse que nous sommes dans le cas 2. Par construction, S est bien un f-arbre, et $\text{Im } S \subseteq \text{Im } T$. Plus encore, Φ_e^P est une fonction totale pour tout $P \in [S]$.

Montrons que Φ_e^P est dominée par une fonction calculable pour tout chemin $P \in [S]$. Soit $g : \mathbb{N} \rightarrow \mathbb{N}$ définie par $g(n) = \max\{\Phi_e^{S(\rho)}(n) : |\rho| = n\}$. Alors, pour tous $n \in \mathbb{N}$ et $P \in [S]$, $\Phi_e^P(n) \leq g(n)$. Le sous-f-arbre S de T satisfait bien le contrat $\mathcal{S}_e$.

Notons enfin pour satisfaire le point (2) ci-dessus que pour tout f-arbre T calculable et tout n , il existe un sous-f-arbre S tel que $S(\epsilon) \geq n$. En effet,

il suffit de fixer une chaîne ρ de longueur n , et de définir $S(\mu) = T(\rho\mu)$. Nous pouvons donc combiner les satisfactions des différents contrats et cette dernière observation pour construire une suite de f-arbres $T_0, T_1, \dots$ satisfaisant les propriétés (1) (2) (3) et (4) données plus haut.

Remarque

Chaque f-arbre T_e de la suite pris indépendamment est calculable, mais la suite $T_0, T_1, \dots$ n'est pas elle-même calculable.

Afin de terminer la preuve, nous avons besoin du lemme suivant.

Lemme 5.7. L'intersection $\bigcap_e [T_e]$ contient exactement un élément. ★

PREUVE. Comme $\text{Im } T_{e+1} \subseteq \text{Im } T_e$ et comme toute chaîne de $\text{Im } T_e$ est une extension de $T_e(\epsilon)$, il est clair que l'on a $T_0(\epsilon) \preceq T_1(\epsilon) \prec T_2(\epsilon) \prec \dots$

Par ailleurs, comme $|T_e(\epsilon)| \geq e$, la suite $T_0(\epsilon) \preceq T_1(\epsilon) \prec T_2(\epsilon) \prec \dots$ converge vers une unique suite infinie $X \in 2^{\mathbb{N}}$. Comme X a une infinité de préfixes dans chaque T_e , alors $X \in [T_e]$ pour tout e , et donc $X \in \bigcap_e [T_e]$. ■

Soit A l'élément de $\bigcap_e [T_e]$. Pour tout $e \in \mathbb{N}$, comme $A \in [T_{2e+1}]$, alors le contrat $\mathcal{R}_e$ est satisfait pour A , donc $W_e \neq A$. De plus, comme $A \in [T_{2e+2}]$, alors le contrat $\mathcal{S}_e$ est satisfait pour A , donc si Φ_e^A est total, Φ_e^A est dominé par une fonction calculable. Il s'ensuit que A est calculatoirement dominé et non calculable. ■

Remarque

Une analyse soigneuse de la construction précédente montre qu'il suffit de l'oracle $\emptyset''$ pour calculer uniformément la suite $(T_e)_{e \in \mathbb{N}}$. En effet, les seules parties non calculables sont les analyses de cas, qui sont des propriétés Σ_2^0 . Ainsi, il existe un degré $\mathbf{d} > \mathbf{0}$ à la fois Δ_3^0 et calculatoirement dominé. Comme nous l'avons vu avec la proposition 4.7, il n'est pas possible d'abaisser cette borne à Δ_2^0 .

Nous retrouverons la structure des f-arbres dans le chapitre 14 pour montrer l'existence de degrés Turing minimaux.

Exercice 5.8. (★★) Construire un f-arbre T , par la méthode des extensions finies, tel que tout $X, Y \in [T]$ sont dans des degrés Turing différents. ◇

Notons que nous avons annoncé l'existence d'une quantité indénombrable de degrés calculatoirement dominés. Cela sera fait avec le théorème 8-5.1. Nous donnons à présent quelques équivalences permettant de mieux cerner cette notion.

5.1. Réduction truth-table

On suppose ci-après que les fonctionnelles considérées essayent de calculer des ensembles d'entiers et non des fonctions, c'est-à-dire que si $\Phi(Y, n) \downarrow$ pour un certain oracle Y et un certain entier n , alors $\Phi(Y, n) \downarrow \in \{0, 1\}$ (si $\Phi(Y, n) \not\downarrow \in \{0, 1\}$, on considérera alors que la fonctionnelle diverge).

Soient une fonctionnelle Φ et un oracle X tels que $\forall n \Phi(X, n) \downarrow \in \{0, 1\}$. Étant donné un autre ensemble $Y \neq X$, il n'y a aucune raison pour que l'on ait également $\forall n \Phi(Y, n) \downarrow$. Une fonctionnelle totale avec un oracle ne l'est pas forcément avec les autres, et il ne devrait pas être très dur pour le lecteur de construire de tels exemples. Mais de telles fonctionnelles sont-elles nécessaires ? Si l'on suppose $X \geq_T Y$, peut-on toujours calculer Y à partir de X via une fonctionnelle totale sur tous les oracles ? Nous allons voir que ce n'est pas forcément le cas via une restriction de la notion de réduction Turing.

Définition 5.9. Pour tous ensembles $X, Y \subseteq \mathbb{N}$, on dit que X est *truth-table réductible* à Y , et l'on écrit $X \leq_{tt} Y$, s'il existe une fonctionnelle Φ telle que $\Phi(Y) = X$ et telle que $\Phi(Z)$ est totale pour tout oracle Z . On écrit $X \equiv_{tt} Y$ si $X \leq_{tt} Y$ et $Y \leq_{tt} X$. On écrit $X <_{tt} Y$ si $X \leq_{tt} Y$ et $Y \not\leq_{tt} X$. On appelle *degrés truth-table* les classes d'équivalences de la relation $\equiv_{tt}$. ◇

Remarquons la notation $\Phi(Y) = X$ signifiant $\forall n \Phi(Y, n) = X(n)$. Nous verrons plusieurs définitions équivalentes à la réduction truth-table, à commencer par celle justifiant son nom, que l'on peut traduire en français par « réduction par table de vérité ».

Définition 5.10. Une réduction par table de vérité est donnée par une suite calculable de paires $(\langle C_{0,n}, C_{1,n} \rangle)_{n \in \mathbb{N}}$ telle que pour tout n l'ensemble $C_{0,n} \cup C_{1,n} \subseteq 2^{<\mathbb{N}}$ contient exactement toutes les chaînes d'une certaine taille m_n , et telle que $C_{0,n} \cap C_{1,n} = \emptyset$. L'ensemble X calcule Y via cette réduction si pour tout n on a $Y(n) = i$ ssi $\sigma \in C_{i,n}$ pour un préfixe σ de X . ◇

Les ensembles $C_{i,n}$ sont les « tables de vérité ». N'importe quel oracle X admet un préfixe dans $C_{0,n} \cup C_{1,n}$. Si le préfixe appartient à $C_{0,n}$, alors X calcule 0 sur l'entrée n ; et, si le préfixe appartient à $C_{1,n}$, alors X calcule 1 sur l'entrée n . Voyons à présent les différentes équivalences à la notion de réduction truth-table.

Théorème 5.11

Soient X, Y des ensembles. Les énoncés suivants sont équivalents.

- (1) $Y \leq_{tt} X$.
- (2) X calcule Y via une réduction par table de vérité.
- (3) Il existe une fonctionnelle Φ et une fonction $b : \mathbb{N} \rightarrow \mathbb{N}$ totale calculable telle que $\Phi(X, n)[b(n)] \downarrow = Y(n)$ pour tout n .

PREUVE. Montrons (1) $\Rightarrow$ (2). Supposons $\Phi(X) = Y$ via une fonctionnelle Φ totale sur tous les oracles. Étant donné n , on cherche le plus petit temps de calcul t_n tel que, pour un certain $m_n \leq t_n$ et pour toute chaîne $\sigma \in 2^{\mathbb{N}}$ de taille m_n , on a $\Phi(\sigma, n)[t_n] \downarrow$. Afin de montrer qu'un tel temps de calcul t_n existe forcément pour tout n , on doit anticiper un peu sur la définition 8-1.1 d'arbre et le lemme 8-1.4 de König à venir. Supposons par l'absurde que, pour un certain n , on ne puisse trouver t_n . Cela implique en particulier que, pour tout m , il existe une chaîne σ de taille m telle que $\forall t \Phi(\sigma, n)[t] \uparrow$. Par ailleurs, si $\forall t \Phi(\sigma, n)[t] \uparrow$ et $\tau \preceq \sigma$, alors aussi $\forall t \Phi(\tau, n)[t] \uparrow$. On peut donc construire un arbre infini T tel que $\sigma \in T$ implique $\forall t \Phi(\sigma, n)[t] \uparrow$. D'après le lemme de König, T contient un chemin infini Y , qui est donc tel que $\Phi(Y, n) \uparrow$, ce qui contredit le fait que Φ soit totale sur tous ses oracles. On peut donc à chaque étape trouver t_n et $m_n \leq t_n$, avec l'ensemble $C_{0,n}$ des chaînes de tailles m_n sur lesquelles le calcul renvoie 0 en t_n étapes, et l'ensemble $C_{1,n}$ des chaînes de tailles m_n sur lesquelles le calcul renvoie 1 en t_n étapes.

Montrons (2) $\Rightarrow$ (3). Étant donné une réduction « table de vérité » donnée par la suite calculable $(\langle C_{0,n}, C_{1,n} \rangle)_{n \in \mathbb{N}}$, pour toute entrée n on peut borner le temps de calcul que la fonctionnelle met pour s'arrêter sur n avec n'importe quel oracle : il s'agit simplement du temps nécessaire pour produire le calcul de $\langle C_{0,n}, C_{1,n} \rangle$.

Montrons (3) $\Rightarrow$ (1). Soit Φ une fonctionnelle, et soit $b : \mathbb{N} \rightarrow \mathbb{N}$ une fonction totale calculable telle que

$$\Phi(Y, n)[b(n)] \downarrow = X(n), \quad \text{pour tout } n.$$

Alors, on construit la fonctionnelle Ψ qui sur tous les oracles Z et sur toute entrée n lance le calcul de $\Phi(Z, n)$ en $b(n)$ étapes. Si le calcul renvoie une valeur en $b(n)$ étapes, alors Ψ renvoie cette valeur, sinon Ψ renvoie 0. Le résultat du calcul est le même entre Φ et Ψ sur l'oracle Y , mais Ψ est maintenant totale sur tous les oracles. ■

Nous montrons à présent que les ensembles X pour lesquels $X \geq_T Y$ implique $X \geq_{tt} Y$ sont exactement les ensembles calculatoirement dominés.

Théorème 5.12 (Jockusch [97], Martin (non publié))

Un ensemble X est calculatoirement dominé ssi $Y \leqslant_T X \Leftrightarrow Y \leqslant_{tt} X$ pour tout $Y \in 2^{\mathbb{N}}$.

PREUVE. Supposons X calculatoirement dominé. Supposons $X \geqslant_T Y$ via la fonctionnelle Φ . Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $\Phi(X, n)[f(n)] \downarrow = Y(n)$ pour tout n . Notons que f est une fonction X -calculable. Il y a donc une fonction calculable $g > f$. On a ainsi $\Phi(X, n)[g(n)] \downarrow = Y(n)$. D'après le théorème 5.11, on a donc $X \geqslant_{tt} Y$.

Supposons maintenant que pour tout Y on ait $Y \leqslant_T X \Leftrightarrow Y \leqslant_{tt} X$. Alors, également pour toute fonction $f : \mathbb{N} \rightarrow \mathbb{N}$, on a $f \leqslant_T X \Leftrightarrow f \leqslant_{tt} X$, via la représentation canonique de f par une suite de $2^{\mathbb{N}}$. Soit $f \leqslant_T X$; montrons que f est dominée par une fonction calculable g . Par hypothèse, $f \leqslant_{tt} X$, donc par le théorème 5.11, il existe une fonctionnelle Φ et une fonction totale calculable $b : \mathbb{N} \rightarrow \mathbb{N}$ telle que $\Phi(X, n)[b(n)] \downarrow = f(n)$ pour tout n . On peut supposer sans perte de généralité que si $\Phi(X, n)[b(n)] \downarrow$, alors l'usage du calcul est inférieur à $b(n)$ (si ce n'est pas le cas, on peut ralentir le calcul pour que cela le soit), donc $\Phi(X \upharpoonright_{b(n)}, n)[b(n)] \downarrow$. Soit $g : \mathbb{N} \rightarrow \mathbb{N}$ la fonction qui pour une entrée n , exécute $\Phi(\sigma, n)[b(n)]$ pour tout $\sigma \in 2^{< \mathbb{N}}$ de longueur $b(n)$, et renvoie le maximum des valeurs obtenues. En particulier,

$$g(n) \geqslant \Phi(X \upharpoonright_{b(n)}, n)[b(n)] = f(n).$$

La fonction g domine f . Il s'ensuit que X est de degré calculatoirement dominé. ■

Nous avons à présent trois notions de réduction : la réduction many-one, la réduction truth-table et la réduction Turing. La proposition suivante récapitule certains résultats vus jusqu'ici, qui attestent qu'aucune ne coïncide avec une autre.

Proposition 5.13. Pour tous X, Y , on a

$$X \leqslant_m Y \Rightarrow X \leqslant_{tt} Y \Rightarrow X \leqslant_T Y.$$

Aucune implication inverse n'est vraie dans le cas général. ★

PREUVE. Les implications sont claires. Montrons qu'aucune implication réciproque ne tient. L'ensemble $\mathbb{N} \setminus \emptyset'$ est tt -réductible à $\emptyset'$ mais non m -réductible à $\emptyset'$, car il serait alors Σ_1^0 d'après la proposition 5-4.3, et donc $\emptyset'$ serait calculable ce qui est faux.

L'existence d'ensembles X, Y tels que $X \geqslant_T Y$ mais $X \not\leqslant_{tt} Y$ est une conséquence du théorème 5.12 et du fait que des degrés non calculatoirement dominés existent (par exemple $\mathbf{0}'$). ■

6. Théorème de domination de Martin

Les fonctions hyperimmunes sont par définition les fonctions qui ne sont dominées par aucune fonction calculable. Il est naturel de se poser la question de la puissance de calcul des fonctions qui dominent toutes les fonctions calculables. Bien entendu, aucune fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ ne domine toutes les fonctions calculables, car la fonction constante $f(0) + 1$ n'est pas dominée par f . On peut cependant affaiblir la propriété, et se poser la question de la puissance calculatoire d'une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que pour toute fonction calculable $g : \mathbb{N} \rightarrow \mathbb{N}$, f domine g *presque partout*, c'est-à-dire partout sauf pour un nombre fini de valeurs.

Notation

On utilisera les notations $\forall^\infty m$ et $\exists^\infty m$ pour signifier respectivement $\exists n \forall m > n$ et $\forall n \exists m > n$. Ainsi, $\forall^\infty m$ signifie « pour presque tout m », et $\exists^\infty m$ signifie « pour une infinité de m ».

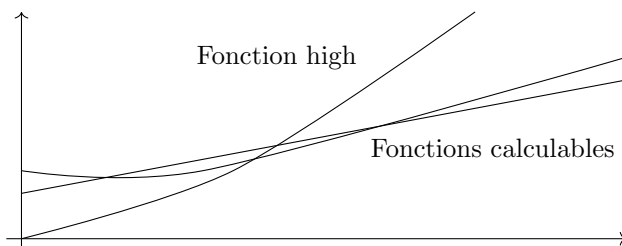


FIGURE 6.1 – Illustration d'une fonction high, qui pour toute fonction calculable f , est toujours au-dessus de f à partir d'une certaine valeur

Le théorème de domination de Martin ((1) $\Leftrightarrow$ (2) dans le théorème suivant [148]) donne une magnifique caractérisation des degrés Turing de ces fonctions. L'équivalence (2) $\Leftrightarrow$ (3) montrée par Jockusch [98] est venue plus tard et présente elle aussi son intérêt. Rappelons qu'un ensemble A est high si $A' \geq_T \emptyset''$ (voir la définition 4-10.1).

Théorème 6.2

Soit $A \subseteq \mathbb{N}$ un ensemble. Les énoncés suivants sont équivalents.

- (1) A est high.
- (2) A calcule une fonction $g : \mathbb{N} \rightarrow \mathbb{N}$ qui domine presque partout toutes les fonctions calculables. Cela signifie que pour toute fonction calculable f , on a $\forall^\infty n \ f(n) \leq g(n)$.
- (3) A calcule une liste $(X_n)_{n \in \mathbb{N}}$ contenant (éventuellement avec répétitions) exactement les ensembles calculables.

PREUVE. Montrons $(1) \Rightarrow (2)$. Supposons que A soit high. On a donc une description $\Delta_2^0(A)$ de $\emptyset''$, c'est-à-dire une fonction A -calculable $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ telle que $\lim_{s \rightarrow \infty} f(n, s) = \emptyset''(n)$ pour tout n . Notons qu'être le code d'une fonction totale est une propriété Π_2^0 . On peut en particulier, en utilisant le fait que $\emptyset''$ soit Σ_2^0 -complet (voir la proposition 5-5.3), calculer pour tout e un code a_e tel que $\emptyset''(a_e) = 0$ ssi Φ_e est une fonction totale.

On définit g de la manière suivante : sur l'entrée t , pour toute fonctionnelle Φ_e pour $e \leq t$, on cherche le plus petit temps de calcul $s \geq t$ tel que $f(a_e, s) = 1$ ou tel que $\Phi_e(t)[s] \downarrow$. Notons qu'un des deux événements arrive forcément : soit Φ_e est totale et donc $\Phi_e(t) \downarrow$, soit Φ_e est partielle et donc $\lim_{s \rightarrow \infty} f(a_e, s) = \emptyset''(a_e) = 1$. Dans le premier cas, on définit $v_{t,e} = 0$, et dans le second, $v_{t,e} = \Phi_e(t)$. On définit finalement $g(t) = \sum_{e \leq t} v_{t,e}$.

Il est clair que pour toute fonction totale Φ_e , à partir du plus petit $t \geq e$ tel que $f(a_e, s) = 0$ pour $s \geq t$, on aura $g(s) \geq \Phi_e(s)$ pour tout $s \geq t$. Donc, g domine presque partout toutes les fonctions calculables.

Montrons $(2) \Rightarrow (3)$. Supposons à présent que A calcule une fonction g qui domine presque partout toute fonction calculable. On utilise le fait que si Φ_e est totale à valeur dans $\{0, 1\}$, alors la fonction calculable $t : \mathbb{N} \rightarrow \mathbb{N}$ qui sur n renvoie le plus petit temps de calcul tel que $\Phi_e(n)[t(n)] \downarrow$, pour tout n , est dominée par g presque partout.

Pour chaque fonctionnelle Φ_e , on calcule l'ensemble Y_e comme étant

$$Y_e(n) = \Phi_e(n)[g(n)] \text{ si } \Phi_e(n)[g(n)] \downarrow \in \{0, 1\}, \text{ et } Y_e(n) = 0 \text{ sinon.}$$

La liste $(Y_e)_{e \in \mathbb{N}}$ contient donc, à modification finie près, exactement les ensembles calculables. Pour les obtenir tous, on calcule en fin de compte la suite $(X_n)_{n \in \mathbb{N}}$ comme étant toutes les modifications finies possibles des ensembles Y_e .

Montrons $(3) \Rightarrow (1)$. Le lecteur peut s'aider de la figure 6.4 pour la compréhension de cette implication. Supposons que A calcule une liste $(X_n)_{n \in \mathbb{N}}$ contenant (éventuellement avec répétitions) exactement les ensembles calculables. Soit $P = \{e : \forall x_1 \exists x_2 R(e, x_1, x_2)\}$ un ensemble Π_2^0 quelconque. Montrons que P est $\Sigma_2^0(A)$. Pour cela, on définit uniformément pour tout e une fonction partielle calculable $f_e : \mathbb{N} \rightarrow \{0, 1\}$ telle que :

- (a) $e \in P$ implique que f_e est une fonction totale calculable ;
- (b) $e \notin P$ implique que f_e est une fonction partielle qui ne peut pas être complétée en une fonction totale calculable.

Soit e fixé. On décrit un processus uniforme en e . À l'étape de calcul t , pour toute valeur $n \leq t$ et telle que f_e ne s'arrête pas pour le moment sur n , on procède comme suit : si $\Phi_n(n)[t] \downarrow \neq 0$, on définit $f_e(n) = 0$.

Sinon, si $\Phi_n(n)[t] \downarrow = 0$, on définit $f_e(n) = 1$. Sinon, si pour tout $k \leq n$ il existe $m_k \leq t$ tel que $R(e, k, m_k)$, alors on définit $f_e(n) = 0$.

Le processus est clairement calculable. Montrons (a). Supposons $e \in P$. Alors, pour tout n , il existe un plus petit t tel que pour tout $k \leq n$ il existe $m_k \leq t$ pour lequel $R(e, k, m_k)$. Quand cela arrive, alors $f_e(n)$ prend une valeur à l'étape t si elle n'en a pas prise jusqu'ici. Donc, f_e est totale. Montrons à présent l'assertion (b). Supposons $e \notin P$. Soit n le plus grand entier tel que pour tout $k \leq n$ il existe m_k pour lequel $R(e, k, m_k)$. Pour $m > n$, $f_e(m)$ s'arrête ssi $\Phi_m(m)$ s'arrête, auquel cas $f_e(m) \neq \Phi_m(m)$. Supposons par l'absurde que f_e a une complétion calculable. Alors, par le lemme de remplissage (le lemme 3-5.1), elle a une complétion calculable de code $a > n$. Dans ce cas, $f_e(a) = \Phi_a(a)$, ce qui contredit la définition de f_e . On a donc (b).

Il s'ensuit que P peut s'écrire comme un ensemble $\Sigma_2^0(A)$ de la manière suivante.

$$P = \{e : \exists n \forall m \forall t f_e(m)[t] \uparrow \vee f_e(m)[t] \downarrow = X_n(m)\}.$$

En effet, si $e \in P$, alors f_e est calculable, et coïncide ainsi avec un certain X_n . À l'inverse, si $e \notin P$, alors f_e n'a pas de complétion calculable, et donc pas de complétion dans $(X_n)_{n \in \mathbb{N}}$. Comme P est $\Sigma_2^0(A)$ et $\overline{P}$ est par définition Σ_2^0 , on a donc que P est $\Delta_2^0(A)$, et P est dès lors A' -calculable. Il suffit d'appliquer cela pour $P = \mathbb{N} \setminus \emptyset''$ pour obtenir que $\emptyset''$ est $\Delta_2^0(A)$, et donc A' -calculable. ■

L'implication (3) $\Rightarrow$ (1) du théorème 6.2 est loin d'être évidente. Nous espérons que le lecteur saura apprécier l'argument, dont la subtile complexité est caractéristique du travail de Jockusch. L'exercice suivant donne des caractérisations alternatives similaires, plus simples à démontrer.

Exercice 6.3. (★) Montrer les équivalences suivantes :

- (1) A calcule une fonction qui domine presque partout toute fonction calculable ;
- (2) A calcule une suite $(f_n)_{n \in \mathbb{N}}$ contenant toutes les fonctions calculables de $\mathbb{N}$ vers $\mathbb{N}$;
- (3) A calcule une suite $(X_n)_{n \in \mathbb{N}}$ ne contenant que des ensembles infinis, et contenant tous les ensembles calculables infinis. ◇

Discutons un peu de la caractérisation (1) $\Leftrightarrow$ (2) du théorème 6.2, illustrée par la figure 6.1. Le fait de pouvoir calculer une fonction de $\mathbb{N}$ dans $\mathbb{N}$ à très forte croissance est susceptible de donner beaucoup de puissance de calcul. Ainsi, par exemple, comme nous l'avons vu, calculer une fonction qui borne le temps d'arrêt des programmes informatiques permet de calculer $\emptyset'$.

f_e	X_0	X_1	X_2	$\dots$
$f_e(0)$	$X_0(0)$	$X_1(0)$	$X_2(0)$	
$f_e(1)$	$X_0(1)$	$X_1(1)$	$X_2(1)$	
$\uparrow$	$X_0(2)$	$X_1(2)$	$X_2(2)$	
$\uparrow$	$X_0(3)$	$X_1(3)$	$X_2(3)$	
$f_e(4)$	$X_0(4)$	$X_1(4)$	$X_2(4)$	
$\uparrow$	$X_0(5)$	$X_1(5)$	$X_2(5)$	
$f_e(6)$	$X_0(6)$	$X_1(6)$	$X_2(6)$	
$\uparrow$	$X_0(7)$	$X_1(7)$	$X_2(7)$	
$\dots$	$\dots$	$\dots$	$\dots$	

FIGURE 6.4 – Illustration de la preuve $(3) \Rightarrow (1)$ du théorème 6.2 : si $e \notin P$, on construit une fonction partielle calculable $f_e : \mathbb{N} \rightarrow \{0, 1\}$ qui n'a pas de complétion totale calculable. Dans l'illustration, la fonction f_e ne peut être égale à aucun des ensembles $X_0, X_1, \dots$ sur toutes les valeurs pour lesquelles elle est définie. En effet, la liste $(X_n)_{n \in \mathbb{N}}$ ne contient que des ensembles calculables. Dans le cas inverse, la fonction f_e est totale calculable, et est donc égale à au moins un des ensembles X_i , puisque la liste $(X_n)_{n \in \mathbb{N}}$ contient tous les ensembles calculables.

Le théorème précédent indique qu'il y a un premier niveau entre les fonctions permettant de calculer l'arrêt simplement parce qu'elles croissent très vite, et les fonctions de croissance calculable : il existe d'après la proposition 4-10.2 des ensembles high ne calculant pas l'arrêt, et donc des fonctions de croissance « intermédiaire ». Nous verrons également que plus une fonction croît rapidement, plus elle a une puissance de calcul importante. Une fonction qui croît suffisamment vite pourra calculer $\emptyset''$, une qui croît encore plus vite pourra calculer $\emptyset'''$, etc. Nous verrons malgré tout avec le théorème 29-5.4 qu'il existe une limite à la puissance de calcul que confère une croissance rapide. La classe des ensembles calculables par n'importe quelle fonction qui croît « suffisamment » vite a une caractérisation précise et reste malgré tout dénombrable.

Exercice 6.5. (★★) La notion d'ensemble c. e. maximal fut introduite dans l'exercice 3-7.13. Un ensemble c. e. X est maximal si $\mathbb{N} \setminus X$ est infini, et si tout ensemble c. e. $Y \supseteq X$ est tel que $Y \setminus X$ est fini ou tel que $\mathbb{N} \setminus Y$ est fini. Soit X un ensemble c. e. maximal ; montrer que X est high. $\diamond$

7. Degrés High ou DNC

Nous terminons ce chapitre par un résultat qui combine degré high et DNC, afin d'obtenir une caractérisation naturelle en termes de puissance de calcul : la possibilité de calculer une fonction qui diffère presque partout de toute fonction calculable.

Théorème 7.1 (Kjos-Hanssen, Merkle et Stephan [113])

Soit $X \in 2^{\mathbb{N}}$. Les énoncés suivants sont équivalents :

- (1) X est de degré high ou DNC ;
- (2) X calcule une fonction qui est différente presque partout de toute fonction calculable.

PREUVE. Montrons (1) $\Rightarrow$ (2). Supposons pour commencer que X est high. Soit $g \leq_T X$ une fonction qui domine presque partout toute fonction calculable. Alors, également, $m \mapsto g(m) + 1$ est différente presque partout de toute fonction calculable. Supposons à présent que X est DNC. Pour chaque n , on calcule le code e_n tel que W_{e_n} énumère toutes les valeurs $\Phi_e(n)$ pour $e \leq n$ pour lesquelles $\Phi_e(n) \downarrow$. D'après le théorème 2.6, X calcule une fonction $h : \mathbb{N}^2 \rightarrow \mathbb{N}$ telle que pour tout $e, n \in \mathbb{N}$, si $|W_e| \leq n$, alors $h(e, n) \notin W_e$. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ la fonction X -calculable définie par $f(n) = h(e_n, n)$. On a alors $f(n) \notin W_{e_n}$ pour tout n . Ainsi, $f(n)$ est différent de $\Phi_0(n), \Phi_1(n), \dots, \Phi_n(n)$ pour tout n , donc f diffère presque partout de toute fonction calculable (et même de toute fonction partielle calculable s'arrêtant sur une infinité de valeurs).

Montrons (2) $\Rightarrow$ (1). Si X est high, il n'y a rien à vérifier.

Supposons donc que X n'est pas high. Soit $g \leq_T X$ telle que Φ_e totale implique $\forall^\infty m \ g(m) \neq \Phi_e(m)$. Nous prétendons avoir également

$$\forall^\infty e \ g(e) \neq \Phi_e(e).$$

Supposons par l'absurde le contraire. Soit f la fonction X -calculable qui sur n renvoie le plus petit temps de calcul t tel que $g(m) = \Phi_m(m)[t] \downarrow$ pour un entier $m > n$. Comme X est non high, il existe une fonction calculable b telle que $\exists^\infty n \ b(n) \geq f(n)$. Notons que l'on peut supposer sans perte de généralité que $b(n) \leq b(n+1)$.

On définit à présent la fonction totale calculable h par $h(n) = \Phi_n(n)[b(n)]$ si $\Phi_n(n)[b(n)] \downarrow$, et $h(n) = 0$ sinon. Supposons à présent $b(n) > f(n)$. Par définition de f , il existe une valeur $m > n$ telle que $\Phi_m(m)[f(n)] \downarrow = g(m)$, et donc telle que $\Phi_m(m)[b(n)] \downarrow = g(m)$. Comme $b(m) \geq b(n)$, on aura

$$h(m) = \Phi_m(m)[b(m)] \downarrow = g(m).$$

Comme on peut recommencer l'argument pour des valeurs de n arbitrairement grandes, on aura $h(m) = g(m)$ pour une infinité de m . Cela contredit

le fait que g diffère presque partout de toute fonction calculable. Donc, nous avons $\forall^\infty e \ g(e) \neq \Phi_e(e)$. Il suffit alors de modifier un nombre fini de valeurs de g pour obtenir une fonction DNC. ■

Notons pour donner tout son éclat au théorème précédent qu'il est possible de construire des degrés DNC qui ne sont pas high (en combinant le corollaire 18-4.3 avec le corollaire 19-3.9 ou plus simplement en considérant le corollaire 8-6.6) tout comme des degrés high qui ne sont pas DNC (voir le corollaire 10-3.34).

Chapitre 8

Classes Π_1^0 et degrés PA

Nous nous sommes jusqu'à présent concentrés principalement sur l'étude des ensembles d'entiers naturels pris individuellement, ou de manière équivalente, des fonctions des entiers vers les entiers. Nous allons maintenant nous tourner vers l'étude de *classes* d'ensembles ou de fonctions, c'est-à-dire d'ensembles d'ensembles d'entiers.

Nous avons déjà vu de nombreuses classes d'ensembles, notamment la classe des ensembles de degré high $\{X \in 2^{\mathbb{N}} : X' \geq_T \emptyset''\}$, ou celle des ensembles de degré low $\{X \in 2^{\mathbb{N}} : X' \equiv_T \emptyset'\}$.

Ensemble vs classe

Afin de distinguer les ensembles d'entiers des sous-ensembles de l'espace de Cantor, nous les appellerons respectivement « ensembles » et « classes » quand nous sommes dans un contexte portant sur l'étude des ensembles $X \in 2^{\mathbb{N}}$ ou des classes $\mathcal{A} \subseteq 2^{\mathbb{N}}$.

Les classes que nous considérerons seront définies par des prédicats, et l'étude de la complexité de ces prédicats permettra de déduire des informations sur les éléments que la classe contient.

L'étude des classes va rapidement se montrer centrale dans l'étude des ensembles d'entiers. On commence ici par les classes de la complexité la plus simple possible : les ouverts et fermés effectifs du Cantor. L'étude de classes de complexité supérieure sera poursuivie dans le chapitre 17. Malgré la simplicité apparente des ouverts et fermés, nous verrons rapidement le large éventail de possibilités et la grande richesse qu'ils renferment.

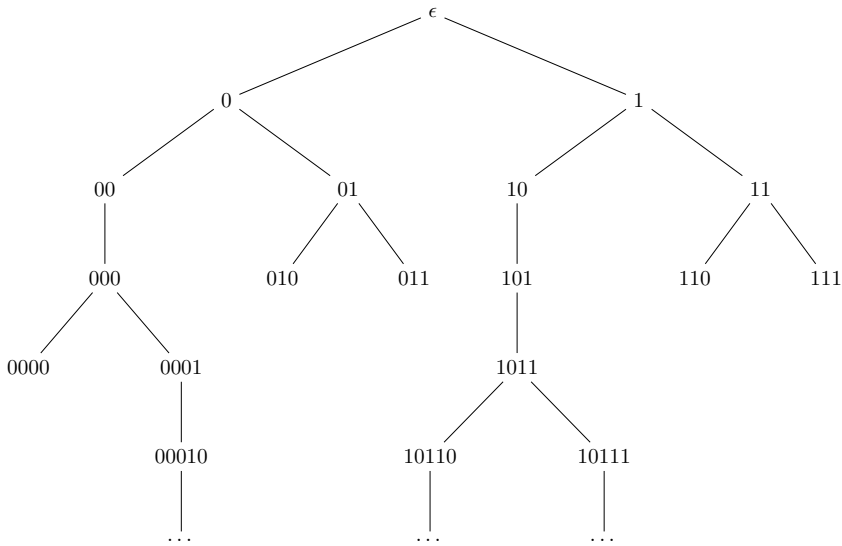


FIGURE 1.2 – Illustration d'un arbre. La racine ϵ est la chaîne vide. Chaque nœud admet éventuellement un successeur gauche, un successeur droit, les deux — auquel cas il est branchant — ou aucun des deux — auquel cas c'est une feuille.

1. Arbres binaires

Nous avons défini la notion de f-arbre afin de montrer l'existence d'un degré calculatoirement dominé non calculable (le théorème 7-5.6). Nous introduisons ici une notion similaire et d'une certaine manière plus primitive, à savoir tout simplement les arbres.

Définition 1.1. Un ensemble $T \subseteq 2^{<\mathbb{N}}$ est un *arbre* si T est clos par préfixe, c'est-à-dire pour tous $\sigma \in T$ et $\tau \preceq \sigma$, alors $\tau \in T$. $\diamond$

Étant donné un arbre $T \subseteq 2^{<\mathbb{N}}$, on appelle un élément $\sigma \in T$ un *nœud* de l'arbre. Un nœud est *branchant* si $\sigma 0, \sigma 1 \in T$. Dans le cas inverse, il est *non branchant*. On considérera par une sorte de convention sur la représentation mentale que l'on peut avoir d'un arbre, qu'une extension $\sigma 0$ de σ « va à gauche dans l'arbre » alors qu'une extension $\sigma 1$ de σ « va à droite ». Si $\sigma 0 \in T$, alors $\sigma 0$ est un *successeur gauche* de σ . Si $\sigma 1 \in T$, alors $\sigma 1$ est un *successeur droit* de σ . Un nœud sans successeur sera appelé une *feuille*.

Définition 1.3. Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre. Un *chemin* à travers l'arbre T est une suite $P \in 2^{\mathbb{N}}$ telle que $P \upharpoonright_n \in T$ pour tout $n \in \mathbb{N}$. On dénote par $[T]$ la classe des chemins de T . $\diamond$

Intuitivement, un chemin P peut être vu comme une suite d'instructions binaires, nous indiquant littéralement « un chemin » à suivre à travers l'arbre. Un bit à 0 dans le chemin nous indique de continuer notre parcours en suivant le successeur gauche et un bit à 1 nous indique de suivre le successeur droit. Nous ne considérons que des chemins infinis.

Remarque

Un chemin n'est pas représenté comme un ensemble de nœuds de l'arbre. Il existe cependant une bijection calculable entre un chemin P et l'ensemble $\{P|_n : n \in \mathbb{N}\}$. Représenter un chemin comme une suite binaire infinie est donc principalement un choix conventionnel qui s'avérera très utile par la suite.

Si T est un arbre fini, c'est-à-dire qui n'a qu'un nombre fini de nœuds, alors $[T]$ est forcément l'ensemble vide. Qu'en est-il de la réciproque ? Il s'agit d'un outil central sur les arbres : le lemme de König, qui stipule que tout arbre infini à branchement fini admet un chemin infini. Notez que les arbres $T \subseteq 2^{<\mathbb{N}}$ sont nécessairement 2-branchants. On a affaire dans ce cas au lemme de König *faible*.

Lemme 1.4 (Lemme de König faible). Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre tel que $|T| = \infty$. Alors, $[T]$ est non vide. ★

PREUVE. On construit un chemin X par récurrence sur n . Comme T est infini, par le principe des tiroirs il existe $i \in \{0, 1\}$ et une infinité de nœuds $\sigma \in T$ qui étendent i (c'est-à-dire avec $i \prec \sigma$). On définit $X(0) = i$. Supposons que $\tau = X(0)X(1) \dots X(n)$ soit défini avec $\tau \in T$ et tel qu'il y a une infinité de nœuds $\sigma \in T$ pour lesquels $\tau \preceq \sigma$. Par le principe des tiroirs, il existe $i \in \{0, 1\}$ et une infinité de nœuds $\sigma \in T$ qui étendent τi . On définit $X(n+1) = i$.

Par récurrence sur n , on définit donc de cette manière un ensemble X tel que $X|_n \in T$ pour tout n . ■

Le lemme de König peut paraître trivial au premier abord. Le lecteur, à l'intuition bien affûtée et à l'aise avec la manipulation d'objets infinis, se sera éventuellement demandé s'il y avait vraiment besoin de ranger cet énoncé dans un lemme. Nous allons voir que malgré les apparences, ce lemme n'est pas aussi trivial que cela. Bien que simple, il constitue un outil central, particulièrement intéressant de par son contenu calculatoire.

1.1. Arbres calculables

Un arbre $T \subseteq 2^{<\mathbb{N}}$ est calculable si l'ensemble T est calculable, autrement dit s'il existe une procédure pour décider si un nœud appartient à l'arbre ou non. Au cours de ce chapitre, nous allons tenter de répondre à la question suivante.

Question 1.5. Étant donné un arbre calculable infini, quelle est la puissance de calcul nécessaire pour calculer un chemin infini de T ? ★

La preuve du lemme de König fournit une construction claire : quand on a calculé un préfixe τ de notre chemin infini, on détermine le prochain bit comme étant 0 si l'arbre contient une infinité de nœuds qui étendent $\tau 0$ et comme étant 1 sinon. Le problème est qu'il n'est *a priori* pas possible de savoir de manière calculable si une infinité de nœuds étendent $\tau 0$: c'est une question pour laquelle l'arrêt des programmes informatiques semble nécessaire. Cela nous conduit à définir la notion de nœud extensible.

Définition 1.6. Un nœud σ d'un arbre $T \subseteq 2^{<\mathbb{N}}$ est *extensible* dans T si l'ensemble $\{\tau \in T : \tau \succeq \sigma\}$ est infini. ◇

Autrement dit, un nœud σ est extensible dans un arbre si le sous-arbre des nœuds compatibles avec σ est infini. Souvenons-nous de la notation $[\sigma]$ qui dénote la classe des ensembles X ayant σ comme préfixe. Par le lemme de König, un nœud σ est extensible dans T si et seulement si $[\sigma] \cap [T] \neq \emptyset$. Notons que dans tout arbre infini, la racine ϵ est un nœud extensible, et que si σ est extensible, alors au moins un nœud parmi $\sigma 0$ et $\sigma 1$ l'est également.

L'exercice suivant montre que les nœuds extensibles sont suffisants pour décrire l'ensemble des chemins d'un arbre.

Exercice 1.7. Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre, et soit S l'ensemble des nœuds extensibles dans T . Montrer que S est un arbre, et que $[T] = [S]$. ◇

Il s'ensuit de la définition de nœud extensible que les feuilles ne sont pas extensibles. En revanche, si l'ensemble des feuilles d'un arbre calculable est décidable, ce n'est pas en général le cas de l'ensemble des nœuds extensibles.

Exercice 1.8. Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre calculable infini ne contenant que des nœuds extensibles. Montrer alors que T contient un chemin infini calculable. ◇

De manière générale, déterminer si un ensemble calculable est infini ou non demande l'oracle $\emptyset''$. Dans le cas des arbres, nous pouvons exploiter la clôture par préfixe, pour ramener la complexité de l'oracle à $\emptyset'$. La notation 2^n de la proposition suivante désigne l'ensemble des chaînes de taille n .

Proposition 1.9. Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre calculable. L'ensemble de ses nœuds extensibles est Π_1^0 . ★

PREUVE. Les arbres étant clos par le bas, $\{\tau \in T : \tau \succeq \sigma\}$ est infini ssi $\forall n > |\sigma| \exists \tau \in 2^n$ tel que $\tau \succeq \sigma$ et $\tau \in T$, ce qui est un prédicat Π_1^0 . Ainsi, l'ensemble des nœuds extensibles de T est l'ensemble Π_1^0 suivant.

$$\{\sigma \in T : \forall n > |\sigma| \exists \tau \in 2^n \text{ tel que } \tau \succeq \sigma \text{ et } \tau \in T\} \quad \blacksquare$$

Par passage au complémentaire, l'ensemble des nœuds non extensibles d'un arbre calculable est Σ_1^0 , ce qui fait que si un nœud est non extensible, on finira par s'en rendre compte au bout d'un temps fini. Nous allons voir comment utiliser la notion de nœud extensible pour créer des arbres calculables infinis n'ayant pas de chemin calculable. Dans la construction d'un arbre calculable, on doit pouvoir décider en un temps fini si un nœud y appartient ou non. Il est en revanche possible de reporter à plus tard la décision de rendre ou non un nœud extensible, en lui ajoutant par défaut des descendants au cours du temps, jusqu'à décider à un instant t de cesser de lui en rajouter pour le rendre non extensible. Cette technique que l'on appelle « l'astuce du temps » permet de montrer le résultat suivant.

Proposition 1.10. Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre Π_1^0 . Il existe un arbre calculable $S \subseteq 2^{<\mathbb{N}}$ tel que $[T] = [S]$. ★

PREUVE. Soit $(T_n)_{n \in \mathbb{N}}$ une approximation Π_1^0 de T , c'est-à-dire une suite uniformément calculable d'ensembles décroissante par la relation d'inclusion (avec $T_{n+1} \subseteq T_n$) telle que $\bigcap_n T_n = T$.

Soit $S = \{\sigma \in 2^{<\mathbb{N}} : \forall \tau \preceq \sigma \tau \in T_{|\sigma|}\}$. L'ensemble S est calculable. Montrons que S est clos par préfixe. Soit $\sigma \in S$, et soit $\rho \preceq \sigma$. Par définition de S , $\forall \tau \preceq \sigma \tau \in T_{|\sigma|}$. Comme $|\rho| \leq |\sigma|$, $T_{|\rho|} \supseteq T_{|\sigma|}$, donc $\forall \tau \preceq \sigma \tau \in T_{|\rho|}$. En particulier, pour tout $\tau \preceq \rho$, également $\tau \preceq \sigma$, donc $\tau \in T_{|\rho|}$. Ainsi, par définition de S , $\rho \in S$.

Montrons maintenant que $[S] = [T]$. Nous avons $P \in [S]$ ssi $\forall \sigma \prec P \sigma \in S$ ssi $\forall \sigma \prec P \forall \tau \preceq \sigma \tau \in T_{|\sigma|}$ ssi $\forall \tau \prec P \forall n \geq |\tau| \tau \in T_n$ ssi $\forall \sigma \prec P \sigma \in T$ ssi $P \in [T]$. ■

Ce chapitre porte principalement sur l'étude des classes d'ensembles correspondant aux chemins d'arbres calculables. Nous verrons avec la proposition 3.5 qu'il existe des arbres calculables infinis ne contenant aucun chemin calculable infini, puis nous déterminerons la puissance de calcul exacte qui est nécessaire au calcul d'un chemin dans n'importe quel arbre calculable infini. Cette étude constitue une des briques de base des mathématiques à rebours, que nous verrons dans la partie III.

2. Topologie sur l'espace de Cantor

La topologie est une branche des mathématiques qui formalise de manière abstraite les notions de limite et de continuité, et qui par extension étudie les propriétés d'objets géométriques invariantes par déformation continue. Le lecteur qui n'a jamais étudié cette branche peut se rassurer : nous n'avons besoin pour le développement des chapitres à venir que d'éléments très basiques de cette théorie, que nous présentons ici.

Notation

On notera i^∞ la suite infinie qui répète le bit $i \in \{0, 1\}$.

2.1. Ouverts et fermés

Comme nous l'avons déjà mentionné, l'espace de Cantor $2^{\mathbb{N}}$ est similaire à l'ensemble des réels de l'intervalle $[0, 1]$. Un élément $X \in 2^{\mathbb{N}}$ peut aussi être vu comme le développement binaire du réel $0.X(0)X(1)X(2)\dots$, avec toutefois une différence subtile : les éléments $\sigma 10^\infty$ et $\sigma 01^\infty$ sont deux éléments distincts de $2^{\mathbb{N}}$ mais correspondent au même réel. Cette différence mise à part, on peut voir $2^{\mathbb{N}}$ comme un intervalle, et les sous-ensembles les plus simples de $2^{\mathbb{N}}$ seront simplement les intervalles de la forme $[\sigma]$, que l'on appellera aussi *cylindre*.

Notation

Étant donné une chaîne $\sigma \in 2^{<\mathbb{N}}$, on écrit $[\sigma]$ pour l'ensemble

$$\{X \in 2^{\mathbb{N}} : X \succeq \sigma\}.$$

On appellera *cylindre* un ensemble de la forme $[\sigma]$.

Étant donné une chaîne $\sigma \in 2^{<\mathbb{N}}$, on peut voir $[\sigma]$ comme un intervalle de suites binaires infinies : celles qui sont lexicographiquement comprises entre $\sigma 0^\infty$ et $\sigma 1^\infty$. Avec cette vision en tête, les classes dites *ouvertes* du Cantor sont simplement les réunions quelconques d'intervalles.

Définition 2.1. Les classes *ouvertes* du Cantor $2^{\mathbb{N}}$ sont les réunions quelconques de cylindres, c'est-à-dire les ensembles de la forme $\bigcup_{\sigma \in W} [\sigma]$ pour un ensemble $W \subseteq 2^{<\mathbb{N}}$. Les classes *fermées* sont les complémentaires des classes ouvertes. $\diamond$

Le lecteur non habitué à ces concepts pourra, pour se faire la main, démontrer sur les définitions que les réunions finies de cylindres sont à la fois des ouverts et des fermés.

Notation

Étant donné un ensemble $W \subseteq 2^{<\mathbb{N}}$, on écrira $[W]$ pour dénoter son ouvert correspondant, c'est-à-dire la classe $\bigcup_{\sigma \in W} [\sigma]$.

Il est clair d'après la définition que les ouverts sont clos par réunion quelconque, et donc par passage au complémentaire que les fermés sont clos par intersection quelconque. Nous introduisons ci-après un élément de vocabulaire qui reviendra souvent dans la manipulation des ouverts et des fermés.

Notation

Étant donné une réunion dénombrable d'ensembles $\bigcup_n \mathcal{B}_n$, on dira que la réunion est *croissante* si $\mathcal{B}_n \subseteq \mathcal{B}_{n+1}$ pour tout n . De la même manière, on dira qu'une intersection $\bigcap_n \mathcal{B}_n$ est *décroissante* si $\mathcal{B}_{n+1} \subseteq \mathcal{B}_n$ pour tout n .

La représentation mentale d'un ensemble ouvert devrait être à peu près claire pour le lecteur : les réunions de briques simples que sont les cylindres. Voici un exemple illustratif.



FIGURE 2.2 – Illustration de l'ouvert $[01] \cup [101] \cup [1101] \cup [11101] \cup \dots$

Nous montrons à présent que l'on peut considérer sans perte de généralité que les intersections d'ouverts sont décroissantes et les réunions de fermés croissantes (en particulier, car $\bigcap_n \mathcal{U}_n = \bigcap_n (\bigcap_{m \leq n} \mathcal{U}_m)$) :

Proposition 2.3. Une intersection finie d'ouverts est un ouvert. Par passage au complémentaire, une réunion finie de fermés est fermée. ★

PREUVE. Soient $\mathcal{U}_0, \mathcal{U}_1 \subseteq 2^{\mathbb{N}}$ deux classes ouvertes. Un ensemble X appartient à $\mathcal{U}_0 \cap \mathcal{U}_1$ ssi il appartient à un cylindre $[\sigma_0] \subseteq \mathcal{U}_0$ ainsi qu'à un cylindre $[\sigma_1] \subseteq \mathcal{U}_1$. Donc, $\mathcal{U}_0 \cap \mathcal{U}_1 = \bigcup_{[\sigma_0] \subseteq \mathcal{U}_0, [\sigma_1] \subseteq \mathcal{U}_1} [\sigma_0] \cap [\sigma_1]$. ■

Les ensembles fermés s'avèrent plus délicats à décrire. Cela n'est pas forcément surprenant. En guise d'analogie, disons que l'on peut bien connaître son quartier, sans avoir une idée précise du reste du monde. La prise de conscience de cette complexité et la manière de l'appréhender figurent déjà dans les travaux de Cantor, par exemple au travers du fameux théorème de Cantor-Bendixson. Il existe toutefois une manière simple de représenter les fermés de $2^{\mathbb{N}}$: en tant que chemins infinis d'un arbre.

Proposition 2.4. Une classe $\mathcal{P} \subseteq 2^{\mathbb{N}}$ est fermée si, et seulement si, il existe un arbre $T \subseteq 2^{<\mathbb{N}}$ tel que $\mathcal{P} = [T]$. ★

PREUVE. Soit $\mathcal{P}$ une classe fermée, et soit $\mathcal{U} = \bigcup_{\sigma \in W} [\sigma]$ son complémentaire, avec $W \subseteq 2^{<\mathbb{N}}$. On définit l'arbre $T \subseteq 2^{<\mathbb{N}}$ comme étant l'ensemble des chaînes σ n'ayant aucun préfixe dans W . Par définition, T est clos par préfixe, et est donc un arbre. Montrons $[T] = \mathcal{P}$. On a $X \in [T]$ ssi aucun préfixe $\sigma \prec X$ n'est dans W ssi $X \notin \bigcup_{\sigma \in W} [\sigma]$ ssi $X \in \mathcal{P}$. Donc, $[T] = \mathcal{P}$.

Inversement, si $T \subseteq 2^{<\mathbb{N}}$ est un arbre, la classe $[T]$ de ses chemins est un fermé, car elle est le complémentaire de la classe $\bigcup_{\sigma \notin T} [\sigma]$. ■

À titre d'exemple, le lecteur peut consulter la figure 2.5, où se trouve l'arbre qui représente le complémentaire de l'ouvert décrit dans la figure 2.2.

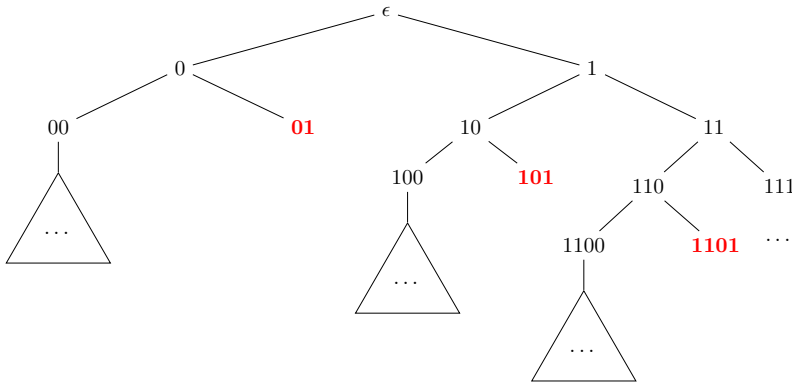


FIGURE 2.5 – Illustration de l'arbre représentant le complémentaire dans $2^{\mathbb{N}}$ de l'ouvert $[01] \cup [101] \cup [1101] \cup [11101] \cup \dots$

Les nœuds en gras correspondent aux cylindres constituant les briques de base de l'ouvert. Les triangles représentent un sous-arbre « plein » à partir du nœud où ils se trouvent.

2.2. Compacité

La *compacité* est une notion fondamentale de topologie. Elle est généralement définie via la propriété de Borel-Lebesgue, qui dans l'espace de Cantor se formule comme suit.

Définition 2.6. Une classe $\mathcal{P} \subseteq 2^{\mathbb{N}}$ a la *propriété de Borel-Lebesgue*^a si pour toute collection d'ouverts $(\mathcal{O}_n)_{n \in \mathbb{N}}$ telle que $\mathcal{P} \subseteq \bigcup_n \mathcal{O}_n$, il existe un ensemble fini $F \subseteq \mathbb{N}$ tel que $\mathcal{P} \subseteq \bigcup_{n \in F} \mathcal{O}_n$. On dira qu'une classe

possédant la propriété de Borel-Lebesgue est *compacte*. ◇

a. Appelée aussi « propriété de Heine-Borel ».

Nous montrons avec la proposition suivante que dans l'espace de Cantor, les classes compactes sont exactement les fermés, et le lecteur pourra constater en lisant la preuve que le lemme de König faible peut être vu comme une reformulation du fait que les classes fermées sont compactes.

Proposition 2.7. Une classe de $2^{\mathbb{N}}$ est fermée si, et seulement si, elle a la propriété de Borel-Lebesgue. ★

PREUVE. Soit $\mathcal{P} \subseteq 2^{\mathbb{N}}$ un fermé, et soit $(\mathcal{O}_n)_{n \in \mathbb{N}}$ une collection d'ouverts telle que $\mathcal{P} \subseteq \bigcup_n \mathcal{O}_n$, et soit $T \subseteq 2^{<\mathbb{N}}$ un arbre tel que $[T] = \mathcal{P}$. Soit $S \subseteq 2^{<\mathbb{N}}$ l'arbre des chaînes $\sigma \in T$ telles que $[\sigma] \not\subseteq \bigcup_{n < |\sigma|} \mathcal{O}_n$. Si l'arbre S est fini, il existe une longueur ℓ telle que pour tout $\sigma \in T$ tel que $|\sigma| = \ell$,

$$[\sigma] \subseteq \bigcup_{n < \ell} \mathcal{O}_n.$$

Il s'ensuit que $[T] \subseteq \bigcup_{\sigma \in T, |\sigma| = \ell} [\sigma] \subseteq \bigcup_{n < \ell} \mathcal{O}_n$. Si S est infini, par le lemme faible de König, $[S] \neq \emptyset$. Soit $P \in [S]$. Montrons que $P \notin \bigcup_n \mathcal{O}_n$ pour en déduire une contradiction, car $[S] \subseteq [T] \subseteq \bigcup_n \mathcal{O}_n$. Soit $i \in \mathbb{N}$. Par définition de $[S]$, pour tout ℓ , on a $P \upharpoonright_{\ell} \in S$, et donc par définition de S on a $[P \upharpoonright_{\ell}] \not\subseteq \bigcup_{n < \ell} \mathcal{O}_n$; en particulier, pour tout $\ell > i$, $[P \upharpoonright_{\ell}] \not\subseteq \mathcal{O}_i$. Comme $\mathcal{O}_i$ est un ouvert, il s'ensuit que $P \notin \mathcal{O}_i$.

Supposons à présent qu'une classe $\mathcal{B}$ admette la propriété de Borel-Lebesgue. Pour tout $X \notin \mathcal{B}$, soit $\mathcal{O}_X$ l'ouvert correspondant au complémentaire de la classe $\{X\}$ (il s'agit de la réunion des cylindres $[\sigma i]$ pour toute chaîne σ et tout i tel que $X(|\sigma|) \neq i$). En particulier, on a $\mathcal{B} = \bigcap_{X \notin \mathcal{B}} \mathcal{O}_X$. Notons que chaque $\mathcal{O}_X$ est une réunion de cylindres et que chaque cylindre est ouvert. Donc, par la propriété de Borel-Lebesgue, on peut trouver pour tout X un ensemble fini de cylindres F_X tel que $\mathcal{B} \subseteq \bigcup_{\sigma \in F_X} [\sigma] \subseteq \mathcal{O}_X$. En particulier, $\mathcal{B} = \bigcap_{X \notin \mathcal{B}} \bigcup_{\sigma \in F_X} [\sigma]$. Chaque réunion $\bigcup_{\sigma \in F_X} [\sigma]$ est une classe fermée en tant que réunion finie de cylindres. Comme une intersection arbitraire de fermés est une classe fermée, on en déduit que $\mathcal{B}$ est une classe fermée. ■

En pratique, nous utiliserons la conséquence suivante de la compacité : toute intersection dénombrable et décroissante de fermés non vides, est non vide, ce que nous démontrons ici.

Proposition 2.8. Soit $\mathcal{P}_0 \supseteq \mathcal{P}_1 \supseteq \dots$ une suite décroissante de fermés non vides. Alors $\bigcap_n \mathcal{P}_n$ est non vide. ★

PREUVE. Soit T_n un arbre tel que $[T_n] = \mathcal{P}_n$. On peut supposer, sans perte de généralité, que $T_{n+1} \subseteq T_n$. Montrons que $\bigcap_n [T_n] = [\bigcap_n T_n]$. On a $X \in \bigcap_n [T_n]$ ssi $X \upharpoonright_m \in T_n$ pour tout m, n , ou encore ssi $X \in [\bigcap_n T_n]$. Donc, $\bigcap_n [T_n] = [\bigcap_n T_n]$. Soit $T = \bigcap_n T_n$. En particulier, $[T] = \bigcap_n \mathcal{P}_n$.

Supposons par l'absurde que $[T] = \bigcap_n \mathcal{P}_n$ est vide. D'après le lemme de König, il y a donc un entier a tel qu'aucune chaîne σ de taille supérieure ou égale à a n'est dans T . Comme les chaînes de taille a sont en quantité finie et que la suite $(T_n)_{n \in \mathbb{N}}$ est décroissante par l'inclusion, il doit donc y avoir entier m tel qu'aucune de ces chaînes n'appartient à T_m . Ainsi, $[T_m]$ est vide, ce qui contredit les hypothèses. ■

2.3. Continuité

Abordons une autre notion topologique que nous mentionnerons ça et là dans les chapitres à venir. La continuité, autre notion centrale de topologie, se cache de manière inattendue en calculabilité sous l'idée suivante.

Une fonctionnelle Turing Φ peut être vue comme une fonction partielle de $2^{\mathbb{N}}$ dans $2^{\mathbb{N}}$, dont l'entrée est un oracle X , et le résultat, que nous notons Φ^X ou $\Phi(X)$, est l'ensemble Y tel que $\Phi^X(n) \downarrow = Y(n)$. Cette fonction de $2^{\mathbb{N}}$ vers $2^{\mathbb{N}}$ n'est bien entendu définie que pour les oracles X tels que $\forall n \ \Phi^X(n) \downarrow \in \{0, 1\}$.

Nous avons vu que quand $\Phi^X(n) \downarrow = v$, par la propriété de l'usage (voir la définition 4-4.2), seul un segment initial fini σ de l'oracle X est utilisé. Plus généralement, si $\Phi^X \succeq \tau$ (ce qui signifie $\forall n < |\tau| \ \Phi^X(n) \downarrow = \tau(n)$), seule une partie finie σ de l'oracle est utilisée pour s'en rendre compte. Il s'ensuit que pour tout $X \in [\sigma]$, $\Phi^X \succeq \tau$, et donc $\{\Phi^X : X \in [\sigma]\} \subseteq [\tau]$.

Le lecteur ayant suivi un cours d'introduction à la topologie reconnaîtra dans cette idée la notion de continuité : pour tout ouvert de l'espace d'arrivée aussi « petit » que l'on veut — en pratique un cylindre $[\tau]$, il existe un ouvert de l'espace de départ suffisamment « petit » — en pratique un cylindre $[\sigma]$ — tel que tout $X \in [\sigma]$ est envoyé à l'intérieur de $[\tau]$: concrètement, la chaîne σ est « envoyée » vers la chaîne τ .

Définition 2.9. Une fonction (éventuellement partielle) $f : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ est *continue en* $X \in \text{dom } f$ si pour tout cylindre $[\tau]$ contenant $f(X)$, il existe un cylindre $[\sigma]$ contenant X tel que $f([\sigma]) \subseteq [\tau]$. On dira qu'une fonction est *continue* si elle est continue en X pour tout $X \in \text{dom } f$. ◇

On considérera en général des fonctions continues sur tout leur domaine de définition, qui admettent alors une caractérisation équivalente en termes de pré-image d'ouverts.

Proposition 2.10. Une fonction (éventuellement partielle) $f : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ est continue si, et seulement si, il existe pour tout ouvert $\mathcal{U} \subseteq 2^{\mathbb{N}}$ un ouvert $\mathcal{V} \subseteq 2^{\mathbb{N}}$ tel que $f^{-1}(\mathcal{U}) = \mathcal{V} \cap \text{dom } f$. Si la fonction est totale, on a alors $f^{-1}(\mathcal{U})$ ouvert pour tout ouvert $\mathcal{U}$. ★

PREUVE. Soient $f : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ une fonction continue et $\mathcal{U} \subseteq 2^{\mathbb{N}}$ un ouvert. Comme $\mathcal{U}$ est ouvert, pour tout $X \in f^{-1}(\mathcal{U})$ il existe un cylindre $[\tau_X]$ contenant $f(X)$ tel que $[\tau_X] \subseteq \mathcal{U}$. Par continuité de f , pour tout X , il existe un cylindre $[\sigma_X]$ contenant X tel que $f([\sigma_X]) \subseteq [\tau_X]$. Alors,

$$\text{dom } f \cap \bigcup_{X \in f^{-1}(\mathcal{U})} [\sigma_X] = f^{-1}(\mathcal{U}).$$

Réciproquement, supposons que pour tout ouvert $\mathcal{U} \subseteq 2^{\mathbb{N}}$, il existe un ouvert $\mathcal{V}$ tel que $\text{dom } f \cap \mathcal{V} = f^{-1}(\mathcal{U})$. Soit $Y \in \text{Im } f$, et soit $[\tau]$ un cylindre contenant Y . Soit un ouvert $\mathcal{V}$ tel que $\text{dom } f \cap \mathcal{V} = f^{-1}([\tau])$. Comme $\mathcal{V}$ est un ouvert, il existe $W \subseteq 2^{<\mathbb{N}}$ tel que $\bigcup_{\sigma \in W} [\sigma] = \mathcal{V}$. Notons que $W \neq \emptyset$, car $Y \in \text{Im } f \cap [\tau]$, donc il existe un cylindre $\sigma \in W$. On a

$$f([\sigma]) = f(\text{dom } f \cap [\sigma]) \subseteq [\tau]. \quad \blacksquare$$

Une fonctionnelle calculable Φ est donc toujours aussi une fonction continue sur son domaine de définition, c'est-à-dire sur l'espace des X tels que $\Phi(X, n) \downarrow \in \{0, 1\}$ pour tout n . En revanche, une fonction continue n'a *a priori* aucune raison d'être calculable : il se peut que n'importe quel morceau fini de la sortie de la fonction puisse être déterminé par un morceau fini de l'entrée, mais que ce « déterminisme » ne soit pas calculable. À titre d'exemple, considérons la fonction Φ qui sur n'importe quel ensemble X associe $X \oplus \emptyset'$. Une telle fonction Φ est continue, mais pas calculable. Elle est en revanche calculable avec l'aide de $\emptyset'$.

En calculabilité, la fonction non continue par excellence est celle qui à X associe X' : en effet, pour savoir si $n \in X'$, il faut savoir si $\Phi_n(X, n) \downarrow$, et pour cela potentiellement connaître une infinité de bits de X (en particulier, si $\Phi_n(X, n) \uparrow$). Nous verrons des versions effectives de certains théorèmes bien connus d'analyse, qui stipulent que toute fonction non continue, mais pas trop complexe, par exemple $X \mapsto X'$, est malgré tout continue sur un « large » ensemble de points, en particulier sur une classe co-maigre (voir le théorème 10-3.20) et sur une classe de mesure arbitrairement grande (voir le théorème 19-3.8).

2.4. Classes parfaites

Une dernière notion topologique que nous utiliserons est celle de classes parfaites. Ce sont les classes qui sont l'image d'une injection continue

de $2^{\mathbb{N}}$ vers $2^{\mathbb{N}}$, c'est-à-dire exactement les classes de la forme $[T]$ pour un f -arbre $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ (voir la section 7-5). Ces classes-là sont donc toujours fermées et peuvent se représenter par un arbre. Par extension, on parlera donc aussi d'arbre parfait.

Définition 2.11. Un arbre non vide $T \subseteq 2^{<\mathbb{N}}$ est parfait si tout nœud de T a deux extensions incompatibles dans T . $\diamond$

Nous avons vu avec l'exercice 1.7 qu'étant donné un fermé $\mathcal{F}$ représenté par un arbre T , on peut considérer sans perte de généralité — si l'on ne s'occupe pas de l'effectivité — que T ne contient que des nœuds extensibles. En revanche, un nœud extensible n'a pas nécessairement deux extensions incompatibles dans le cas général. Quand cela arrive, cela signifie qu'il y a exactement un chemin infini passant par ce nœud. On appelle de tels chemins des *points isolés*. Pour une classe $\mathcal{A}$ arbitraire, la définition correspondante est la suivante.

Définition 2.12. Soit $\mathcal{A} \subseteq 2^{\mathbb{N}}$. Un élément $X \in \mathcal{A}$ est un *point isolé* s'il existe un préfixe $\sigma \prec X$ tel que $[\sigma] \cap \mathcal{A} = \{X\}$. $\diamond$

La définition usuelle de classe parfaite découle alors de celle de point isolé.

Définition 2.13. Une classe non vide $\mathcal{F} \subseteq 2^{\mathbb{N}}$ est *parfaite* si elle est fermée et n'a pas de point isolé. De manière équivalente, $\mathcal{F} = [T]$ pour un arbre parfait $T \subseteq 2^{<\mathbb{N}}$. $\diamond$

Les classes parfaites sont d'une grande importance, notamment car elles permettent de construire des arguments de cardinalité : toute classe parfaite est par définition en bijection continue avec $2^{\mathbb{N}}$, et a donc la même cardinalité que $2^{\mathbb{N}}$. Par ailleurs, si une classe $\mathcal{A} \subseteq 2^{\mathbb{N}}$ contient une classe parfaite, alors on a une injection de $2^{\mathbb{N}}$ dans $\mathcal{A}$. L'injection identité de $\mathcal{A}$ dans $2^{\mathbb{N}}$ nous donne alors $|\mathcal{A}| = |2^{\mathbb{N}}|$. Il s'agit en fait grosso modo *de la seule manière* de montrer qu'une classe $\mathcal{A} \subseteq 2^{\mathbb{N}}$ a la puissance du continu. Nous en reparlerons dans la section 9-4, ainsi que dans la section 30-4.

Voici pour terminer une application simple de la notion de classe parfaite à l'étude de la cardinalité.

Proposition 2.14. Toute classe fermée dénombrable et non vide $\mathcal{F}$ contient des points isolés. On peut injecter $2^{\mathbb{N}}$ dans toute classe fermée non vide ne contenant pas de point isolé. $\star$

PREUVE. Supposons que $\mathcal{F}$ ne contienne pas de point isolé. Soit $\mathcal{F} = [T]$ pour un arbre T n'ayant que des nœuds extensibles. Comme $\mathcal{F}$ ne contient pas de point isolé, tous les nœuds de T ont alors deux extensions incompatibles. On a alors une injection de $2^{\mathbb{N}}$ dans $\mathcal{F}$. Si une classe fermée $\mathcal{F}$

est dénombrable, on ne peut injecter $2^{\mathbb{N}}$ dans $\mathcal{F}$, et elle contient donc par contraposée des points isolés. ■

Corollaire 2.15 (Cantor)

L'hypothèse du continu est vraie pour les classes fermées de $2^{\mathbb{N}}$: elles sont soit finies, soit dénombrables, soit de cardinalité $|2^{\mathbb{N}}|$.

PREUVE. Soit $\mathcal{F}$ un fermé, et soit $W \subseteq 2^{<\mathbb{N}}$ l'ensemble des chaînes σ telles que $\mathcal{F} \cap [\sigma]$ est fini ou dénombrable. Soit $\mathcal{F}' = \mathcal{F} \setminus \bigcup_{\sigma \in W} [\sigma]$. Notons que $\mathcal{F}' \subseteq \mathcal{F}$ est toujours une classe fermée. Si $\mathcal{F}'$ est vide, alors d'après le lemme de König (ou la compacité) il n'est besoin d'enlever à $\mathcal{F}$ qu'un nombre fini de chaînes $\sigma_0, \dots, \sigma_n \in W$ pour que $\mathcal{F}' = \mathcal{F} \setminus ([\sigma_0] \cup \dots \cup [\sigma_n])$ soit vide. Comme chaque classe $\mathcal{F} \cap [\sigma_i]$ est dénombrable, on a vidé $\mathcal{F}$ en lui enlevant une quantité dénombrable de points. Donc, $\mathcal{F}$ est dénombrable. Sinon, $\mathcal{F}'$ n'est pas vide, et en plus de cela il ne peut contenir de points isolés (car si $\mathcal{F}' \cap [\sigma]$ ne contient qu'un seul élément, alors $\mathcal{F} \cap [\sigma]$ est dénombrable). Par la proposition 2.14, on a une injection de $2^{\mathbb{N}}$ vers $\mathcal{F}'$. ■

Ce résultat sera étendu avec le corollaire 30-3.3.

3. Classes Π_1^0

Nous nous intéressons à présent à la version *effective* des classes ouvertes et fermées. On emploie souvent le terme *effectif* plutôt que calculable pour ce genre d'objets, car ils ne sont pas nécessairement calculables dans le sens où l'on peut précisément en connaître les moindres détails, mais ils admettent malgré tout une certaine description tangible, effective, fournie par un algorithme.

Définition 3.1. Une classe $\mathcal{U} \subseteq 2^{\mathbb{N}}$ est dite Σ_1^0 s'il existe un ensemble c. e. $W \subseteq 2^{<\mathbb{N}}$ tel que $\mathcal{U} = \bigcup_{\sigma \in W} [\sigma]$. Une classe $\mathcal{P} \subseteq 2^{\mathbb{N}}$ est dite Π_1^0 si son complémentaire est une classe Σ_1^0 . ◇

Les Σ_1^0 et Π_1^0 sont respectivement les ouverts et fermés *effectifs* de l'espace de Cantor. On appellera *code* d'une classe Σ_1^0 $\mathcal{U}$ un entier e tel que $\mathcal{U} = \bigcup_{\sigma \in W_e} [\sigma]$. De même, le *code* d'une classe Π_1^0 $\mathcal{P}$ est le code de la classe Σ_1^0 $\mathcal{U} = 2^{\mathbb{N}} \setminus \mathcal{P}$. Cela nous permet de parler de calculabilité uniforme sur les suites de classes Σ_1^0 ou Π_1^0 en considérant la calculabilité de leur suite de codes. Les classes Π_1^0 sont un exemple important et très étudié en calculabilité.

Remarque

Il convient de bien distinguer les ensembles d'entiers Σ_1^0 et Π_1^0 , qui sont les premiers niveaux de la hiérarchie arithmétique (voir le chapitre 5) des classes Σ_1^0 et Π_1^0 qui sont les ouverts et fermés effectifs respectifs de l'espace de Cantor. Il existe cependant des liens entre ces notions.

Commençons par le fait que la proposition 2.3, qui stipule qu'une intersection finie d'ouverts est un ouvert, fonctionne également avec les classes Σ_1^0 .

Proposition 3.2. Les classes Σ_1^0 sont closes par intersection finie. Par passage au complémentaire, les classes Π_1^0 sont closes par réunion finie. ★

PREUVE. Soient $\mathcal{U}_0 = \bigcup_{\sigma \in W_0} [\sigma]$ et $\mathcal{U}_1 = \bigcup_{\sigma \in W_1} [\sigma]$ deux classes Σ_1^0 . Alors, l'ouvert $\mathcal{U}_0 \cap \mathcal{U}_1$ est décrit par l'ensemble c.e. qui énumère la plus longue chaîne parmi σ_0, σ_1 pour tous $\sigma_0 \in W_0$ et $\sigma_1 \in W_1$ telles que $\sigma_0 \preceq \sigma_1$ ou telles que $\sigma_1 \preceq \sigma_0$. La chaîne ainsi énumérée correspond à $[\sigma_0] \cap [\sigma_1]$. ■

La première étape pour mieux appréhender la nature des classes Π_1^0 est sans doute de prouver la version effective du théorème 2.4 : les Π_1^0 sont exactement les chemins infinis d'arbres calculables.

Proposition 3.3. Une classe $\mathcal{P}$ est Π_1^0 si, et seulement si, il existe un arbre calculable $T \subseteq 2^{<\mathbb{N}}$ tel que $[T] = \mathcal{P}$. ★

PREUVE. Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre calculable. La classe $[T] = \{X : \forall n X \upharpoonright_n \in T\}$ est alors bien Π_1^0 . En effet, son complémentaire est la classe Σ_1^0 décrite par la réunion des cylindres $[\sigma]$ tels que $\sigma \notin T$.

Supposons que $\mathcal{P}$ est Π_1^0 , et soit $\mathcal{U} = \bigcup_{\sigma \in W} [\sigma]$ son complémentaire. On calcule l'arbre $T \subseteq 2^{<\mathbb{N}}$ suivant : à l'étape de calcul t , pour toute chaîne $\sigma \in 2^{<\mathbb{N}}$ de taille t , on décide $\sigma \in T$ ssi pour tout préfixe $\tau \preceq \sigma$ on a $\tau \notin W[t]$.

Il est clair que T est clos par préfixe : si σ de taille t est dans T , alors aucun préfixe de σ n'est dans W à l'étape de calcul t , donc pour $s \leq t$, aussi aucun préfixe de $\sigma \upharpoonright_s$ n'est dans W à l'étape de calcul s . À présent, si $X \in \mathcal{P}$, alors aucun préfixe σ de X n'est dans W , et chacun de ces préfixes sera donc dans T . À l'inverse, si $X \notin \mathcal{P}$, alors un préfixe σ de X rentre dans W à une certaine étape t . Par construction, aucune chaîne $\tau \succeq \sigma$ de taille supérieure à t ne sera dans T . Donc, $\mathcal{P} = [T]$. ■

Le code d'un arbre calculable $T \subseteq 2^{<\mathbb{N}}$ est un entier e tel que $\Phi_e = T$. Notons que la preuve de la proposition 3.3 est uniforme, et permet de passer calculatoirement d'un code d'une classe Π_1^0 à un code de l'arbre correspondant, et inversement. On pourra donc considérer indistinctement le code des classes Π_1^0 et des arbres calculables dans les preuves à venir. La

proposition suivante établit un lien avec les classes Σ_1^0 et Π_1^0 et la hiérarchie arithmétique.

Proposition 3.4. Soit $\mathcal{P} \subseteq 2^{\mathbb{N}}$ une classe.

- (1) La classe $\mathcal{P}$ est Σ_1^0 si, et seulement si, $\mathcal{P} = \{X \in 2^{\mathbb{N}} : \exists n \, R(X \upharpoonright_n)\}$ pour un prédicat calculable $R \subseteq 2^{<\mathbb{N}}$.
- (2) La classe $\mathcal{P}$ est Π_1^0 si, et seulement si, $\mathcal{P} = \{X \in 2^{\mathbb{N}} : \forall n \, R(X \upharpoonright_n)\}$ pour un prédicat calculable $R \subseteq 2^{<\mathbb{N}}$. ★

PREUVE. Pour (2), étant donné $\mathcal{P}$ une classe Π_1^0 , il suffit de considérer l'arbre calculable T tel que $[T] = \mathcal{P}$. Le prédicat calculable est simplement T . À l'inverse, si $\mathcal{P} = \{X \in 2^{\mathbb{N}} : \forall n \, R(X \upharpoonright_n)\}$, alors l'arbre calculable donné par $\sigma \in T$ ssi $\forall \tau \preceq \sigma \, R(\tau)$ est tel que $[T] = \mathcal{P}$.

On obtient (1) par passage au complémentaire. ■

Voyons à présent quelques généralités, en premier lieu la preuve que le lemme de König ne relève pas des mathématiques calculables : certaines classes Π_1^0 non vides — et donc certains arbres calculables infinis — ne contiennent aucun point calculable. Nous verrons tout au long des chapitres à venir de nombreux exemples de classes Π_1^0 ne contenant aucun point calculable. Nous anticipons en particulier pour la proposition suivante sur l'exemple le plus simple à définir : la classe des ensembles DNC_2 de la proposition 6.4.

Proposition 3.5. Il existe des classes Π_1^0 non vides ne contenant aucun ensemble calculable. ★

PREUVE. On définit la classe

$$\mathcal{P} = \{X \in 2^{\mathbb{N}} : \forall e \, \forall t \, \Phi_e(e)[t] \uparrow \vee \Phi_e(e)[t] \downarrow \neq X(e)\}.$$

La classe $\mathcal{P}$ contient tous les ensembles X tels que $X(n)$ peut prendre n'importe quelle valeur si $\Phi_n(n) \uparrow$, et qui sont toujours différents de $\Phi_n(n)$ si $\Phi_n(n) \downarrow$. Il est clair que cette classe est non vide (l'arrêt des programmes informatique calcule par exemple facilement un élément de $\mathcal{P}$). Par la proposition 3.4, $\mathcal{P}$ est une classe Π_1^0 . De plus, la classe $\mathcal{P}$ ne contient aucun ensemble calculable : si X est calculable, alors il doit exister e tel que $\Phi_e(e) \downarrow = X(e)$. ■

Continuons à présent sur un autre aspect clef : les exemples de classes Π_1^0 non vides ne contenant aucun point calculable sont nécessairement indénombrables. Ils ne peuvent pas contenir de *points isolés*, c'est-à-dire des ensembles X tels que pour un certain n , aucun autre ensemble que X et étendant $X \upharpoonright_n$ n'appartient au Π_1^0 .

Proposition 3.6. Si $\mathcal{P}$ est une classe Π_1^0 contenant exactement un élément X , alors, X est calculable. ★

PREUVE. Soit $T \subseteq 2^{<\mathbb{N}}$ l'arbre calculable tel que $[T] = \mathcal{P}$. Soit l'algorithme suivant : on cherche le plus petit t tel que soit pour toute chaîne $\sigma \succeq 0$ de taille t on a $\sigma \notin T$, soit pour toute chaîne $\sigma \succeq 1$ de taille t on a $\sigma \notin T$. Notez qu'exactement un des deux événements doit nécessairement arriver : si les deux événements arrivent, la classe est vide. Si aucun des deux n'arrive, il y a une infinité de chaînes dans T qui étendent 0 et aussi une infinité qui étendent 1. D'après le lemme de König, T contient donc au moins deux chemins infinis : un qui étend 0 et un qui étend 1, ce qui contredit les hypothèses sur $\mathcal{P}$.

Une fois un des deux événements arrivés, on sait donc si X commence par 0 ou 1. On voit aisément comment continuer par récurrence : une fois $X \upharpoonright_n$ calculé, on cherche le plus petit t tel que pour toute chaîne $\sigma \succeq X \upharpoonright_n 0$ de taille t on a $\sigma \notin T$, ou pour toute chaîne $\sigma \succeq X \upharpoonright_n 1$ de taille t on a $\sigma \notin T$. Une fois l'un des deux événements arrivé, la valeur de $X(n)$ est connue. ■

Corollaire 3.7

Les points isolés de toute classe Π_1^0 sont calculables.

PREUVE. Soit $\mathcal{P}$ une classe Π_1^0 , et soit $X \in \mathcal{P}$ un point isolé. Par définition, on a un préfixe $\sigma \prec X$ tel que $[\sigma] \cap \mathcal{P} = \{X\}$. En particulier, $[\sigma] \cap \mathcal{P}$ est une classe Π_1^0 contenant exactement un élément, et cet élément est donc calculable. ■

Corollaire 3.8

Toute classe Π_1^0 dénombrable contient un ensemble calculable.

PREUVE. Par la proposition 2.14 et le corollaire 3.7. ■

4. Théorèmes de base

Les membres d'une classe Π_1^0 peuvent être de degrés Turing très différents. Par exemple, l'espace de Cantor $2^{\mathbb{N}}$ est une classe Π_1^0 contenant des ensembles de chaque degré Turing.

Étant donné une classe Π_1^0 non vide, on s'intéresse principalement au degré de difficulté du calcul de l'un de ses membres.

Définition 4.1. Une *base* pour les classes Π_1^0 est une classe d'ensembles $\mathcal{C}$ telle que toute classe Π_1^0 non vide contient un élément de $\mathcal{C}$. $\diamond$

Dans cette section, nous allons prouver un certain nombre de théorèmes qui étant donné une propriété de faiblesse P sont de la forme « Toute classe Π_1^0 non vide contient un membre satisfaisant P . » Ces théorèmes sont appelés « théorèmes de bases », car ils établissent que les membres de P forment une base pour les classes Π_1^0 . Inversement, les « théorèmes d'anti-bases » énoncent l'existence d'une classe Π_1^0 non vide ne contenant aucun membre satisfaisant une propriété de faiblesse. Le tout premier théorème de base est dû à Kreisel [124], et est laissé en exercice.

Exercice 4.2. (★) Montrer que toute classe Π_1^0 non vide contient un élément $\emptyset'$ -calculable. $\diamond$

On peut faire encore mieux que l'exercice 4.2 via le théorème central dit de « base low », qui stipule que toute classe Π_1^0 non vide contient un ensemble low. Ce théorème a une importance fondamentale en calculabilité et en mathématique à rebours, notamment pour fournir nombre d'exemples et contre-exemples.

Théorème 4.3 (Jockusch et Soare [104])

Toute classe Π_1^0 non vide contient un ensemble low.

PREUVE. Soit $\mathcal{P}$ une classe Π_1^0 non vide. Nous allons utiliser $\emptyset'$ pour calculer un élément $Z \in \mathcal{P}$, tout en calculant son saut Turing Z' . Pour cela, définissons une suite décroissante uniformément $\emptyset'$ -calculable de classes Π_1^0 non vides $\mathcal{P} = \mathcal{P}_0 \supseteq \mathcal{P}_1 \supseteq \mathcal{P}_2 \supseteq \dots$ comme suit. Soit $\mathcal{P}_0 = \mathcal{P}$; supposons $\mathcal{P}_n$ défini, et considérons la classe

$$\mathcal{B}_n = \{X \in 2^{\mathbb{N}} : \forall t \Phi_n(X, n)[t] \uparrow\} \cap \mathcal{P}_n.$$

Notons que $\mathcal{B}_n$ est aussi une classe Π_1^0 , et que le code d'un arbre calculable T_n tel que $\mathcal{B}_n = [T_n]$ est calculable uniformément en n .

On pose à $\emptyset'$ la question de savoir si $\mathcal{B}_n$ est vide : d'après le lemme de König, c'est le cas ssi il existe m tel qu'aucune chaîne σ de taille m n'appartient à T_n , ce qui est bien un événement Σ_1^0 . Si $\emptyset'$ répond positivement, on note $Y(n) = 1$ et l'on définit $\mathcal{P}_{n+1} = \mathcal{P}_n$. Notons que, dans ce cas, tous les éléments $X \in \mathcal{P}_{n+1}$ sont tels que $\Phi_n(X, n) \downarrow$. Dans le cas inverse, on note $Y(n) = 0$ et l'on définit $\mathcal{P}_{n+1} = \mathcal{B}_n$. Notons que, dans ce cas-ci, tous les éléments de $X \in \mathcal{P}_{n+1}$ sont tels que $\Phi_n(X, n) \uparrow$.

Pour chaque n , $\mathcal{P}_n$ est un fermé non vide, et $\bigcap_n \mathcal{P}_n$ est donc non vide. Par construction, l'élément Y calculé par $\emptyset'$ correspond au saut Turing de n'importe quel élément de $\bigcap_n \mathcal{P}_n$. ■

Nous avons déjà vu avec la proposition 4-9.1 l'existence d'ensembles low et non calculable. Nous en avons à présent une preuve alternative en combinant le théorème 4.3 et la proposition 3.5.

Nous nous attelons à présent au deuxième grand théorème de base pour les classes Π_1^0 : les ensembles calculatoirement dominés. Nous avons besoin pour cela d'un lemme, qui a aussi son intérêt propre.

Lemme 4.4. Soit $\mathcal{P}$ une classe Π_1^0 non vide. Supposons qu'une fonctionnelle Φ est totale sur tous les chemins de $\mathcal{P}$. Alors, on peut définir uniformément en un code de la classe $\mathcal{P}$ une fonction calculable g qui domine $n \mapsto \Phi(X, n)$ pour tout $X \in \mathcal{P}$. ★

PREUVE. Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre calculable tel que $[T] = \mathcal{P}$. Montrons que pour tout entier n il existe t tel que $\Phi(\sigma, n)[|\sigma|] \downarrow$ pour toute chaîne $\sigma \in T$ de taille t . En effet, dans le cas inverse, il existe un entier n tel que l'ensemble $\{\sigma \in T : \Phi(\sigma, n)[|\sigma|] \uparrow\}$ contient pour tout t une chaîne de taille t et est donc un sous-arbre infini de T , qui contient donc par le lemme de König un chemin infini X . On a donc $\forall t \Phi(X, n)[t] \uparrow$, ce qui contredit la totalité de Φ sur tous les oracles de $[T]$.

On peut donc calculer la fonction g qui pour n cherche le plus petit t tel que $\Phi(\sigma, n)[t] \downarrow = v_\sigma$ pour toute chaîne $\sigma \in T$ de taille t . Une fois t trouvé, on définit $g(n) = \sum_{|\sigma|=t} v_\sigma + 1$. Il est clair que g domine toutes les fonctions calculables via Φ par un oracle de $[T]$. ■

Le théorème suivant est connu sous le nom de « théorème de base calculatoirement dominée ». Avec l'existence d'une classe Π_1^0 non vide ne possédant pas de membre calculable, ce théorème nous fournit une preuve alternative de l'existence d'ensembles calculatoirement dominés non calculables.

Théorème 4.5 (Jockusch et Soare [104])

Toute classe Π_1^0 non vide contient un ensemble calculatoirement dominé.

PREUVE. Soit $\mathcal{P}$ une classe Π_1^0 non vide. Nous allons définir une suite infinie décroissante de classes Π_1^0 non vides $\mathcal{P} = \mathcal{P}_0 \supseteq \mathcal{P}_1 \supseteq \mathcal{P}_2 \supseteq \dots$ de telle sorte que $\bigcap_n \mathcal{P}_n$ ne contienne que des ensembles calculatoirement dominés. Soit $\mathcal{P}_0 = \mathcal{P}$. Supposons $\mathcal{P}_n$ défini. Soit $\mathcal{B}_{n,m} = \{X : \Phi_n(X, m) \uparrow\}$. Notons que chaque classe $\mathcal{B}_{n,m}$ est Π_1^0 . Supposons qu'il existe un entier m tel que $\mathcal{P}_n \cap \mathcal{B}_{n,m} \neq \emptyset$. Alors, on définit $\mathcal{P}_{n+1} = \mathcal{P}_n \cap \mathcal{B}_{n,m}$. Notons que pour tout $X \in \mathcal{P}_{n+1}$ on a $\Phi_n(X, m) \uparrow$. Supposons à présent que pour tout m on a $\mathcal{P}_n \cap \mathcal{B}_{n,m} = \emptyset$. Cela implique que la fonctionnelle Φ_n est totale pour tout $X \in \mathcal{P}_n$. On définit alors $\mathcal{P}_{n+1} = \mathcal{P}_n$. D'après le lemme 4.4, il existe

donc une fonction calculable $g : \mathbb{N} \rightarrow \mathbb{N}$ qui domine $m \mapsto \Phi_n(X, m)$ pour tout $X \in \mathcal{P}_{n+1}$.

En tant qu'intersection décroissante de fermés non vides, la classe $\bigcap_n \mathcal{P}_n$ est non vide. Soit $X \in \bigcap_n \mathcal{P}_n$. Par construction, pour tout n , si Φ_n est totale sur l'oracle X , alors $m \mapsto \Phi_n(X, m)$ est bornée par une fonction calculable. Donc, X est calculatoirement dominé. ■

Nous voyons à présent un dernier théorème de base dit « d'évitement de cône » : étant donné un ensemble X , on appelle *cône supérieur* de X la classe $\mathcal{C}_X = \{Y \in 2^{\mathbb{N}} : Y \geq_T X\}$. Jockusch et Soare [104] ont prouvé que, pour chaque ensemble non calculable X , la classe $2^{\mathbb{N}} \setminus \mathcal{C}_X$ est une base pour les classes Π_1^0 . En d'autres termes, si X est un ensemble non calculable, toute classe Π_1^0 non vide possède un élément qui ne calcule pas X . La contraposée, plus naturelle, énonce que si un ensemble est calculable par tous les membres d'une classe Π_1^0 non vide, il est nécessairement calculable. Notons que si une classe Π_1^0 a un membre calculable, le résultat est évident, et il ne devient intéressant que pour les classes Π_1^0 non vides qui n'en ont pas. Tout comme pour le théorème de base calculatoirement dominée, nous avons besoin d'un lemme pour régler le cas d'une fonctionnelle fixée.

Lemme 4.6. Soient X un ensemble, $\mathcal{P}$ une classe Π_1^0 non vide et Φ une fonctionnelle. Si $\Phi^Y = X$ pour tout $Y \in \mathcal{P}$, alors X est calculable. ★

PREUVE. Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre calculable tel que $[T] = \mathcal{P}$. Supposons que pour tout $Y \in [T]$, $\Phi^Y = X$. Montrons que pour tout n , il existe un $t \in \mathbb{N}$ tel que $\Phi(\sigma, n)[|\sigma|] \downarrow = X(n)$ pour toute chaîne $\sigma \in T$ de taille t . En effet, dans le cas contraire, l'ensemble $S = \{\sigma \in T : \Phi(\sigma, n)[|\sigma|] \neq X(n)\}$ est un sous-arbre de T qui contient des éléments de chaque longueur, et il existe donc par le lemme de König un chemin $Y \in [S] \subseteq [T]$ tel que $\Phi^Y(n) \neq X(n)$, contredisant ainsi notre hypothèse.

Soit $g : \mathbb{N} \rightarrow \mathbb{N}$ la fonction calculable qui sur l'entrée n cherche $t, v_n \in \mathbb{N}$ tels que $\Phi(\sigma, n)[|\sigma|] \downarrow = v_n$ pour toute chaîne $\sigma \in T$ de taille t , et renvoie v_n . Nous avons montré que cette fonction était totale. De plus, on a nécessairement $v_n = X(n)$ pour tout n , car sinon chaque élément de $\mathcal{P}$ calcule autre chose que X sur le bit n . Ainsi, X est calculé par g , et est donc calculable. ■

Théorème 4.7 (Jockusch et Soare [104])

Soient X un ensemble non calculable et $\mathcal{P}$ une classe Π_1^0 non vide. Alors, il existe un élément de $\mathcal{P}$ qui ne calcule pas X .

PREUVE. Nous allons définir une suite infinie décroissante de classes Π_1^0 non vides $\mathcal{P} = \mathcal{P}_0 \supseteq \mathcal{P}_1 \supseteq \mathcal{P}_2 \supseteq \dots$ de telle sorte qu'aucun élément de $\bigcap_n \mathcal{P}_n$ ne calcule X . Soit $\mathcal{P}_0 = \mathcal{P}$. Supposons $\mathcal{P}_n$ défini. Soit

$$\mathcal{B}_{n,m} = \{Y : \Phi_n(Y, m) \uparrow \vee \Phi_n(Y, m) \neq X(m)\}.$$

Notons que chaque classe $\mathcal{B}_{n,m}$ est Π_1^0 (pas uniformément, bien sûr, car on ne connaît pas X). Montrons qu'il existe m tel que $\mathcal{P}_n \cap \mathcal{B}_{n,m} \neq \emptyset$. Si ce n'était pas le cas, on aurait alors $\Phi_n^Y = X$ pour tout $Y \in \mathcal{P}_n$, contredisant dès lors le lemme 4.6. Il existe donc m tel que $\mathcal{P}_n \cap \mathcal{B}_{n,m} \neq \emptyset$, et l'on définit alors $\mathcal{P}_{n+1} = \mathcal{P}_n \cap \mathcal{B}_{n,m}$ pour un tel entier m , ce qui nous assure $\Phi_n(X, m) \uparrow$ ou $\Phi_n(X, m) \downarrow \neq X(m)$ pour tout $X \in \mathcal{P}_{n+1}$.

En tant qu'intersection décroissante de fermés non vides, la classe $\bigcap_n \mathcal{P}_n$ est non vide. Soit $Y \in \bigcap_n \mathcal{P}_n$. Par construction, pour tout n , $\Phi_n^Y \neq X$, car $Y \in \mathcal{P}_n$. Donc, $X \not\leq_T Y$. ■

Hirschfeldt [87] a donné une élégante preuve alternative du théorème d'évitement de cône, comme simple conséquence du théorème 4.3 de base low et du théorème 4.5 de base calculatoirement dominée.

PREUVE ALTERNATIVE DU THÉORÈME 4.7. Deux cas se présentent.

- ▷ Cas 1. L'ensemble X est Δ_2^0 . En particulier, par la proposition 7-4.7, X est hyperimmune. Par le théorème 4.5 de base calculatoirement dominé, la classe $\mathcal{P}$ contient un ensemble calculatoirement dominé P . En particulier, P ne calcule pas X .
- ▷ Cas 2. L'ensemble X n'est pas Δ_2^0 . Par le théorème 4.3 de base low, la classe $\mathcal{P}$ contient un ensemble low P , donc Δ_2^0 . En particulier, P ne calcule pas X .

Dans chacun des cas, $\mathcal{P}$ contient un élément qui ne calcule pas X . ■

Nous verrons dans les chapitres à venir de nombreux autres théorèmes concernant les classes Π_1^0 .

5. Bases pour les classes Π_1^0 parfaites

Nous avons vu que les classes Π_1^0 sans élément calculable sont nécessairement des classes parfaites. Ces classes admettent des théorèmes de base renforcés, et l'on peut notamment y construire des sous-classes parfaites dont tous les éléments ont une propriété de faiblesse fixée à l'avance. Nous voyons ici un exemple avec les ensembles calculatoirement dominés.

L'idée est de recommencer la preuve de base calculatoirement dominée, mais en dupliquant la construction étape après étape.

Théorème 5.1

Soit $\mathcal{P}$ une classe Π_1^0 non vide ne contenant aucun élément calculable. Il existe une classe parfaite $\mathcal{B} \subseteq \mathcal{P}$ qui ne contient que des ensembles calculatoirement dominés.

PREUVE. Soit $P_\epsilon = \mathcal{P}$. Supposons que pour n et chaque $\sigma \in 2^{<\mathbb{N}}$ de taille n on ait défini des classes Π_1^0 deux à deux disjointes et non vides $\mathcal{P}_\sigma \subseteq \mathcal{P}$. On répète la construction du théorème 4.5 pour définir pour chaque σ une classe Π_1^0 non vide $\mathcal{Q}_\sigma \subseteq \mathcal{P}_\sigma$ telle que soit l'on a un m tel que $\Phi_n(X, m) \uparrow$ pour tout $X \in \mathcal{Q}_\sigma$, soit l'on a une fonction calculable $g : \mathbb{N} \rightarrow \mathbb{N}$ telle que $\Phi_n(X, m) < g(m)$ pour tout m et pour tout $X \in \mathcal{Q}_\sigma$. Comme $\mathcal{P}$ ne contient aucun point calculable, alors pour tout σ , $\mathcal{Q}_\sigma \subseteq \mathcal{P}$ non plus. Donc, d'après le corollaire 3.7, il doit exister τ_0, τ_1 incomparables telles que les classes $\mathcal{Q}_\sigma \cap [\tau_0]$ et $\mathcal{Q}_\sigma \cap [\tau_1]$ sont toutes les deux non vides. On définit alors $\mathcal{P}_{\sigma 0} = \mathcal{Q}_\sigma \cap [\tau_0]$ et $\mathcal{P}_{\sigma 1} = \mathcal{Q}_\sigma \cap [\tau_1]$.

Pour chaque $X \in 2^\mathbb{N}$, la classe $\bigcap_n \mathcal{P}_{X \upharpoonright n} \subseteq \mathcal{P}$ contient exactement un élément G_X ; cet élément est calculatoirement dominé, et par construction $X \neq Y$ implique $G_X \neq G_Y$. La classe des G_X pour $X \in 2^\mathbb{N}$ forme en fait un arbre parfait, dont les nœuds sont déterminés par le choix des extensions incomparables τ_0, τ_1 . ■

La technique de duplication du théorème précédent peut également être appliquée au théorème de base d'évitement de cône, mais il ne peut bien entendu pas être utilisé avec le théorème de la base low, car la classe des ensembles low est dénombrable. Le lecteur pourra essayer de l'appliquer quand même, afin de voir ce qui coince.

Notons enfin qu'il n'est bien entendu pas nécessaire de passer par des classes Π_1^0 pour construire une classe parfaite d'ensembles calculatoirement dominés, et l'on peut appliquer, comme suit, la même idée de duplication de construction à la preuve des f-arbres.

Exercice 5.2. (★) Construire une classe parfaite d'ensembles calculatoirement dominés via des f-arbres. ◇

Exercice 5.3. (★) Soit $\mathcal{P}$ une classe Π_1^0 non vide sans point calculable. Mélanger la construction ci-dessus avec la preuve du lemme 4.6 pour obtenir une sous classe parfaite de $\mathcal{P}$ dont tous les éléments sont calculatoirement dominés, et dont les degrés Turing sont deux à deux incomparables. ◇

Exercice 5.4. (★★) Soit $\mathcal{P}$ une classe parfaite. Construire une sous-classe parfaite de $\mathcal{P}$ dont les éléments sont deux à deux incomparables en termes de degrés Turing. ◇

Exercice 5.5. (★) Soit $\mathcal{P}$ une classe Π_1^0 non vide sans point isolé. Montrer que $\emptyset'$ calcule un élément non calculable de $\mathcal{P}$. $\diamond$

Notons que le dernier exercice utilise nécessairement le fait que la classe $\mathcal{P}$ ne contient pas de point isolé. Nous verrons avec la proposition 30-3.5 une technique simple, mais très puissante, permettant de construire des classes Π_1^0 — avec points isolés — dont les éléments sont soit des ensembles finis, soit des ensembles de complexité calculatoire « très élevée ».

6. Degrés PA

Nous prenons ici un peu d'avance sur le chapitre 9, dans lequel nous exposons les notions de théorie logique du premier ordre, de système formel de l'arithmétique de Peano ainsi que du premier théorème d'incomplétude de Gödel : la notion de degré PA est née en lien direct avec ces notions. Nous allons toutefois rapidement nous abstraire de cet aspect historique pour donner avec le théorème 6.2 une caractérisation équivalente des degrés PA ne faisant appel qu'aux notions de calculabilité vues jusqu'ici.

L'étude des degrés PA — acronyme de « Peano Arithmetic » — remonte aux travaux de Gödel et de son fameux théorème d'incomplétude : il n'existe pas d'extension calculable, complète et cohérente des axiomes de l'arithmétique de Peano¹. L'étude des degrés Turing se développant, la question de la puissance nécessaire pour calculer une telle extension s'est tout naturellement posée. Nous allons voir que les développements autour de cette question ont abouti à l'un des concepts les plus riches de la calculabilité, qui a sans doute trouvé l'apogée de sa force et de son intérêt à travers l'étude des mathématiques à rebours.

Dans ce qui suit, fixons une énumération calculable $\psi_0, \psi_1, \psi_2, \dots$ de toutes les formules de l'arithmétique. Supposons également qu'il existe une fonction calculable $\mathbf{neg} : \mathbb{N} \rightarrow \mathbb{N}$ telle que $\psi_{\mathbf{neg}(n)} = \neg\psi_n$. Pour le théorème qui suit, nous appellerons *théorie* un ensemble $T \subseteq \mathbb{N}$ tel que pour tout m , si $\{\psi_n : n \in T\} \vdash \psi_m$, alors $m \in T$.

En d'autres termes, une théorie est un ensemble de formules de l'arithmétique clos par conséquence logique. Une théorie T est *cohérente* si le code de la formule « $0 = 1$ » n'appartient pas à T . Une théorie T est *complète* si pour tout n : soit $n \in T$, soit $\mathbf{neg}(n) \in T$. Le lecteur qui aborde ces notions pour la première fois trouvera plus de détails dans le chapitre 9.

1. Cette version du théorème est en fait un renforcement de celui de Gödel, qui fut établi par Rosser.

Définition 6.1. Une *complétion de l'arithmétique de Peano* est une théorie complète T contenant $\{n \in \mathbb{N} : PA \vdash \psi_n\}$. Un degré Turing est PA s'il contient une complétion cohérente de l'arithmétique de Peano. $\diamond$

Les degrés PA étant clos par le haut, il est équivalent pour un degré d'être PA et de contenir un ensemble qui calcule une complétion cohérente de l'arithmétique de Peano.

Nous montrons dès à présent une équivalence qui nous servira de caractérisation pour les degrés PA.

Théorème 6.2 (Jockusch et Soare [96], Solovay (non publié))
Soit X un ensemble. Les deux énoncés suivants sont équivalents.

- (1) X est de degré PA.
- (2) X est de degré DNC_2 , c'est-à-dire que l'ensemble X calcule une fonction $f : \mathbb{N} \rightarrow \{0, 1\}$ telle que $f(n) \neq \Phi_n(n)$ pour tout n .

Avant de passer à la preuve, nous renvoyons le lecteur à la définition 7-2.7, qui introduisait la notion de degré DNC_f pour une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $2 \leq f(n) \leq f(n+1)$. La notion de degré DNC_2 est la plus forte possible de cet ordre : la fonction f calculée ne dispose que de deux possibilités (0 ou 1) pour différer de chaque $\Phi_n(n)$. Nous verrons avec les corollaires 18-4.3 et 19-1.8 que de nombreux ensembles de degré DNC ne sont pas DNC_2 .

PREUVE. L'équivalence montrée par Jockusch et Soare utilise le théorème de base de Scott [194] pour les degrés PA, qui stipule que tout degré PA calcule un chemin infini dans toute classe Π_1^0 non vide. Nous montrons ici directement l'équivalence.

L'implication (1) $\Rightarrow$ (2) est essentiellement le théorème de Gödel-Rosser, qui étend le premier théorème d'incomplétude de Gödel, et qui sera démontrée formellement avec le théorème 9-3.10 et le corollaire 9-3.11.

Montrons (2) $\Rightarrow$ (1). Soit $f \leq_T X$ une fonction à valeurs dans $\{0, 1\}$ telle que $f(n) \neq \Phi_n(n)$ pour tout n . On définit $T_0 = PA$. Supposons T_n cohérente, définie à l'étape n . On considère la formule de l'arithmétique ψ_n de code n , et l'on définit le code de machine e_n tel que

$$\Phi_{e_n}(e_n) = 1, \text{ si } T_n + \psi_n \vdash 0 = 1, \quad \text{et} \quad \Phi_{e_n}(e_n) = 0, \text{ si } T_n + \neg\psi_n \vdash 0 = 1.$$

Si $\Phi_{e_n}(e_n) \downarrow = 0$, alors $T_n + \neg\psi_n$ est incohérente, et $T_n + \psi_n$ est donc cohérente. Si $\Phi_{e_n}(e_n) \downarrow = 1$, alors $T_n + \psi_n$ est incohérente, et $T_n + \neg\psi_n$ est donc cohérente. Si $\Phi_{e_n}(e_n) \uparrow$, alors $T_n + \psi_n$ et $T_n + \neg\psi_n$ sont toutes les deux cohérentes. À présent, comme $f(e_n) \neq \Phi_{e_n}(e_n)$, on peut ainsi définir $T_{n+1} = T_n + \psi_n$ si $f(e_n) = 1$ et $T_{n+1} = T_n + \neg\psi_n$ si $f(e_n) = 0$. Dans tous les cas, on aura une théorie cohérente.

La théorie $T = \bigcup_n T_n$ est donc cohérente, et elle est par construction également complète. ■

Remarque

Notons que la direction (2) $\rightarrow$ (1) du théorème 6.2 fonctionne pour toute théorie T_0 cohérente dont les axiomes sont calculables. Ainsi, toute fonction DNC_2 est capable de calculer une complétion de toute théorie cohérente dont les axiomes sont calculables. La direction (1) $\rightarrow$ (2) est plus spécifique à l'arithmétique de Peano, car elle requiert une théorie suffisamment expressive pour coder des calculs par des formules.

Notons que $\emptyset'$ peut calculer une fonction DNC_2 et est donc de degré PA. La proposition suivante permet de déduire qu'il n'est pas du tout nécessaire d'être Turing complet pour calculer une extension complète et cohérente de l'arithmétique de Peano.

Définition 6.3. Le *spectre de degrés* d'une classe $\mathcal{P} \subseteq 2^{\mathbb{N}}$ est l'ensemble

$$\deg \mathcal{P} = \{\deg_T X : X \in [\mathcal{P}]\}.$$
 ◇

Proposition 6.4. Il existe une classe Π_1^0 dont le spectre de degrés correspond aux degrés PA. ★

PREUVE. Il s'agit d'une simple constatation, qui a déjà été utilisée pour la preuve du théorème 3.5. La classe des ensembles DNC_2 se décrit de la manière suivante :

$$\mathcal{P} = \{X \in 2^{\mathbb{N}} : \forall e \forall t \Phi_e(e)[t] \uparrow \vee \Phi_e(e)[t] \downarrow \neq X(e)\}. \quad \blacksquare$$

Corollaire 6.5

Il existe des degrés PA qui sont low.

PREUVE. D'après la proposition 6.4 et le théorème 4.3. ■

Corollaire 6.6

Il existe des degrés PA calculatoirement dominés.

PREUVE. D'après la proposition 6.4 et le théorème 4.5. ■

Corollaire 6.7

Soit A un ensemble non calculable. Alors, il existe un ensemble X de degré PA qui ne calcule pas A .

PREUVE. D'après la proposition 6.4 et le théorème 4.7. ■

Voyons à présent une autre caractérisation importante des degrés PA, qui stipule que ces derniers capturent la puissance de calcul nécessaire et suffisante pour le lemme faible de König.

Théorème 6.8

Soit $X \subseteq \mathbb{N}$. Les deux énoncés suivants sont équivalents.

- (1) *X est de degré PA.*
 - (2) *X calcule un ensemble dans chaque classe Π_1^0 non vide.*
- De plus, pour (2), le calcul est uniforme en un code de la classe Π_1^0 .*

PREUVE. Pour (2) $\Rightarrow$ (1), il suffit de remarquer qu'il existe une classe Π_1^0 non vide ne contenant que des ensembles de degrés PA (voir la proposition 6.4).

Montrons à présent (1) $\Rightarrow$ (2). Soit $f \leq_T X$ à valeurs dans $\{0, 1\}$, telle que $f(n) \neq \Phi_n(n)$ pour tout n . Soit $\mathcal{P}$ une classe Π_1^0 non vide, et soit T un arbre calculable tel que $[T] = \mathcal{P}$. Soit $\sigma_0 = \epsilon$. Étant donné σ_n défini tel que $[\sigma_n] \cap [T]$ est non vide, on calcule $\sigma_{n+1} = \sigma_n i$ pour $i \in \{0, 1\}$ de la manière suivante : on calcule d'abord le code e_n d'un programme qui sur toute entrée m cherche le plus petit t tel que pour $i = 0$ ou $i = 1$ aucune chaîne σ de taille t avec $\sigma \succeq \sigma_n i$ n'appartient à T . D'après le lemme de König, cette condition est équivalente au fait que $[\sigma_n i] \cap [T]$ soit vide. Si le programme repère un des deux événements, il s'arrête avec comme valeur i . Il suffit alors de regarder la valeur de $f(n)$. On définit simplement $\sigma_{n+1} = \sigma_n f(n)$. Comme $f(e_n) \neq \Phi_{e_n}(e_n)$, on a la garantie que $[\sigma_{n+1}] \cap [T]$ est non vide. ■

Classe universelle

Notons que d'après la proposition 6.4, il existe une classe Π_1^0 non vide dont tous les membres sont de degré PA, et que d'après le théorème 6.8, tout degré PA calcule un membre de chaque classe Π_1^0 non vide. Une telle classe est donc « maximale » en termes de complexité calculatoire, au sens où si l'on sait calculer un membre de cette classe, alors on sait calculer un membre de toute classe Π_1^0 non vide.

On appelle *classe Π_1^0 universelle* une classe Π_1^0 non vide dont tous les membres sont de degré PA.

Dans la même veine que le théorème 7-7.1, on termine par une caractérisation qui combine à présent le fait d'être de degré high ou PA. Notez la différence avec (1) $\leftrightarrow$ (3) du théorème 7-6.2, au sein duquel on considère une suite $(X_n)_{n \in \mathbb{N}}$ contenant exactement les ensembles calculables, alors qu'ici l'on considère seulement qu'elle contient les ensembles calculables.

Théorème 6.9 (Jockusch [98])

Soit $X \subseteq \mathbb{N}$. Les deux énoncés suivants sont équivalents.

- (1) L'ensemble X est de degré high ou PA.
- (2) L'ensemble X calcule une suite $(X_n)_{n \in \mathbb{N}}$ contenant tous les ensembles calculables.

PREUVE. Montrons d'abord (1) implique (2). Si X est high, alors l'implication est claire d'après le théorème 7-6.2. Supposons à présent que X est de degré PA. Soit $g \leq_T X$ telle que $g(n) \neq \Phi_n(n)$ pour tout n . Notons que l'ensemble X calcule aussi la fonction f qui inverse les valeurs de g , c'est-à-dire telle que $\Phi_n(n) \downarrow \in \{0, 1\}$ implique $f(n) = \Phi_n(n)$. Étant donné une fonction calculable Φ_e et un entier n , on peut calculer le code a_n tel que $\Phi_{a_n}(a_n) = \Phi_e(n)$. En utilisant ce procédé et le fait que $\Phi_{a_n}(a_n) \downarrow \in \{0, 1\}$ implique $f(a_n) = \Phi_{a_n}(a_n) = \Phi_e(n)$, on calcule aisément un ensemble X_e tel que si $n \mapsto \Phi_e(n)$ est totale et à valeurs dans $\{0, 1\}$ alors $X_e(n) = \Phi_e(n)$ pour tout n . On peut donc calculer notre suite $(X_e)_{e \in \mathbb{N}}$ contenant tous les ensembles calculables.

Montrons à présent (2) implique (1). Soit $(X_n)_{n \in \mathbb{N}}$ une suite X -calculable contenant tous les ensembles calculables. L'idée est de procéder au départ comme dans la preuve de (3) $\Rightarrow$ (1) du théorème 7-6.2. Étant donné un prédicat Π_2^0 de la forme

$$P = \{e : \forall x_1 \exists x_2 R(e, x_1, x_2)\},$$

l'idée était de définir uniformément en e une fonction partielle calculable f_e telle que :

- (a) $e \in P$ implique que f_e est une fonction totale calculable ;
- (b) $e \notin P$ implique que f_e est une fonction partielle qui n'a aucune complétion totale calculable.

Il suffit de remarquer que dans le théorème 7-6.2, la définition de f_e qui est donnée est telle que dans le cas (b), non seulement aucune complétion de f_e n'est calculable, mais en plus une telle complétion est forcément de degré PA. La définition était la suivante.

Soit e fixé. À l'étape de calcul t , pour toute valeur n plus petite que t et telle que f_e ne n'arrête pour le moment pas sur n , on procède comme suit : si $\Phi_n(n)[t] \downarrow \neq 0$, on définit $f_e(n) = 0$. Sinon, si $\Phi_n(n)[t] \downarrow \neq 1$, on définit $f_e(n) = 1$. Sinon, si pour tous $k \leq n$ il existe $m_k \leq t$ tel que $R(e, k, m_k)$, alors on définit $f_e(n) = 0$.

Tout comme dans la preuve du théorème 7-6.2, si $e \in P$ alors f_e est une fonction totale. Dans le cas contraire, on s'aperçoit que pour presque toutes les valeurs de n telles que $\Phi_n(n) \downarrow$ on a $f_e(n) \neq \Phi_n(n)$. Toute complétion

de f_e est donc une fonction DNC_2 , modulo un nombre fini de valeurs, et est donc de degré PA.

Il y a à présent deux possibilités :

- ▷ soit $(X_n)_{n \in \mathbb{N}}$ contient un ensemble de degré PA, auquel cas on a (1) ;
- ▷ soit ce n'est pas le cas, et l'on peut alors donner une définition $\Sigma_2^0(X)$ de P comme dans la preuve du théorème 7-6.2, ce qui implique, appliquée à $P = \mathbb{N} \setminus \emptyset''$, que $\emptyset''$ est $\Delta_2^0(X)$, et que X est donc high. ■

Nous terminons cette section par un exercice qui constitue une caractérisation alternative et bien connue des degrés PA.

Exercice 6.10. (★) Montrer que X est PA si, et seulement si, pour tous ensembles c. e. A, B avec $A \cap B = \emptyset$, il existe un ensemble X -calculable C tel que $A \subseteq C$ et $C \cap B = \emptyset$. ◇

7. Arbres à branchement fini

Nous introduisons ici l'*espace de Baire* : la classe $\mathbb{N}^{\mathbb{N}}$ de toutes les suites infinies à valeurs dans $\mathbb{N}$, ou autrement dit la classe de toutes les fonctions de $\mathbb{N}$ dans $\mathbb{N}$. Tout comme l'espace de Cantor a son ensemble de chaînes $2^{<\mathbb{N}}$, l'espace de Baire a son ensemble de chaînes $\mathbb{N}^{<\mathbb{N}}$: les suites finies à valeurs dans $\mathbb{N}$. Les différentes opérations que nous avons vues sur les chaînes binaires (préfixe, concaténation, longueur, ...) s'étendent sans problème aux chaînes de l'espace de Baire. En particulier, étant donné une chaîne $\sigma \in \mathbb{N}^{<\mathbb{N}}$, on dénote par $[\sigma]$ la classe des suites $P \in \mathbb{N}^{\mathbb{N}}$ telles que $\sigma \prec P$. La notion d'arbre s'étend également aux sous-ensembles de $\mathbb{N}^{<\mathbb{N}}$ comme suit.

Définition 7.1. Un ensemble $T \subseteq \mathbb{N}^{<\mathbb{N}}$ est un *arbre* si T est clos par préfixe, c'est-à-dire pour tout $\sigma \in T$ et tout $\tau \preceq \sigma$, alors $\tau \in T$. ◇

Contrairement aux arbres binaires, les nœuds peuvent avoir une infinité de successeurs. Un nœud $\sigma \in T$ est *branchant* s'il a au moins deux successeurs. Un *chemin* de T est une suite $P \in \mathbb{N}^{\mathbb{N}}$ dont tous les segments initiaux sont dans T . On dénote par $[T]$ la classe des chemins de T . Le lemme de König ne fonctionne plus sur les arbres de l'espace de Baire, comme le montre contre-exemple suivant.

Exemple 7.2. Soit $T = \{\sigma \in \mathbb{N}^{<\mathbb{N}} : \forall n < |\sigma| \sigma(n) \geq |\sigma|\}$. L'arbre T contient des nœuds de longueur arbitraire et est infini, mais $[T] = \emptyset$.

La puissance calculatoire des chemins d'un arbre calculable arbitraire de l'espace de Baire sera étudiée dans la partie **IV** sur l'hypercalculabilité. Dans cette section, nous allons nous restreindre à une sous-catégorie de ces arbres tombant sous la coupe du lemme de König. Les arbres $T \subseteq \mathbb{N}^{<\mathbb{N}}$ à *branchement fini*, c'est-à-dire au sein desquels chaque nœud a un nombre fini de successeurs.

Lemme 7.3 (Lemme de König). Soit $T \subseteq \mathbb{N}^{<\mathbb{N}}$ un arbre à branchement fini tel que $|T| = \infty$. Alors, $[T]$ est non vide. ★

PREUVE. On construit un chemin X par récurrence sur n . Comme T est infini, mais que la racine ϵ n'a qu'un nombre fini de successeurs, il existe d'après le principe des tiroirs un $i \in \mathbb{N}$ et une infinité de nœuds $\sigma \in T$ qui étendent i (c'est-à-dire avec $i \prec \sigma$). On définit $X(0) = i$. Supposons que $\tau = X(0)X(1) \dots X(n)$ soit défini avec $\tau \in T$ et tel qu'il y a une infinité de nœuds $\sigma \in T$ pour lesquels $\tau \preceq \sigma$. Le nœud σ n'ayant qu'un nombre fini de successeurs, le principe des tiroirs confirme qu'il existe $i \in \mathbb{N}$ et une infinité de nœuds $\sigma \in T$ qui étendent τi . On définit $X(n+1) = i$.

Par récurrence sur n , on définit donc de cette manière un ensemble X tel que $X \upharpoonright_n \in T$ pour tout n . ■

L'espace de Baire

Comme pour l'espace de Cantor, les ouverts de l'espace de Baire sont les classes $\mathcal{O} \subseteq \mathbb{N}^{<\mathbb{N}}$ de la forme $\mathcal{O} = \bigcup_{\sigma \in W} [\sigma]$ pour un ensemble $W \subseteq \mathbb{N}^{<\mathbb{N}}$, et les fermés $\mathcal{P} \subseteq \mathbb{N}^{<\mathbb{N}}$ sont de la forme $[T]$ pour un arbre $T \subseteq \mathbb{N}^{<\mathbb{N}}$. En revanche, contrairement à l'espace de Cantor, les fermés de l'espace de Baire ne sont pas compacts en général. Les compacts de l'espace de Baire sont précisément les fermés $\mathcal{P}$ de la forme $[T]$ pour un arbre $T \subseteq \mathbb{N}^{<\mathbb{N}}$ à branchement fini.

La preuve du lemme de König est quasiment la même que celle de sa version faible, et l'on pourrait s'attendre à première vue à ce que la puissance calculatoire nécessaire pour calculer un chemin d'un arbre calculable à branchement fini soit celle des degrés PA. Ce n'est cependant pas le cas, comme le montre la proposition 7.4.

Proposition 7.4. Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre binaire Δ_2^0 . Il existe un arbre calculable à branchement fini S tel que $\deg([T]) = \deg([S])$. ★

PREUVE. Soit $(T_n)_{n \in \mathbb{N}}$ une approximation Δ_2^0 de l'arbre T . On peut supposer sans perte de généralité que pour tout entier n , T_n est clos par préfixe et $T_n \subseteq 2^{\leq n}$ (l'ensemble des chaînes de taille inférieure ou égale à n). On montre aisément que toute réunion d'arbres est un arbre, ce qui implique que $\bigcup_n T_n$ est un arbre.

Montrons que $[\bigcup_n T_n] = [T]$. Clairement, $T \subseteq \bigcup_n T_n$, donc $[T] \subseteq [\bigcup_n T_n]$. Soit $P \in [\bigcup_n T_n]$. Soit s un entier ≥ 0 ; montrons que $P \upharpoonright_s \in T$. L'approximation $(T_n)_{n \in \mathbb{N}}$ de T étant Δ_2^0 , on a $P \upharpoonright_s \in T$ ssi $\forall t \exists n \geq t \ P \upharpoonright_s \in T_n$. Soit $t \geq s$. Comme $P \upharpoonright_t \in \bigcup_n T_n$ et comme par hypothèse $\bigcup_{n < t} T_n \subseteq 2^{<t}$, on a $P \upharpoonright_t \in T_n$ pour $n \geq t$. Par clôture par le bas de T_n , il vient $P \upharpoonright_s \in T_n$. Donc, $\forall t \exists n \geq t \ P \upharpoonright_s \in T_n$. Ainsi, $P \upharpoonright_s \in T$.

Soient $\sigma, \tau \in \mathbb{N}^{<\mathbb{N}}$ de même longueur. On note $\langle \sigma, \tau \rangle$ la chaîne ρ de longueur $|\sigma|$ telle que pour tout $n < |\rho|$, $\rho(n) = \langle \sigma(n), \tau(n) \rangle$. L'opération s'étend naturellement aux suites infinies P, Q pour lesquelles on écrira $\langle P, Q \rangle$.

Nous allons construire un arbre calculable $S \subseteq \mathbb{N}^{<\mathbb{N}}$ à branchement fini dont tous les chemins seront de la forme $\langle P, Q \rangle$ avec $P \in [\bigcup_n T_n] = [T]$ et Q un « témoin » de $P \in [\bigcup_n T_n]$, au sens où pour tout s , $P \upharpoonright_s \in T_{Q(s)}$.

Définissons une fonction partielle calculable $f : \bigcup_n T_n \rightarrow \mathbb{N}^{<\mathbb{N}}$ qui envoie des chaînes vers des chaînes de même longueur inductivement comme suit.

▷ $f(\epsilon) = \epsilon$.

▷ Si $\sigma i \in \bigcup_n T_n$, alors $f(\sigma i) = f(\sigma) \frown s$, où s est le plus petit entier tel que $\sigma i \in T_s$ et $\frown$ la concaténation.

▷ Par continuité, la fonction f s'étend aux suites infinies de $[\bigcup_n T_n] = [T]$.

Soit $S = \{\langle \sigma, f(\sigma) \rangle : \sigma \in \bigcup_n T_n\}$. Notons que la réunion $\bigcup_n T_n$ n'est pas calculable en général, mais que S l'est, car il est facile de vérifier que pour tout $\rho = \langle \sigma, \mu \rangle$, on a $f(\sigma) = \mu$. L'ensemble S est clos par préfixe, car $\bigcup_n T_n$ l'est également et $f(\sigma i) \upharpoonright_{|\sigma|} = f(\sigma)$ pour tout $\sigma \in \bigcup_n T_n$. Ainsi, S est un arbre calculable. Notons également que S est 2-branchant, donc à branchement fini.

Montrons que $\deg([T]) = \deg([S])$. Soit $P \in [T]$; alors, $\langle P, f(P) \rangle \in [S]$ et $P \equiv_T \langle P, f(P) \rangle$. Soit $R \in [S]$; alors, $R = P \oplus f(P)$ pour un P dans $[\bigcup_n T_n] = [T]$. De même, $R \equiv_T P$. Cela conclut la preuve de la proposition 7.4. ■

Corollaire 7.5

Il existe un arbre calculable $T \subseteq \mathbb{N}^{<\mathbb{N}}$ à branchement fini et un degré PA P ne calculant pas de chemin à travers T .

PREUVE. Soit $S = \{\emptyset' \upharpoonright_n : n \in \mathbb{N}\}$ l'arbre binaire Δ_2^0 ayant $\emptyset'$ pour unique chemin infini. Par la proposition 7.4, il existe un arbre calculable $T \subseteq \mathbb{N}^{<\mathbb{N}}$ à branchement fini tel que $\deg([T]) = \deg([S])$. En particulier, tout chemin de T calcule $\emptyset'$. Par le corollaire 6.5, il existe un degré à la fois PA et low. En particulier, ce degré ne calcule pas de chemin à travers T . ■

On peut relativiser la notion d'être DNC_2 relativement à un oracle X : on demande le calcul d'une fonction $f : \mathbb{N} \rightarrow \{0, 1\}$ telle que $f(n) \neq \Phi_n(X, n)$ pour tout n . Le théorème 6.8 se relativise bien au sens où les degrés DNC_2 relativement à X , que l'on appellera aussi degrés PA relativement à X ou $\text{PA}(X)$, coïncident avec ceux permettant de calculer un chemin dans toute classe $\Pi_1^0(X)$ non vide.

Exercice 7.6. Soit Y un ensemble $\text{PA}(X)$. Montrer que $Y \geq_T X$. $\diamond$

On en déduit qu'un degré PA relativement à $\emptyset'$ est nécessaire pour calculer un chemin dans tout arbre calculable infini à branchement fini. Notons que la situation de l'exercice 7.6 est différente lorsque l'on considère les degrés DNC au lieu des degrés DNC_2 . Plus précisément, si X est un ensemble non calculable, il existe un ensemble Y de degré DNC relativement à X qui ne calcule pas X (voir le corollaire 18-4.4). Ce résultat fait intervenir des notions de théorie de l'aléatoire que nous aborderons au chapitre le chapitre 18.

Arbre binaire vs arbre 2-branchant

L'arbre $T \subseteq \mathbb{N}^{<\mathbb{N}}$ construit dans la preuve de la proposition 7.4 est 2-branchant, au sens où chaque nœud a au plus deux successeurs. D'un point de vue purement structurel, il est donc isomorphe à un arbre binaire $S \subseteq 2^{<\mathbb{N}}$. Pourtant, il existe des degrés PA qui ne calculent pas de chemin dans cet arbre. La différence entre la puissance calculatoire de cet arbre et celle d'un arbre binaire ne provient donc pas d'une différence combinatoire, mais relève tout simplement d'un manque d'information sur les successeurs d'un nœud : étant donné un arbre calculable à branchement fini, on ne peut pas borner de manière calculable et uniforme la valeur maximale du successeur d'un nœud.

La remarque précédente nous conduit à la définition qui suit.

Définition 7.7. Un arbre $T \subseteq \mathbb{N}^{<\mathbb{N}}$ est *calculatoirement borné* s'il existe une fonction calculable $g : \mathbb{N} \rightarrow \mathbb{N}$ telle que pour tout $\sigma \in T$, et pour tout $n < |\sigma|$, on a l'inégalité $\sigma(n) < g(n)$. $\diamond$

Il est clair que tout arbre calculatoirement borné est à branchement fini. La proposition suivante permet de réconcilier l'idée selon laquelle des objets combinatoirement similaires devraient avoir la même puissance calculatoire, en montrant que dès que l'arbre à branchement fini est accompagné d'une borne calculable sur son branchement, alors la puissance calculatoire nécessaire pour calculer un chemin est exactement celle des degrés PA.

Étant donné une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$, nous noterons $f^{<\mathbb{N}}$ l'ensemble des chaînes $\sigma \in \mathbb{N}^{<\mathbb{N}}$ telles que pour tout $n < |\sigma|$, $\sigma(n) < f(n)$.

Proposition 7.8. Pour tout arbre $T \subseteq \mathbb{N}^{<\mathbb{N}}$ calculable et calculatoirement borné, il existe un arbre binaire $S \subseteq 2^{<\mathbb{N}}$ tel que $\deg([T]) = \deg([S])$. ★

PREUVE. L'idée de la preuve est tout simplement de définir un codage binaire des chaînes, en utilisant la borne calculatoire pour savoir combien de bits allouer à chaque niveau. Pour retirer toute ambiguïté, nous noterons $2^{=n}$ l'ensemble des chaînes binaires de longueur n , au lieu de 2^n , qui désignera la n -ième puissance de 2. Soit $g : \mathbb{N} \rightarrow \mathbb{N}$ une fonction calculable telle que $T \subseteq g^{<\mathbb{N}}$. Sans perte de généralité, nous pouvons supposer que $g(n) = 2^{h(n)}$ pour tout n , avec $h : \mathbb{N} \rightarrow \mathbb{N}$ une fonction calculable.

Pour tout entier n , soit $e_n : 2^n \rightarrow 2^{=n}$ la bijection canonique. Par exemple, la fonction $e_2 : 4 \rightarrow \{00, 01, 10, 11\}$ vérifie

$$e_2(0) = 00, \quad e_2(1) = 01, \quad e_2(2) = 10 \quad \text{et} \quad e_2(3) = 11.$$

Ce codage s'étend en une bijection calculable $e : g^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ définie par

$$e(\sigma) = e_{h(0)}(\sigma(0)) \frown e_{h(1)}(\sigma(1)) \frown \cdots \frown e_{h(|\sigma|-1)}(\sigma(|\sigma|-1)),$$

où l'on dénote ici —comme d'usage— la concaténation par $\frown$. Par exemple, si $h(n) = n + 1$, alors $g(n) = 2^{h(n)} = 2^{n+1}$, et

$$e(032) = e_1(0) \frown e_2(3) \frown e_3(2) = 0 \frown 11 \frown 010 = 011010.$$

Notons que l'ensemble $\widehat{S} = \{e(\sigma) : \sigma \in T\}$ n'est pas un arbre, car il n'est clos par préfixe que pour les segments initiaux de longueur exactement $\text{Im } g$. Nous devons donc définir l'arbre S comme la clôture par préfixe de $\widehat{S}$, autrement dit $S = \{e(\sigma) \upharpoonright_n : \sigma \in T \wedge n \in \mathbb{N}\}$. La fonction de codage e étant monotone sur les longueurs, et l'ensemble T étant clos par préfixe, $\rho \in S$ si, et seulement si, il existe une chaîne $\sigma \in g^{<\mathbb{N}}$ de longueur au plus $|\rho|$ telle que $\rho \prec e(\sigma)$. Ainsi, S est un arbre binaire calculable infini, dont les chemins sont exactement les suites infinies de la forme $e_{h(0)}(P(0)) \frown e_{h(1)}(P(1)) \frown \cdots$, pour $P \in [T]$. Dès lors, $\deg([T]) = \deg([S])$. ■

Corollaire 7.9

Soit $T \subseteq \mathbb{N}^{<\mathbb{N}}$ un arbre infini calculable et calculatoirement borné. Tout degré PA calcule un chemin de T .

PREUVE. Par la proposition 7.8, il existe un arbre binaire $S \subseteq 2^{<\mathbb{N}}$ tel que $\deg([T]) = \deg([S])$. Par le théorème 6.8, tout degré PA calcule un chemin de S , donc tout degré PA calcule un chemin de T . ■

Nous voyons à présent la réciproque de la proposition 7.4, qui montre que les degrés PA relativement à $\emptyset'$ sont exactement ceux permettant de calculer un chemin dans un arbre calculable à branchement fini.

Proposition 7.10. Soit $T \subseteq \mathbb{N}^{<\mathbb{N}}$ un arbre calculable infini à branchement fini. Il existe $S \subseteq 2^{<\mathbb{N}}$ un arbre binaire Δ_2^0 tel que $\deg([T]) = \deg([S])$. ★

PREUVE. Notons tout d'abord que tout arbre calculable infini à branchement fini $T \subseteq \mathbb{N}^{<\mathbb{N}}$ est $\emptyset'$ -calculatoirement borné, c'est-à-dire qu'il existe une fonction $\emptyset'$ -calculable $g : \mathbb{N} \rightarrow \mathbb{N}$ telle que $T \subseteq g^{<\mathbb{N}}$. La preuve de la proposition 7.8 se relativise à $\emptyset'$, et permet de définir $S \subseteq 2^{<\mathbb{N}}$ un arbre binaire Δ_2^0 tel que $\deg([T]) = \deg([S])$. ■

Donnons pour finir quelques exercices. La *clôture par préfixe* d'un ensemble $S \subseteq \mathbb{N}^{<\mathbb{N}}$ est l'ensemble

$$\widehat{S} = \{\tau \in \mathbb{N}^{<\mathbb{N}} : \exists \sigma \in S \ \tau \preceq \sigma\}.$$

Exercice 7.11. Montrer que pour tout ensemble infini $S \subseteq 2^{<\mathbb{N}}$, sa clôture par préfixe admet un chemin. ◇

Exercice 7.12. Montrer qu'il existe un ensemble infini calculable $S \subseteq 2^{<\mathbb{N}}$ tel que $[\widehat{S}] = \{\emptyset'\}$. ◇

Exercice 7.13. (★★) Montrer que pour tout arbre $T \subseteq 2^{<\mathbb{N}}$ infini et $\emptyset'$ -calculable, il existe un ensemble infini $S \subseteq 2^{<\mathbb{N}}$ contenant exactement une chaîne de chaque longueur, tel que $[T] = [\widehat{S}]$. ◇

Schéma Récapitulatif

Voici une figure qui récapitule les différentes notions abordées jusqu'ici.

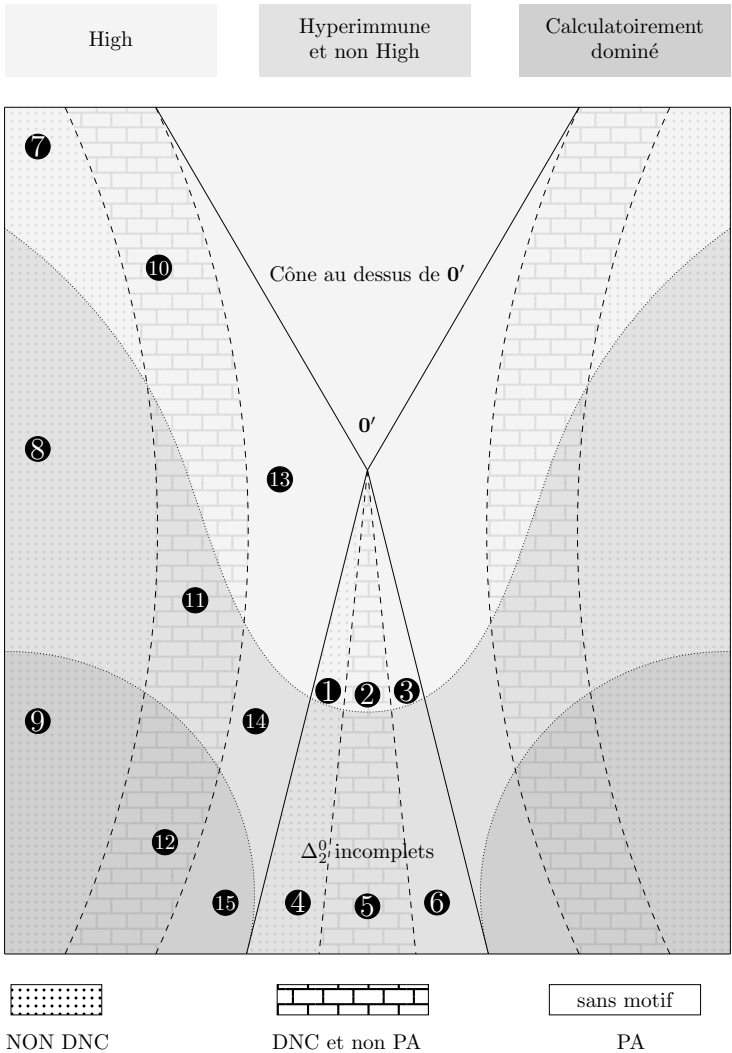


FIGURE 7.14 – Récapitulatif sur les degrés Turing vus jusqu'ici. Les points (1) à (6) renvoient vers des exemples d'ensembles de chaque type. Les points de (7) à (15) renvoient vers l'existence d'une classe parfaite d'ensembles de chaque type (et donc d'après l'exercice 5.4 vers l'existence d'une classe parfaite de degrés Turing de chaque type).

Les points de (1) à (15) qui suivent utilisent pour certains des concepts qui ne seront vus que dans les chapitres suivants.

- (1) Il existe un degré Turing Δ_2^0 non DNC et high : il suffit de mixer la preuve de l'exercice 4-10.6 avec celle de l'exercice 7-2.9 pour obtenir un ensemble high qui en plus d'être incomplet, ne soit pas DNC.
- (2) Il existe un degré Turing Δ_2^0 DNC, non PA et high : On part d'un ensemble X DNC, non PA et low (par le théorème 18-4.1 et le corollaire 18-2.3, on peut prendre par exemple un aléatoire au sens de Martin-Löf low). On peut ensuite broder sur la construction de l'exercice 4-10.6 et sur celle aussi de l'exercice 7-2.9 pour construire un ensemble $Y \Delta_2^0$ high tel que $X \oplus Y$ est non PA (en utilisant en particulier le fait que $X' \leq_T \emptyset'$).
- (3) Il existe un degré Turing Δ_2^0 incomplet, PA et high : on part d'un ensemble X PA et low (voir le corollaire 6.5). On brode ensuite sur la construction de l'exercice 4-10.6 pour construire un ensemble $Y \Delta_2^0$ high tel que $X \oplus Y$ est incomplet.
- (4) Il existe un degré Turing Δ_2^0 non DNC, hyperimmune et non high : il suffit de construire un ensemble non DNC et low, en mélangeant la proposition 4-9.1 et l'exercice 7-2.9.
- (5) Il existe un degré Turing Δ_2^0 DNC, non PA, hyperimmune et non high : il suffit de considérer un ensemble aléatoire au sens de Martin-Löf et low. D'après le théorème 18-4.1, un tel ensemble est DNC. D'après le théorème 19-1.7, il n'est pas PA. D'après la proposition 7-4.7, il est hyperimmune. Enfin, comme il est low il ne peut être high.
- (6) Il existe un degré Turing Δ_2^0 PA, hyperimmune et non high : d'après la proposition 7-4.7, il suffit de considérer un degré PA low donné par le corollaire 6.5.
- (7) Il existe une classe parfaite d'ensembles high et non DNC. Il suffit d'utiliser le théorème 10-3.21 et de broder sur Posner/Robinson (voir le corollaire 10-3.34) pour construire une classe parfaite d'ensembles 1-génériques et high.
- (8) Il existe une classe parfaite d'ensembles hyperimmunes, non high et non DNC. D'après le théorème 10-3.2, la proposition 10-3.38 et le corollaire 10-3.34 tout ensemble suffisamment générique sera dans ce cas-là.
- (9) Il existe une classe parfaite d'ensembles calculatoirement dominés et non DNC. Il faut reprendre la construction d'ensembles calculatoirement dominés via des f -arbres, et la modifier pour produire des ensembles non DNC à la manière dont cela est fait dans l'exercice 7-2.9.
- (10) Il existe une classe parfaite d'ensembles high, DNC et non PA. On peut appliquer le théorème 18-3.4 de Kučera/Gács relativisé à $\emptyset'$ sur l'arbre des 2-aléatoires pour construire des ensembles high et 2-aléatoires. Par

le théorème 18-4.1, de tels ensembles sont DNC. Par le théorème 19-1.7, ils ne sont pas PA.

- (11) Il existe une classe parfaite d'ensembles hyperimmunes, non high, DNC et non PA. D'après le théorème 18-4.1, le corollaire 19-3.9 et le corollaire 19-1.8, c'est le cas pour tout ensemble suffisamment aléatoire.
- (12) Il existe une classe parfaite d'ensembles calculatoirement dominés, DNC et non PA. Il s'agit du théorème 5.1 appliqué à une classe Π_1^0 ne contenant que des aléatoires au sens de Martin-Löf. D'après le théorème 19-1.7, les ensembles MLR et calculatoirement dominés ne peuvent être PA.
- (13) Il existe une classe parfaite d'ensembles incomplets, high et PA. On fixe un ensemble X high et incomplet. On élabore ensuite sur le théorème 4.7 de base d'évitement de cône relativisé à X , pour construire une classe parfaite d'ensembles PA Y tel que $X \oplus Y$ est incomplet.
- (14) Il existe une classe parfaite d'ensembles hyperimmunes, non high et PA. Soit X un ensemble hyperimmune et non high. On utilise la relativisation à X du théorème 5.1, appliqué à la classe Π_1^0 des ensembles DNC_2 , pour construire une classe parfaite d'ensembles Y tels que toute fonction calculée par $X \oplus Y$ est dominée par une fonction calculée par X .
- (15) Il existe une classe parfaite d'ensembles calculatoirement dominés et PA. Il s'agit du théorème 5.1 appliqué à la classe Π_1^0 des ensembles DNC_2 .

Interlude formel

1. Un peu d'histoire : la crise des fondements

Les mathématiques se sont développées naturellement au fil des siècles en tant qu'outil au service d'une représentation abstraite de la réalité. Cet état de fait est par exemple flagrant en sciences physiques pour lesquelles les mathématiques rendent compte avec précision d'un ensemble varié de phénomènes. Avec le temps, les notions étudiées sont devenues de plus en plus complexes, de plus en plus abstraites, et les connexions entre les mathématiques et le monde réel sont devenues de plus en plus incertaines. Pendant longtemps, la discipline a cependant pu compter sur le sens logique inné de l'esprit humain pour parler de choses que l'on ne « voyait plus » tout en gardant un cadre rigoureux. Les nombres complexes en constituent un exemple frappant : les nombres de carré négatif qui n'existent *a priori* que dans l'imagination du mathématicien sont baptisés en 1545 par Cardan « quantités sophistiquées ». De la sophistication il en fallait sans doute pour accepter cet ovni comme objet d'étude sérieux. Pourtant ces « quantités sophistiquées » trouvent leur utilité dans la résolution de problèmes, eux, très concrets. Raphaël Bombelli en donne une première formalisation en 1572, et montre comment utiliser ces nombres pour résoudre certaines équations du troisième degré. Ils trouveront au fil des siècles de nombreuses utilités en mathématiques, ainsi qu'en physique, où ils sont utilisés avec succès au sein d'équations représentant le monde réel.

Il a fallu un certain saut conceptuel pour accepter le développement d'un cadre mathématique rigoureux et cohérent autour des nombres complexes. Disons, malgré tout, que ce concept pour aussi surprenant qu'il soit, reste encore « relativement simple ». Les véritables problèmes arrivent avec les

travaux de Cantor sur la cardinalité et les nombres transfinis. Cantor ouvre grand la porte sur un gouffre sans fond, qui nous emmène bien au-delà de ce que peut appréhender avec confiance l'esprit humain : l'étude de l'infini. Bien sûr, l'infini est présent en mathématique depuis l'antiquité, en premier lieu via la considération qu'il n'existe pas de plus grand nombre entier. Mais la révolution épistémologique de Cantor consiste à considérer l'infini comme un objet d'étude à part entière. Cette considération amènera aux balbutiements de ce qui deviendra un siècle plus tard la Théorie des ensembles avec un grand 'T'.

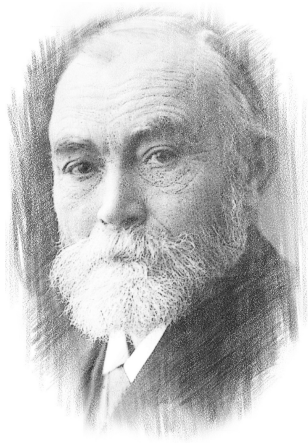
Avec les travaux de Cantor, la question de savoir *ce qu'est* l'activité mathématique s'est faite de plus en plus pressante : peut-on raisonner vraiment sur tout, et même sur l'infini, concept qui nous dépasse ? Mais si l'on accepte, comme c'est le cas aujourd'hui que l'on peut raisonner sur l'infini, on ne peut certainement pas le faire n'importe comment. Au fond, qu'est-ce *faire* des mathématiques ? En particulier quand on commence à manipuler des objets sur lesquels nous n'avons plus tellement d'intuition, comment être sûr que ce dont on parle a réellement un sens ? Ces considérations ont trouvé leur apogée lors de la fameuse « crise des fondements », qui va de la fin du XIX^e siècle au début du XX^e. Il s'agit alors de définir des règles pour encadrer l'activité mathématique. Il s'agit en fait de définir précisément l'étude mathématique non pas d'objets comme les entiers, les réels ou encore les fonctions, mais de définir *l'étude mathématique des mathématiques elles-mêmes*. Il est *a posteriori* remarquable de constater le succès de cette entreprise : les mathématiques sont un outil suffisamment puissant pour pouvoir se définir et s'étudier elles-mêmes, avec la rigueur inhérente à la discipline ! Ce ne fut évidemment pas un chemin aisé. Dans cette entreprise, les trois mousquetaires — qui comme on le sait sont en fait au nombre de quatre — s'appellent Frege, Russell, Zermelo et Hilbert.

Frege

Philosophe et mathématicien allemand de la fin du XIX^e siècle, Gottlob Frege est mu par une certitude : la logique précède les mathématiques. Mais la logique de l'époque est encore très pauvre, et se cantonne essentiellement aux travaux de George Boole sur ce que l'on appelle aujourd'hui *le calcul propositionnel* : la manipulation de propositions, vraies ou fausses, que l'on peut connecter entre elles via les « et » et « ou » logiques, bien connus des informaticiens. Pour l'ambition de Frege, ce système est trop restreint pour asseoir les mathématiques sur des fondations logiques. En particulier, rien dans le calcul propositionnel ne permet de parler d'objets spécifiques à travers les relations qu'ils entretiennent les uns avec les autres. Il formalise alors dans son ouvrage le « Begriffsschrift » un nouveau langage afin de

palier aux carences de la logique de l'époque, langage qui évoluera pour devenir ce que l'on appelle aujourd'hui *le calcul des prédicats*, omniprésent en mathématiques.

Frege est aujourd'hui considéré comme le père de la logique moderne, notamment via son concept de variables quantifiées $\forall x \dots$, $\exists x \dots$. Il utilise son formalisme pour s'atteler dans ses ouvrages suivants, « Fondements de l'arithmétique » (1884) et « Lois fondamentales de l'arithmétique I et II » (parus en 1893 et 1903), à fonder l'arithmétique sur la logique. Il crée pour cela une définition des entiers naturels qui peut être vue aujourd'hui comme reposant sur le concept *d'ensemble* et sur celui du *schéma d'axiomes de compréhension* : si $\Psi(x)$ est une formule mathématique qui peut être vraie ou fausse en fonction de x , alors l'ensemble $\{x : \Psi(x)\}$ des éléments qui satisfont cette formule est bien défini.

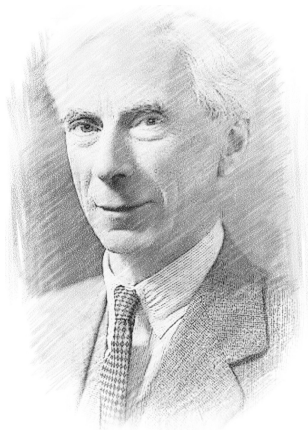


Gottlob Frege, 1848–1925

Russell

En 1902, Russell dans une lettre adressée à Frege lui fait part d'un doute concernant ses travaux. Soit y l'ensemble des ensembles qui ne s'appartiennent pas : $y = \{x : x \notin x\}$. A-t-on alors $y \in y$ ou bien $y \notin y$? On comprend aisément la situation paradoxale à laquelle on arrive. Cet exemple restera célèbre comme étant *le paradoxe de Russell*. À 53 ans, Frege comprend alors que le paradoxe de Russell implique l'effondrement du système qu'il a mis des années à échafauder. Un coup dur dont il aura beaucoup de mal à se remettre.

Malgré ce paradoxe, Russell accueille avec beaucoup d'enthousiasme les travaux de Frege, et contribue largement à en faire reconnaître la valeur. Tout comme Frege, Russell sent le besoin d'appuyer les mathématiques sur des bases solides. Tout comme Frege,



Bertrand Russell, 1872–1970

Russell a cette intuition que la logique précède les mathématiques. Il va s'atteler dix ans durant, avec son ancien professeur Alfred Whitehead, à cette recherche de fondements logiques, qui aboutiront à leur fameux ouvrage « Principia Mathematica » : un travail titanesque qui s'étend sur plus de 2000 pages, dont l'ambition est de décrire un ensemble d'axiomes logiques et de règles d'inférence à partir desquelles toute vérité mathématique pourrait être démontrée. Ces travaux posent les fondements de ce que l'on appelle aujourd'hui *la théorie des types*, un système encore étudié aujourd'hui, présentant des liens très forts avec les langages de programmation. En parallèle, s'est développée la théorie des ensembles dont l'axiomatisation a été initiée par Zermelo, et complétée plus tard par Fraenkel et Skolem indépendamment, pour donner le système axiomatique ZF, du nom de Zermelo-Fraenkel.

Zermelo

Un fait absolument remarquable est que le système ZF, auquel il est parfois nécessaire de rajouter l'axiome du choix, donne un cadre au sein duquel peuvent se formaliser *la totalité des mathématiques modernes*, si l'on exclut les récents développements de la théorie des ensembles, dont l'objectif est justement de sortir de ce système. Zermelo trouve le moyen d'éviter le fameux paradoxe de Russell — comme Russell lui-même avec sa théorie des types — en bornant l'axiome de compréhension. L'ensemble $\{x : \Psi(x)\}$ n'est désormais plus valide, il convient de partir d'un ensemble existant y , auquel cas on peut à présent définir l'ensemble des éléments



Ernst Zermelo, 1871–1953

de y qui satisfont $\Psi : \{x \in y : \Psi(x)\}$. Pourtant, cette théorie ne fait pas tout de suite consensus. Poincaré, s'il n'a jamais activement pris part à la crise des fondements, en a suivi les développements avec intérêt. Comme tous les protagonistes de l'époque, il est très conscient du danger qui se cache derrière le paradoxe de Russell. Il théorise même le problème sous le nom d'*imprédictivité*¹ : une définition imprédictive est en substance une définition circulaire dans laquelle l'objet qui est défini est lui-même susceptible d'être utilisé dans la définition. C'est ce qui se passe quand on définit $y = \{x : x \notin x\}$: l'ensemble y défini à gauche de l'égalité est aussi

1. Utilisé auparavant par Russell dans un sens légèrement différent.

concerné à droite du signe égal, puisque l'on considère potentiellement tous les ensembles. Si l'axiomatique de Zermelo évite le paradoxe de Russell, il reste néanmoins, indirectement imprédicatif, comme le remarquera Poincaré qui écrira [176] :

« *En posant d'avance son ensemble M [Poincaré parle alors de la borne utilisée par Zermelo dans l'axiome de compréhension, qui permet de définir pour un ensemble M existant $\{x \in M : \Psi(x)\}$, il [Mr. Zermelo] a élevé un mur de clôture qui arrête les gêneurs qui pourraient venir du dehors. Mais il ne se demande pas s'il peut y avoir des gêneurs du dedans qu'il a enfermés avec lui dans son mur. Si l'ensemble M a une infinité d'éléments, cela veut dire non que ces éléments puissent être conçus comme existant d'avance tout à la fois, mais qu'il peut sans cesse en naître de nouveaux ; ils naîtront à l'intérieur du mur, au lieu de naître dehors, voilà tout.* »

Le système de Zermelo considère que si un ensemble A existe, alors l'ensemble de ses parties $\mathcal{P}(A)$ existe également. Cet axiome combiné avec l'axiome de compréhension restreint permet de faire des définitions circulaires. Ainsi, dans la définition $A = \{n \in \mathbb{N} : \forall S \in \mathcal{P}(\mathbb{N}) \ n \notin S\}$, le quantificateur $\forall S$ va prendre pour valeur tous les sous-ensembles de $\mathbb{N}$, et en particulier l'ensemble A lui-même. L'ensemble A est donc défini en fonction de lui-même. Poincaré ajoute alors à la fin de son argumentation :

« *Mais s'il [Mr. Zermelo] a bien fermé sa bergerie, je ne suis pas sûr qu'il n'y ait pas enfermé le loup. Je ne serais tranquille que s'il avait démontré qu'il est à l'abri de la contradiction.* »

On peut difficilement donner tort à Poincaré : comment être sûr au fond qu'un paradoxe de Russell ne surgira pas de nulle part, au détour d'une définition circulaire cachée ? Zermelo lui-même est conscient du problème, et cherchera à démontrer sans succès que son système axiomatique est *cohérent*, c'est-à-dire exempt de paradoxe. Cette recherche de preuve de la cohérence des mathématiques a connu son apogée vers 1920, sous l'impulsion de David Hilbert.

Hilbert

Hilbert est certainement — tout comme Poincaré — un des derniers mathématiciens à avoir une connaissance approfondie de l'ensemble des mathématiques de son époque. Son œuvre est considérable, et il a profondément influencé les développements de la discipline durant le XX^e siècle. Il participe activement à la crise des fondements en opposant à Russell une vision *formaliste* des mathématiques, plutôt qu'une vision *logiciste*. Pour Hilbert, les mathématiques doivent pouvoir se réduire à un ensemble de règles, que l'on doit pouvoir appliquer de façon purement mécanique et déconnectée de toute psychologie du mathématicien. Il imagine ainsi des systèmes de preuve, au sein desquels il différencie *les axiomes*, qui sont les

phrases mathématiques que l'on suppose vraies, par exemple les axiomes de Zermelo-Fraenkel, et les *règles de déduction*, qui permettent de combiner les axiomes entre eux pour en déduire des théorèmes. C'est la vision de Hilbert qui finira par s'imposer, même si cela ne se sera pas accompli sans remous. L'objectif ultime pour Hilbert est de montrer via des systèmes de déduction considérés comme sûr — en particulier via des raisonnements finis sur des objets finis — que l'ensemble des mathématiques, qui elles font appel à des objets infinis dont la pertinence est sujette à caution, forme un système cohérent, c'est-à-dire exempt de paradoxe :

c'est ce que l'on appellera le *programme de Hilbert*, dont le *Entscheidungsproblem* évoqué dans la section 6-1 constitue l'un des aspects. Ce programme connaîtra un coup d'arrêt brutal avec les travaux de Gödel, qui démontre dix ans plus tard son fameux théorème d'incomplétude : l'arithmétique elle-même est impuissante à démontrer qu'elle est exempte de paradoxe.



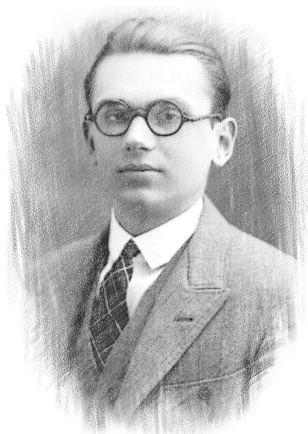
David Hilbert, 1862–1943

Le cinquième mousquetaire

Les travaux de Gödel apportent une conclusion aussi magistrale qu'inattendue à la crise des fondements. Gödel montre deux choses : même les systèmes les plus simples et les mieux compris, comme l'arithmétique, qui ne parle que d'objets finis, contiennent des vérités indémonstrables. En particulier, et à supposer qu'il soit vrai que les axiomes de l'arithmétique forment un système exempt de paradoxes, alors la cohérence de l'arithmétique est elle-même l'une de ces vérités indémonstrables. Gödel montre enfin — avec l'aide de Rosser — que l'ajout d'axiomes ne change rien à l'affaire : n'importe quel système d'axiomes cohérent, contenant l'arithmétique, et « dont on peut connaître les axiomes » ne peut démontrer sa propre cohérence. Gödel développe pour cela les premières versions de ce qui sera plus tard la calculabilité : « dont on peut connaître les axiomes » signifie calculable, dans un sens similaire au sens moderne.

Le retentissement dans le monde mathématique est colossal. Le programme de Hilbert est à terre, et les mathématiques n'auront jamais de fondations entièrement satisfaisantes.

Encore aujourd'hui, on ne sait pas si le système ZF qui axiomatise l'ensemble des mathématiques est cohérent : et pour cause, si comme on l'espère il est bien exempt de paradoxe, on ne pourra jamais le démontrer mathématiquement, ou en tout cas tant que l'on se cantonne aux axiomes de ZF. L'impact épistémologique est considérable. Les mathématiques, mère des sciences exactes, sont non seulement tributaires d'une *croyance* , mais sont en plus capables de démontrer qu'il en sera toujours ainsi !



Kurt Gödel, 1906–1978

2. La logique du premier ordre

De Frege et Russell on retiendra le langage logique moderne utilisé en mathématiques, de Hilbert on retiendra un système de preuve à base d'axiomes et de règles de déductions applicables aux énoncés mathématiques, et de Zermelo on retiendra le système axiomatique ZF ou ZFC, suffisant pour formaliser l'ensemble des mathématiques traditionnelles. Nous présentons à présent sans rentrer dans trop de détails les principes de base de la logique du premier ordre, notre objectif étant de présenter de manière un peu plus précise le théorème de Gödel et ses conséquences. Pour cette raison, le fil rouge de notre présentation sera l'exemple spécifique de l'arithmétique de Peano.

2.1. Langage de l'arithmétique

La première étape pour formaliser nos démonstrations mathématiques est de fixer le langage utilisé. Nous allons pour cela définir le langage de l'arithmétique de Peano.

Définition 2.1. Le langage $\mathcal{L}_{PA}$ de l'arithmétique de Peano comprend les symboles spécifiques au calcul des prédicats :

- (1) des symboles de *variables* $x, y, z, \dots$: elles représentent des entiers naturels ;
- (2) les parenthèses $()$ et les symboles de *connecteurs logiques* : $\wedge, \vee, \rightarrow, \neg$;
- (3) des symboles de *quantificateurs* : $\forall, \exists$.

Et ceux spécifiques à l'arithmétique de Peano :

- (1) les symboles de *fonctions binaires* suivantes : $+$, $\times$;
- (2) les symboles de *relations binaires* suivantes : $=$, $<$;
- (3) des symboles de *constantes* $\dot{0}$, $\dot{1}$. ◇

Un langage n'est rien d'autre qu'une liste de symboles, mais ces symboles ont vocation à être utilisés avec un sens précis. En ce qui concerne le calcul des prédicats, il s'agit du sens usuel : par exemple, $\ll \wedge \gg$ est le *et* logique ou encore $\ll \exists \gg$ est la quantification existentielle. En ce qui concerne les symboles spécifiques à l'arithmétique de Peano, il y a d'abord les fonctions $+$ et $\times$ qui représentent respectivement l'addition et la multiplication, les symboles d'égalité et d'inégalité qui ont leurs sens usuels sur les entiers, et enfin les constantes $\dot{0}$, $\dot{1}$ qui chacune représente l'entier respectif correspondant.

Langages du premier ordre

On voit sans peine comment généraliser la définition précédente pour obtenir d'autres langages. Les symboles spécifiques au calcul des prédicats sont les mêmes dans tous les langages du premier ordre, auxquels on ajoute un nombre arbitraire de symboles de fonctions (n -aire pour des entiers n arbitraires), un nombre arbitraire de symboles de relations (également n -aire pour des entiers n arbitraires) et un nombre arbitraire de symboles de constantes.

Les symboles de fonctions sont soumis à des règles d'agencement pour former ce que l'on appelle les *termes* du langage.

Définition 2.2. Les *termes* du calcul des prédicats de l'arithmétique sont définis inductivement de la manière suivante.

- (1) Une variable ou une constante est un terme.
- (2) Si t_1, t_2 sont des termes, alors $(t_1 + t_2)$ et $(t_1 \times t_2)$ sont des termes. ◇

Exemple 2.3. Les expressions suivantes sont des termes :

$$x, \quad (((x + \dot{1}) + \dot{1}) + \dot{1} + \dot{1}), \quad (\dot{1} + \dot{0}), \quad (x + (y \times z)).$$

Comme nous pouvons le voir, le langage de l'arithmétique est assez minimaliste, et les expressions valides sont très structurées pour ôter toute ambiguïté.

En pratique, on utilisera un certain nombre de raccourcis de notation pour améliorer la lisibilité, tant que la traduction en termes valides est non ambiguë. Par exemple, $t_0 + t_1 + t_2$ est un raccourci pour $((t_0 + t_1) + t_2)$. De même, $x + \dot{3}$ est un raccourci pour $(x + ((\dot{1} + \dot{1}) + \dot{1}))$.

Définition 2.4. Un terme est *clos* s'il ne contient aucune variable, et donc uniquement des constantes et les opérations $+$ et $\times$. $\diamond$

Intuitivement, un terme clos de l'arithmétique est une manière de représenter un entier naturel. Par exemple, $(\dot{1} + \dot{1}) \times \dot{0}$ est un nom pour l'entier 0. Les entiers possèdent chacun une infinité de noms.

Exemple 2.5. Le terme $x + \dot{1}$ n'est pas clos, contrairement à $(\dot{1} + \dot{1}) \times \dot{0}$.

Si les symboles de fonctions sont utilisés pour créer les termes du langage, les symboles de relations sont eux utilisés pour créer les *formules* du langage.

Définition 2.6. Les *formules de l'arithmétique* sont définies comme suit.

- (1) Pour tous termes t_1, t_2 , alors $t_1 = t_2$ et $t_1 < t_2$ sont des formules. Ces formules sont appelées *formules atomiques*. Les formules atomiques auxquelles on ajoute leurs négations, ici, $\neg t_1 = t_2$ et $\neg t_1 < t_2$, sont appelées *littéraux*.
- (2) Pour toutes formules F_1, F_2 , alors $(F_1 \wedge F_2), (F_1 \vee F_2), (F_1 \rightarrow F_2)$ et $\neg F_1$ sont des formules.
- (3) Pour toute formule F , alors $\forall x F$ et $\exists x F$ sont des formules. $\diamond$

Là encore, on aura recours au sucre syntaxique en notant $t_1 \leq t_2$ la formule $(t_1 < t_2) \vee (t_1 = t_2)$ et $F_1 \leftrightarrow F_2$ la formule $(F_1 \rightarrow F_2) \wedge (F_2 \rightarrow F_1)$.

Formules de premier ordre

Ici aussi, on généralise sans peine la formation de formules et de termes dans n'importe quel langage : les symboles de fonction du langage servent à créer les termes, qui servent ensuite avec l'aide des symboles de relation à créer des formules atomiques, qui peuvent ensuite être composées entre elles avec l'aide des symboles du calcul des prédicats comme dans (2) et (3) de la définition précédente.

Avec les quantificateurs apparaissent les notions de variables *libres* et de variables *liées* : les variables *liées* sont sans surprise celles qui sont liées à un quantificateur, et les variables libres sont celles qui ne le sont pas. La définition formelle est assez lourde, mais quelques exemples suffisent à s'en créer une intuition.

Exemple 2.7. Dans la formule $\ll \forall x \exists y y = x + \dot{1} \gg$, les variables x et y sont liées, alors que dans la formule $\ll \exists y y = x + 1 \gg$ seule la variable y est liée, contrairement à la variable x qui est libre.

Définition 2.8. Une *formule close* ou un *énoncé* est une formule dans laquelle aucune variable n'est libre. ◇

Notation

Étant donné une formule F ayant pour variables libres $x_1, \dots, x_n$, on écrira $F(x_1, \dots, x_n)$ pour signifier que les variables libres de F sont $x_1, \dots, x_n$.

Intuitivement, une formule close est une affirmation qui aura une valeur de vérité (vraie ou fausse) une fois évaluée sur les entiers. Les formules possédant des variables libres définissent quant à elles des prédicats sur les entiers.

2.2. Les systèmes de déduction à la Hilbert

Afin de formaliser mathématiquement la notion de démonstration, Hilbert a imaginé un système de règles bien précises qui suffisent à montrer « tout ce qui est démontrable ». Comment le sait-on ? Cette idée sera rendue précise avec le théorème de complétude de Gödel à venir. Par la suite, de nombreux autres systèmes de démonstration ont été développés, tous équivalents et plus ou moins adaptés à certains objectifs.

2.2.1. Axiomes et règles

Dans un système à la Hilbert, une démonstration est une liste finie de phrases mathématiques $F_0, F_1, F_2, \dots, F_n$ — des formules dans le langage considéré — satisfaisant les règles suivantes : pour tout $i \leq n$, soit F_i est un axiome, soit F_i est produit à partir de règles d'inférence appliquées à des formules $F_{j_1}, \dots, F_{j_m}$ pour $j_1, \dots, j_m < i$. Chaque phrase F_i dans cette liste sera alors démontrée, l'objectif étant normalement d'obtenir F_n , la dernière d'entre elles. Voyons à présent un exemple précis de système à la Hilbert suffisamment puissant pour démontrer tout ce qui est démontrable.

Les axiomes : les axiomes que nous pouvons toujours utiliser sont les tautologies de la logique du premier ordre. Ainsi, par exemple $A \vee \neg A$ pourra être utilisé comme axiome. On en distingue trois types.

1. Les tautologies de la logique propositionnelle. Par exemple, $(F \rightarrow G) \rightarrow (\neg G \rightarrow \neg F)$ est une tautologie de la logique propositionnelle : elle sera vraie pour n'importe quelle formule F ou G indépendamment de leur valeur de vérité.
2. Les tautologies du calcul des prédicats. En pratique, seuls quatre schémas d'axiomes sont nécessaires :
 - (a) $\forall x(F \rightarrow G) \rightarrow (F \rightarrow \forall xG)$ pour toute formule F ne contenant pas la variable x , et toute formule G ;

- (b) $\exists x(F \rightarrow G) \rightarrow (\exists xF \rightarrow G)$ pour toute formule F , et toute formule G ne contenant pas la variable x ;
- (c) $\forall xF \rightarrow F_{t/x}$ pour tout terme t et toute formule F ne contenant aucune variable de t ;
- (d) $F_{t/x} \rightarrow \exists xF$ pour tout terme t et toute formule F ne contenant aucune variable de t .

Ci-dessus, $F_{t/x}$ désigne la formule F pour laquelle chaque occurrence de x est remplacée par le terme t .

3. Les axiomes de l'égalité :

- (e) $t = t$ pour tout terme t ;
- (f) $t_1 = q_1 \wedge \dots \wedge t_n = q_n \rightarrow f(t_1, \dots, t_n) = f(q_1, \dots, q_n)$ pour tout n , tous termes $(t_i)_{1 \leq i \leq n}, (q_i)_{1 \leq i \leq n}$ et tout symbole de fonction n -aire f ;
- (g) $t = q \rightarrow (F(t/z) \rightarrow F(q/z))$ pour tous termes t, q et toute formule $F(z)$ ne faisant pas intervenir des variables de t ou q .

Ci-dessus, $F(t/z)$ et $F(q/z)$ désignent la formule F dans laquelle chaque occurrence de z est remplacée respectivement par t et q .

Notons que ces schémas d'axiomes dépendent du langage considéré, chaque langage utilisant des symboles de fonctions et relations qui leur sont spécifiques pour construire respectivement les termes et les formules atomiques.

Symbole d'égalité

Nous considérons ici que le symbole d'égalité fait nécessairement partie du langage que l'on utilise, et aura toujours son sens usuel, ce qui justifie les axiomes de l'égalité mentionnés ci-dessus.

Les règles d'inférence. Les règles d'inférence permettent de combiner des phrases déjà démontrées dans notre liste, pour en obtenir de nouvelles. Les deux règles suivantes sont suffisantes.

1. *Règle 1 : le Modus Ponens*, à la base de tout raisonnement déductif. Si $A \rightarrow B$ est démontré et si A est démontré, alors on peut en déduire B .
2. *Règle 2 : la généralisation*. Si $F(x)$ est démontré pour une variable x libre dans F , on peut alors en déduire $\forall xF(x)$. Cette règle est très utilisée en mathématiques : si l'on veut prouver par exemple que pour tous rationnels $x < y$, il existe un rationnel z tel que $x < z < y$, on commence par fixer des variables rationnelles x, y sur lesquelles on ne suppose rien d'autre que $x < y$. Si l'on arrive à déduire l'existence d'un rationnel z tel que $x < z < y$, sans utiliser aucune propriété spécifique de x, y , on en déduit par la règle de généralisation que pour tous rationnels $x < y$, il existe un rationnel z tel que $x < z < y$.

Cela conclut la description de notre système à la Hilbert. Voyons tout de suite un exemple de démonstration.

Exemple 2.9. Montrons $\forall xF(x) \rightarrow \exists xF(x)$. Pour plus de lisibilité, on notera $A \equiv \forall xF(x)$, $B \equiv F(y)$ et $C \equiv \exists xF(x)$.

- (1) $A \rightarrow B$ (axiome (c)).
- (2) $B \rightarrow C$ (axiome (d)).
- (3) $(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow ((A \rightarrow B) \wedge (B \rightarrow C)))$ (tautologie).
- (4) $(B \rightarrow C) \rightarrow ((A \rightarrow B) \wedge (B \rightarrow C))$ (Modus Ponens sur (1) et (3)).
- (5) $(A \rightarrow B) \wedge (B \rightarrow C)$ (Modus Ponens sur (2) et (4)).
- (6) $((A \rightarrow B) \wedge (B \rightarrow C)) \rightarrow (A \rightarrow C)$ (tautologie).
- (7) $A \rightarrow C$ (Modus Ponens sur (5) et (6)).

— Quid des univers vides ? —

Le lecteur pourra être surpris par la phrase $\forall xF(x) \rightarrow \exists xF(x)$ — que l'on a démontrée. Que se passe-t-il si l'on se place dans un univers vide ? À ce moment, $\forall xF(x)$ est bien vérifié, mais pas $\exists xF(x)$. L'axiome (d) ci-dessus implique effectivement que les formules démontrées ne seront valides que s'il existe au moins un élément dans notre univers. La notion d'univers sera rendue précise dans la section 2.4 à venir. Il est de fait nécessaire d'avoir une telle restriction si l'on veut démontrer que toute formule est équivalente à une formule sous forme prénexe (voir la définition 2.12).

2.2.2. Premiers outils

Nous prétendons que le système à la Hilbert décrit ci-dessus permet de montrer tout ce qui est démontrable, et nous le verrons formellement avec le théorème de complétude à venir. Le système est volontairement minimaliste, et difficile à manipuler tel quel. Le lecteur pourra par exemple essayer de démontrer $\neg\forall xF(x) \rightarrow \exists x \neg F(x)$ pour se rendre compte de la difficulté d'utilisation du système en l'état. Le mathématicien qui veut faire ce genre de preuve procèdera naturellement comme pour toute démonstration : en supposant que $\neg\forall xF(x)$ est vrai, et en essayant d'en déduire $\exists x \neg F(x)$. Le problème est que si l'on veut respecter le formalisme d'un système à la Hilbert, $\neg\forall xF(x)$ n'est pas nécessairement un axiome que l'on peut supposer vrai afin d'en dériver une conclusion. Nous voyons alors notre premier outil fondamental, qui va nous permettre de procéder comme on en a l'habitude.

Lemme 2.10 (Lemme de déduction). Soit F une formule close. Si l'on peut faire une démonstration de G en utilisant F comme axiome, il existe alors une démonstration de $F \rightarrow G$ (qui n'utilise pas F comme axiome). ★

PREUVE. Soit $G_1, \dots, G_n$ une démonstration de G_n utilisant F comme axiome. Montrons par récurrence sur la taille d'une démonstration que l'on peut faire quelques insertions dans la suite $F \rightarrow G_1, \dots, F \rightarrow G_n$ afin d'en faire une démonstration valide n'utilisant pas F comme axiome.

Si G_i est l'énoncé F , alors $F \rightarrow F$ est un axiome de la logique propositionnelle. Si G_i est un axiome de la logique propositionnelle, alors c'est aussi le cas de $F \rightarrow G_i$. Si G_i est l'un des axiomes (a), (b), (c), (d) du calcul des prédicats, alors $G_i \rightarrow (F \rightarrow G_i)$ est un axiome de la logique propositionnelle. En utilisant le Modus Ponens sur G_i et $G_i \rightarrow (F \rightarrow G_i)$, on obtient bien $F \rightarrow G_i$. Si $G_i = \forall x G_j$ pour $j < i$ est obtenu par la règle de généralisation, alors $\forall x (F \rightarrow G_j)$ est obtenu à partir de $F \rightarrow G_j$ (que l'on a par hypothèse de récurrence) par la règle de généralisation. Par l'axiome (a), on a

$$\forall x (F \rightarrow G_j) \rightarrow (F \rightarrow \forall x G_j),$$

de sorte que par Modus Ponens on obtient $F \rightarrow \forall x G_j$. Enfin, si $G_i = G_b$ est obtenu par Modus Ponens sur G_a , alors $G_a \rightarrow G_b$ pour $a, b < i$. Alors, on a $F \rightarrow G_a$ et $F \rightarrow (G_a \rightarrow G_b)$, par hypothèse de récurrence. La formule

$$(F \rightarrow (G_a \rightarrow G_b)) \rightarrow ((F \rightarrow G_a) \rightarrow (F \rightarrow G_b))$$

est une tautologie du calcul des prédicats.

Par Modus Ponens, on en déduit $(F \rightarrow G_a) \rightarrow (F \rightarrow G_b)$ et, par une deuxième application du Modus Ponens, on en déduit $F \rightarrow G_b$. ■

Voyons tout de suite un exemple d'application du lemme de déduction pour démontrer $\neg \forall x F(x) \rightarrow \exists x \neg F(x)$.

Exemple 2.11. Montrons $\neg \exists x F(x) \rightarrow \forall x \neg F(x)$. D'après le lemme de déduction, on peut supposer $\neg \exists x F(x)$ comme axiome.

- (1) $\neg \exists x F(x)$ (axiome).
- (2) $F(x) \rightarrow \exists x F(x)$ (axiome (d)).
- (3) $(F(x) \rightarrow \exists x F(x)) \rightarrow (\neg \exists x F(x) \rightarrow \neg F(x))$ (tautologie).
- (4) $\neg \exists x F(x) \rightarrow \neg F(x)$ (Modus Ponens sur (2) et (3)).
- (5) $\neg F(x)$ (Modus Ponens sur (1) et (4)).
- (6) $\forall x \neg F(x)$ (généralisation sur (5)).

Montrons à présent $\forall x \neg \neg F(x) \rightarrow \forall x F(x)$.

- (1) $\forall x \neg \neg F(x)$ (axiome).
- (2) $\forall x \neg \neg F(x) \rightarrow \neg \neg F(x)$ (axiome (c)).
- (3) $\neg \neg F(x)$ (Modus Ponens sur (1) et (2)).
- (4) $\neg \neg F(x) \rightarrow F(x)$ (tautologie).
- (5) $F(x)$ (Modus Ponens sur (3) et (4)).
- (6) $\forall x F(x)$ (généralisation sur (5)).

On laisse au lecteur le soin d'utiliser la contraposée pour en déduire

$$\neg \forall x F(x) \rightarrow \exists x \neg F(x).$$

2.2.3. Forme prénexe

Ce système de démonstration nous permet d'établir que toute formule — dans un langage quelconque — est prouvablement équivalente à une formule sous *forme prénexe*.

Définition 2.12. Une formule est sous *forme prénexe* si elle est de la forme $Q_1 x_1 \dots Q_n x_n F(x_1, \dots, x_n, y_1, \dots, y_m)$ où chaque Q_i est un quantificateur $\forall$ ou $\exists$ et $F(x_1, \dots, x_n, y_1, \dots, y_m)$ est une formule sans quantificateur. $\diamond$

On laisse au lecteur le soin de montrer les équivalences suivantes :

- ▷ $\forall x F \wedge G \equiv \forall x (F \wedge G)$;
- ▷ $\forall x F \vee G \equiv \forall x (F \vee G)$;
- ▷ $\exists x F \wedge G \equiv \exists x (F \wedge G)$;
- ▷ $\exists x F \vee G \equiv \exists x (F \vee G)$.

Chacune de ces équivalences, couplées à l'exemple 2.11, permet de transformer n'importe quelle formule en une formule prénexe prouvablement équivalente dans notre système de déduction, en déplaçant petit à petit les quantificateurs vers la gauche.

Notons que les équivalences ci-dessus ne tiennent que si l'on se place dans un univers possédant au moins un élément. Ainsi, par exemple, aura-t-on $(\forall x x = x) \wedge (\exists y y \neq y)$ manifestement faux dans un univers vide, alors que $\forall x (x = x \wedge (\exists y y \neq y))$ est toujours vrai.

2.3. Théories logiques et arithmétique de Peano

Une fois un langage fixé — dans notre cas celui de l'arithmétique — et le système de démonstration spécifié, on peut alors considérer une *théorie mathématique* dans ce langage, et l'utiliser pour démontrer des théorèmes concernant la structure décrite par cette théorie.

Définition 2.13. Une *théorie* T dans un langage $\mathcal{L}$ est une collection de formules closes de ce langage. On utilise aussi souvent le terme *système axiomatique* ou plus simplement *système* pour désigner une théorie. $\diamond$

La théorie est alors vue comme une liste d'axiomes, que l'on peut utiliser dans nos démonstrations, en plus des axiomes présents dans le système de démonstration.

Voyons tout de suite les axiomes de l'arithmétique qui furent mis au point par Peano vers la fin du XIX^e siècle.

2.3.1. Axiomes de l'arithmétique de Peano

Les axiomes de Peano permettent de spécifier le comportement des entiers naturels. La première série d'axiomes définit le comportement des entiers vis-à-vis du successeur.

- (1) $\forall x \neg(x + \dot{1} = \dot{0})$: 0 n'a pas de prédécesseur.
- (2) $\forall x (x = \dot{0} \vee \exists y (x = y + \dot{1}))$: tout entier différent de 0 a un prédécesseur.
- (3) $\forall x \forall y (x + \dot{1} = y + \dot{1} \rightarrow x = y)$: la fonction successeur pour les entiers est injective.

Les axiomes suivants donnent des règles pour calculer l'addition et la multiplication :

- (4) $\forall x (x + \dot{0} = x)$;
- (5) $\forall x \forall y (x + (y + \dot{1}) = (x + y) + \dot{1})$;
- (6) $\forall x (x \times \dot{0} = \dot{0})$;
- (7) $\forall x \forall y (x \times (y + \dot{1}) = (x \times y) + x)$.

Enfin, on définit le comportement des entiers vis-à-vis de l'ordre :

- (8) $\forall x \forall y (x < y \leftrightarrow (\exists z (z \neq \dot{0} \wedge x + z = y)))$.

Notation

On note **Q** la théorie composée des axiomes (1)-(8), qui forment ce que l'on appelle l'*arithmétique de Robinson*.

Pour obtenir l'arithmétique de Peano, on ajoute l'axiome suivant, pour toute formule de l'arithmétique $F(x)$:

- (9) $(F(\dot{0}) \wedge (\forall x (F(x) \rightarrow F(x + \dot{1})))) \rightarrow \forall x F(x)$

Notons que l'axiome (9) n'est pas un unique axiome. Tout comme pour les axiomes (a), (b), (c), (d) de notre système de démonstration, il s'agit d'un *schéma d'axiomes*, c'est-à-dire d'une infinité d'axiomes paramétrés par une formule, ici $F(x)$.

L'énoncé (9) est l'axiome bien connu de l'induction sur les entiers : si une formule F est vraie pour l'entier 0, et si le fait qu'elle soit vraie pour n implique qu'elle le soit pour $n + 1$, alors elle est vraie pour tout entier n .

Notation

On note **PA** la théorie composée de **Q** et du schéma d'axiome (9) pour toute formule de l'arithmétique. C'est cette théorie que l'on appelle *arithmétique de Peano*.

Nous verrons dans le chapitre 23 comment utiliser les axiomes de PA pour montrer quelques faits élémentaires sur les entiers naturels. Nous verrons

en particulier que le schéma d'induction est équivalent au schéma suivant : pour toute formule F vraie pour au moins un entier, il existe un plus petit entier x tel que $F(x)$ est vrai. Cela peut bien entendu sembler parfaitement évident, car nous avons en tête la structure des entiers naturels $\mathbb{N}$ que nous connaissons bien, mais une démonstration n'utilise pas cette structure : elle utilise uniquement les axiomes, et dans le cas de l'arithmétique, ceux-ci sont précisément faits pour que toute structure mathématique les vérifiant se comporte comme les entiers naturels. Cela nous amènera à la notion de *modèle*, dans la section suivante.

2.3.2. Démonstrations dans une théorie

Une fois que l'on a fixé une théorie, on peut en utiliser les axiomes au sein d'un système à la Hilbert pour démontrer des énoncés mathématiques.

Notation

On écrira $T \vdash F$ pour signifier qu'il existe une démonstration de la formule F à partir des axiomes de T via le système à la Hilbert exposé dans la section 2.2. S'il n'existe pas de telle démonstration, on écrira alors $T \not\vdash F$.

Parmi les tautologies de la logique propositionnelle, on trouve pour toutes formules F, G la formule $(F \wedge \neg F) \rightarrow G$, appelée « ex falso quodlibet », signifiant qu'à partir d'une contradiction $(F \wedge \neg F)$ on peut déduire ce que l'on veut. Il s'ensuit que si une théorie permet de prouver une formule et son contraire, tout énoncé est prouvable dans cette théorie, ce qui lui retire tout intérêt. On attendra donc avant tout d'une théorie qu'elle soit *cohérente*.

Définition 2.14. Une théorie T est *cohérente* s'il n'existe aucune formule F telle que $T \vdash F \wedge \neg F$. 

Notation

On écrira $T \vdash \perp$ pour signifier $T \vdash F \wedge \neg F$ pour une certaine formule F , la notation $T \not\vdash \perp$ signifiant alors logiquement que pour toute formule F on a $T \not\vdash F \wedge \neg F$. Comme le symbole d'égalité fait partie de notre langage, on peut considérer sans perte de généralité $\perp$ comme étant $\neg x = x$. Comme $x = x$ est un axiome, si $T \vdash \neg x = x$, alors $T \vdash x = x \wedge \neg x = x$.

Nous avons vu dans l'introduction de notre interlude que l'incohérence trouvée par Russell dans le travail de Frege fut une pierre angulaire dans la crise des fondements. Aussi les mathématiciens aimeraient-ils autant que possible être certains de ne travailler qu'avec des théories cohérentes. Mais comment peut-on vérifier la cohérence d'une théorie ? Sans même parler de théorie, qu'en est-il de la cohérence du système à la Hilbert lui-même et

de ses axiomes de la logique, présentés dans la section 2.2? La notion de *modèle* permet de résoudre ces questions.

2.4. Structures, modèles et théorème de cohérence

La notion de structure peut se définir de manière très générale pour tout langage fixé. Commençons par l'exemple qui nous intéresse plus spécifiquement, à savoir le langage de l'arithmétique.

Définition 2.15. Une *structure* $\mathcal{M} = (M, +^{\mathcal{M}}, \times^{\mathcal{M}}, <^{\mathcal{M}}, =^{\mathcal{M}}, 0^{\mathcal{M}}, 1^{\mathcal{M}})$ dans $\mathcal{L}_{\text{PA}}$ est donnée par :

- ▷ nn ensemble non vide M ;
- ▷ des fonctions $+^{\mathcal{M}}, \times^{\mathcal{M}} : M \times M \rightarrow M$ correspondant aux symboles de fonction $+, \times$;
- ▷ une relation $<^{\mathcal{M}} \subseteq M \times M$ correspondant au symbole de relation $<$;
- ▷ une relation $=^{\mathcal{M}} \subseteq M \times M$ correspondant au symbole de relation $=$, et qui correspond à « la vraie égalité », c'est-à-dire telle que
$$(x, y) \in =^{\mathcal{M}} \leftrightarrow x = y.$$
- ▷ Des éléments $0^{\mathcal{M}}, 1^{\mathcal{M}} \in M$ correspondant respectivement aux symboles de constante 0 et 1 . ◇

Une structure est donc un ensemble, ainsi que des fonctions, relations et constantes sur cet ensemble, constituant une interprétation des symboles du langage.

Par abus de notation, on confondra parfois $\mathcal{M}$ avec son *ensemble sous-jacent* M (on notant par exemple $x \in \mathcal{M}$). Pour simplifier les notations, on supprimera parfois les exposants $^{\mathcal{M}}$ lorsqu'il sera clair que l'on parle des fonctions et relations de la structure et non de symboles du langage.

Structure du premier ordre

On voit sans peine comment généraliser la définition précédente pour obtenir des structures pour tout langage du premier ordre. On a toujours un ensemble M non vide. Chaque symbole de fonction n -aire f correspond à une fonction de $f^{\mathcal{M}} : M^n \rightarrow M$, chaque symbole de relation n -aire R correspond à une relation $R^{\mathcal{M}} \subseteq M^n$ et chaque symbole de constante c correspond à une constante $c^{\mathcal{M}} \in M$. La relation d'égalité sera toujours présente et correspondra toujours à « la vraie égalité ».

Une formule, comme par exemple

$$F(x) = \exists y \, y \times (\dot{1} + \dot{1}) = x,$$

n'est qu'une suite de symboles. Une fois fixée une structure $\mathcal{M}$, chaque symbole a vocation à être interprété par l'objet qui lui correspond dans $\mathcal{M}$; de plus, les variables libres pourront également être remplacées par divers *paramètres* — c'est-à-dire divers éléments — de la structure.

Définition 2.16 (Formules paramétrées)

Soit $\mathcal{L}$ un langage, et soit $\mathcal{M}$ une structure pour $\mathcal{L}$. Étant donné une formule $F(x_1, \dots, x_n)$ de $\mathcal{L}$ ayant comme variables libres $x_1, \dots, x_n$, et étant donné $a_1, \dots, a_n \in \mathcal{M}$, l'expression $F(a_1, \dots, a_n)$ désigne une *formule paramétrée* par $a_1, \dots, a_n$: il s'agit de la formule F au sein de laquelle chaque occurrence libre de x_i est remplacée par a_i pour $1 \leq i \leq n$. Une formule paramétrée sans variable libre sera une *formule paramétrée close*. $\diamond$

Une formule paramétrée close n'est plus une simple suite de symboles, mais un énoncé qui sera *vrai* ou *faux* dans la structure considérée.

Remarque

Notons que la notion de formule paramétrée induit celle de termes paramétrés. Ainsi, dans la structure usuelle des entiers pour le langage de l'arithmétique, $(5 + 4) \times 2$ sera un terme paramétré par les éléments 5, 4 et 2 de notre structure.

Nous définissons à présent la satisfaction dans une structure, pour les formules closes ou bien paramétrées dans cette structure.

Définition 2.17. Soit $\mathcal{L}$ un langage, et soit $\mathcal{M} = (M, \dots)$ une structure pour $\mathcal{L}$. On dit qu'une formule $F(x_1, \dots, x_n)$ de $\mathcal{L}$ est *vraie* dans $\mathcal{M}$ pour des paramètres $a_1, \dots, a_n \in \mathcal{M}$, et l'on note $\mathcal{M} \models F(a_1, \dots, a_n)$, si $F(a_1, \dots, a_n)$ est de fait vérifiée dans la structure. La définition se fait formellement par induction sur les formules (dans ce qui suit, $\bar{x}$ et $\bar{a}$ sont des raccourcis pour $x_1, \dots, x_n$ et $a_1, \dots, a_n$).

- ▷ Cas de base : $\mathcal{M} \models R(t_1(\bar{a}), \dots, t_m(\bar{a}))$, où R est un symbole de relation m -aire du langage correspondant à la relation $R^{\mathcal{M}} \subseteq M^m$ et chaque $t_i(\bar{x})$ est un terme, ssi $(t_1^{\mathcal{M}}(\bar{a}), \dots, t_n^{\mathcal{M}}(\bar{a})) \in R^{\mathcal{M}}$, où chaque $t_i^{\mathcal{M}}(\bar{a})$ est l'élément de M obtenu en appliquant les fonctions correspondant à chaque symbole de fonction utilisé dans $t_1(\bar{a})$.
- ▷ Quantification universelle : $\mathcal{M} \models \forall y G(y, \bar{a})$ ssi $\mathcal{M} \models G(b, \bar{a})$ pour tout $b \in M$.
- ▷ Quantification existentielle : $\mathcal{M} \models \exists y G(y, \bar{a})$ ssi il existe $b \in M$ tel que $\mathcal{M} \models G(b, \bar{a})$.

- ▷ Négation : $\mathcal{M} \models \neg G(\bar{a})$ ssi $\mathcal{M} \not\models G(\bar{a})$.
- ▷ Conjonction : $\mathcal{M} \models G_1(\bar{a}) \wedge G_2(\bar{a})$ ssi $\mathcal{M} \models G_1(\bar{a})$ et $\mathcal{M} \models G_2(\bar{a})$.
- ▷ Disjonction : $\mathcal{M} \models G_1(\bar{a}) \vee G_2(\bar{a})$ ssi $\mathcal{M} \models G_1(\bar{a})$ ou $\mathcal{M} \models G_2(\bar{a})$.

La satisfaction de l'implication se déduit de celle de la négation et de la disjonction. $\diamond$

Une fois fixée une théorie, on peut considérer des modèles de cette théorie, c'est-à-dire des structures au sein desquelles chaque axiome de la théorie sera vrai.

Définition 2.18. Soit T une théorie dans un langage $\mathcal{L}$. Une structure $\mathcal{M}$ dans le langage $\mathcal{L}$ est un *modèle* de T si chaque axiome de T est vrai dans $\mathcal{M}$. $\diamond$

Exemple 2.19. ▷ L'ensemble $\mathbb{Z}$ muni des opérations usuelles n'est pas un modèle de l'arithmétique de Peano, car il ne satisfait pas l'axiome (1) : 0 n'a pas de prédécesseur.

- ▷ L'ensemble $2\mathbb{N}$ des entiers pairs, où $\hat{0}$ est interprété par 0 et $\hat{1}$ est interprété par 2, n'est pas non plus un modèle de l'arithmétique de Peano, car il ne vérifie pas l'axiome (7) : $2 \times (2 + 2) \neq (2 \times 2) + 2$.
- ▷ Le modèle par excellence de l'arithmétique de Peano est bien entendu celui des entiers naturels $\mathbb{N}$, munis des opérations et relations usuelles.

Les théories forment l'aspect *syntactique* des mathématiques, les modèles en forment eux l'aspect *sémantique*. Les avantages à réfléchir sur les modèles d'une théorie sont multiples.

En premier lieu, les modèles sont ce sur quoi ont naturellement travaillé les mathématiciens depuis les origines. Aujourd'hui, les mathématiques sont tellement avancées dans l'abstraction que c'est un aspect des choses que l'on perd parfois de vue, mais cette science n'est au départ pas si éloignée que cela de la physique, en ce sens qu'il s'agit d'abord d'un travail *d'observation* de la réalité de certains phénomènes afin d'en extraire les lois logiques qui les régissent. Chaque mathématicien sait par expérience qu'il ne décide pas de la vérité, celle-ci réside parfois bien cachée dans les structures abstraites étudiées, autrement dit dans les modèles. Cette méthodologie d'observation et de recherche basée sur la logique donne son caractère universel et transcendant à la vérité mathématique, constituant en quelque sorte le ciment qui lie la communauté des mathématiciens. Si la syntaxe est bien sûr importante, car elle constitue le langage permettant de transmettre

les mathématiques, *la sémantique précède cette syntaxe*² : c'est de là que partent les intuitions et sur elle que portent les théorèmes.

Il y a enfin un avantage plus prosaïque à l'étude des modèles : par essence, si une formule close F est vraie dans un modèle, alors sa négation $\neg F$ ne peut pas y être vraie. Un modèle est toujours une structure cohérente et complète : chaque formule close y est soit vraie soit fausse, et aucune formule ne peut y être vraie en même temps que sa négation. On peut utiliser cela pour montrer le théorème de cohérence.

Théorème 2.20

Soit T une théorie, et soit $F(x_1, \dots, x_n)$ une formule dans le langage de cette théorie. Si $T \vdash F(x_1, \dots, x_n)$, alors tout modèle de T est aussi un modèle de $\forall x_1 \dots \forall x_n F(x_1, \dots, x_n)$.

PREUVE. On procède sans peine par induction sur la taille d'une démonstration. Supposons que ce soit le cas pour toute démonstration $G_1, \dots, G_n$ dans T . Soit $G_1, \dots, G_{n+1}$ une démonstration dans T . Par hypothèse d'induction, tout modèle de T est modèle de chaque formule $\forall \bar{x} G_i$ pour $i < n+1$, où la notation $\forall \bar{x} G_i$ signifie que l'on quantifie universellement sur chaque variable libre de G_i (quand il y en a). Soit $\mathcal{M}$ un modèle de T . Si G_{n+1} est un axiome de T , alors c'est une formule close, et $\forall \bar{x} G_{n+1}$ est bien évidemment vraie dans $\mathcal{M}$. Si G_{n+1} est un axiome de l'égalité (de type (e) (f) ou (g), ci-dessus), alors $\forall \bar{x} G_{n+1}$ est vraie dans $\mathcal{M}$ par le fait que l'égalité de $\mathcal{M}$ est toujours la vraie égalité.

Si G_{n+1} est une tautologie de la logique propositionnelle ou un axiome de type (a)(b)(c) ou (d) du calcul des prédicats, alors $\forall \bar{x} G_{n+1}$ est vraie par définition de la satisfaction dans un modèle (les détails sont laissés au lecteur ; notons que pour (d) on utilise le fait que le modèle soit non vide).

Si G_{n+1} est obtenue par généralisation sur G_i pour $i < n+1$ — en particulier, G_{n+1} est de la forme $\forall y G_i$ —, alors $\mathcal{M}$, vérifiant $\forall \bar{x} G_i$, vérifie également $\forall \bar{x} G_{n+1}$ (qui est en fait la même formule). Enfin, si G_{n+1} est obtenu par Modus Ponens via $G_i \rightarrow G_{n+1}$ et G_i pour $i < n+1$, alors $\mathcal{M}$ vérifie $\forall \bar{x} G_i$ et $\forall \bar{x} (G_i \rightarrow G_{n+1})$. Par définition de la satisfaction, $\mathcal{M}$ vérifie donc $\forall \bar{x} G_{n+1}$. ■

Le théorème précédent peut se résumer de la manière suivante : « on ne peut démontrer que des choses vraies ». C'est une bonne nouvelle, de laquelle on peut déduire notre théorème de cohérence.

2. Aphorisme cher au professeur René Cori, qui enseigna aux auteurs de ce livre les principes du présent chapitre.

Corollaire 2.21 (Théorème de cohérence)

Si un système axiomatique admet un modèle, alors il est cohérent.

PREUVE. On le montre par contraposée. Si $T \vdash F \wedge \neg F$ pour une formule close F , alors tout modèle de T est modèle de $F \wedge \neg F$. Comme il n'existe aucun modèle de $F \wedge \neg F$, alors T n'a pas de modèle. ■

On vérifie sans peine l'existence d'un modèle des axiomes de la logique : l'ensemble $\{1\}$ avec la relation d'égalité $1 = 1$. Cela montre que les axiomes de la logique sont cohérents et ne peuvent démontrer $\perp$.

On vérifie aussi sans peine l'existence d'un modèle pour l'arithmétique de Peano, à savoir $\mathbb{N}$ muni des fonctions usuelles d'addition et de multiplication, ainsi que des relations usuelles $<$ et $=$ sur les entiers. Voilà une deuxième bonne nouvelle, les axiomes de l'arithmétique de Peano sont eux aussi cohérents. Nous examinerons le sens de cette affirmation un peu plus loin, notamment à la lumière du deuxième théorème d'incomplétude de Gödel et de ses implications.

2.5. Modèles et théorème de complétude

Si le système de déduction à la Hilbert que nous avons donné — avec ses axiomes de la logique et ses règles de déduction — est bien cohérent, comment savoir en revanche qu'il est suffisamment puissant ? Après tout, les deux règles d'inférence semblent former un outil de travail bien maigre. Peut-on réellement tout démontrer avec ce système ? Nous allons voir que c'est le cas, via un théorème démontré, par Gödel dans sa thèse de doctorat, qui peut être vu comme la réciproque du théorème de cohérence : tout ce qui est universellement vrai est démontrable.

Théorème 2.22 (Théorème de complétude de Gödel)

Soit T une théorie dans un langage dénombrable. Si T est cohérente, alors T a un modèle.

Notons que le théorème de complétude pour des langages indénombrables peut aussi se démontrer en utilisant l'axiome du choix. Nous n'en montrons que la version dénombrable, qui nous suffira amplement. Le théorème de complétude de Gödel est plus difficile à montrer que le théorème de cohérence, lequel est une simple vérification de routine. Il s'agit ici de construire un modèle d'une théorie T , à partir du simple fait que $T \not\vdash \perp$. L'idée de la preuve passe par la création d'une théorie dite *complète*.

Définition 2.23. Une théorie T est dite *complète* si $T \vdash F$ ou $T \vdash \neg F$ pour toute formule close F dans le langage de T . $\diamond$

Proposition 2.24. Soit $\mathcal{L}$ un langage dénombrable. Toute théorie de $\mathcal{L}$ cohérente peut être étendue en une théorie complète et cohérente. $\star$

La preuve de la proposition ci-dessus utilise le lemme 2.10 que nous reformulons ici avec la notation $\vdash$ introduite depuis.

Lemme (Lemme 2.10 de déduction). Soit $T \cup \{F\}$ une théorie et G une formule. Supposons $T \cup \{F\} \vdash G$. Alors, $T \vdash F \rightarrow G$. $\star$

PREUVE DE LA PROPOSITION 2.24. Étant donné une théorie T cohérente dans un langage dénombrable, on en construit inductivement une extension T' complète et cohérente. Soit $T_0 = T$. À l'étape n , supposons qu'une théorie cohérente T_n soit définie. Soit F_n la n -ième formule close de notre langage. Si $T_n \vdash F_n$, on définit $T_{n+1} = T_n \cup \{F_n\}$. Si $T_n \vdash \neg F_n$, on définit $T_{n+1} = T_n \cup \{\neg F_n\}$. Dans ces deux premiers cas, la cohérence de T_{n+1} découle de la cohérence de T_n et du lemme de déduction.

Si jamais T_n ne prouve ni F_n ni $\neg F_n$, alors on définit $T_{n+1} = T_n \cup \{F_n\}$. Supposons par l'absurde que $T_n \cup \{F_n\} \vdash \perp$. Alors, d'après le lemme de déduction, $T_n \vdash F_n \rightarrow \perp$, et donc $T_n \vdash \neg F_n$ (par contraposée et Modus Ponens), ce qui contredit le fait que T ne prouve pas $\neg F_n$. Notons que l'on pourrait tout aussi bien définir $T_{n+1} = T_n \cup \{\neg F_n\}$.

On définit enfin $T' = \bigcup_n T_n$. La cohérence de T' vient alors du fait qu'une démonstration dans une théorie n'utilise qu'un nombre fini de ses axiomes : si $T' \vdash \perp$, alors il existe forcément n tel que $T_n \vdash \perp$. Comme chaque théorie T_n est cohérente, alors T' doit être cohérente. $\blacksquare$

PREUVE DU THÉORÈME DE COMPLÉTUDE. La preuve que nous donnons est due à Léon Henkin [85], et repose sur la création d'une théorie complète, cohérente, et possédant ce que l'on appelle des *témoins de Henkin*. Nous allons d'abord montrer l'affirmation suivante.

« Soit T une théorie dans un langage $\mathcal{L}$, et soit c un symbole de constante qui n'apparaît pas dans $\mathcal{L}$. Supposons $T \cup \{\exists x F(x) \rightarrow F(c)\} \vdash \perp$. On a alors $T \vdash \perp$. »

Comme $T \cup \{\exists x F(x) \rightarrow F(c)\} \vdash \perp$, on a alors, par le lemme de déduction, contraposée et Modus Ponens, $T \vdash \exists x F(x) \wedge \neg F(c)$. En particulier, $T \vdash \neg F(c)$. Soit à présent une variable z qui n'intervient pas dans la preuve de $\neg F(c)$. Comme la constante c n'apparaît pas dans la théorie T , elle ne peut être introduite dans la démonstration que par un axiome de la logique. Chacun de ces axiomes reste valide en remplaçant c par z . On

laisse au lecteur le soin de vérifier que si l'on remplace c par z pour chaque étape de la démonstration, on obtient une démonstration valide de $\neg F(z)$. Par la règle de généralisation, on obtient finalement $T \vdash \forall x \neg F(x)$. Comme également $T \vdash \exists x F(x)$, alors $T \vdash \perp$.

Passons à présent à la preuve du théorème de complétude. Soit T une théorie dans un langage dénombrable $\mathcal{L}$. Montrons comment construire un modèle. On définit $T_0 = T$ et $\mathcal{L}_0 = \mathcal{L}$. À l'étape $n \in \mathbb{N}$, supposons que l'on a défini une théorie cohérente T_{2n} dans un langage $\mathcal{L}_{2n}$. Soit $\mathcal{L}_{2n+1}$ le langage $\mathcal{L}_{2n}$ auquel on ajoute un nouveau symbole de constante c_G pour toute formule G de T_{2n} de la forme $\exists x F(x)$. Soit T_{2n+1} la théorie T_{2n} à laquelle on ajoute les énoncés $\exists x F(x) \rightarrow F(c_G)$ pour tout énoncé G de T_{2n} de la forme $\exists x F(x)$. Notons que par l'affirmation ci-dessus, comme T_{2n} est cohérente, à chaque ajout d'axiome de la forme $\exists x F(x) \rightarrow F(c_G)$, la théorie reste cohérente. Donc, T_{2n+1} est cohérente. Finalement, soit $\mathcal{L}_{2n+2}$ égal à $\mathcal{L}_{2n+1}$, et en utilisant la proposition 2.24, soit T_{2n+2} la complétion de T_{2n+1} pour le langage $\mathcal{L}_{2n+2}$. Soient $\mathcal{T}_\omega = \bigcup_n T_n$ et $\mathcal{L}_\omega = \bigcup_n \mathcal{L}_n$. Notons que $\mathcal{T}_\omega$ est une théorie complète et cohérente dans le langage $\mathcal{L}_\omega$, qui contient de plus un énoncé de la forme $\exists x F(x) \rightarrow F(c)$ pour chacun des énoncés de la forme $\exists x F(x)$ de T_ω , où c est un symbole de constante de $\mathcal{L}_\omega$. Les fameux *témoins de Henkin* sont les nouveaux symboles de constantes ainsi introduits.

L'ensemble sous-jacent M de notre modèle $\mathcal{M}$ est l'ensemble des termes clos de $\mathcal{L}_\omega$, quotienté par la relation d'égalité. Formellement, si $(t_n)_{n \in \mathbb{N}}$ est la liste des termes clos de $\mathcal{L}_\omega$, alors $M = \{t_n : \forall i < n \ (\neg t_i = t_n) \in T_\omega\}$.

Notons que les symboles de fonctions de $\mathcal{L}$ ont une interprétation claire dans M . Par exemple, si f est un symbole de fonction unaire de $\mathcal{L}$, alors sa fonction correspondante $f^{\mathcal{M}} : M \rightarrow M$ est définie par $f^{\mathcal{M}}(t) = q$ pour q l'élément de M qui est égal au terme clos $f(t) \in \mathcal{L}_\omega$, c'est-à-dire tel que $(f(t) = q) \in T_\omega$.

La théorie T_ω étant complète et cohérente, pour tout symbole de relation m -aire R de $\mathcal{L}$ et tout élément $t_1, \dots, t_m \in M$, exactement un des énoncés parmi $R(t_1, \dots, t_m)$ ou $\neg R(t_1, \dots, t_m)$ est dans T_ω . Cela induit une interprétation $R^{\mathcal{M}}$ du symbole de relation R de $\mathcal{L}$ dans $\mathcal{M}$.

Il reste à montrer par induction sur les formules que tous les énoncés de T_ω sont satisfaits dans $\mathcal{M}$ (et donc aussi ceux de T). Sans perte de généralité, on ne traite que les formules sous forme prénexe et où le symbole de négation n'apparaît que devant les formules atomiques. Par définition des relations dans $\mathcal{M}$, c'est bien le cas pour les formules atomiques et leurs négations. Si $F_1 \wedge F_2 \in T_\omega$, alors comme T_ω est complète $F_1, F_2 \in T_\omega$. Par hypothèse d'induction, $\mathcal{M} \models F_1$ et $\mathcal{M} \models F_2$, et donc $\mathcal{M} \models F_1 \wedge F_2$.

Si $F_1 \vee F_2 \in T_\omega$, alors comme T est complète $F_1 \in T_\omega$ ou $F_2 \in T_\omega$ (si non, par complétude, $\neg F_1, \neg F_2 \in T_\omega$, ce qui contredit $F_1 \vee F_2$). Par hypothèse d'induction, $\mathcal{M} \models F_1 \vee F_2$. Si $\forall x F(x) \in T_\omega$, alors comme T_ω est complète $F(t)$ est dans T_ω pour tout terme clos t de $\mathcal{L}_\omega$, et donc tout élément de M . Par hypothèse d'induction, $\mathcal{M} \models F(t)$ pour tout $t \in M$, et donc $\mathcal{M} \models \forall x F(x)$. Si $\exists x F(x) \in T_\omega$, alors $\exists x F(x) \rightarrow F(c) \in T_\omega$ pour un symbole de constante $c \in \mathcal{L}_\omega$. En particulier, $F(c) \in T_\omega$ par Modus Ponens. Soit $t \in M$ un terme clos de $\mathcal{L}_\omega$ tel que $(t = c) \in T_\omega$. Par les axiomes de l'égalité, on a $F(t)$. Par hypothèse d'induction, $\mathcal{M} \models F(t)$. Donc, $\mathcal{M} \models \exists x F(x)$. ■

Le théorème de complétude est souvent utilisé sous la forme du corollaire suivant.

Corollaire 2.26

Si tout modèle d'une théorie T est modèle d'une formule F , alors $T \vdash F$.

PREUVE. Si l'on a $T \cup \{\neg F\} \vdash \perp$, alors $T \vdash \neg F \rightarrow \perp$ par le lemme de déduction, et donc $T \vdash F$.

Supposons à présent $T \not\vdash F$; alors, par la ligne ci-dessus $T \cup \{\neg F\} \not\vdash \perp$. D'après le théorème de complétude, il existe donc un modèle de $T \cup \{\neg F\}$, c'est-à-dire un modèle de T qui ne soit pas modèle de F . ■

Le théorème de complétude implique en particulier que l'on ne peut pas faire mieux que le système à la Hilbert que nous avons présenté : supposons que dans ce système les axiomes seuls de la logique ne suffisent pas à démontrer une formule F , autrement dit $\not\vdash F$ (à partir d'une théorie vide). Supposons qu'un système de démonstration plus puissant et cohérent existe tel que $\vdash^* F$, où $\vdash^*$ est la notion de preuve dans ce système. Alors également, $\vdash^* \neg F \rightarrow \perp$, et donc $\neg F \vdash^* \perp$. À présent, comme $\not\vdash F$, il existe d'après le théorème de complétude un modèle de $\neg F$, et notre modèle est donc d'après le théorème de cohérence pour $\vdash^*$ un modèle de $\perp$, ce qui est impossible.

3. Théorèmes d'incomplétudes de Gödel

Nous rentrons à présent dans le vif du sujet, via les théorèmes d'incomplétudes de Gödel, qui reposent entre autres choses sur un codage des ensembles calculatoirement énumérables par des formules de l'arithmétique.

3.1. Formules de l'arithmétique de Peano

Nous avons défini dans le chapitre 5 une hiérarchie de complexité sur les ensembles, appelée *hiérarchie arithmétique*. Nous allons maintenant donner tout son sens à cette appellation en définissant une hiérarchie syntaxique des formules de l'arithmétique, qui coïncide avec la hiérarchie arithmétique, dans le sens où un ensemble A est Σ_n^0 ssi il est définissable par une formule Σ_n de l'arithmétique de Peano.

Définition 3.1. Une formule de l'arithmétique de Peano est Δ_0 si les quantifications qu'elle comporte sont toutes bornées, c'est-à-dire de la forme $\exists x < t$ et $\forall x < t$, avec t un terme où la variable x n'apparaît pas librement. Notons que les formules $\exists x < t F(x)$ et $\forall x < t F(x)$ se traduisent respectivement par $\exists x(x < t \wedge F(x))$ et $\forall x(x < t \rightarrow F(x))$. $\diamond$

Étant donné $F(x_1, \dots, x_n)$ une formule Δ_0 de l'arithmétique de Peano, la restriction sur les quantifications fait de l'ensemble

$$\{(x_1, \dots, x_n) \in \mathbb{N} : F(x_1, \dots, x_n)\}$$

un ensemble calculable. Cela découle par exemple directement de la clôture des prédicats primitifs récurrents par conjonction, disjonction et quantification bornée (voir l'exemple 6-3.16 et l'exercice 6-3.19). On définit une hiérarchie de complexité sur les formules de l'arithmétique de Peano analogue à la hiérarchie arithmétique de la définition 5-1.1.

Définition 3.2

1. Une formule $F(x_1, \dots, x_m)$ de l'arithmétique de Peano est Σ_n si

$$F(x_1, \dots, x_m) = \overbrace{\exists y_1 \forall y_2 \dots Q y_n}^{n \text{ quantificateurs}} G(x_1, \dots, x_m, y_1, \dots, y_n)$$

pour $G(x_1, \dots, x_m, y_1, \dots, y_n)$ une formule Δ_0 , où Q vaut $\exists$ si n est impair, et $\forall$ si n est pair.

2. Une formule $F(x_1, \dots, x_m)$ de l'arithmétique de Peano est Π_n si

$$F(x_1, \dots, x_m) = \overbrace{\forall y_1 \exists y_2 \dots Q y_n}^{n \text{ quantificateurs}} G(x_1, \dots, x_m, y_1, \dots, y_n)$$

pour $G(x_1, \dots, x_m, y_1, \dots, y_n)$ une formule Δ_0 , où Q vaut $\forall$ si n est impair, et $\exists$ si n est pair. $\diamond$

Nous verrons que les ensembles définissables par une formule Σ_n (resp. Π_n) de l'arithmétique de Peano coïncident avec les ensembles Σ_n^0 (resp. Π_n^0) de la définition 5-1.1. Nous commençons pour cela par montrer quelques propriétés de clôture analogues à celles des propositions 5-1.6 à 5-1.9.

Proposition 3.3. Soient $F(\bar{a}, x)$, $F_1(\bar{a}, x)$ et $F_2(\bar{a}, x)$ des formules Σ_n (resp Π_n). Alors, chacune des formules suivantes est prouvablement équivalente (en usant des axiomes de l'arithmétique) à une formule Σ_n (resp. Π_n) :

- ▷ $F_1(\bar{a}, x) \wedge F_2(\bar{a}, x)$, $F_1(\bar{a}, x) \vee F_2(\bar{a}, x)$;
- ▷ $\exists x < b \ F(\bar{a}, x)$, $\forall x < b \ F(\bar{a}, x)$;
- ▷ $\exists x \ F(\bar{a}, x)$ (resp. $\forall x \ F(\bar{a}, x)$). ★

PREUVE. Soient les équivalences suivantes :

$$F(\bar{a}, x) \equiv \exists y G(\bar{a}, x, y), \quad F_1(\bar{a}, x) \equiv \exists y G_1(\bar{a}, x, y) \quad \text{et} \quad F_2(\bar{a}, x) \equiv \exists y G_2(\bar{a}, x, y).$$

Il s'en déduit alors les nouvelles équivalences suivantes :

$$\begin{aligned} F_1(\bar{a}, x) \wedge F_2(\bar{a}, x) &\leftrightarrow \exists y \exists y_1, y_2 < y \ (G_1(\bar{a}, x, y_1) \wedge G_2(\bar{a}, x, y_2)) \\ F_1(\bar{a}, x) \vee F_2(\bar{a}, x) &\leftrightarrow \exists y \ (G_1(\bar{a}, x, y) \vee G_2(\bar{a}, x, y)) \\ \exists x < b \ F(\bar{a}, x) &\leftrightarrow \exists y \exists x < b \ G(\bar{a}, x, y) \\ \forall x < b \ F(\bar{a}, x) &\leftrightarrow \exists z \forall x < b \ \exists y < z \ G(\bar{a}, x, y) \\ \exists x \ F(\bar{a}, x) &\leftrightarrow \exists z \exists x < z \ \exists y < z \ G(\bar{a}, x, y). \end{aligned}$$

À présent, si F, F_1, F_2 sont Σ_1 avec $G, G_1, G_2 \Delta_0$, les équivalences ci-dessus montrent la proposition pour le cas Σ_1 . Par passage à la négation, les équivalences tiennent aussi pour le cas Π_1 . Supposons la proposition vraie pour les cas Σ_n et Π_n . Alors, les équivalences ci-dessus pour F, F_1, F_2 des formules Σ_{n+1} , avec $G, G_1, G_2 \Pi_n$, impliquent — en utilisant les hypothèses d'induction sur G, G_1, G_2 — la proposition pour le cas Σ_{n+1} . Par négation, la proposition est vraie pour le cas Π_{n+1} . ■

Notons que la quatrième équivalence dans la preuve ci-dessus, en l'occurrence l'équivalence $\forall x < b \ F(\bar{a}, x) \leftrightarrow \exists z \forall x < b \ \exists y < z \ G(\bar{a}, x, y)$, est la moins triviale de toutes : les quatre autres utilisent simplement le fait que deux entiers ont toujours un majorant, alors que celle-ci requiert l'utilisation de l'induction. C'est quelque chose que nous étudierons en détail dans la section 23-3.

Passons à présent à l'équivalence annoncée. Étant donné $\exists y \ F(x_1, \dots, x_n, y)$ une formule Σ_1 de l'arithmétique de Peano avec F qui est Δ_0 , l'ensemble

$$\{(x_1, \dots, x_n) \in \mathbb{N} : \exists y \ F(x_1, \dots, x_n, y)\}$$

est un ensemble calculatoirement énumérable : on teste petit à petit la formule F sur tous les $(n+1)$ -uplets $x_1, \dots, x_n, y$ et, quand on en trouve un pour lequel F est vraie, on énumère $(x_1, \dots, x_n)$. Gödel a montré que tout ensemble calculatoirement énumérable pouvait en fait être représenté sous cette forme.

Théorème 3.4 (Gödel)

Un ensemble d'entiers $A \subseteq \mathbb{N}$ est c.e. si, et seulement si, il existe $F(n)$ une formule Σ_1 de $\mathcal{L}_{PA}$ telle que $n \in A$ ssi $\mathbb{N} \models F(n)$.

Nous allons montrer le théorème 3.4 en nous basant sur le modèle des fonctions générales récurives, qui coïncident, comme nous l'avons vu dans le chapitre 6, avec les fonctions calculables. Nous allons montrer que toute fonction partielle générale réursive $f : \mathbb{N}^k \rightarrow \mathbb{N}$ est représentée par une formule Σ_1 de l'arithmétique $F(n_1, \dots, n_k)$, c'est-à-dire

$$\{(\bar{n}, r) \in \mathbb{N}^{k+1} : f(\bar{n}) \downarrow = r\} = \{(\bar{n}, r) \in \mathbb{N}^{k+1} : \mathbb{N} \models F(\bar{n}, r)\}.$$

Comme tout ensemble c.e. est le domaine d'une fonction partielle, cela démontre le théorème. La difficulté principale se trouve dans la gestion du schéma de récursion primitive, pour lequel nous avons besoin de coder des listes d'entiers par des formules de l'arithmétique. Gödel a recours pour cela à une utilisation astucieuse d'un résultat d'arithmétique modulaire : le théorème des restes chinois.

Lemme 3.5 (Théorème des restes chinois). Soit une suite quelconque d'entiers $(a_0, \dots, a_n)$, et soit $(p_0, \dots, p_n)$ une suite d'entiers deux à deux premiers entre eux avec $p_i \geq a_i$ pour $i \leq n$. Alors, il existe un entier b tel que a_i est le reste de la division euclidienne de b par p_i pour tout i . ★

Le théorème des restes chinois va permettre à Gödel de coder des listes d'entiers de taille arbitraire. Notons que ce n'est pas la seule manière d'établir un système de codage/décodage des listes par des formules de l'arithmétique, et nous en verrons une autre dans la section 23-4.

Lemme 3.6 (Fonction β de Gödel). Il existe une fonction

$$\beta : \mathbb{N}^3 \rightarrow \mathbb{N}$$

représentée par une formule Δ_0 telle que pour tout n et toute suite d'entiers $(a_0, \dots, a_n)$ il existe des entiers $a, b \in \mathbb{N}$ pour lesquels $\beta(a, b, i) = a_i$ pour tout $i \leq n$. ★

PREUVE. La formule $B(a, b, i, r)$ qui représente β est la suivante : « r est le reste de la division euclidienne de b par $a(i+1)+1$ ». La formule est bien Δ_0 : $r < a \times (i+1) + 1 \wedge \exists c < b \ c \times (a \times (i+1) + 1) + r = b$. Soit une suite d'entiers $(a_0, \dots, a_n)$. Montrons l'existence d'entiers a, b tels que cette formule définit bien la fonction $(a, b, i) \mapsto a_i$.

Soit m tel que $m > \max\{a_i : i \leq n\}$ et $m > n$. Posons $a = m!$. Nous allons utiliser le théorème des restes chinois avec $p_i = a(i+1) + 1$. Montrons que ces nombres sont deux à deux premiers.

Supposons par l'absurde qu'un nombre premier p divise p_i et p_j avec $i < j$. Alors aussi, p divise $p_j - p_i = a(j - i)$. Comme p est premier, alors p divise a ou p divise $j - i$. Comme $a = m!$ avec $m > n \geq j > i$ alors $j - i$ divise a et donc dans tous les cas p divise a . Donc, p divise $a(i + 1)$. Comme p divise également $a(i + 1) + 1$, alors, p divise $a(i + 1) + 1 - a(i + 1) = 1$, ce qui est une contradiction. Les nombres p_i sont donc premiers entre eux.

On a bien $p_i = a(i + 1) + 1 > m > a_i$ pour tout i . D'après le théorème des restes chinois, il existe un entier b tel que pour tout i l'entier a_i est le reste de la division euclidienne de b par $a(i + 1) + 1$. ■

PREUVE DU THÉORÈME 3.4. On montre que toute fonction générale réursive partielle est représentée par une formule Σ_1 de l'arithmétique. On laisse au lecteur le soin de montrer que c'est bien le cas pour les fonctions de base (projections, fonctions constantes et fonction successeur). On utilise sans le mentionner pour chacun des trois schémas à venir la proposition 3.3 afin d'obtenir une formule Σ_1 qui représente notre fonction.

Schéma de composition. Soit

$$f(\bar{x}) = g(h_1(\bar{x}), \dots, h_k(\bar{x}))$$

pour des fonctions $g, h_1, \dots, h_k$ représentées par des formules $G, H_1, \dots, H_k$. Alors, f est représentée par la formule

$$F(\bar{x}, r) \equiv \exists y_1, \dots, y_k \ H_1(\bar{x}, y_1) \wedge \dots \wedge H_k(\bar{x}, y_k) \wedge G(y_1, \dots, y_k, r).$$

Schéma de minimisation. Soit

$$f(\bar{x}) = \min\{a \in \mathbb{N} : \forall i \leq a \ g(\bar{x}, i) \downarrow \wedge g(\bar{x}, a) = 0\}$$

pour g représentée par une formule G . Alors, f est représentée par la formule :

$$F(\bar{x}, a) \equiv G(\bar{x}, a, 0) \wedge \forall i < a \ \exists r \neq 0 \ G(\bar{x}, i, r).$$

Schéma de récursion primitive. C'est ici que nous aurons besoin de la fonction β de Gödel, représentée par la formule B . Soit f défini par :

$$\begin{aligned} f(\bar{x}, 0) &= g(\bar{x}) \\ f(\bar{x}, n + 1) &= h(\bar{x}, n, f(\bar{x}, n)) \end{aligned}$$

pour des fonctions g, h représentées par des formules G, H . Alors, f est représentée par la formule $F(\bar{x}, n, r)$ suivante :

$$\begin{aligned} \exists a, b \quad & B(a, b, n, r) \wedge \exists a_0 \ (B(a, b, 0, a_0) \wedge G(\bar{x}, a_0)) \wedge \forall i < n \\ & \exists a_i, a_{i+1} \ (B(a, b, i, a_i) \wedge B(a, b, i + 1, a_{i+1}) \wedge H(\bar{x}, i, a_i, a_{i+1})). \end{aligned}$$

Afin de voir que f est bien représentée par F , il nous faut remarquer que B tel que définie dans le lemme précédent est toujours une formule fonctionnelle.

En effet, quelles que soient les valeurs de a, b, i , il y a toujours au plus un élément r tel que $B(a, b, i, r)$ est vrai. Ainsi, si la formule $F(\bar{x}, n, r)$ est vérifiée, il existe bien une suite $a_0, \dots, a_n$ telle que $g(\bar{x}) = a_0$ et $h(\bar{x}, i, a_i) = a_{i+1}$, avec $a_n = r$, le résultat de $f(\bar{x}, n)$. D'après le lemme précédent, il existe bien pour tout n des entiers a, b qui codent via la fonction β la suite de valeurs $(f(\bar{x}, 0), f(\bar{x}, 1), \dots, f(\bar{x}, n))$.

La formule F représente donc bien la fonction f . ■

On montre facilement à partir du théorème 3.4 qu'un ensemble d'entiers est Σ_n^0 (resp. Π_n^0) si, et seulement si, il est décrit par une formule Σ_n (resp. Π_n) de l'arithmétique.

Codage des suites finies

Il existe de nombreuses manières de coder des suites finies d'entiers à l'aide d'entiers naturels. La plupart de ces techniques ont recours à des fonctions primitives récursives, ce qui demande de montrer au préalable qu'elles sont représentables par des formules simples de l'arithmétique. Le théorème des restes chinois permet de réaliser un codage simple des suites finies à base de la division euclidienne, qui s'exprime par un prédicat Δ_0 immédiat (voir le lemme 3.6).

3.2. Démonstrations et calcul

Une démonstration dans notre système de déduction à la Hilbert repose sur un système de règles bien précis, et il est aisé de créer un programme informatique qui prend en paramètre une démonstration — via un codage approprié — et qui vérifie en un temps fini si la démonstration est valide ou non. En effet, les règles d'inférence sont claires, quant aux axiomes de la logique, nous avons les tautologies du calcul propositionnel, quatre schémas d'axiomes pour le calcul des prédicats, et trois schémas d'axiomes pour l'égalité. Il est aisé de vérifier si une phrase du calcul propositionnel — par exemple, de la forme $(F \rightarrow G) \leftrightarrow (\neg G \rightarrow \neg F)$ — est une tautologie, en s'assurant que la phrase est toujours vraie pour n'importe quelle valeur de vérité pour F et G . Il est également facile de vérifier si une phrase correspond à l'un des schémas d'axiomes de l'égalité ou du calcul des prédicats. On en déduit le théorème suivant.

Théorème 3.7 (Gödel)

Étant donné une théorie calculatoirement énumérable T dans le langage $\mathcal{L}_{PA}$, l'ensemble des formules F telles que $T \vdash F$ est calculatoirement énumérable.

Il suffit en effet de faire une recherche sur toutes les démonstrations possibles à partir des axiomes de T et de lister toutes les formules qu'elles démontrent.

L'objectif de Hilbert était de montrer que toute vérité arithmétique était prouvable, l'arithmétique de Peano étant supposément un système suffisant pour le faire. Si tel était le cas, alors on pourrait créer un algorithme permettant de décider si un énoncé mathématique F est prouvable ou réfutable : il suffirait de lister tous les énoncés prouvés par l'arithmétique de Peano jusqu'à trouver F ou $\neg F$.

Gödel a montré que ce n'était pas possible, ni pour l'arithmétique de Peano, ni pour aucune théorie calculatoirement énumérable et cohérente contenant l'arithmétique de Peano (il fallut toutefois l'aide de Rosser pour cette dernière étape). Nous commençons par prouver un lemme qui nous aidera dans la suite.

Lemme 3.8. Si $\mathbb{N} \models F$ où F est une formule close Σ_1 , alors $\text{PA} \vdash F$. ★

PREUVE. La formule F est de la forme $\exists x_1 \dots \exists x_n G(x_1, \dots, x_n)$ pour G une formule Δ_0 . Si F est vraie dans $\mathbb{N}$, alors il existe des entiers $a_1, \dots, a_n \in \mathbb{N}$ tels que $G(a_1, \dots, a_n)$ est vrai dans $\mathbb{N}$. On montre facilement par induction que si une formule Δ_0 et paramétrée dans $\mathbb{N}$ est vraie dans $\mathbb{N}$, alors elle est démontrable dans PA. À titre informel, pour une quantification existentielle bornée, il suffit de prendre un entier témoin pour cette quantification et de montrer la formule avec ce témoin ; et, pour une quantification universelle bornée par un entier a , il suffit de montrer que la formule est vraie pour chaque entier inférieur à a . ■

Nous avons à présent les ingrédients nécessaires pour montrer le premier théorème d'incomplétude.

Théorème 3.9 (Premier théorème d'incomplétude de Gödel)

Soit $T \supseteq \text{PA}$ une théorie c.e. cohérente, telle que si T démontre une formule Σ_1 , alors $\mathbb{N}$ est modèle de cette formule. Alors, il existe F une formule Σ_1 telle que $T \not\vdash F$ et $T \not\vdash \neg F$.

PREUVE. Soit $(\Phi_e)_{e \in \mathbb{N}}$ une énumération des fonctions calculables. D'après le théorème 3.4, il existe $F(e)$ une formule Σ_1 de $\mathcal{L}_{\text{PA}}$ telle que

$$\{e \in \mathbb{N} : \mathbb{N} \models F(e)\} = \{e \in \mathbb{N} : \exists t \Phi_e(e)[t] \downarrow\}.$$

Supposons que pour tout e on ait $T \vdash F(e)$ ou $T \vdash \neg F(e)$. Notons que si $\exists t \Phi_e(e)[t] \downarrow$, alors $\mathbb{N} \models F(e)$, et donc $\text{PA} \vdash F(e)$ d'après le lemme 3.8. Comme $\text{PA} \subseteq T$, on a aussi $T \vdash F(e)$.

À présent, si $\forall t \Phi_e(e)[t] \uparrow$, alors on a $\mathbb{N} \models \neg F(e)$. D'après notre hypothèse, on ne peut avoir $T \vdash F(e)$, car on aurait alors $\mathbb{N} \models F(e)$ ce qui contredit $\mathbb{N} \models \neg F(e)$. Comme T est complète, on a donc $T \vdash \neg F(e)$.

Il s'ensuit que l'ensemble c. e. $\{e \in \mathbb{N} : T \vdash \neg F(e)\}$ coïncide avec l'ensemble $\{e \in \mathbb{N} : \forall t \Phi_e(e)[t] \uparrow\}$, ce qui fait du complémentaire de l'arrêt un ensemble c. e. On a là une contradiction. ■

Nous remarquons plusieurs choses. En premier lieu, nous avons été obligés de nous restreindre aux théories T dites 1-cohérentes, c'est-à-dire ne démontrant pas de formules Σ_1 fausses dans $\mathbb{N}$. Nous verrons bientôt qu'il existe bien d'autres modèles possibles que $\mathbb{N}$ pour PA, et autant de théories non 1-cohérentes que l'on souhaite. Il s'agit donc d'une restriction bien ennuyeuse.

Ensuite, la preuve montre en substance qu'une théorie complète et 1-cohérente permet de calculer $\emptyset'$. Nous avons déjà vu que n'importe quel degré PA permet de calculer une extension complète et cohérente de PA, et comme il y a des degrés PA qui ne calculent pas $\emptyset'$, il n'y a en fait aucun espoir de montrer le théorème de Gödel en réduisant le problème de l'arrêt à toute théorie complète et cohérente qui étend PA. L'astuce pour se sortir de cette situation fut trouvée par Rosser, l'idée étant en substance d'utiliser le fait que si une théorie démontre $\exists t \Phi_e(e)[t] \downarrow = 0$ (que ce soit vrai dans $\mathbb{N}$ ou pas), alors elle ne peut pas montrer dans le même temps $\exists t \Phi_e(e)[t] \downarrow = 1$, à supposer bien entendu que les formules Σ_1 permettant de parler des fonctions calculables intègrent bien le fait qu'une fonction a au plus une valeur sur son entrée, ce qui est bien le cas en pratique.

Théorème 3.10 (Théorème d'incomplétude de Gödel-Rosser)

Soit $T \supseteq \text{PA}$ une théorie c. e. et cohérente. Alors, il existe F une formule Σ_1 telle que $T \not\vdash F$ et $T \not\vdash \neg F$.

PREUVE. Soit $(\Phi_e)_{e \in \mathbb{N}}$ une énumération des fonctions calculables. Supposons par l'absurde que l'on ait $T \vdash F$ ou $T \vdash \neg F$ pour toute formule $F \Sigma_1$. Nous allons alors calculer une fonction totale $f : \mathbb{N} \rightarrow \{0, 1\}$ qui est DNC₂, c'est-à-dire telle que pour tout entier e on ait $\Phi_e(e) \downarrow$ implique $f(e) \neq \Phi_e(e)$. Notons qu'il s'agit alors d'une contradiction : si f est calculable, alors il existe un code e tel que $\Phi_e(n) \downarrow = f(n)$ pour tout n , et donc en particulier tel que $\Phi_e(e) \downarrow = f(e)$.

Les ensembles $\{\exists t \Phi_e(e)[t] \downarrow = 0\}$ et $\{\exists t \Phi_e(e)[t] \downarrow = 1\}$ sont c. e., et donc d'après le théorème 3.4 il existe $F_0(e)$ et $F_1(e)$ des formules Σ_1 telles que

$$\begin{aligned} \{e \in \mathbb{N} : \mathbb{N} \models F_0(e)\} &= \{e \in \mathbb{N} : \exists t \Phi_e(e)[t] \downarrow = 0\} \\ \{e \in \mathbb{N} : \mathbb{N} \models F_1(e)\} &= \{e \in \mathbb{N} : \exists t \Phi_e(e)[t] \downarrow = 1\}. \end{aligned}$$

Notons que PA démontre aussi $F_0(e) \rightarrow \neg F_1(e)$ et $F_1(e) \rightarrow \neg F_0(e)$: autrement dit, $e \mapsto \Phi_e(e)$ est une fonction partielle qui ne peut avoir à la fois 0 et 1 comme valeur pour le même élément.

Pour calculer la valeur de $f(e)$, on énumère alors à l'aide du théorème 3.7 toutes les formules démontrées par T jusqu'à tomber sur $F_0(e)$ ou $\neg F_0(e)$. Par hypothèse, une de ces deux éventualités arrive forcément. Si $T \vdash F_0(e)$, alors on définit $f(e) = 1$. Sinon, on définit $f(e) = 0$.

Montrons que notre fonction a la propriété attendue.

Si $\exists t \Phi_e(e)[t] \downarrow = 0$, alors $\mathbb{N} \models F_0(e)$, et donc d'après le lemme 3.8 $T \vdash F_0(e)$. Ainsi, $f(e) = 1 \neq \Phi_e(e)$.

Si à présent $\exists t \Phi_e(e)[t] \downarrow = 1$, alors $\mathbb{N} \models F_1(e)$, et l'on a donc d'après le lemme 3.8 $T \vdash F_1(e)$. On a alors $T \not\vdash F_0(e)$, et donc $T \vdash \neg F_0(e)$. Ainsi, $f(e) = 0 \neq \Phi_e(e)$.

Enfin, si $\forall t \Phi_e(e)[t] \uparrow$, la valeur de f n'importe pas. ■

Corollaire 3.11

Soit $T \supseteq \text{PA}$ une théorie complète et cohérente. Alors, T calcule un ensemble DNC_2 .

PREUVE. D'après la preuve du théorème précédent. ■

Arrêtons-nous quelques instants sur ce que nous dit le théorème de Gödel-Rosser : il existe F une formule Σ_1 qui n'est ni démontrable ni réfutable dans PA. D'après le lemme 3.8, si une formule Σ_1 est vraie dans $\mathbb{N}$, elle est démontrable dans PA. On en déduit $\mathbb{N} \not\models F$, et donc $\mathbb{N} \models \neg F$: cela fait de $\neg F$ une formule vraie, dans le sens où elle est vraie dans $\mathbb{N}$, mais que l'on ne peut démontrer à l'aide des axiomes de PA.

Le théorème nous dit enfin que rajouter des axiomes est sans espoir : tant que l'on garde la théorie calculatoirement énumérable, elle restera incomplète. Notons que l'on a montré avec la proposition 2.24 que T pouvait tout à fait être étendue en une théorie complète et cohérente, mais ce sera au prix de ne plus en connaître les axiomes, et l'on serait alors bien en peine de l'utiliser pour démontrer quoi que ce soit...

Passons à présent au deuxième théorème de Gödel, plus surprenant encore que le premier, et qui mit un coup d'arrêt brutal au programme de Hilbert.

Notation

Pour une théorie calculatoirement énumérable T dans $\mathcal{L}_{PA}$, soit $\text{Coh}(T)$ la formule close Π_1 correspondant à « T est une théorie cohérente », c'est-à-dire « pour toute preuve p dans T , p n'est pas une preuve de $\perp$ ».

L'existence d'une telle formule découle de la correspondance entre ensembles Σ_n^0/Π_n^0 et formules Σ_n/Π_n .

Théorème 3.12 (Second théorème d'incomplétude de Gödel)

Soit T une théorie cohérente calculatoirement énumérable contenant PA. Alors, $T \not\vdash \text{Coh}(T)$.

PREUVE. La première étape est de voir que la preuve du théorème 3.10 peut être formalisée, via un codage approprié, dans l'arithmétique de Peano : il existe F une formule Σ_1 telle que

$$\text{PA} \vdash \text{Coh}(T) \rightarrow (\ulcorner T \not\vdash F \urcorner \wedge \ulcorner T \not\vdash \neg F \urcorner).$$

Les notations $\ulcorner \Psi \urcorner$ indiquent la transformation de Ψ en un énoncé de l'arithmétique.

Supposons par l'absurde $T \vdash \text{Coh}(T)$. Alors, par la règle du Modus Ponens, on a $T \vdash \ulcorner T \not\vdash F \urcorner \wedge \ulcorner T \not\vdash \neg F \urcorner$, et donc $T \vdash \ulcorner T \not\vdash \neg F \urcorner$ ainsi que $T \vdash \ulcorner T \not\vdash F \urcorner$.

Il faut ensuite constater que la preuve du lemme 3.8 également peut se faire dans l'arithmétique de Peano, c'est-à-dire que l'arithmétique de Peano montre que si une formule Σ_1 fixée est vraie, alors elle est démontrable dans l'arithmétique de Peano — et donc dans T . Cela donne donc formellement avec la formule F

$$T \vdash F \rightarrow \ulcorner T \vdash F \urcorner.$$

On a alors par contraposée (en utilisant l'équivalence entre la négation et le codage de la négation) :

$$T \vdash \ulcorner T \not\vdash F \urcorner \rightarrow \neg F.$$

Comme $T \vdash \ulcorner T \not\vdash F \urcorner$, alors par Modus Ponens on obtient

$$T \vdash \neg F.$$

Il suffit enfin de voir que si T démontre une formule quelle qu'elle soit, alors PA — et donc T — démontre que T démontre cette formule. C'est encore une fois une application de la formalisation de la preuve du lemme 3.8 dans T , la phrase $T \vdash F$ étant Σ_1 . On a donc finalement

$$T \vdash \ulcorner T \vdash \neg F \urcorner,$$

ce qui contredit $T \vdash \ulcorner T \not\vdash \neg F \urcorner$. ■

3.3. Conséquence des théorèmes d'incomplétude

Souvenons-nous du théorème, lui, de complétude de Gödel : $T \vdash F$ ssi tout modèle de T est un modèle de F . Vu le second théorème d'incomplétude, aucune théorie T cohérente et calculatoirement énumérable contenant l'arithmétique ne peut démontrer sa propre cohérence : $T \not\vdash \text{Coh}(T)$. On en déduit donc par exemple qu'il existe des modèles de l'arithmétique de Peano au sein desquels la phrase $\text{Coh}(\text{PA})$ est fausse : dans ces modèles, il existe en

particulier une démonstration de $0 = 1$. Nous avons pourtant montré que PA avait des modèles, et donc, d'après le théorème de cohérence, que PA ne pouvait pas démontrer $0 = 1$. Cette situation qui semble paradoxale est résolue avec la considération suivante : la preuve de $0 = 1$ dans un modèle de $PA \cup \{\neg \text{Coh}(PA)\}$ n'est pas une vraie preuve. La phrase $\text{Coh}(PA)$ une fois exprimée dans le langage de l'arithmétique est de la forme : « il existe un entier qui code pour une démonstration valide de $0=1$ dans PA ». Un tel modèle contiendra donc un tel entier, mais cet entier ne sera pas un véritable entier — sinon, en déroulant la démonstration codée par cet entier, on aurait une vraie démonstration de $0 = 1$.

Un tel modèle de PA est dit *non standard* : il comporte des entiers dits eux aussi non standard. Tout modèle de PA contient 0 et 1. Par les axiomes régissant l'addition, on montre aisément que tout modèle de PA contient bien sûr les entiers standard : 0, 1, 2, 3, 4, ... Un modèle non standard de PA contiendra des entiers *plus grands* que tous les entiers standard, et du point de vue du modèle, rien ne permet de distinguer ces entiers-là des autres.

Prenons un entier non standard a d'un tel modèle. En utilisant les axiomes de l'arithmétique de Peano, on voit que les entiers $a + 1, a + 2, a + 3, \dots$ existent également dans le modèle. Par l'axiome qui stipule que tout entier autre que 0 admet un prédécesseur, on voit aussi que les différents entiers $a - 1, a - 2, a - 3, \dots$ sont dans le modèle. Comme $a > n$ pour tout $n \in \mathbb{N}$, alors également $a + a > a + n$ pour tout $n \in \mathbb{N}$. Il existe donc aussi un entier non standard plus grand que tous les $a + n$. En jouant ainsi avec les axiomes de PA, on arrive au théorème suivant.

Théorème 3.13

L'ordre $<$ pour les modèles dénombrables et non standard de l'arithmétique est constitué d'une copie de $\mathbb{N}$, suivie de $\mathbb{Q}$ copies de $\mathbb{Z}$.

On connaît donc bien à quoi ressemble l'ordre des éléments dans un modèle non standard. On ne connaît en revanche malheureusement pas à quoi ressemblent ni l'addition ni la multiplication.

Théorème 3.14 (Tennenbaum)

Soit $\mathcal{M}$ un modèle non standard dénombrable de PA. Soit

$$+, \times : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

des fonctions qui représentent les fonctions d'addition et de multiplication de $\mathcal{M}$, une fois établie une bijection entre $\mathcal{M}$ et $\mathbb{N}$. Alors, ni $+$ ni $\times$ ne sont calculables.

Nous reparlerons un peu des modèles non standard de l'arithmétique dans l'étude des mathématiques à rebours, et notamment dans la section [23-3](#).

4. Système ZFC

Nous avons donné une preuve de la cohérence de l'arithmétique de Peano, en utilisant le théorème de cohérence (voir le corollaire 2.21) et en fournissant un modèle pour cette théorie. D'après le deuxième théorème d'incomplétude de Gödel, nous avons nécessairement utilisé une théorie plus puissante que PA pour créer ce modèle. De quelle théorie s'agit-il ? On peut donner plusieurs réponses à cette question. La théorie la plus naturelle permettant de montrer la consistance de l'arithmétique du premier ordre est sans doute la théorie de l'arithmétique *du second ordre*, qui sera abordée plus en détail dans le chapitre 22.

4.1. Motivations

En arithmétique du second ordre, on s'autorise non seulement à travailler avec des entiers, mais également avec des ensembles d'entiers. L'existence d'ensembles d'entiers arbitraires est sujette à caution, davantage en tout cas que l'existence des entiers eux-mêmes : il s'agit d'objets infinis, qui sont en quantité indénombrable, et s'il y a un aspect « légitime » à accepter l'existence d'ensembles d'entiers calculables (après tout, on peut écrire des algorithmes qui les produisent, et leur existence théorique est donc doublée d'une certaine forme d'existence pratique), on peut questionner plus facilement celle des autres ensembles. Certains d'entre eux sont malgré tout accessibles, dans le sens où ils ont une définition claire, par exemple le saut Turing. Ce qui rend $\emptyset'$ plus légitime qu'un autre ensemble non calculable arbitraire, c'est le fait qu'il soit *définissable*, et qui plus est par une formule assez simple — en particulier $\Sigma_1 : \{e \in \mathbb{N} : \Phi_e(e) \downarrow\}$. En arithmétique du second ordre, on s'autorisera l'existence de tous les ensembles définissables par une formule de l'arithmétique arbitraire, en particulier les formules Σ_n^0 pour un certain n : cela s'appelle *l'axiome de compréhension*. Une fois que l'on a admis l'existence d'un ensemble X , il serait absurde de ne pas accepter l'existence d'ensembles calculables avec X comme oracle, et si l'on accepte l'axiome de compréhension, il serait tout aussi absurde de ne pas accepter l'existence des ensembles définissables par une formule de l'arithmétique arbitraire, qui pourrait utiliser X comme oracle.

L'arithmétique du second ordre consiste donc en l'arithmétique de Peano, à laquelle on ajoute l'axiome de compréhension qui permet de valider l'existence des ensembles définissables via une formule de l'arithmétique, éventuellement à l'aide d'un oracle — un ensemble déjà existant. Une très grande partie des mathématiques peut déjà se formaliser de cette manière, mais pas toutes les mathématiques. En particulier, rien ne permet dans

l'arithmétique du second ordre de parler de l'ensemble de tous les sous-ensembles d'entiers, ou même de sous-ensembles arbitraires de ce dernier. On a besoin pour cela d'un autre axiome : étant donné un ensemble X , l'ensemble $\mathcal{P}(X)$ des parties de X — c'est-à-dire l'ensemble de tous les sous-ensembles de X — est légitime, il existe et on peut l'utiliser. L'ensemble des parties de X n'est pas de même nature que X . L'un est un ensemble d'entiers et l'autre un ensemble d'ensembles d'entiers. La théorie des ensembles permet de traiter ces deux éléments de manière homogène : tout élément ne sera plus qu'ensemble, y compris les entiers. Informellement, l'entier 0 sera l'ensemble vide, l'entier 1 sera l'ensemble qui contient 0, l'entier 2 sera l'ensemble qui contient 0 et 1, et inductivement l'ensemble $n + 1$ sera l'ensemble qui contient m pour tout $m \leq n$. On peut bien sûr imaginer d'autres manières de représenter les entiers par des ensembles, et avec la pratique ces dernières n'ont aucune importance, mais il faut bien en choisir une, et cette manière-là est celle qui est devenue standard, sous l'impulsion du mathématicien John von Neumann. Nous en reparlerons avec l'étude des ordinaux dans le chapitre 27.

4.2. Système de Zermelo

À présent que l'on ne travaille plus qu'avec des ensembles, nous ne sommes plus dans le langage de l'arithmétique. Notre unique symbole de relation est celui de l'appartenance $\in$ — auquel on ajoute tout de même celui de l'égalité. Il nous faut quelques axiomes basiques afin de régir les manipulations élémentaires des ensembles.

- (1) *L'axiome de l'ensemble vide* : l'ensemble vide existe.
- (2) *L'axiome de paire* : si a et b sont des ensembles, alors $\{a, b\}$ est un ensemble.
- (3) *L'axiome de réunion* : l'idée est que si $(a_i)_{i \in I}$ est une collection d'ensembles indexée par un ensemble I , alors l'ensemble $\bigcup_{i \in I} a_i$ existe. La notion « indexée » n'étant pas formellement définie en théorie des ensembles, on dira à la place que si b est un ensemble dont les éléments sont les a_i , alors la réunion de tous ces a_i existe. Pour être tout à fait formel, l'axiome est : $\forall b \exists c \forall x (x \in c \leftrightarrow \exists a \in b \ x \in a)$.

On a également besoin d'un axiome qui régisse l'égalité entre deux ensembles.

- (4) *L'axiome d'extensionnalité* : deux ensembles sont égaux si, et seulement si, ils ont les mêmes éléments.

Il y a finalement notre axiome fauteur de troubles, qui nous a poussé dans cette théorie des ensembles permettant — par exemple — de traiter de l'ensemble des parties de $\mathbb{N}$.

- (5) *L'axiome de l'ensemble des parties* : pour tout ensemble a , l'ensemble des parties de a , noté $P(a)$ existe. Formellement :

$$\forall a \exists b \forall c (c \in b \leftrightarrow c \subseteq a),$$

où $c \subseteq a$ peut s'écrire comme $\forall x(x \in c \rightarrow x \in a)$.

Nous pouvons enfin ajouter notre axiome de compréhension, permettant de construire des ensembles à partir de formules du premier ordre utilisant éventuellement d'autres ensembles comme paramètres. Comme il existe une infinité de formules du premier ordre, il ne s'agit pas d'un seul axiome, mais d'un schéma d'axiome : un axiome par formule.

- (6) *Le schéma d'axiome de compréhension* : pour $F(y, x_1, \dots, x_n)$, formule fixée dans le langage de la théorie des ensembles, pour tout n -uplet d'ensembles $b_1, \dots, b_n$ et pour tout ensemble a , l'ensemble

$$\{y \in a : F(y, b_1, \dots, b_n)\}$$

existe.

On peut vérifier sans peine que les axiomes précédents impliquent l'existence de tous les ensembles héréditairement finis, c'est-à-dire des ensembles finis, dont les éléments sont eux-mêmes des ensembles finis, etc., jusqu'à arriver en déroulant l'arbre décrivant les relations d'appartenance d'un ensemble avec ses éléments, à l'ensemble vide pour toute feuille de cet arbre. Rien ne nous permet pour le moment de parler de l'ensemble de tous les entiers. Nous avons de fait besoin pour cela d'un axiome.

- (7) *L'axiome de l'infini* : il existe un ensemble infini. Formellement :

$$\exists x (\emptyset \in x \wedge \forall y \in x \ y \cup \{y\} \in x).$$

L'axiome de l'infini affirme moralement l'existence de $\mathbb{N}$ en tant qu'ensemble. Via le codage que nous avons donné des entiers, on peut vérifier que l'ensemble codant l'entier $n + 1$ est égal à l'ensemble $n \cup \{n\}$, où n est l'ensemble codant l'entier n . L'axiome de l'infini nous dit donc qu'il existe un ensemble contenant tous les entiers. Il pourrait éventuellement contenir d'autres éléments, mais en utilisant les autres axiomes on définit $\mathbb{N}$ comme le plus petit ensemble x — plus petit pour l'inclusion — tel que

$$\emptyset \in x \wedge \forall y \in x \ y \cup \{y\} \in x.$$

4.3. L'axiome de remplacement et les jeux boréliens

Les axiomes de (1) à (7) forment la théorie Z de Zermelo. Ils sont suffisants pour développer une large partie des mathématiques. Fraenkel et Skolem vont introduire un nouvel axiome, à la fois intuitif et formellement

nécessaire pour développer les théories des ordinaux et des hiérarchies d'infinis. Il s'agit plus exactement un schéma d'axiomes, qui dit essentiellement que l'image d'une formule fonctionnelle existe. Une formule $F(y, r)$ est fonctionnelle si pour tout y la formule $F(y, r_y)$ est satisfaite pour exactement un ensemble r_y . Il s'énonce formellement comme suit.

- (8) *Le schéma d'axiome de remplacement* : pour une formule fonctionnelle $F(y, x_1, \dots, x_n, z)$ fixée dans le langage de la théorie des ensembles, pour tout n -uplet d'ensembles $b_1, \dots, b_n$, pour tout ensemble a , l'ensemble

$$\{z : \exists y \in a \, F(y, b_1, \dots, b_n, z)\}$$

existe.

La théorie des ensembles devient strictement plus puissante avec le schéma d'axiome de remplacement, qui permet en particulier de montrer l'existence de l'ensemble $\mathbb{N} \cup P(\mathbb{N}) \cup P(P(\mathbb{N})) \dots$, dont la construction est impossible sans cet axiome.

Il est remarquable de noter qu'une utilisation conjointe de l'axiome de l'ensemble des parties et du schéma d'axiome de remplacement est indispensable pour construire certains ensembles d'entiers — qui seront nécessairement d'une complexité extrême en termes de degré Turing. L'exemple emblématique est le théorème de détermination des jeux boréliens de Martin. Voyons de quoi il s'agit : considérons une classe $\mathcal{B} \subseteq 2^{\mathbb{N}}$ pour le moment arbitraire, et considérons le jeu à deux joueurs suivant : le joueur 1 choisit un bit $x_0 \in \{0, 1\}$, puis le joueur 2 choisit à son tour un bit $x_1 \in \{0, 1\}$, et ainsi de suite, à l'étape $2n$ le joueur 1 choisit le bit x_{2n} et à l'étape $2n + 1$ le joueur 2 choisit le bit x_{2n+1} . À « la fin » du jeu, on obtient un ensemble $X = x_0 x_1 x_2 \dots$. Le joueur 1 gagne le jeu si $X \in \mathcal{B}$, sinon c'est le joueur 2 qui remporte la partie. La question est alors : un des deux joueurs a-t-il une stratégie gagnante ? Une stratégie pour le joueur 1 est une fonction f qui prend en paramètre une chaîne σ de taille paire, correspondant à ce qui a été joué jusque-là, le dernier coup étant le dernier bit de σ joué par le joueur 2, et qui renvoie le coup suivant $f(\sigma)$. Une telle stratégie est gagnante si l'ensemble obtenu en jouant les coups donnés par la fonction f est toujours dans $\mathcal{B}$, quels que soient les coups joués par le joueur 2.

Nous verrons dans le chapitre 17 des classes $\mathcal{B} \subseteq 2^{\mathbb{N}}$ — d'une variété fort riche — ne nécessitant pas l'axiome de l'ensemble des parties pour être manipulées : nous en avons déjà un exemple avec les classes Π_1^0 ou Σ_1^0 . Nous mènerons des constructions itérées de classes de plus en plus complexes, qui peuvent se coder par un objet dénombrable, d'une manière analogue au codage des classes Π_1^0 par des arbres : il s'agira des classes dites *boréliennes*. Le mathématicien Donald A. Martin, dont nous reparlerons dans le chapitre 12, a montré le remarquable théorème suivant.

Théorème 4.1 (Martin [149])

Soit $\mathcal{B}$ une classe borélienne. Alors, pour le jeu décrit ci-dessus avec la classe $\mathcal{B}$, un des deux joueurs a une stratégie gagnante.

La stratégie d'un jeu borélien est une fonction $f : 2^{<\mathbb{N}} \rightarrow \{0, 1\}$, qui peut donc être représentée par un ensemble d'entiers. Friedman [67] a montré que pour certains boréliens, de complexité relativement simple, une telle stratégie ne pouvait être construite sans l'utilisation de l'axiome de l'ensemble des parties, et même sans itération arbitraire de l'application de cet axiome. Ainsi, pour montrer l'existence de certains réels, nous faut-il faire appel à l'axiome de l'ensemble des parties utilisé conjointement à celui de remplacement afin de construire $\mathbb{N} \cup P(\mathbb{N}) \cup P(P(\mathbb{N})) \dots$, indispensable à la définition de notre réel — la stratégie gagnante pour un certain borélien.

4.4. L'axiome de fondation et le codage

Un autre axiome a été rajouté par Fraenkel et Skolem, ainsi que par von Neumann. Contrairement aux autres axiomes, ce dernier est à peu près inutile pour la construction de l'univers mathématique, mais on l'ajoute simplement parce que cela correspond à notre conception des choses : l'axiome dit en substance qu'étant donné un ensemble x , si l'on considère un élément $x_1 \in x$, puis un élément $x_2 \in x_1$, en continuant ainsi inductivement avec un élément $x_{n+1} \in x_n$, on arrivera nécessairement au bout d'un certain $n \in \mathbb{N}$ à l'ensemble vide : il n'y a pas d'autres ensembles que ceux que l'on peut construire inductivement via les autres axiomes en partant de l'ensemble vide. En particulier, il n'y a aucun ensemble x tel que $x \in x$. Cela peut paraître évident ou pas selon chacun, mais reflète en tout cas la conception généralement adoptée dans la communauté sur la nature des ensembles.

Afin d'étayer notre propos, rappelons que l'un des intérêts de la théorie des ensembles est de pouvoir y formaliser la totalité des mathématiques. Cette formalisation passe par le codage des structures mathématiques usuelles par des ensembles, et ce codage ne se fait de toute façon qu'avec des ensembles qui respectent l'axiome de fondation — que ce dernier soit adopté ou pas. Aussi, s'il n'est pas contradictoire de penser qu'il existe d'autres ensembles qui ne respectent pas cet axiome, on n'en a en pratique pas besoin, et de tels objets ne correspondent *a priori* à rien de tangible.

Terminons en insistant sur le fait que si l'on peut coder le reste des mathématiques par des ensembles, ce n'est en revanche pas une raison pour le faire, et l'on est évidemment bien plus à l'aise à travailler avec des entiers, des réels ou autre. Ce qui est intéressant c'est l'*existence* de ce codage, et le fait que si un énoncé est indécidable en théorie des ensembles, il en devient alors indécidable « tout court ».

4.5. L'axiome du choix et la cardinalité

La théorie Z de Zermelo plus l'axiome de remplacement et celui de fondation donne la théorie ZF, de Zermelo/Fraenkel.

Un dernier axiome, sans doute le plus célèbre, est l'axiome du choix, qui faisait originellement partie de la théorie de Zermelo. La nécessité de cet axiome devrait être aisée à comprendre via l'analogie que l'on peut en faire en calculabilité. Nous avons par exemple vu que toute classe Π_1^0 non vide et dénombrable contient un ensemble calculable. En revanche, il n'est pas possible, étant donné le code d'une classe Π_1^0 non vide, de trouver uniformément un algorithme permettant d'en calculer un élément. En particulier, étant donné une suite $(\mathcal{F}_n)_{n \in \mathbb{N}}$ de classes Π_1^0 dénombrables non vides, il n'existe pas nécessairement de fonction calculable permettant de *choisir* un élément dans chacune de ces classes. La question clef est ici celle de l'uniformité. Évidemment, à l'aide de $\emptyset'$, on pourra créer cette fonction de choix, mais que se passe-t-il si l'on considère des classes plus complexes ? Peut-on toujours construire une fonction de choix ? La réponse est non, sans l'aide d'un nouvel axiome.

- (9) *L'axiome du choix* : étant donné une collection $(A_i)_{i \in I}$ d'ensembles non vides, il existe une fonction $f : I \rightarrow \bigcup_{i \in I} A_i$ telle que $f(i) \in A_i$ pour tout i .

L'axiome du choix apparaît nécessaire pour développer une théorie complète de la cardinalité des ensembles. En particulier, avec l'axiome du choix, on peut montrer que pour tous ensembles A, B on a $|A| \leq |B|$ ou $|B| \leq |A|$ (nous en reparlerons avec l'étude détaillée des ordinaux dans le chapitre 27). Ce n'est plus du tout vrai sans l'axiome du choix, l'exemple par excellence en calculabilité étant certainement celui des degrés Turing. Il est facile de construire une injection de $2^{\mathbb{N}}$ dans les degrés Turing, mais il est en revanche impossible de construire une injection des degrés Turing dans $2^{\mathbb{N}}$ sans l'axiome du choix (il est en particulier impossible de montrer l'existence d'une fonction $f : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ telle que $X \equiv_T Y \leftrightarrow f(X) = f(Y)$).

Une première réaction est de se simplifier la vie et d'utiliser l'axiome du choix si nécessaire. Cette approche n'est toutefois pas dans l'esprit de la calculabilité, de laquelle la théorie des ensembles est moins éloignée que l'on ne pourrait le croire : les axiomes autres que l'axiome du choix nous permettent de *construire* des objets de plus en plus complexes à partir d'objets existants, un peu à la manière dont on construit des degrés Turing de plus en plus complexes en itérant le saut. L'axiome du choix est quant à lui fondamentalement non constructif, et n'est de ce point de vue là pas aussi légitime que les autres. Il conduit en plus à des théorèmes qui semblent paradoxaux, l'exemple le plus connu étant le *paradoxe de Banach/Tarski*, une

construction qui à l'aide de l'axiome du choix montre comment découper une boule de l'espace $\mathbb{R}^3$ en un nombre fini de morceaux, et comment ré-assembler ces morceaux pour obtenir deux boules strictement identiques à la première.

En n'acceptant pas l'axiome du choix, on sort bien entendu de ce monde idéal où la cardinalité de tout ensemble est comparable, mais il s'agit en fait d'une situation « artificielle », dont nous n'avons pas réellement besoin.

4.6. Résultats d'indépendances

La théorie obtenue avec les axiomes (1)-(8) est la théorie ZF, et si on lui ajoute l'axiome du choix, on a alors la théorie dite ZFC.

4.6.1. L'axiome du choix

La question de savoir si l'axiome du choix est démontrable à partir des autres axiomes, ou même celle de savoir s'il ne risque pas d'introduire de contradiction, est longtemps restée ouverte. Le théorème de complétude de Gödel permet la technique suivante : si l'on fait l'hypothèse que ZF est cohérent, et donc a un modèle, et qu'à l'aide de ce modèle on peut construire un modèle de ZFC, on aura montré que la cohérence de ZF implique celle de ZFC, et en particulier que ZF ne peut pas démontrer que l'axiome du choix est faux (à moins bien sûr que ZF ne soit incohérent). C'est exactement ce qu'a fait Gödel quelques années plus tard [74], via son modèle dit de *l'univers des constructibles*. Ce n'est qu'encore plus tard, en 1962, avec sa fameuse technique de Forcing dont nous verrons certains aspects dans le chapitre 11, que Cohen [37] réussira le tour de force de construire un modèle de ZF dans lequel l'axiome du choix est faux : l'axiome du choix ne peut donc être ni démontré ni réfuté dans ZF.

4.6.2. L'hypothèse du continu

La question qui obséda Cantor tout au long de sa vie (et de nombreux mathématiciens durant presque un siècle) est elle aussi indépendante des autres axiomes de la théorie des ensembles (avec ou sans l'axiome du choix). L'univers des constructibles de Gödel constitue également un modèle de ZFC dans lequel l'hypothèse du continu est vérifiée, c'est-à-dire qu'il n'existe aucun ensemble A pour lequel $|\mathbb{N}| < |A| < |2^{\mathbb{N}}|$. Toujours avec sa technique de forcing, Cohen a construit des modèles de ZFC dans lesquels on a un nombre arbitraire d'infinis strictement compris entre $|\mathbb{N}|$ et $|2^{\mathbb{N}}|$.

Notons ici que l'on peut donner plusieurs versions de l'hypothèse du continu. Celle formulée originellement par Cantor portait sur les ensembles de réels : existe-t-il un ensemble $\mathcal{A} \subseteq \mathbb{R}$ tel que $|\mathbb{N}| < |\mathcal{A}| < |\mathbb{R}|$? Plus tard, la question sera étendue à n'importe quel ensemble, et en particulier aux ordinaux, que

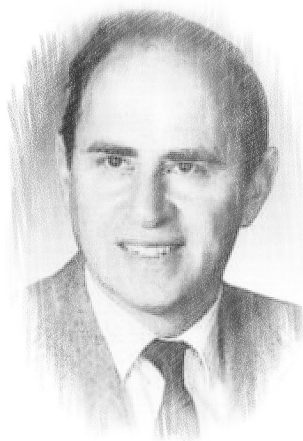
nous verrons formellement dans le chapitre 26. Quelle que soit la version de l'hypothèse du continu, il s'agit d'une question indépendante des autres axiomes de la théorie des ensembles.

Chapitre 10

Forcing de Cohen

Le forcing est une technique inventée par le mathématicien Paul Cohen au début des années 60 pour montrer l'indépendance de l'hypothèse du continu et de l'axiome de choix de la théorie de Zermelo/Fraenkel. Cette technique a révolutionné la théorie des ensembles, et joue également un rôle prépondérant en calculabilité, où elle s'est imposée comme l'une des deux techniques principales de construction d'ensembles, aux côtés de la méthode de priorité (voir le chapitre 13).

Historiquement, cette technique a été conçue pour étendre un modèle $\mathcal{M}$ de la théorie de ZF en lui ajoutant un nouvel objet G pour former un nouveau modèle $\mathcal{M}[G]$ tout en permettant aux éléments du modèle $\mathcal{M}$ de contrôler les propriétés du modèle étendu $\mathcal{M}[G]$. Plus généralement, la technique de forcing permet de créer des objets mathématiques à l'aide d'approximations de plus en plus précises, tout en étant capable de contrôler certaines propriétés de l'objet final avant que la construction ne soit terminée. Une notion de forcing est avant tout la définition d'un ordre partiel d'approximations $(\mathbb{P}, \leq)$, où la relation $d \leq c$ signifie que l'approximation d est plus précise que l'approximation c . Lorsqu'une propriété (ou contrat) $\mathcal{R}$ sur l'objet final est déjà déterminée par



Paul Joseph Cohen, 1934–2007

une approximation $c \in \mathbb{P}$, autrement dit lorsque quelle que soit la suite de la construction, l'objet final satisfera $\mathcal{R}$, on dit alors que c *force* $\mathcal{R}$. Le cœur de la technique du forcing réside dans la capacité à forcer des propriétés complexes sur l'objet final à partir de simples approximations.

Le forcing est souvent considéré comme une technique difficile à appréhender aux premiers abords. Son application à la calculabilité, plus simple, permet d'une part de l'aborder en douceur, et d'autre part de comprendre plus facilement le détail de ses mécanismes sous-jacents. Cette simplicité est due essentiellement aux deux choses suivantes.

- ▷ À l'instar de la méthode des extensions finies, nous allons principalement nous servir du forcing pour créer des ensembles d'entiers, tout en contrôlant leurs puissances calculatoires. On s'intéressera donc à forcer des formules simples, du type $\langle \Phi^G(n) \downarrow \rangle$ ou $\langle \Phi^G(n) \uparrow \rangle$, où G désigne l'ensemble final. Il s'agit donc de forcer des formules Σ_1^0 ou Π_1^0 . Nous verrons que plusieurs constructions, notamment celles avec la méthode des extensions finies, sont des utilisations simplifiées du forcing. Nous verrons également comment étendre le forcing à des formules de complexité arbitraires, ce qui présente un premier niveau de complexité supplémentaire.
- ▷ Contrairement à la théorie des ensembles, nous ne forcerons que des propriétés dont les quantificateurs portent sur les entiers. La création d'un ensemble d'entiers par forcing ne modifie pas la nature des entiers, et donc la portée des quantificateurs. En théorie des ensembles en revanche, les quantificateurs portent sur les ensembles, et la création d'un nouvel ensemble ajoute une grande quantité d'ensembles au modèle, et change donc la portée des quantificateurs. Il est alors nécessaire d'utiliser des *noms*, pour parler des objets de notre modèle étendu à l'intérieur de notre modèle de base. Nous n'aurons pas besoin d'avoir recours aux noms en calculabilité, ce qui rend la présentation plus abordable.

1. Formules de l'arithmétique du second ordre

Les énoncés que l'on « force » en calculabilité sont toujours des formules de l'arithmétique du premier ordre *avec des variables libres d'ensembles*. Ces formules sont un cas particulier de l'arithmétique du second ordre, dont nous parlerons plus en détail dans le chapitre 22. Une formule de l'arithmétique du second ordre possède deux types de variables : des variables dites *du premier ordre* représentant des entiers et que l'on notera en minuscule, et des variables dites *du second ordre* représentant des ensembles d'entiers et que l'on notera en majuscule. Le langage se voit augmenté du symbole d'appartenance $\in$, et de la formule atomique

« $x \in A$ ». En arithmétique du second ordre, les quantifications peuvent être sur les entiers, et sur les ensembles d'entiers. Par exemple, l'énoncé « $\forall A \forall n \exists m (m > n \wedge m \in A)$ » affirme que tout ensemble d'entiers est infini.

L'arithmétique du premier ordre avec variables libres d'ensembles est une restriction de l'arithmétique du second ordre, où seules les quantifications sur les entiers sont autorisées¹. Lors de l'évaluation d'une formule de l'arithmétique du second ordre dans un modèle donné, ses variables libres d'ensembles sont remplacées par des paramètres, c'est-à-dire des éléments du modèle considéré. Ainsi, l'énoncé $F(G) = \forall n \exists m (m > n \wedge m \in G)$ est une formule de l'arithmétique du premier ordre avec G comme variables libres d'ensemble, et tout ensemble infini $X \subseteq \mathbb{N}$ est un paramètre pour lequel $F(X)$ est vrai.

Nous avons vu dans la section 9-3 les formules Δ_0^0 de l'arithmétique : celles ne contenant que des quantifications bornées, c'est-à-dire des quantifications de la forme $\forall x \leq y$ et $\exists x \leq y$. Nous pouvons définir une hiérarchie sur les formules de l'arithmétique avec variables libres d'ensembles, similaire à la hiérarchie de la définition 9-3.2.

Définition 1.1

1. Une formule $F(Y_1, \dots, Y_k, x_1, \dots, x_m)$ de l'arithmétique du second ordre est Σ_n^0 si

$$F(Y_1, \dots, Y_k, x_1, \dots, x_m) = \overbrace{\exists y_1 \forall y_2 \dots Q y_n}^{n \text{ quantificateurs}} G(Y_1, \dots, Y_k, x_1, \dots, x_m, y_1, \dots, y_n),$$

pour $G(Y_1, \dots, Y_k, x_1, \dots, x_m, y_1, \dots, y_n)$ une formule Δ_0^0 , où Q vaut $\exists$ si n est impair, et $\forall$ si n est pair.

2. Une formule $F(Y_1, \dots, Y_k, x_1, \dots, x_m)$ de l'arithmétique du second ordre est Π_n^0 si

$$F(Y_1, \dots, Y_k, x_1, \dots, x_m) = \overbrace{\forall y_1 \exists y_2 \dots Q y_n}^{n \text{ quantificateurs}} G(Y_1, \dots, Y_k, x_1, \dots, x_m, y_1, \dots, y_n),$$

pour $G(Y_1, \dots, Y_k, x_1, \dots, x_m, y_1, \dots, y_n)$ une formule Δ_0^0 , où Q vaut $\forall$ si n est impair, et $\exists$ si n est pair. $\diamond$

Nous avons jusqu'ici implicitement utilisé les formules de l'arithmétique du second ordre, via l'utilisation de fonctionnelles. Le théorème 9-3.4 se décline en l'équivalence suivante.

1. Nous ne verrons que très tard dans cet ouvrage, dans la partie IV, la grande complexité calculatoire qui se cache derrière les quantifications du second ordre.

Théorème 1.2

Soient $A, Z \in 2^{\mathbb{N}}$. Les énoncés suivants sont équivalents.

- (1) $A \subseteq \mathbb{N}$ est Z -c. e.
- (2) Il existe $F(X, n)$ une formule Σ_1^0 de l'arithmétique du second ordre telle que $A = \{n \in \mathbb{N} : \mathbb{N} \models F(Z, n)\}$.
- (3) Il existe une fonctionnelle Turing $\Phi(Z, n)$ telle que

$$A = \{n \in \mathbb{N} : \Phi(Z, n) \downarrow\}.$$

2. Forcing Σ_1^0/Π_1^0

Nous allons maintenant commencer notre immersion dans l'univers du forcing en étudiant une notion de forcing spécifique, à savoir le forcing de Cohen. Comme expliqué dans l'introduction, une notion de forcing est spécifiée par la donnée de son ordre partiel d'approximations. Dans notre cas, il s'agira de l'ordre partiel des chaînes, muni de la relation de préfixe. Il s'agit d'une des notions de forcing les plus simples conceptuellement, mais qui contient déjà les concepts fondamentaux du forcing.

Il existe plusieurs manières d'aborder le forcing, avec différents niveaux d'abstraction. Il est possible de le voir comme une élaboration de la méthode des extensions finies, cherchant à systématiser la satisfaction de contrats, et à en extraire une construction générale. Nous présenterons cette approche dans la section 2.1.

Il est également possible de formuler le forcing dans un cadre topologique. La topologie permet de définir une notion de classe d'ensembles « négligeable », ou *maigre*. La construction d'un ensemble par forcing consiste alors à choisir un élément « typique », c'est-à-dire évitant un ensemble négligeable de propriétés indésirables. Nous verrons cette approche dans la section 2.2.

2.1. L'approche par extensions finies

Reconsidérons à présent la méthode des extensions finies développée dans la section 4-8. Le but est de construire un ensemble $G \in 2^{\mathbb{N}}$ satisfaisant une infinité de contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$. Chaque contrat est traité indépendamment, et doit donc être satisfait tout en laissant suffisamment de degrés de liberté dans la construction pour satisfaire les autres contrats.

Point sur les contrats

Les contrats — que nous avons vus jusqu'ici de manière informelle — peuvent être vus comme des formules dans le langage de l'arithmétique du premier ordre avec variables d'ensembles, comportant une variable libre du second ordre représentant l'ensemble G qui doit satisfaire le contrat. La méthode des extensions finies consiste à construire $G \in 2^{\mathbb{N}}$ en spécifiant des segments initiaux de plus en plus longs, représentés sous forme de chaînes $\sigma \in 2^{<\mathbb{N}}$. Les contrats Σ_1^0 sont ceux correspondant à des formules Σ_1^0 ou de manière équivalente ceux que l'on peut toujours mettre sous la forme $\ll \Phi_e(G, 0) \downarrow \gg$ pour une fonctionnelle Φ_e . Les contrats Π_1^0 sont ceux correspondant à des formules Π_1^0 ou de manière équivalente ceux que l'on peut toujours mettre sous la forme $\ll \Phi_e(G, 0) \uparrow \gg$ pour une fonctionnelle Φ_e .

Satisfaction d'un contrat. La procédure générale pour satisfaire un contrat $\mathcal{R}_e$ est la suivante : étant donné une chaîne $\sigma \in 2^{<\mathbb{N}}$ représentant un segment initial de A déjà spécifié pour satisfaire les contrats précédents, il s'agit de trouver une chaîne $\tau \succeq \sigma$ telle que le contrat $\mathcal{R}_e$ est satisfait. L'ensemble final G n'étant pas alors connu, le contrat doit être satisfait quelle que soit la suite de la construction, autrement dit il doit être satisfait pour tout $G \in [\tau] = \{X \in 2^{\mathbb{N}} : \tau \prec X\}$.

Ordre partiel des chaînes. Prenons de la hauteur, et considérons la méthode des extensions finies d'un point de vue plus abstrait. Nous avons un ordre partiel $(2^{<\mathbb{N}}, \preceq)$ qui correspond à l'ensemble $2^{<\mathbb{N}}$ des chaînes finies, muni de la relation de préfixe. Nous avons aussi une fonction d'interprétation $[\cdot] : 2^{<\mathbb{N}} \rightarrow \mathcal{P}(2^{\mathbb{N}})$ définie par $[\sigma] = \{X \in 2^{\mathbb{N}} : \sigma \prec X\}$. Intuitivement, les éléments de $2^{<\mathbb{N}}$ représentent des approximations de l'ensemble G en construction. Étant donné une approximation σ , $[\sigma]$ est la classe des ensembles que l'on pourrait potentiellement obtenir à la fin de la construction. Plus on avance dans la construction, plus l'approximation s'affine et la classe des ensembles candidats se restreint. Ainsi, nous avons la propriété suivante : si $\sigma \preceq \tau$, alors $[\tau] \subseteq [\sigma]$. Voyons une première définition de la relation de forcing.

Définition 2.1. Soit $\mathcal{R}$ un contrat Σ_1^0 ou Π_1^0 . Une chaîne σ *force* $\mathcal{R}$, que l'on notera par $\sigma \Vdash^* \mathcal{R}$, si le contrat est satisfait pour tout $G \in [\sigma]$. $\diamond$

Densité. Nous pouvons nous abstraire de la notion de contrat en représentant un contrat $\mathcal{R}_e$ comme l'ensemble $P_e \subseteq 2^{<\mathbb{N}}$ des chaînes qui le forcent. Notons que si σ force $\mathcal{R}_e$, alors tout $\tau \succeq \sigma$ force également $\mathcal{R}_e$, car $[\tau] \subseteq [\sigma]$. L'ensemble P_e est donc clos par suffixe. La procédure de satisfaction du contrat $\mathcal{R}_e$ consiste à montrer que pour toute chaîne $\sigma \in 2^{<\mathbb{N}}$,

il existe une extension $\tau \succeq \sigma$ telle que $\tau \in P_e$. Si c'est le cas, on dira que l'ensemble P_e est dense.

Définition 2.2. Un ensemble $W \subseteq 2^{<\mathbb{N}}$ est *dense* si pour tout $\sigma \in 2^{<\mathbb{N}}$, il existe une extension $\tau \succeq \sigma$ telle que $\tau \in W$. $\diamond$

Intuitivement, un ensemble $W \subseteq 2^{<\mathbb{N}}$ est dense si quelle que soit la construction en cours $\sigma_0 \preceq \sigma_1 \preceq \dots \preceq \sigma_n$ à l'aide de la méthode des extensions finies, il n'est jamais trop tard pour trouver une extension $\sigma_{n+1} \succeq \sigma_n$ dans W . Il s'ensuit que si nous avons un ensemble dénombrable de contrats représentés par des ensembles $(P_n)_{n \in \mathbb{N}}$, dès lors que ces ensembles sont denses, il existe une suite infinie $\sigma_0 \preceq \sigma_1 \preceq \sigma_2 \preceq \dots$ telle que pour tout n , il existe un entier m pour lequel $\sigma_m \in P_n$. En particulier, soit $\{G\} = \bigcap_n [\sigma_n]$, alors G possèdera des segments initiaux dans chaque ensemble P_e .

Remarque

Si les ensembles $(P_n)_{n \in \mathbb{N}}$ sont uniformément c.e, alors la construction de la suite infinie $\sigma_0 \preceq \sigma_1 \preceq \sigma_2 \preceq \dots$ peut se faire de manière calculable, et l'ensemble G résultant l'est également.

Généricité. Nous pouvons formaliser la construction de la méthode des extensions finies en un théorème trivial au vu des intuitions précédentes.

Définition 2.3. On dit qu'un ensemble $G \in 2^{\mathbb{N}}$ *rencontre* un ensemble $W \subseteq 2^{<\mathbb{N}}$ si $G \upharpoonright_n \in W$ pour un certain $n \in \mathbb{N}$. Soit $\vec{D} = (D_n)_{n \in \mathbb{N}}$ une suite d'ensembles de chaînes; un ensemble $G \in 2^{\mathbb{N}}$ est *$\vec{D}$ -générique* s'il rencontre chaque D_n . $\diamond$

Théorème 2.4

Soit $\vec{D} = (D_n)_{n \in \mathbb{N}}$ une suite dénombrable d'ensembles de chaînes denses et soit $\sigma \in 2^{<\mathbb{N}}$. Il existe un ensemble $\vec{D}$ -générique qui étend σ .

PREUVE. Soit $\sigma_0 \prec \sigma_1 \prec \sigma_2 \prec \dots$ la suite infinie strictement croissante de chaînes définies inductivement comme suit. Initialement, $\sigma_0 = \sigma$. Si σ_n est défini, σ_{n+1} est une extension stricte de σ_n dans D_n . Une telle extension existe par densité de D_n . Soit $\{G\} = \bigcap_n [\sigma_n]$; alors, G est $\vec{D}$ -générique et étend σ . $\blacksquare$

Nous verrons la contrepartie topologique du théorème précédent avec le lemme 2.14. Le théorème 2.4 dit entre autre que si $(D_n)_{n \in \mathbb{N}}$ est une suite d'ensembles denses correspondants aux contrats $(\mathcal{R}_n)_{n \in \mathbb{N}}$, alors il existe des ensembles $\vec{D}$ -génériques, qui satisfont donc tous les contrats simultanément.

Il existe une quantité indénombrable d'ensembles de chaînes denses, et un ensemble $G \in 2^{\mathbb{N}}$ ne peut pas être générique pour tous ces ensembles simultanément, simplement parce que l'ensemble $\{\sigma \in 2^{<\mathbb{N}} : \sigma \not\prec G\}$ est un ensemble dense que G ne rencontre pas. La notion de généricité est donc dépendante d'une collection $\vec{D}$ dénombrable d'ensembles de chaînes denses.

Suffisamment générique

Il est courant d'énoncer des résultats de la forme « tout ensemble *suffisamment générique* satisfait telle propriété ». Cela signifie qu'il existe une suite dénombrable $\vec{D} = (D_n)_{n \in \mathbb{N}}$ d'ensembles de chaînes denses telle que tout ensemble $\vec{D}$ -générique satisfait la propriété. Notons qu'étant donné n'importe quelle autre suite dénombrable d'ensembles de chaînes denses $\vec{E} = (E_n)_{n \in \mathbb{N}}$, la suite $\{\vec{D}, \vec{E}\}$ est toujours une suite dénombrable d'ensembles de chaînes denses et l'on pourra donc produire un ensemble $\{\vec{D}, \vec{E}\}$ -générique. C'est cela qui justifie l'appellation de *suffisamment générique*.

Nous allons reformuler la preuve de la proposition 4-8.2 en termes de densité et de généricité. Nous avons besoin pour cela d'étendre la relation de forcing aux contrats Σ_2^0 , ce qui ne présente pas de difficulté : une chaîne σ *force* un tel contrat si ce dernier est satisfait pour tout $G \in [\sigma]$. Nous verrons dans la section 4 que cette idée ne fonctionne plus pour les contrats Π_2^0 .

Proposition (4-8.2). Pour tout ensemble non calculable A , il existe un ensemble B tel que $B \not\prec_T A$ et $A \not\prec_T B$. ★

PREUVE. Soit A un ensemble non calculable. Nous voulons construire un ensemble B satisfaisant les contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$ et $(\mathcal{S}_e)_{e \in \mathbb{N}}$:

$$\mathcal{R}_e : \exists x \Phi_e^A(x) \uparrow \vee \exists x \Phi_e^A(x) \downarrow \neq B(x) \quad \mathcal{S}_e : \exists x \Phi_e^B(x) \uparrow \vee \exists x \Phi_e^B(x) \downarrow \neq A(x).$$

Soient $R_e \subseteq 2^{<\mathbb{N}}$ et $S_e \subseteq 2^{<\mathbb{N}}$ l'ensemble des chaînes forçant respectivement $\mathcal{R}_e$ et $\mathcal{S}_e$.

Densité de l'ensemble R_e . Soit $\sigma \in 2^{<\mathbb{N}}$, et soit $x = |\sigma|$. Deux cas se présentent.

- ▷ Cas 1. On a $\Phi_e^A(x) \downarrow = i$ pour $i \in \{0, 1\}$. Il suffit alors de définir τ comme l'unique chaîne de longueur $|\sigma| + 1$ étendant σ telle que $\tau(x) = 1 - i$. Pour tout $X \in [\tau]$, $X(x) = 1 - i \neq \Phi_e^A(x)$, donc $\tau \in R_e$.
- ▷ Cas 2. On a $\Phi_e^A(x) \uparrow$. Dans ce cas, $R_e = 2^{<\mathbb{N}}$, et $\sigma \in R_e$.

Densité de l'ensemble S_e . Soit $\sigma \in 2^{<\mathbb{N}}$. Trois cas se présentent.

- ▷ Cas 1. Il existe une entrée x et un ensemble $X \succeq \sigma$ tels que

$$\Phi_e^X(x) \downarrow \neq A(x).$$

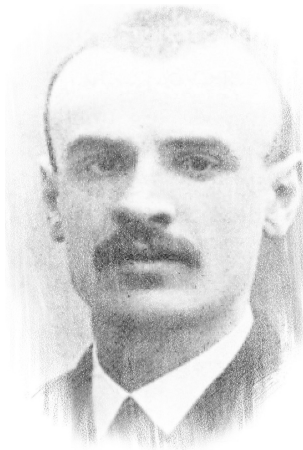
Dans ce cas, par la propriété de l'usage, il existe une chaîne finie $\tau \succeq \sigma$ telle que $\Phi_e^X(x) \downarrow \neq A(x)$ pour tout $X \in [\tau]$. La chaîne τ force donc le contrat $\mathcal{S}_e$, donc $\tau \in S_e$.

- ▷ Cas 2. Il existe une entrée x telle que pour tout ensemble $X \succeq \sigma$, on a $\Phi_e^X(x) \uparrow$. Dans ce cas, la chaîne σ force déjà le contrat $\mathcal{S}_e$ en s'assurant que $\Phi_e^B(x) \uparrow$. Donc, $\sigma \in S_e$.
- ▷ Cas 3. Aucun des deux cas précédents n'apparaît. Nous avons alors montré dans la preuve initiale de la proposition 4-8.2 que ce cas ne pouvait pas arriver, car l'ensemble A serait calculable, contrairement à notre hypothèse.

Soit B un ensemble $(R_e, S_e)_{e \in \mathbb{N}}$ -générique. Un tel ensemble existe par le théorème 2.4. En particulier, B satisfait tous les contrats $\mathcal{R}_e$ et $\mathcal{S}_e$ simultanément, donc $B \not\leq_T A$ et $A \not\leq_T B$. Cela conclut la preuve de la proposition 4-8.2. ■

2.2. L'approche topologique

La genèse de la méthode des extensions finies, de la généricité, et plus généralement du forcing, se trouve dans les travaux de René Baire, au début du XX^e siècle. Une des motivations de Baire est à l'époque de comprendre un peu mieux certaines fonctions « bizarres », mais qui surviennent naturellement en analyse. Si nous revenons un peu en arrière, dans la première moitié du XIX^e siècle émerge la prise de conscience, notamment aux travers des travaux de Cauchy et Bolzano, qu'une suite $(f_n : \mathbb{R} \rightarrow \mathbb{R})_{n \in \mathbb{N}}$ de fonctions continues qui converge point par point n'a pas nécessairement pour limite une fonction continue, et pour cause, une fois les notions de calculabilité correcte-



René Baire, 1874–1932

ment étendues pour travailler dans $\mathbb{R}$, les limites de fonctions *effectivement continues* — c'est-à-dire calculables — sont exactement les fonctions Δ_2^0 , c'est-à-dire celles pour lesquelles $f(r)$ est calculable uniformément en r' , le saut de r . Il s'agit d'une simple application du lemme de Schoenfield.

Quelques décennies plus tard, Baire se penche sur ce phénomène, et essaye de mieux comprendre ces fonctions que sont les limites de fonctions continues, et dont les irrégularités les rendent difficile à manipuler et même

à appréhender avec clarté. Il présentera ses résultats dans le cours Pecot « leçons sur les fonctions discontinues », où il développe son fameux outillage mathématique des *catégories de Baire* pour montrer notamment qu'une fonction limite de fonctions continues, si elle n'est plus forcément continue, le sera malgré tout sur un « grand ensemble de points ». Selon la terminologie moderne, les points de discontinuité d'une telle fonction seront une classe *maigre* ou encore de *catégorie 1*. Le théorème 3.20 à venir peut être vu comme une version effective de ce résultat.

Nous avons défini dans la section 8-2 la notion d'ouvert de l'espace de Cantor, comme étant une classe de la forme $\bigcup_{\sigma \in U} [\sigma]$ pour un ensemble $U \subseteq 2^{<\mathbb{N}}$ quelconque. La densité d'un ensemble de chaînes se traduit par une notion de densité de l'ouvert correspondant dans l'espace de Cantor.

Définition 2.7. Une classe $\mathcal{B} \subseteq 2^{\mathbb{N}}$ est dite *dense* si elle intersecte tout cylindre $[\sigma]$, c'est-à-dire $[\sigma] \cap \mathcal{B} \neq \emptyset$ pour tout $\sigma \in 2^{<\mathbb{N}}$. $\diamond$

L'exercice suivant lie la notion de densité sur les ouverts de l'espace de Cantor, et sur l'ordre partiel des chaînes binaires.

Exercice 2.8. Étant donné $U \subseteq 2^{<\mathbb{N}}$, on note $U^{\prec}$ la clôture par suffixe de U , c'est-à-dire $U^{\prec} = \{\tau \in 2^{<\mathbb{N}} : \exists \sigma \in U \tau \succeq \sigma\}$.

Soit $U \subseteq 2^{<\mathbb{N}}$. Montrer que $U^{\prec}$ est dense dans $2^{<\mathbb{N}}$ ssi l'ouvert $\bigcup_{\sigma \in U} [\sigma]$ est dense dans l'espace de Cantor. $\diamond$

Notre but est de définir une notion de classe « négligeable » ou *maigre* pour donner une définition topologique du forcing. On part de l'intuition qu'un cylindre $[\sigma]$ n'est pas maigre. La notion de classe négligeable devrait être close par sous-classe. Ainsi, si une classe contient un ouvert non vide de l'espace de Cantor, elle n'est pas négligeable. On introduit pour formaliser cela la notion d'intérieur.

Définition 2.9. On appelle *intérieur* d'une classe $\mathcal{F} \subseteq 2^{\mathbb{N}}$ — que l'on note $\text{int}(\mathcal{F})$ — le plus grand ouvert inclus dans $\mathcal{F}$, c'est-à-dire la réunion de tous les cylindres $[\sigma]$ tels que $[\sigma] \subseteq \mathcal{F}$. $\diamond$

Une classe négligeable doit donc en particulier posséder un intérieur vide.

Exemple 2.10. La classe $\{X \in 2^{\mathbb{N}} : \forall n \, X(2n) = 0\}$ est un fermé d'intérieur vide : il est en effet clair qu'elle ne contient aucun cylindre $[\sigma]$, car il existe toujours des $X \in [\sigma]$ tels que $X(2n) = 1$ pour n suffisamment grand. Son complémentaire $\{X \in 2^{\mathbb{N}} : \exists n \, X(2n) \neq 0\}$ est donc un ouvert dense, décrit par la réunion des cylindres $[\sigma 1]$ pour toute chaîne σ de taille paire.

Nous avons maintenant les éléments en main pour définir la notion de classe maigre.

Définition 2.11. Une classe $\mathcal{B} \subseteq 2^{\mathbb{N}}$ est dite *maigre* si $\mathcal{B}$ est incluse dans une réunion dénombrable de classes fermées d'intérieur vide. Le complémentaire d'une classe maigre est dite *co-maigre*. $\diamond$

Notons que, par passage au complémentaire, une classe est co-maigre si elle contient une intersection dénombrable d'ouverts denses. On laisse au lecteur le soin de montrer dans les deux exercices suivants que pour une suite d'ensembles $\vec{W} = (W_n)_{n \in \mathbb{N}}$ de chaînes denses, la classe des ensembles $\vec{W}$ -génériques est une classe co-maigre.

Exercice 2.12. Soit $\vec{W} = (W_n)_{n \in \mathbb{N}}$ une suite d'ensembles de chaînes telles que chaque $W_n^{\prec}$ est dense (en reprenant la notation de l'exercice 2.8).

Montrer que la classe $\bigcap_n [W_n]$ est co-maigre, où $[W_n] = \bigcup_{\sigma \in W_n} [\sigma]$. $\diamond$

Exercice 2.13. Soit $\vec{W} = (W_n)_{n \in \mathbb{N}}$ une suite d'ensembles de chaînes denses. Montrer que la classe $\bigcap_n [W_n]$ contient exactement les ensembles $\vec{W}$ -génériques. $\diamond$

Les deux exercices précédents établissent donc un lien entre l'approche du forcing par la méthode des extensions finies et l'approche topologique : si $(\mathcal{R}_n)_{n \in \mathbb{N}}$ est une suite de contrats tels que les ensembles de chaînes correspondants $(W_n)_{n \in \mathbb{N}}$ sont denses, alors la classe des ensembles $\vec{W}$ -génériques — qui satisfont tous les contrats simultanément — est co-maigre.

Digression

Historiquement, Baire appelle *classes de catégorie 1* les classes maigres, et *classes de catégorie 2* celles qui ne le sont pas. Cela donnera le nom de « théorie des catégories de Baire » à l'étude de ces notions. Notons qu'une classe n'est pas forcément maigre ou co-maigre. En particulier, les classes de catégorie 2 ne sont pas nécessairement co-maigres. En ce qui nous concerne, ce sont réellement les notions de maigre et co-maigre qui nous intéressent, et c'est donc ce vocabulaire que nous utiliserons.

Fait

D'après la définition 2.11, il est clair qu'une réunion dénombrable de classes maigres est maigre, et qu'une intersection dénombrable de classes co-maigres est co-maigre.

Une intuition que nous allons étayer dans les développements à venir est que les classes maigres sont « petites » et les classes co-maigres sont « grosses ». Il ne faut pas prendre l'attribution de ces adjectifs comme étant absolue. Il y a d'autres manières de juger de la taille des classes, qui ne coïncident en rien avec le fait que les maigres soit petites et les co-maigres grosses. On peut par exemple trouver des classes maigres de mesure 1 et des classes co-maigre de mesure 0 (voir la partie II).

Ce qu'il faut comprendre plutôt, c'est que les classes co-maigres sont suffisamment grosses pour toujours être stables par intersection dénombrable, et dans le même temps toujours denses, et même dans un sens fort : si $\mathcal{B}$ est une classe co-maigre, alors $\mathcal{B} \cap [\sigma]$ est indénombrable pour tout cylindre $[\sigma]$. Il s'agit en fait d'un renforcement du théorème 2.4. Pour s'en convaincre, se souvenir du fait suivant démontré avec la proposition 8-2.3 et la proposition 8-3.2.

Fait

L'intersection d'un nombre fini d'ouverts est un ouvert. Par conséquent, on peut toujours supposer qu'une intersection dénombrable d'ouverts $\bigcap_n \mathcal{U}_n$ est décroissante. Par passage au complémentaire, on peut toujours supposer qu'une réunion dénombrable de fermés est croissante.

Rappelons qu'une classe $\mathcal{F} \subseteq 2^{\mathbb{N}}$ est *parfaite* si elle est l'image d'une injection continue de $2^{\mathbb{N}}$ vers $2^{\mathbb{N}}$ (voir la section 8-2.4), c'est-à-dire que $\mathcal{F}$ est de la forme $[T]$ pour un f-arbre $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ (voir la section 7-5). Le lemme suivant montre qu'une classe co-maigre a la puissance du continu.

Lemme 2.14. Une classe co-maigre de l'espace de Cantor contient une sous-classe parfaite de points dans chaque cylindre $[\sigma]$. ★

PREUVE. Soit $\mathcal{B}$ une classe co-maigre, et soit $\bigcap_n \mathcal{U}_n \subseteq \mathcal{B}$ une intersection d'ouverts denses. On peut supposer sans perte de généralité que $\mathcal{U}_{n+1} \subseteq \mathcal{U}_n$. Le lecteur peut s'aider de la figure 2.2 pour une représentation graphique de la construction qui suit.

Soit $\sigma \in 2^{<\mathbb{N}}$. Comme l'ouvert $\mathcal{U}_0$ est dense, $\mathcal{U}_0 \cap [\sigma 0]$ est non vide, et il y a donc une chaîne $\sigma_0 \succ \sigma 0$ telle que $[\sigma_0] \subseteq \mathcal{U}_0$. De la même manière il y a une chaîne $\sigma_1 \succ \sigma 1$ telle que $[\sigma_1] \subseteq \mathcal{U}_0$. Supposons que pour toute chaîne τ de taille $n+1$ on ait défini des chaînes $\sigma_\tau \succeq \sigma$ deux à deux incomparables et telles que $[\sigma_\tau] \subseteq \mathcal{U}_n$. Pour chacune de ces chaînes τ et pour chaque $i \in \{0, 1\}$, on définit $\sigma_{\tau i}$ comme étant une chaîne qui étend $\sigma_\tau i$ et telle que $[\sigma_{\tau i}] \subseteq \mathcal{U}_{n+1}$, ce qui est possible car $\mathcal{U}_{n+1}$ est dense.

Il est clair que pour tout ensemble X la classe $\bigcap_{\tau \prec X} [\sigma_\tau] \subseteq \bigcap_n \mathcal{U}_n$ contient un unique élément $Y_X \in [\sigma] \cap \bigcap_n \mathcal{U}_n$. On vérifie sans peine que la fonction $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ définie par $T(\tau) = \sigma_\tau$ est un f-arbre dont les chemins sont les éléments $Y_X = T(\sigma_0) \prec T(\sigma_1) \prec \dots$ pour $X = \sigma_0 \prec \sigma_1 \prec \dots$.

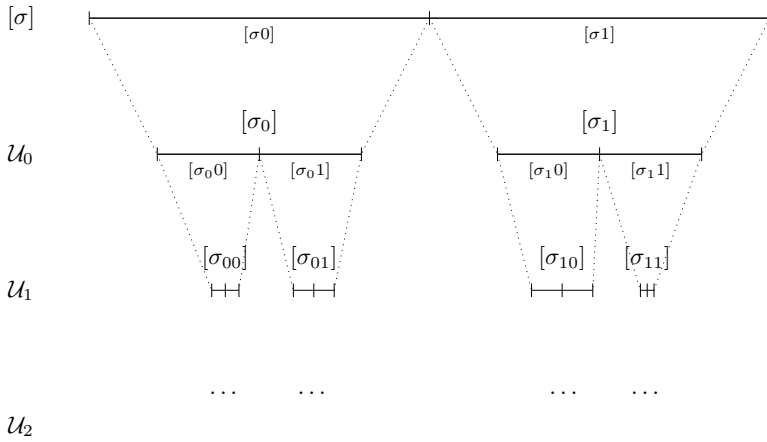


FIGURE 2.15 – *Illustration de la construction d’une sous-classe parfaite de points dans $[\sigma] \cap \bigcap_n \mathcal{U}_n$. Comme chaque ouvert $\mathcal{U}_n$ est dense, on peut trouver pour toute chaîne $\sigma_\tau i$ une extension $\sigma_{\tau i} \succeq \sigma_\tau i$ telle que $[\sigma_{\tau i}] \subseteq \mathcal{U}_n$.*

Ainsi, $\mathcal{B}$ contient une sous-classe parfaite de points dans le cylindre $[\sigma]$, ce qui achève la démonstration du lemme 2.14. ■

Notons en particulier comme conséquence du lemme précédent qu’une réunion dénombrable de fermés d’intérieur vide est d’intérieur vide : si une telle réunion de fermés contenait un cylindre $[\sigma]$, son complémentaire — une classe co-maigre — ne pourrait contenir de point dans $[\sigma]$, ce qui serait une contradiction.

Nous voyons à présent l’équivalent topologique de la définition 2.1 de forcing pour les contrats Σ_1^0 et Π_1^0 .

Définition 2.16. Soit $\mathcal{R}$ un contrat Σ_1^0 ou Π_1^0 . Soit $\mathcal{B}_{\mathcal{R}}$ la classe des éléments qui satisfont $\mathcal{R}$. Alors, σ force $\mathcal{R}$ ssi $[\sigma] \subseteq \mathcal{B}_{\mathcal{R}}$. Notons que si $\mathcal{R}$ est Σ_1^0 , alors $\mathcal{B}_{\mathcal{R}}$ est un ouvert effectif, et que si $\mathcal{R}$ est Π_1^0 , alors $\mathcal{B}_{\mathcal{R}}$ est un fermé effectif. ◇

Notons par exemple que si la classe des éléments satisfaisant un contrat Π_1^0 est d’intérieur vide, alors aucune chaîne ne force ce contrat : la classe des éléments ne satisfaisant pas le contrat est un ouvert dense, et contiendra tout élément suffisamment générique. Cela nous amène à l’étude des ensembles qui sont dans « suffisamment d’ouverts denses » : les 1-génériques et les faiblement 1-génériques, que nous voyons à présent.

3. Généricité effective

Jockusch fut sans doute l'un des premiers à comprendre l'utilité que pouvaient avoir les idées de Cohen en calculabilité, et il initia l'étude d'une version effective des concepts de Cohen, avec les notions de 1-généricité et 1-généricité faible, résultant de l'application du forcing à tous les contrats Σ_1^0 et Π_1^0 .

La généricité peut être vue à la fois comme une notion de force et de faiblesse. Un ensemble suffisamment générique sera par exemple toujours de degré hyperimmune. En revanche, nous verrons que les ensembles suffisamment génériques ne peuvent pas calculer l'arrêt, ni même de fonction DNC. De manière générale, la méthode des extensions finies satisfait des propriétés de force et de faiblesse de la même manière : en prouvant la densité de certains ensembles bien choisis.

3.1. Ensembles faiblement 1-génériques

Nous commençons par présenter les ensembles faiblement 1-génériques, introduits par Kurtz durant sa thèse de doctorat, effectuée sous la direction de Jockusch.

Définition 3.1 (Kurtz [126]). Un ensemble $G \in 2^{\mathbb{N}}$ est *faiblement 1-générique* s'il est générique pour tous les ensembles c. e. denses. Autrement dit, G est faiblement 1-générique si G appartient à toutes les classes Σ_1^0 denses de l'espace de Cantor. $\diamond$

Notons qu'il n'existe qu'une quantité dénombrable de classes Σ_1^0 denses. La notion d'ensemble faiblement 1-générique s'avère n'être pas assez restrictive pour receler des propriétés normalement inhérentes aux génériques, mais en termes de degré Turing, la notion présente un certain intérêt, notamment via la caractérisation suivante.

Théorème 3.2 (Kurtz [126])

Soit $G \subseteq \mathbb{N}$. Les énoncés suivants sont équivalents.

- (1) G est de degré hyperimmune.
- (2) G calcule une fonction qui est égale infiniment souvent à toute fonction calculable.
- (3) G calcule un ensemble faiblement 1-générique.

PREUVE. Montrons d'abord (1) $\rightarrow$ (3), l'implication la plus difficile.

Soit $f \leq_T G$ une fonction qui n'est bornée par aucune fonction calculable. On suppose sans perte de généralité que f est croissante. Notons que pour toute fonction calculable g , il y a une infinité de valeurs n telles

que $f(n) > g(n)$. On calcule à partir de la fonction f un ensemble $G \in 2^{\mathbb{N}}$ qui appartient à toute classe Σ_1^0 dense. Soit $(W_e)_{e \in \mathbb{N}}$ une énumération des sous-ensembles Σ_1^0 de $2^{<\mathbb{N}}$. On construit G par approximations successives $\sigma_0 \preceq \sigma_1 \preceq \sigma_2 \preceq \dots$.

On décrit d'abord une procédure récursive à effectuer à chaque fois que l'on veut concaténer une chaîne τ à une chaîne σ que l'on a jusqu'à présent calculé. Cette procédure, que nous nommerons R , prend un troisième paramètre : un entier e qui correspond au plus petit entier tel que $\sigma\tau$ est énuméré dans W_e à l'étape de calcul $f(|\sigma|)$. On notera $R(\sigma, \tau, e)$ pour le résultat de l'appel à cette procédure. Notons enfin que certains entiers sont marqués comme « satisfaits » au moment où la procédure est appelée : ce sont les entiers e tels que σ étend une chaîne de W_e à l'étape de calcul courante.

La procédure $R(\sigma, \tau, e)$ fait la chose suivante : pour chaque préfixe $\tau' \preceq \tau$ dans l'ordre, elle cherche le plus petit entier $e' < e$ qui n'est pas satisfait et tel qu'une chaîne de la forme $\sigma\tau'\rho$ soit énumérée dans $W_{e'}[f(|\sigma\tau'|)]$. Si un tel entier est trouvé, la procédure renvoie alors le résultat de l'appel récursif à $R(\sigma\tau', \rho, e')$. Sinon, elle renvoie $\sigma\tau$. Notons que la diminution de la valeur du dernier paramètre dans les appels récursifs fait que la procédure s'arrête nécessairement.

À l'étape 0, on définit $\sigma_0 = \epsilon$. Supposons σ_t défini à l'étape t . À l'étape $t+1$, on cherche le plus petit entier $e \leq t+1$ non satisfait tel qu'une chaîne de la forme $\sigma_t\tau$ soit énumérée dans $W_e[f(|\sigma_t|)]$. Si l'on trouve un tel entier e , on définit σ_{t+1} comme étant $R(\sigma_t, \tau, e)$. Sinon, σ_{t+1} comme étant σ_0 . Cela conclut la construction.

Notons que si W_e décrit un ouvert dense, alors la fonction f_e qui à n associe le plus petit temps de calcul t tel que toutes les chaînes de taille n ont une extension dans $W_e[t]$ est une fonction calculable et totale. On a en particulier $f_e(n) < f(n)$ pour une infinité de valeurs n . Supposons que W_e décrit un ouvert dense, que e n'est pas satisfait au temps t et que tous les $e' < e$ qui sont satisfait à un moment de la construction sont satisfaits au temps t . Soit n le plus petit entier supérieur ou égal à $|\sigma_t|$ tel que $f(n) > f_e(n)$. Soit $s \geq t$ le plus petit entier tel que $|\sigma_s| \leq n < |\sigma_{s+1}|$. Si $|\sigma_s| = n$, alors par minimalité de e l'algorithme définit $\sigma_{s+1} = \sigma_s\tau$ avec $\sigma_s\tau \in W_e[f(n)]$. Sinon alors, par construction, au moment de définir $\sigma_{s+1} = \sigma_s\tau$ pour une certaine chaîne τ , l'algorithme vérifie pour tout préfixe $\tau' \preceq \tau$, que l'on n'a pas une extension de $\sigma_s\tau'$ énumérée dans $W_e[f(|\sigma_s\tau'|)]$, et en particulier pour le préfixe τ' tel que $|\sigma_s\tau'| = n$. Si c'est le cas, l'algorithme est relancé sur cette extension. Comme c'est effectivement le cas par hypothèse, et par minimalité de e , on aura en fait $\sigma_s\tau \in W_e[f(n)]$ pour $\sigma_{s+1} = \sigma_s\tau$. On en conclut que l'ensemble $G = \sigma_0 \prec \sigma_1 \prec \sigma_2 \prec \dots$ appartient à tous les ouverts denses.

Montrons l'implication (3) $\rightarrow$ (2).

Soit G un ensemble faiblement 1-générique. Notons que si f est une fonction totale calculable alors l'ensemble $W_f = \{\sigma 0^{f(|\sigma|)} 1 : \sigma \in 2^{<\mathbb{N}}\}$ décrit une classe Σ_1^0 dense. On calcule à partir de G la fonction g qui à n associe le nombre maximal $g(n)$ de 0 tel que $G \upharpoonright_n 0^{g(n)} \prec G$. Comme pour toute fonction totale calculable f l'ensemble G appartient à W_f , il est clair que g est égale au moins une fois à toute fonction calculable. Si g n'était égale qu'un nombre fini de fois à une fonction calculable donnée, on pourrait modifier un nombre fini de valeurs de cette fonction pour avoir une fonction calculable qui n'est jamais égale à g . Donc, g est égale infiniment souvent à toute fonction calculable.

Montrons enfin (2) $\rightarrow$ (1).

Soit f une fonction égale infiniment souvent à toute fonction calculable. Alors, $f + 1$ est infiniment souvent au-dessus de toute fonction calculable. ■

On peut relativiser la notion de 1-généricité faible à n'importe quel oracle.

Définition 3.3 (Kurtz [126]). Soit $A \in 2^{\mathbb{N}}$. Un ensemble G est *faiblement 1-générique relativement à A* si G rencontre W pour tout ensemble de chaînes $\Sigma_1^0(A)$ dense W . ◇

La hiérarchie des sauts Turing permet de définir une hiérarchie de généricité comme suit.

Définition 3.4 (Kurtz [126]). Un ensemble $G \in 2^{\mathbb{N}}$ est *faiblement n -générique* s'il est faiblement 1-générique relativement à $\emptyset^{(n-1)}$, c'est-à-dire s'il rencontre tous les ensembles de chaînes $\Sigma_1^0(\emptyset^{(n-1)})$ denses, ou encore tous les ensembles de chaînes Σ_n^0 denses. ◇

Certaines implications du théorème 3.2 se généralisent à tout oracle A .

Exercice 3.5. Montrer que tout ensemble G faiblement 1-générique relativement à A calcule une fonction $g : \mathbb{N} \rightarrow \mathbb{N}$ qui est égale infiniment souvent à toute fonction A -calculable. ◇

Exercice 3.6. Soit $n \geq 1$, et soit $G \in 2^{\mathbb{N}}$. Montrer que si G calcule une fonction qui est égale infiniment souvent à toutes les fonctions A -calculables, alors G calcule une fonction A -hyperimmune. ◇

La direction (1) $\rightarrow$ (3) du théorème 3.2 ne se généralise pas en règle générale. Dans la hiérarchie des sauts Turing, cela fonctionne uniquement à la première étape.

Proposition 3.7 (Andrews, Gerdes et Miller [7]).

Toute fonction hyperimmune relativement à $\emptyset'$ calcule un ensemble faiblement 2-générique. ★

L'idée pour montrer la proposition précédente est d'utiliser le fait que les objets $\emptyset'$ -calculables sont calculables à la limite. En revanche, à partir de $n \geq 3$, les ensembles faiblement n -génériques ne peuvent plus être construits simplement à l'aide de fonctions échappant à des collections de fonctions, au sens suivant.

Définition 3.8. Soit $\mathcal{F}$ une collection de fonctions $\mathbb{N} \rightarrow \mathbb{N}$. Une fonction f est $\mathcal{F}$ -échappante si pour tout $g \in \mathcal{F}$, il existe un entier n tel que $f(n) > g(n)$. ◇

Le théorème suivant exprime une profonde différence structurelle entre les fonctions échappantes et les degrés génériques, au sens où quel que soit un oracle A , il existe une fonction A -hyperimmune qui ne calcule pas d'ensemble faiblement 3-générique.

Théorème 3.9 (Andrews, Gerdes et Miller [7])

Pour toute collection dénombrable de fonctions $\mathcal{F}$, il existe une fonction $\mathcal{F}$ -échappante qui n'est pas de degré faiblement 3-générique.

PREUVE. Nous utiliserons une variante de la notion de f -arbre définie dans la section 7-5. Nous allons définir une fonction Δ_3^0 totale $T : \mathbb{N}^{<\mathbb{N}} \rightarrow \mathbb{N}^{<\mathbb{N}}$ telle que :

- (1) $\text{dom } T$ (le domaine de T) est un ensemble clos par préfixe ;
- (2) pour tous $\sigma, \tau \in \text{dom } T$, $\sigma \preceq \tau$ si et seulement si $T(\sigma) \preceq T(\tau)$;
- (3) pour tout $\sigma \in \mathbb{N}^{<\mathbb{N}}$ et $n \in \mathbb{N}$, il existe un $m \geq n$ tel que $T_s(\sigma n)$ étend $T_s(\sigma)m$.

La fonction T s'étend en une fonction de $\mathbb{N}^{\mathbb{N}}$ vers $\mathbb{N}^{\mathbb{N}}$, en définissant pour tout $X \in \mathbb{N}^{\mathbb{N}}$, $T(X)$ comme l'unique élément de la classe $\bigcap_{\sigma \prec X} [T(\sigma)]$. Un *chemin* de T est une suite $P \in \mathbb{N}^{\mathbb{N}}$ dont une infinité de segments initiaux appartiennent à $\text{Im } T$. Autrement dit, un chemin est une suite $P \in 2^{\mathbb{N}}$ de la forme $P = T(X)$ pour un $X \in \mathbb{N}^{\mathbb{N}}$. On notera $[T]$ l'ensemble des chemins de T . Par (3), pour toute collection dénombrable de fonctions $\mathcal{F}$, il existe un chemin $f \in [T]$ qui est $\mathcal{F}$ -échappant. Nous allons construire T de telle sorte que pour tout chemin $f \in [T]$, f n'est pas de degré faiblement 3-générique.

Le lecteur peut s'aider de la figure 3.10 pour comprendre la construction qui suit. Soit $(\sigma_s)_{s \in \mathbb{N}}$ une énumération calculable de $\mathbb{N}^{<\mathbb{N}}$ qui ne fait apparaître chaque chaîne qu'après avoir énuméré ses préfixes. En particulier, $\sigma_0 = \epsilon$. Au début de l'étape s , nous aurons déjà défini T sur $\sigma_0, \dots, \sigma_s$.

Nous supposons également défini pour chaque $t \leq s$, un réservoir c. e. infini $V_{t,s} \subseteq \mathbb{N}^{<\mathbb{N}}$ de chaînes étendant $T(\sigma_s)$. Nous allons nous assurer que pour tout $n \in \mathbb{N}$, $T(\sigma_s n)$ étend une chaîne de $V_{t,s}$. Simultanément, nous allons créer pour chaque e un ensemble $\Sigma_1^0(\emptyset'')$ dense $U_e \subseteq 2^{<\mathbb{N}}$ tel que si Φ_e^P est totale pour un chemin $P \in [T]$, alors P ne rencontre pas U_e . Ainsi, quel que soit le chemin $P \in [T]$, Φ_e^P ne sera pas un ensemble faiblement 3-générique.

Initialement, $f(\epsilon) = \epsilon$ et $V_{0,0} = \mathbb{N}$. À l'étape $s = \langle e, i \rangle$, nous allons nous assurer que toute chaîne de longueur i a une extension dans U_e . L'ensemble $\{\sigma_t : t \leq s\}$ forme un arbre fini, et pour chaque feuille τ de cet arbre, $\Phi_e^{T(\tau)}$ peut avoir des valeurs différentes. L'ensemble U_e doit donc ajouter des extensions à toutes les chaînes de longueur i , tout en s'assurant qu'elle évitera $\Phi_e^{T(\sigma_t)}$ pour tout $t \leq s$. Nous fixons donc une longueur suffisamment grande, $k = i + s + 1$, de telle sorte que si l'on construit un ensemble de chaînes binaires « interdites » $\rho_0, \dots, \rho_s$ chacune de longueur k , toute chaîne binaire de longueur i admet une extension de longueur k évitant les chaînes interdites.

Pour chaque $t \leq s$, on demande à $\emptyset''$ s'il existe une chaîne ρ_s de taille k telle qu'il existe une infinité de chaînes binaires $\mu \in V_{t,s}$ ayant une extension $\mu' \succeq \mu$ pour laquelle $\Phi_e^{\mu'} \upharpoonright_k = \rho_s$. Si c'est le cas, on définit $V_{t,s+1}$ comme l'ensemble de ces μ' . Notons que $V_{t,s+1}$ est alors encore calculatoirement énumérable. Sinon, pour toute chaîne ρ de taille k , seul un nombre fini de chaînes de $V_{t,s}$ possède une extension qui sera envoyée vers une extension de ρ via Φ_e . En particulier, on peut enlever un nombre fini de chaînes de $V_{t,s}$ de manière à ce qu'aucune extension des chaînes restantes ne sera jamais envoyé vers une chaîne de taille plus grande que k . On prend alors une chaîne ρ_s arbitraire, et l'on définit $V_{t,s+1}$ comme la restriction de $V_{t,s}$ aux chaînes μ qui n'ont pas d'extension μ' envoyée vers une chaîne de taille plus grande que k via Φ . Comme nous retirons un nombre fini de chaînes, $V_{s,t+1}$ est encore c. e.

Nous nous retrouvons donc avec un ensemble de chaînes binaires $\rho_0, \dots, \rho_s$, chacune de longueur k , de telle sorte que pour tout chemin $P \in [T]$, si Φ_e^P est total, alors $\Phi_e^P \upharpoonright_k = \rho_t$ pour un $t \leq s$. On énumère dans U_e toutes les chaînes de longueur k autres que $\rho_0, \dots, \rho_s$. En excluant ces chaînes, on s'assure que pour tout chemin $P \in [T]$, si Φ_e^P est total, alors il ne rencontrera pas U_e . La longueur k étant suffisamment grande, toute chaîne de longueur i a une extension dans U_e .

Enfin, nous définissons $T(\sigma_{s+1})$. Supposons que $\sigma_{s+1} = \sigma_t n$ pour un $t \leq s$ et $n \in \mathbb{N}$. On choisit $\tau \in V_{t,s+1}$, on le retire de $V_{t,s+1}$, et l'on pose $T(\sigma_{s+1}) = \tau$. Enfin, on définit $V_{s+1,s+1} = \{\tau m : m \in \mathbb{N}\}$. Cela termine la preuve du théorème 3.9. ■

Ci-dessous, l'énumération $(\sigma_s)_{s \in \mathbb{N}}$ commence par $\epsilon, 0, 1, 2, 00, 01, 10, 11, \dots$

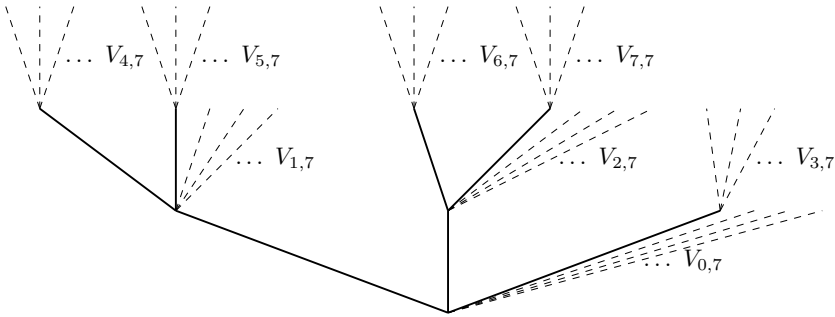


FIGURE 3.10 – Illustration de la preuve : à l'étape 8, on va restreindre chacun des réservoirs de $V_{0,7}$ à $V_{7,7}$ en supprimant certaines de leurs branches, et en étendant d'autres, de manière à ce que la fonctionnelle courante Φ_e évite, pour chacune des branches des réservoirs $V_{i,8}$, un ouvert dense que l'on est en train d'énumérer à l'aide du double saut. Une fois les réservoirs restreints, on complète notre arbre en prenant une extension dans chacun d'entre eux, ce qui crée pour chaque extension un nouveau réservoir, et ainsi de suite.

Exercice 3.11. (★★) Modifier la preuve du théorème précédent pour montrer que pour toute collection dénombrable de fonctions $\mathcal{F}$, il existe une fonction $\mathcal{F}$ -échappante qui ne calcule aucune fonction qui est égale infiniment souvent à toute fonction $\emptyset''$ -calculable. $\diamond$

3.2. Ensembles 1-génériques

Nous voyons à présent la 1-généricité, une notion un peu plus forte et beaucoup plus riche que la 1-généricité faible. Les ensembles faiblement 1-génériques sont ceux résultant du forcing pour les contrats Σ_1^0 denses. Mais qu'en est-il des contrats Σ_1^0 qui ne sont pas denses ?

La 1-généricité correspond au premier niveau de forcing permettant de s'attaquer aux contrats Σ_1^0/Π_1^0 quelconques, via le théorème suivant, qui sera démontré à la fin de la sous-section qui suit (nous renvoyons le lecteur à la définition 2.1 pour la notation $\Vdash^*$).

Théorème 3.12

Soit G un ensemble 1-générique. Soit $\mathcal{R}$ un contrat Σ_1^0 ou Π_1^0 . Alors, G satisfait $\mathcal{R}$ si, et seulement si, il existe un préfixe $\sigma \prec G$ tel que $\sigma \Vdash^* \mathcal{R}$.

3.2.1. Nature des ensembles 1-génériques

Définition 3.13. Une chaîne $\sigma \in 2^{<\mathbb{N}}$ *évite* un ensemble $W \subseteq 2^{<\mathbb{N}}$ (noté $\sigma \perp W$) si non seulement $\sigma \notin W$, mais également aucune extension de σ n'est dans W . $\diamond$

Notation

On note $W^\perp = \{\tau \in 2^{<\mathbb{N}} : \tau \perp W\}$.

Lemme 3.14. Soit $W \subseteq 2^{<\mathbb{N}}$ un ensemble arbitraire.

Alors, l'ensemble $W \cup W^\perp$ est dense. $\star$

PREUVE. Soit $\sigma \in 2^{<\mathbb{N}}$. Soit σ possède une extension dans W , et donc dans $W \cup W^\perp$, soit σ évite W , auquel cas $\sigma \in W^\perp$. $\blacksquare$

Notons que si W est un ensemble dense, alors $W^\perp = \emptyset$. Souvenons-nous que la densité d'un ensemble de chaînes W clos par suffixe sur $2^{<\mathbb{N}}$ correspond à la densité sur $2^{\mathbb{N}}$ de son ouvert correspondant $[W] = \bigcup_{\sigma \in W} [\sigma]$. L'ensemble $W^\perp$ correspond également à un ouvert : l'intérieur du complémentaire de l'ensemble $[W]$, c'est-à-dire la réunion de tous les cylindres $[\sigma]$ inclus dans le complémentaire de $[W]$.

Si on le reformule dans l'espace de Cantor, le lemme 3.14 énonce que pour n'importe quelle classe ouverte $\mathcal{U} \subseteq 2^{\mathbb{N}}$, la classe $\mathcal{U} \cup \text{int}(2^{\mathbb{N}} \setminus \mathcal{U})$ est un ouvert dense. Si $\mathcal{U}$ est à la base dense, alors $\text{int}(2^{\mathbb{N}} \setminus \mathcal{U}) = \emptyset$, sinon on « densifie » $\mathcal{U}$ en y ajoutant l'intérieur de son complémentaire. Nous sommes maintenant prêts à définir la notion d'ensemble 1-générique.

Définition 3.15 (Jockusch [100]). Un ensemble $G \in 2^{\mathbb{N}}$ est 1-générique s'il est $\{W_e \cup W_e^\perp : e \in \mathbb{N}\}$ -générique, où W_e est l'ensemble de chaînes calculatoirement énumérable de code e . De manière équivalente, G est 1-générique si $G \in \mathcal{U} \cup \text{int}(2^{\mathbb{N}} \setminus \mathcal{U})$ pour toute classe Σ_1^0 . $\diamond$

Comme signalé plus haut, si $W \subseteq 2^{<\mathbb{N}}$ est un ensemble dense, alors $W^\perp = \emptyset$. Il s'ensuit que tout ensemble 1-générique rencontre tout ensemble Σ_1^0 dense, et est donc faiblement 1-générique. En particulier, les degrés des ensembles faiblement 1-génériques coïncidant avec les degrés hyperimmunes, tout ensemble 1-générique est de degré hyperimmune, et donc non calculable.

Si l'on regarde la contraposée de la notion de 1-généricité, un ensemble G n'est pas 1-générique s'il existe un ensemble c.e. $W \subseteq 2^{<\mathbb{N}}$ ne contenant aucun préfixe de G et tel que $W \ll$ est dense le long de $G \gg$, c'est-à-dire que pour tout $\sigma \prec G$ il existe $\tau \succeq \sigma$ avec $\tau \in W$ et $\tau \not\prec G$. Cette idée est illustrée dans la figure 3.16.

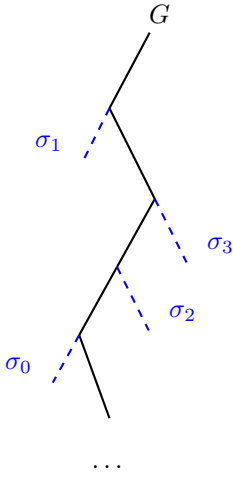


FIGURE 3.16 – Une illustration d'un ensemble G qui n'est pas 1-générique : on peut énumérer des chaînes $\sigma_0, \sigma_1, \dots$ densément le long de G , et sans jamais énumérer un préfixe de G .

Nous voyons à présent formellement pourquoi le théorème 3.12 annoncé est vrai.

Proposition 3.17. Soit $\mathcal{R}_e$ un contrat Σ_1^0 . Alors, l'ensemble $W \subseteq 2^{<\mathbb{N}}$ des chaînes forçant $\mathcal{R}_e$ est Σ_1^0 . De plus, $W^\perp$ est l'ensemble des chaînes forçant $\neg\mathcal{R}_e$ et il est Π_1^0 . ★

PREUVE. Le contrat $\mathcal{R}_e$ étant Σ_1^0 , il peut s'écrire de la forme $\Phi(G, 0)\downarrow$ pour une fonctionnelle Φ . Une chaîne σ force $\mathcal{R}_e$ si $\Phi(X, 0)\downarrow$ pour tout $X \in [\sigma]$. Par le lemme de König, σ force $\mathcal{R}_e$ s'il existe une longueur n telle que pour tout $\tau \succeq \sigma$ de longueur n , $\Phi(\tau, 0)\downarrow$. L'ensemble

$$W = \{\sigma \in 2^{<\mathbb{N}} : \exists n \forall \tau \in 2^n (\tau \succeq \sigma \rightarrow \Phi(\tau, 0)\downarrow)\}$$

est Σ_1^0 (où 2^n désigne l'ensemble des chaînes de taille n).

Montrons que $W^\perp$ est l'ensemble des chaînes σ qui forcent $\neg\mathcal{R}_e$. Si σ ne force pas $\neg\mathcal{R}_e$, alors il existe un $X \in [\sigma]$ tel que $\Phi(X, 0)\downarrow$. Par la propriété de l'usage, pour n suffisamment grand et pour tout $Y \in [X \upharpoonright_n]$, $\Phi(Y, 0)\downarrow$. On peut supposer $n \geq |\sigma|$. Soit $\tau = X \upharpoonright_n$. Alors, $\tau \in W$, donc $\sigma \notin W^\perp$. Par contraposée, si $\sigma \in W^\perp$, alors σ force $\neg\mathcal{R}_e$. Supposons maintenant que $\sigma \notin W^\perp$. Alors, il existe $\tau \succeq \sigma$ tel que $\tau \in W$. En particulier, il existe un $X \in [\tau] \subseteq [\sigma]$ tel que $\Phi(X, 0)\downarrow$. Il s'ensuit que σ ne force pas $\neg\mathcal{R}_e$.

Comme l'ensemble W est Σ_1^0 , il est clair que l'ensemble $W^\perp$ est Π_1^0 . ■

Corollaire 3.18

Soit $\mathcal{R}_e$ un contrat Σ_1^0 . Alors, l'ensemble D des chaînes qui forcent $\mathcal{R}_e$ ou qui forcent $\neg\mathcal{R}_e$ est dense.

PREUVE. Immédiat par le lemme 3.14 et la proposition 3.17. ■

Notons que tout ensemble satisfait la formule $\ll \mathcal{R}_e \vee \neg \mathcal{R}_e \gg$, et donc que toute chaîne force $\ll \mathcal{R}_e \vee \neg \mathcal{R}_e \gg$. En revanche, il est beaucoup plus fort pour une chaîne σ de forcer $\mathcal{R}_e$ ou de forcer $\neg \mathcal{R}_e$, car tout ensemble $A \in [\sigma]$ doit avoir le même comportement vis-à-vis de $\mathcal{R}_e$.

On peut à présent montrer le théorème 3.12 annoncé : si G est un ensemble 1-générique et $\mathcal{R}$ un contrat Σ_1^0 ou Π_1^0 , alors G satisfait $\mathcal{R}$ ssi il existe un préfixe $\sigma \prec G$ tel que $\sigma \Vdash^* \mathcal{R}$.

PREUVE DU THÉORÈME 3.12. Soit G un ensemble 1-générique. Soit $\mathcal{R}$ un contrat Σ_1^0 . Si un préfixe $\sigma \prec G$ force $\mathcal{R}$ ou $\neg \mathcal{R}$, alors par définition G satisfait respectivement $\mathcal{R}$ ou $\neg \mathcal{R}$.

Réciproquement, supposons que G satisfasse $\mathcal{R}$. Soit W l'ensemble c. e. des chaînes qui forcent $\mathcal{R}$. Comme G est 1-générique, il rencontre $W \cup W^\perp$. Si G rencontre $W^\perp$, alors σ force $\neg \mathcal{R}$ pour un préfixe $\sigma \prec G$, et G satisfait donc $\neg \mathcal{R}$, ce qui est une contradiction. Ainsi, G rencontre W , et un préfixe $\sigma \prec G$ force donc $\mathcal{R}$.

Symétriquement, si G satisfait $\neg \mathcal{R}$, alors un préfixe $\sigma \prec G$ force $\neg \mathcal{R}$. ■

3.2.2. Propriétés des ensembles 1-génériques

De manière générale, la fonction qui à X associe X' n'est pas continue. En effet, il est parfois nécessaire de connaître une infinité de bits de X pour savoir si $n \in X'$. René Baire a montré — sous une autre forme bien sûr — que cette fonction était en revanche continue sur une classe co-maigre de points. La 1-généricité est le niveau de généricité requis pour rendre compte de ce théorème.

Définition 3.19. Un ensemble $G \in 2^\mathbb{N}$ est *low généralisé* si

$$G' \leq_T G \oplus \emptyset'.$$

◇

Si un ensemble G est low généralisé, alors non seulement la fonction qui à X associe X' est continue en G , mais plus encore elle est calculable en $G \oplus \emptyset'$. Un ensemble quelconque n'a *a priori* aucune raison d'être low généralisé. À titre d'exemple, $\emptyset'$ ne l'est pas, mais c'est le cas pour les ensembles 1-génériques.

Théorème 3.20

Les ensembles 1-génériques sont low généralisés.

PREUVE. Pour tout entier e , on définit la classe $\mathcal{U}_e = \{X : \Phi_e(X, e) \downarrow\}$. Soit $W_e \subseteq 2^{<\mathbb{N}}$ un ensemble Σ_1^0 qui représente $\mathcal{U}_e$, c'est-à-dire tel que

$$[W_e] = \mathcal{U}_e, \quad \text{où } [W_e] = \bigcup_{\sigma \in W_e} [\sigma].$$

Notons que $e \in G'$ ssi $G \in \mathcal{U}_e$. On a par ailleurs $G \in \mathcal{U}_e$ ssi il existe $\tau \preceq G$ tel que $\tau \in W_e$, et par définition de la 1-généricité de G on a $G \notin \mathcal{U}_e$ ssi il existe $\tau \preceq G$ tel que $\sigma \notin W_e$ pour tout σ comparable avec τ . La question de savoir si $\sigma \notin W_e$ pour tout σ comparable avec τ est Π_1^0 uniformément en τ , et donc peut être posée à $\emptyset'$. Pour savoir si $e \in G'$, il suffit donc de chercher un préfixe τ de G tel que l'on est dans un cas ou dans l'autre, ce qui arrivera nécessairement. ■

Nous avons vu que tout ensemble 1-générique était de degré hyperimmune, et donc non calculable. Il ne s'agit donc pas d'une propriété de faiblesse. Nous allons maintenant voir que ce n'est pas non plus une notion de force, au sens où certaines puissances calculatoires ne pourront jamais être atteintes par les degrés 1-génériques. En particulier, aucun degré 1-générique ne calcule $\emptyset'$.

Théorème 3.21 (Demuth et Kučera [45])

Un ensemble 1-générique n'est pas de degré DNC.

PREUVE. Supposons que $n \mapsto \Phi(G, n)$ soit une fonction DNC. On considère la classe $\mathcal{U} = \{X : \exists n \Phi(X, n) \downarrow = \Phi_n(n) \downarrow\}$. Par hypothèse, $G \notin \mathcal{U}$. Considérons une chaîne σ . On définit par le théorème de point fixe de Kleene le code e_σ de la fonction qui l'entrée e_σ cherche une extension $\tau_\sigma \succeq \sigma$ telle que $\Phi(\tau_\sigma, e_\sigma) \downarrow$ et assigne $\Phi(\tau_\sigma, e_\sigma)$ à $\Phi_{e_\sigma}(e_\sigma)$. Le processus étant uniforme, on énumère toutes les chaînes de la forme τ_σ dans un ensemble c. e. W .

Notons que pour tout $\sigma \prec G$ la fonction de code e_σ s'arrête sur l'entrée e_σ , car il existe au moins une extension de σ — à savoir G lui-même — pour laquelle Φ s'arrête sur l'entrée e_σ . Comme $n \mapsto \Phi(G, n)$ est DNC, on a dès lors $\Phi(G, e_\sigma) \neq \Phi_{e_\sigma}(e_\sigma)$, et donc $\sigma \prec \tau_\sigma \not\prec G$.

On a ainsi un ensemble c. e. W dont aucun élément n'est un préfixe de G , mais qui contient une extension de chaque préfixe de G . Il s'ensuit que G n'est pas 1-générique. ■

La restriction de la généricité permet de construire des ensembles bénéficiant de certains avantages de la généricité tout en n'étant pas trop complexes d'un point de vue calculatoire.

Exercice 3.22. (★) Montrer par un argument direct qu'aucun ensemble 1-générique n'est calculable. ◇

Exercice 3.23. (★) Montrer qu'il existe un ensemble 1-générique Δ_2^0 . En déduire qu'il existe un ensemble 1-générique de degré low. ◇

Nous allons voir dans la section suivante avec le corollaire 3.34 qu'il existe aussi des ensembles 1-génériques de degrés high. Terminons par un exercice intéressant, qui demande d'élaborer sur les techniques du théorème 3.28, et qui utilise la notion de jointure effective de la définition 4-5.6, mais étendue à une suite dénombrable d'ensembles :

Définition 3.24. La *jointure effective* $\bigoplus_{n \in \mathbb{N}} A_n$ d'une suite $(A_n)_{n \in \mathbb{N}}$ d'ensembles est l'ensemble Y tel que $\langle n, m \rangle \in Y$ ssi $m \in A_n$. $\diamond$

Dans les exercices qui suivent, la notation $\bigoplus_{j \neq i} G_j$ correspond donc à la jointure effective de la suite $(G_n)_{n \in \mathbb{N}}$ de laquelle on enlève l'ensemble G_i .

Exercice 3.25. (**) Soit $G = \bigoplus_{n \in \mathbb{N}} G_n$ un ensemble 1-générique. Montrer que G_i est hyperimmune relativement à $\bigoplus_{j \neq i} G_j$ pour tout $i \in \mathbb{N}$. $\diamond$

Exercice 3.26. (**) (Miller). Soit $X = \bigoplus_{n \in \mathbb{N}} X_n$ un ensemble tel que X_i est hyperimmune relativement à $\bigoplus_{j \neq i} X_j$ pour tout $i \in \mathbb{N}$. Montrer que X calcule un ensemble 1-générique. $\diamond$

3.2.3. Relativisation des ensembles 1-génériques

Tout comme on a relativisé la 1-généricité faible, on relativise à présent la 1-généricité.

Définition 3.27 (Jockusch [100]). Soit $A \in 2^{\mathbb{N}}$. Un ensemble $G \in 2^{\mathbb{N}}$ est 1-générique relativement à A si $G \in \mathcal{U} \cup \text{int}(2^{\mathbb{N}} \setminus \mathcal{U})$ pour toute classe $\mathcal{U}$ qui est $\Sigma_1^0(A)$. On dira que G est n -générique s'il est 1-générique relativement à $\emptyset^{(n-1)}$, ou de manière équivalente s'il rencontre $W \cup W^\perp$ pour tout W ensemble de chaîne Σ_n^0 . $\diamond$

Nous étudierons cette relativisation plus en détail dans la section 5. Voyons pour le moment un premier théorème clef.

Théorème 3.28

Soit X non calculable, et soit G un ensemble 1-générique relativement à X . Alors, $G \not\leq_T X$.

PREUVE. Soit Φ une fonctionnelle Turing. Nous construisons $\mathcal{U}$, une classe $\Sigma_1^0(X)$ telle que $Y \in \mathcal{U} \cup \text{int}(\mathbb{N} \setminus \mathcal{U})$ implique $\Phi(Y) \neq X$.

On énumère simplement dans l'ensemble X -c.e. qui décrit $\mathcal{U}$, toutes les chaînes σ telles que $\exists n \Phi(\sigma, n) \downarrow \neq X(n)$. Supposons à présent que τ est une chaîne pour laquelle $[\tau] \subseteq \mathbb{N} \setminus \mathcal{U}$, c'est-à-dire qu'aucune extension $\sigma \succeq \tau$ ne soit telle que $\exists n \Phi(\sigma, n) \downarrow \neq X(n)$. Montrons alors que, pour tout $Y \succeq \tau$,

on a $\exists n \ \Phi(Y, n) \uparrow$. Supposons que ce ne soit pas le cas. Alors, on peut calculer X de la manière suivante : pour connaître $X(n)$, il suffit de chercher une extension $\sigma \succeq \tau$ telle que $\Phi(\sigma, n) \downarrow$. Par hypothèse, une telle extension existe, et toujours par hypothèse, elle est telle que $\Phi(\sigma, n) \downarrow = X(n)$. Comme c'est vrai pour tout n , cela contredit le fait que X est incalculable. Donc, si $[\tau] \subseteq \mathbb{N} \setminus \mathcal{U}$, alors pour tout $Y \succeq \tau$ on a $\exists n \ \Phi(Y, n) \uparrow$. On en déduit que, pour tout $Y \in \mathcal{U} \cup \text{int}(\mathbb{N} \setminus \mathcal{U})$, on a $\Phi(Y) \neq X$. Comme G est 1-générique relativement à X , alors $G \in \mathcal{U} \cup \text{int}(\mathbb{N} \setminus \mathcal{U})$, et la même chose est vrai pour toute fonctionnelle Φ . Donc, $G \not\leq_T X$. ■

Voyons à présent les différentes implications entre les notions de n -généricité et n -généricité faible.

Proposition 3.29. Soit G un ensemble. Alors, pour tout $n > 0$, G faiblement $(n+1)$ -générique implique G n -générique, lequel implique à son tour G faiblement n -générique. Les implications sont strictes. ★

PREUVE. Si G est faiblement $(n+1)$ -générique, il rencontre tout ensemble $\Sigma_1^0(\emptyset^{(n)})$ dense. Pour tout ensemble $\Sigma_1^0(\emptyset^{(n-1)})$ W , la réunion $W \cup W^\perp$ est dense et $\Sigma_1^0(\emptyset^{(n)})$, donc rencontre G . Ainsi, G est n -générique. Si G est n -générique, alors pour tout ensemble $\Sigma_1^0(\emptyset^{(n-1)})$, G rencontre $W \cup W^\perp$. Si W est dense, alors $W^\perp = \emptyset$, donc G rencontre W . Ainsi, G est faiblement n -générique.

Pour voir que la première implication est stricte, il suffit de construire un ensemble n -générique $\emptyset^{(n)}$ -calculable et de voir qu'aucun ensemble faiblement $(n+1)$ -générique n'est $\emptyset^{(n)}$ -calculable, car $\{\sigma : \sigma \not\leq X\}$ est un ensemble de chaînes Σ_{n+1}^0 dense pour tout ensemble X qui est Δ_{n+1}^0 . On pourra consulter l'exercice 3.30 pour un exemple d'ensemble n -générique qui n'est pas faiblement n -générique. ■

Exercice 3.30. (★★) Un ensemble X est *approachable par la gauche* relativement à A s'il existe une suite A -calculable d'ensembles $(X_s)_{s \in \mathbb{N}}$ telle que X_s est lexicographiquement plus petit que X_{s+1} pour tout s , et tel que $\lim_s X_s = X$.

Montrer que pour tout A il existe un ensemble faiblement 1-générique relativement à A et approachable par la gauche relativement à A . Montrer qu'aucun ensemble 1-générique relativement à A n'est approachable par la gauche relativement à A . ◇

3.3. Théorème de Posner/Robinson

Dans un article de 1981, Posner et Robinson étudient les degrés strictement sous l'arrêt, mais dont la jointure permet de calculer l'arrêt. Une version généralisée et moderne de leur théorème principal est la suivante : pour tout ensemble A non calculable — et en particulier aussi « faible » que possible calculatoirement —, il existe un ensemble G tel que $A \oplus G \geq_T G'$. Informellement, il existe toujours un ensemble G dont la distance calculatoire entre lui-même et son saut est « réduite » à A , et ce pour n'importe quel A .

La présentation moderne du théorème est celle de Jockusch et Shore, qui montrent quelque chose de plus général :

Théorème 3.31 (Jockusch et Shore [102])

Soient A, Z des ensembles non calculables. Il existe un ensemble 1-générique G tel que $A \oplus G \geq_T Z$. De plus, on peut obtenir G de manière calculable à partir de $A \oplus Z \oplus \emptyset'$.

PREUVE. L'idée est de construire un ensemble 1-générique G , qui va encoder Z , de manière à ce que G et A permettent de retrouver le déroulement de la construction. La construction elle-même sera calculable en $A \oplus Z \oplus \emptyset'$. On peut supposer sans perte de généralité que A n'est pas un ensemble c. e. (dans le cas contraire, on remplace A par son complémentaire). Soit $(W_e)_{e \in \mathbb{N}}$ une énumération des sous-ensembles Σ_1^0 de $2^{<\mathbb{N}}$.

On définit $\sigma_0 = \epsilon$, le mot vide. Supposons σ_e défini. On considère l'ensemble

$$D_e = \{m : \exists \tau \text{ tel que } \sigma_e Z(e) 0^m 1 \tau \in W_e\}.$$

Notons que D_e est un ensemble c. e. En particulier, comme A n'est pas c. e., il existe $m \in D_e$, avec $m \notin A$, ou alors il existe $m \notin D_e$ avec $m \in A$. On considère le plus petit m tel que l'on est dans un cas ou dans l'autre. Notons que $\emptyset' \oplus A$ permet de trouver uniformément cet entier m .

Dans le premier cas, on définit σ_{e+1} comme étant $\sigma_e Z(e) 0^m 1 \tau$ pour la première chaîne τ telle que $\sigma_e Z(e) 0^m 1 \tau$ est énumérée dans W_e . Dans le deuxième cas, on définit σ_{e+1} comme étant $\sigma_e Z(e) 0^m 1$. Notons que dans ce cas aucune chaîne de W_e ne peut étendre σ_{e+1} . On définit G comme étant $\sigma_0 \preceq \sigma_1 \preceq \sigma_2 \preceq \dots$. Cela termine la construction.

Il est clair que G est 1-générique et calculable en $A \oplus Z \oplus \emptyset'$. Comment fait-on à présent pour calculer Z à partir de $G \oplus A$? Supposons que l'on connaisse la chaîne σ_e . On connaît alors nécessairement le e -ième bit de Z : il s'agit du bit i tel que $\sigma_e i \prec G$. On peut ensuite trouver σ_{e+1} de la manière suivante : on regarde le nombre m de 0 qui suit $\sigma_e i$ dans G .

Si $m \in A$, cela signifie que $\sigma_{e+1} = \sigma_e i 0^m 1$. Si $m \notin A$, cela signifie que $\sigma_{e+1} = \sigma_e i 0^m 1 \tau$ pour la première chaîne τ que l'on trouve dans W_e . Trouver τ est alors un procédé calculable. On peut donc dans tous les cas trouver σ_{e+1} , et en répétant le processus, calculer Z à partir de $A \oplus G$. ■

Corollaire 3.32 (Posner et Robinson [180])

Soit A un ensemble non calculable. Il existe un ensemble G tel que

$$A \oplus G \geq_T G'.$$

Si A est Δ_2^0 , on a alors $A \oplus G \equiv_T G'$.

PREUVE. On applique le théorème précédent avec $Z = \emptyset'$. On a donc un ensemble 1-générique G tel que $G \leq_T \emptyset' \oplus A$ et tel que $G \oplus A \geq_T \emptyset'$. En utilisant le fait que les 1-génériques sont low généralisés, on a donc

$$G \oplus A \geq_T G \oplus \emptyset' \equiv_T G'.$$

Il est clair que si A est Δ_2^0 , alors $A \oplus G \equiv_T G'$. ■

Un autre corollaire intéressant est le théorème d'inversion du saut : tout ensemble qui calcule l'arrêt peut être vu comme le degré Turing du saut d'un ensemble.

Corollaire 3.33 (Théorème d'inversion du saut, Friedberg [62])

Soit $Z \geq_T \emptyset'$. Il existe un ensemble G tel que $Z \equiv_T G'$.

PREUVE. On applique le théorème précédent avec Z et $A = \emptyset'$. On a donc un ensemble 1-générique G tel que

$$G \oplus \emptyset' \leq_T Z \oplus \emptyset' \equiv_T Z \quad \text{et tel que} \quad G \oplus \emptyset' \geq_T Z.$$

Usant du fait que les 1-génériques sont low généralisés, on a donc $G' \equiv_T Z$. ■

Le théorème 3.31 permet également de déduire l'existence d'ensembles high et non Turing complets, et même non DNC.

Corollaire 3.34

Il existe un ensemble high, non DNC et en particulier non Turing complet.

PREUVE. Il suffit d'appliquer le théorème 3.31 pour trouver un ensemble 1-générique G tel que $\emptyset' \oplus G \geq_T \emptyset''$, ce qui implique $G' \geq_T \emptyset''$. Notons que comme G est 1-générique, il n'est pas de degré DNC, et ne calcule en particulier pas $\emptyset'$. ■

3.4. Maigreur/co-maigreur des propriétés calculatoires

Revenons un moment aux notions topologiques de classes maigres et de classes co-maigres introduites par Baire. Nous avons déjà mentionné que toute classe n'était pas forcément maigre ou co-maigre. C'est en revanche le cas pour les classes ayant de bonnes propriétés de clôture, et en particulier pour toutes celles dites boréliennes (nous définirons précisément ce terme dans le chapitre 17) et closes par équivalence Turing. Ce sera en particulier le cas de toutes les notions de calculabilité que nous verrons, et nous nous poserons la question de savoir si ces dernières sont maigres ou co-maigres. Aussi une classe est-elle sans perte de généralité co-maigre si elle contient tout élément *suffisamment générique*. En pratique, la 1-généricité est suffisante pour les différentes propriétés calculatoires vues jusqu'ici.

Nous avons les degrés high — les low en tant que classe dénombrable ne nous intéressent pas — les degrés calculatoirement dominés vs hyperimmunes, et enfin les degrés DNC et PA. La classe des ensembles de chacun de ces degrés est-elle maigre ou co-maigre ? Nous avons déjà la réponse en ce qui concerne les degrés calculatoirement dominés et hyperimmunes.

Proposition 3.35. La classe des ensembles calculatoirement dominés est maigre. Celle des ensembles hyperimmunes est co-maigre. ★

PREUVE. D'après le théorème 3.2, si G est faiblement 1-générique, alors il n'est pas de degré calculatoirement dominé. ■

Proposition 3.36. La classe des ensembles DNC est maigre, ainsi bien sûr que la classe des ensembles PA. ★

PREUVE. D'après le théorème 3.21, si X est 1-générique, alors il n'est pas de degré DNC, hors la classe des ensembles 1-génériques est co-maigre. ■

Nous nous attaquons à présent aux degrés high. Nous utilisons pour cela notre théorème 3.28 d'évitement de cône.

Proposition 3.37. Si X est non calculable, alors la classe des ensembles qui calculent X est maigre. ★

PREUVE. La classe des ensembles 1-génériques relativement à X est une intersection d'ouverts denses, et est donc co-maigre. Par le théorème 3.28, aucun d'entre eux ne calcule X , et la classe des ensembles calculant X est donc dans son complémentaire, une classe maigre. ■

Nous pouvons finalement montrer, en combinant ce que nous avons vu, que la classe des ensembles high est elle aussi maigre.

Proposition 3.38. La classe des ensembles high est maigre. ★

PREUVE. On utilise pour cela une version relativisée du théorème 3.28 : si X n'est pas $\emptyset'$ -calculable, alors pour tout ensemble G qui est 1-générique relativement à $X \oplus \emptyset'$ on a $G \oplus \emptyset' \not\geq_T X$. Cette version relativisée se montre de la même manière et ne présente pas de difficulté particulière.

On utilise à présent le théorème 3.20 : si l'ensemble G est 1-générique, on a alors $G' \leq_T G \oplus \emptyset'$. On en déduit que si G est 1-générique, alors G est high ssi $G \oplus \emptyset' \geq_T \emptyset''$. Par ailleurs, d'après la version relativisée du théorème 3.28, aucun 1-générique relativement à $\emptyset''$ n'est tel que $G \oplus \emptyset' \geq_T \emptyset''$. Comme tout ensemble 1-générique relativement à $\emptyset''$ est aussi 1-générique, on a donc qu'aucun 1-générique relativement à $\emptyset''$ n'est high. La classe des ensembles high est donc maigre. ■

4. Forcing Σ_n^0/Π_n^0

Le concept de 1-généricité peut être vu comme un cadre formel effectif autour de la méthode des extensions finies, qui permet pour résumer de contrôler l'arrêt ou non de fonctionnelles, c'est-à-dire de contrôler la valeur de vérité de prédicats Σ_1^0 ou Π_1^0 . Il s'agit d'un premier niveau de forcing : étant donné $\mathcal{R}_e$ un contrat Σ_1^0 ou Π_1^0 , d'après le corollaire 3.18 toute chaîne σ admet une extension $\tau \succeq \sigma$ telle que tout ensemble $X \in [\tau]$ satisfait $\mathcal{R}_e$ ou telle que tout ensemble $X \in [\tau]$ satisfait $\neg \mathcal{R}_e$.

À partir du niveau Σ_2^0/Π_2^0 , les choses deviennent plus complexes et la méthode des extensions finies ne peut plus fonctionner de la même manière, comme en témoigne l'exemple suivant.

Exemple 4.1. Considérons le contrat $\mathcal{R} : \ll \exists x \forall y \geq x G(y) = 0 \gg$, qui exprime la finitude de l'ensemble G . Pour tout $\sigma \in 2^{<\mathbb{N}}$, $[\sigma]$ contient à la fois un ensemble fini et un ensemble infini. Il n'existe donc aucun cylindre dont tous les éléments satisfont $\mathcal{R}$ ou dont tous les éléments satisfont $\neg \mathcal{R}$.

Point sur les contrats

Nous nous attaquons à présent aux contrats Σ_n^0 (resp. Π_n^0) c'est-à-dire aux contrats portant sur un ensemble G et qui s'expriment par une formule Σ_n^0 (resp. Π_n^0) de l'arithmétique du second ordre, avec G comme variable d'ensemble libre. De manière équivalente, un contrat est Σ_n^0 s'il existe une fonctionnelle $\Phi_e(G, x_1, \dots, x_{n-1})$ telle que les ensembles G satisfaisant le contrat sont ceux tels que :

$$\begin{aligned} \exists x_1 \forall x_2 \dots \forall x_{n-1} \Phi_e(G, x_1, x_2, \dots, x_{n-1}) \downarrow & \text{ pour } n \text{ impair} \\ \exists x_1 \forall x_2 \dots \exists x_{n-1} \Phi_e(G, x_1, x_2, \dots, x_{n-1}) \uparrow & \text{ pour } n \text{ pair.} \end{aligned}$$

Les contrats Π_n^0 ont l'équivalence analogue en commençant par une quantification universelle. Par convention, la négation $\neg\mathcal{R}$ devant une formule Σ_n^0 (resp. Π_n^0) ne sera pas considérée comme un symbole du langage, mais comme une opération de transformation $\mathcal{R}$, qui inverse les quantificateurs, et remplace le prédicat Δ_0^0 final par sa négation, pour faire de $\neg\mathcal{R}$ une formule Π_n^0 (resp. Σ_n^0).

Nous devons donc abstraire un peu les choses afin de donner une définition plus générale du forcing permettant de contrôler la valeur de vérité de prédicats de complexité arbitraire. Introduisons avant de commencer une définition qui sera utilisée dans les preuves à venir.

Définition 4.2. Un ensemble $D \subseteq 2^{<\mathbb{N}}$ est *dense sous la chaîne* σ si pour tout $\tau \succeq \sigma$, il existe un $\rho \succeq \tau$ tel que $\rho \in D$. ◇

4.1. L'approche sémantique

La relation de forcing sera définie par induction sur n , entre les chaînes finies et les prédicats Σ_n^0 . Un des objectifs sera alors de conserver la propriété suivante.

(D) : L'ensemble des chaînes forçant $\mathcal{R}$ ou forçant $\neg\mathcal{R}$ est dense.

Afin d'examiner ce dont nous avons besoin, reprenons l'exemple précédent, à savoir un contrat $\mathcal{R} \Sigma_2^0$ arbitraire $\ll \exists x \Phi(G, x) \uparrow \gg$, et regardons ce qui ne fonctionne pas quand on essaye de prouver la densité de l'ensemble $Q \subseteq 2^{<\mathbb{N}}$ des chaînes dont tous les éléments satisfont $\mathcal{R}$ ou dont tous les éléments satisfont $\neg\mathcal{R}$.

Soit $\sigma \in 2^{<\mathbb{N}}$ une chaîne. Faisons une analyse de cas.

▷ Cas 1. Il existe une extension $\tau \succeq \sigma$ et un entier $x \geq 1$ tel que pour tout $\rho \succeq \tau$, $\Phi(\rho, x) \uparrow$. Dans ce cas, par la propriété de l'usage, pour tout $A \in [\tau]$, $\Phi(A, x) \uparrow$. Il s'ensuit que $\mathcal{R}$ est satisfait pour tout $A \in [\tau]$, donc que $\tau \in Q$.

▷ Cas 2. Pour toute extension $\tau \succeq \sigma$ et tout $x \in \mathbb{N}$, il existe une chaîne $\rho \succeq \tau$ telle que $\Phi(\rho, x) \downarrow$. C'est dans ce cas que notre tentative échoue. Pourtant, intuitivement, dans ce cas, nous devrions être capables de poursuivre la construction d'une suite tout en s'assurant que l'ensemble résultant satisfasse $\neg\mathcal{R}$. En effet, quel que soit l'état d'avancement de la construction, nous nous retrouvons avec une chaîne $\tau \succeq \sigma$, donc par hypothèse, pour tout $x \in \mathbb{N}$, il est toujours possible de trouver une extension $\rho \succeq \tau$ telle que $\Phi(G, x) \downarrow$ pour tout $G \in [\rho]$: pour tout x , l'ensemble Q_x des chaînes ρ telles que $\Phi(\rho, x) \downarrow$ est dense sous σ , c'est-à-dire que pour tout $\tau \succeq \sigma$ il

existe $\rho \succeq \tau$ telle que $\rho \in Q_x$. Par conséquent, si un ensemble suffisamment générique G étend σ , alors il rencontrera chacun des Q_x . On aura donc $\forall x \Phi(G, x) \downarrow$, autrement dit G satisfera $\neg \mathcal{R}$.

Cette analyse motive donc la définition suivante pour le forcing de Cohen.

Définition 4.3. Une chaîne σ *force sémantiquement* un contrat $\mathcal{R}$, auquel cas on notera $\sigma \Vdash \mathcal{R}$, si tout ensemble « suffisamment générique » qui étend σ satisfait $\mathcal{R}$: il existe une suite dénombrable d'ensembles de chaînes denses $(W_n)_{n \in \mathbb{N}}$ telle que si $G \in [\sigma]$ rencontre chaque W_n , alors le contrat sera satisfait pour G . $\diamond$

Notons que la relation $\Vdash$ est plus générale que la relation $\Vdash^*$ pour les contrats Σ_1^0 et Π_1^0 . Par exemple, la chaîne vide ϵ force sémantiquement le contrat $\mathcal{R} : \ll \exists x G(x) = 1 \gg$, car l'ensemble des chaînes contenant un 1 est dense, et donc tout ensemble suffisamment générique satisfera $\mathcal{R}$. En revanche, la suite infinie de zéros 0^∞ appartient à $[\epsilon]$ et ne satisfait pas le contrat $\mathcal{R}$.

4.2. L'approche syntaxique

La définition 4.3 est simple à exprimer dans le langage naturel, mais sort de la hiérarchie arithmétique : une traduction directe demande de quantifier existentiellement sur toutes les suites dénombrables d'ensembles de chaînes denses, puis de quantifier universellement sur les ensembles génériques pour ces ensembles de chaînes. Nous allons définir une relation beaucoup plus simple syntaxiquement parlant, qui permettra de raisonner plus facilement et notamment de prouver les propriétés essentielles que l'on attend de la relation de forcing, à savoir qu'elle soit close par extension et que l'ensemble des chaînes forçant un contrat ou sa négation soit dense.

Définissons à présent une relation syntaxique de forcing pour tout contrat arithmétique, par induction sur ses quantifications. Le premier niveau sera celui auquel nous sommes déjà habitué : soit $\mathcal{R}$ un contrat Σ_1^0 — et donc de la forme $\Phi_e(G) \downarrow$ —, alors une chaîne σ force $\mathcal{R}$ si $\Phi_e(\sigma) \downarrow$, et donc si tout $X \in [\sigma]$ satisfait le contrat. Il en va de même pour les contrats Π_1^0 .

Définition 4.4

- (1) $\sigma \Vdash^* \mathcal{R}$ pour $\mathcal{R}$ un contrat Σ_1^0 ou Π_1^0 ssi tout $X \in [\sigma]$ satisfait $\mathcal{R}$.
- (2) $\sigma \Vdash^* \exists x \mathcal{R}(x)$ pour $\mathcal{R}(x)$ un contrat Π_k^0 pour $k > 0$ avec variable libre x ssi il existe un $n \in \mathbb{N}$ tel que $\sigma \Vdash^* \mathcal{R}(n)$.
- (3) $\sigma \Vdash^* \forall x \mathcal{R}(x)$ pour $\mathcal{R}(x)$ un contrat Σ_k^0 pour $k > 0$ avec variable libre x ssi pour tout $\tau \succeq \sigma$ et tout $n \in \mathbb{N}$, $\tau \not\Vdash^* \neg \mathcal{R}(n)$. $\diamond$

Avant toute chose, remarquons que (3) de la définition précédente admet une formulation équivalente, parfois plus adaptée à ce que l'on veut démontrer.

Lemme 4.5. Soit $\mathcal{R}$ un contrat Π_n^0 , on a $\sigma \Vdash^* \mathcal{R}$ ssi $\forall \tau \succeq \sigma \ \tau \nVdash^* \neg \mathcal{R}$. ★

PREUVE. Il s'agit d'une simple reformulation de la définition.

Cas 1. Le contrat $\mathcal{R}$ est Π_1^0 , de la forme $\Phi(G) \uparrow$. Alors, $\sigma \Vdash^* \mathcal{R}$ ssi $\Phi(X) \uparrow$ pour tout $X \in [\sigma]$ ssi $\Phi(X) \uparrow$ pour tout $\tau \succeq \sigma$ et tout $X \in [\tau]$ ssi pour tout $\tau \succeq \sigma$, il existe $X \in [\tau]$ tel que $\Phi(X) \uparrow$ (par la propriété de l'usage) ssi pour tout $\tau \succeq \sigma$, $\tau \nVdash^* \Phi(G) \downarrow$.

Cas 2. Le contrat $\mathcal{R}$ est Π_{k+1}^0 , de la forme $\forall x \mathcal{Q}(x)$ pour $\mathcal{Q}(x)$ un contrat Σ_k^0 pour $k \geq 1$. Alors, $\sigma \Vdash^* \mathcal{R}$ ssi pour tout $\tau \succeq \sigma$ et tout $n \in \mathbb{N}$, $\tau \nVdash^* \neg \mathcal{Q}(n)$ ssi $\forall \tau \succeq \sigma \ \tau \nVdash^* \exists x \neg \mathcal{Q}(x)$ ssi $\forall \tau \succeq \sigma \ \tau \nVdash^* \neg \mathcal{R}$. ■

La première propriété que l'on attend d'une relation de forcing est sa clôture par extension. En effet, « σ force $\mathcal{R}$ » signifie que la propriété $\mathcal{R}$ est déjà décidée sur l'objet final construit, ce qui ne doit pas changer au cours des étapes suivantes de la construction.

Proposition 4.6. Soient $\sigma, \tau \in 2^{<\mathbb{N}}$ et soit $\mathcal{R}$ un contrat arithmétique. Si $\sigma \Vdash^* \mathcal{R}$ et $\sigma \preceq \tau$, alors $\tau \Vdash^* \mathcal{R}$. ★

PREUVE. Par induction sur la complexité arithmétique du contrat.

Cas 1. Le contrat $\mathcal{R}$ est Σ_1^0 ou Π_1^0 . Par définition, pour tout $X \in [\sigma]$, $\mathcal{R}(X)$ est vrai. Comme $\tau \succeq \sigma$, alors $[\tau] \subseteq [\sigma]$, donc pour tout $X \in [\tau]$, $\mathcal{R}(X)$ est vrai. Ainsi, $\tau \Vdash^* \mathcal{R}$.

Cas 2. Le contrat $\mathcal{R}$ est de la forme $\exists x \mathcal{S}(x)$ pour $\mathcal{S}(x)$ un contrat Π_k^0 pour $k > 0$. Par définition, il existe un $n \in \mathbb{N}$ tel que $\sigma \Vdash^* \mathcal{S}(n)$. Par hypothèse d'induction, $\tau \Vdash^* \mathcal{S}(n)$, donc $\tau \Vdash^* \exists x \mathcal{S}(x)$.

Cas 3. Le contrat $\mathcal{R}$ est de la forme $\forall x \mathcal{S}(x)$ pour $\mathcal{S}(x)$ un contrat Σ_k^0 pour $k > 0$. D'après le lemme 4.5, $\sigma \Vdash^* \mathcal{R}$ implique $\forall \rho \succeq \sigma \ \rho \nVdash^* \neg \mathcal{R}$. Si $\tau \succeq \sigma$, alors on a aussi $\forall \rho \succeq \tau \ \rho \nVdash^* \neg \mathcal{R}$, ce qui encore d'après le lemme 4.5 implique $\tau \Vdash^* \mathcal{R}$. ■

La seconde propriété, et peut-être la plus importante, est la densité de l'ensemble des chaînes forçant un contrat ou sa négation. La proposition 4.7 signifie en particulier que la valeur de vérité de tout contrat arithmétique sera décidée au bout d'un moment fini de la construction. C'est ce qui donne toute la puissance de la relation de forcing.

Proposition 4.7. Soit $\mathcal{R}$ un contrat arithmétique. L'ensemble

$$\{\sigma \in 2^{<\mathbb{N}} : \sigma \Vdash^* \mathcal{R} \text{ ou } \sigma \Vdash^* \neg \mathcal{R}\}$$

est dense. ★

PREUVE. On peut supposer sans perte de généralité que $\mathcal{R}$ est Σ_n^0 (dans le cas inverse, on répète l'argument avec $\neg \mathcal{R}$). Soit σ une chaîne. D'après le lemme 4.5, soit $\tau \Vdash^* \mathcal{R}$ pour une extension $\tau \succeq \sigma$, soit $\sigma \Vdash^* \neg \mathcal{R}$. ■

Une dernière propriété que l'on attend également est bien entendu la validité de la définition de la relation de forcing, c'est-à-dire que si une chaîne σ force un contrat, alors ce contrat sera effectivement satisfait pour n'importe quel ensemble suffisamment générique qui étend σ . Insistons encore sur ce que signifie être « suffisamment générique » dans ce contexte : il existe une suite dénombrable d'ensembles de chaînes denses $(W_n)_{n \in \mathbb{N}}$ telle que si $G \in [\sigma]$ rencontre chaque W_n , alors le contrat sera satisfait pour G .

Proposition 4.8. Soit $\mathcal{R}$ un contrat arithmétique et soit $\sigma \in 2^{<\mathbb{N}}$.

Si $\sigma \Vdash^* \mathcal{R}$, alors $\sigma \Vdash \mathcal{R}$: si $G \in [\sigma]$ est suffisamment générique, alors $\mathcal{R}$ est satisfait pour G . ★

PREUVE. Par induction sur la complexité arithmétique du contrat.

Cas 1. Le contrat $\mathcal{R}$ est Σ_1^0 ou Π_1^0 . Supposons que $\sigma \Vdash^* \mathcal{R}$. Par définition, tout ensemble $G \in [\sigma]$ satisfait $\mathcal{R}$, donc *a fortiori* tout ensemble suffisamment générique $G \in [\sigma]$. Ainsi, $\sigma \Vdash \mathcal{R}$.

Cas 2. Le contrat $\mathcal{R}$ est de la forme $\exists x \mathcal{S}(x)$ pour $\mathcal{S}(x)$ un contrat Π_k^0 pour $k > 0$. Supposons $\sigma \Vdash^* \exists x \mathcal{S}(x)$. Par définition, il existe un $n \in \mathbb{N}$ tel que $\sigma \Vdash^* \mathcal{S}(n)$. Par hypothèse d'induction, σ force $\mathcal{S}(n)$, donc $\sigma \Vdash \exists x \mathcal{S}(x)$.

Cas 3. Le contrat $\mathcal{R}$ est de la forme $\forall x \mathcal{S}(x)$ pour $\mathcal{S}(x)$ un contrat Σ_k^0 pour $k > 0$. Supposons que $\sigma \Vdash^* \forall x \mathcal{S}(x)$. Par définition, pour tous $\tau \succeq \sigma$ et $n \in \mathbb{N}$, $\tau \nVdash^* \neg \mathcal{S}(n)$. Par la proposition 4.7, pour tout n , l'ensemble

$$D_n = \{\tau \in 2^{<\mathbb{N}} : \tau \Vdash^* \mathcal{S}(n)\}$$

est dense sous σ : pour tout $\tau \succeq \sigma$ il existe $\rho \succeq \tau$ tel que $\rho \Vdash^* \mathcal{S}(n)$. Soit G un ensemble suffisamment générique étendant σ . On suppose en particulier que le niveau de généricité de G garanti qu'il rencontre chaque ensemble D_n . Donc, pour tout n , il existe un préfixe $\tau_n \prec G$ tel que $\tau_n \Vdash^* \mathcal{S}(n)$. Pour une telle chaîne τ_n , par hypothèse d'induction, il existe une suite dénombrable d'ensemble de chaînes dense $(D_{n,m})_{m \in \mathbb{N}}$ telle que si $G \in [\tau_n]$ rencontre chaque $D_{n,m}$, alors il satisfait $\mathcal{S}(n)$. On a bien $G \in [\tau_n]$, et comme G est suffisamment générique, il rencontre chaque $D_{n,m}$, et satisfait donc $\mathcal{S}(n)$. Comme c'est le cas pour tout n , alors G satisfait $\forall x \mathcal{S}(x)$, donc $\sigma \Vdash \forall x \mathcal{S}(x)$. ■

Le cœur du forcing se trouve sans doute dans la précédente proposition, et en particulier dans le cas 3 de sa preuve : c'est là que s'exerce le mécanisme de la relation $\sigma \Vdash^* \mathcal{R}$, qui garantit que si G appartient à $[\sigma]$ et si G est suffisamment générique, alors le contrat $\mathcal{R}$ sera satisfait pour G . Il s'agit en réalité d'une modification sophistiquée de la méthode des extensions finies, le point clef étant le suivant : peu importe le préfixe σ de G que l'on a construit jusqu'ici, on peut étendre σ pour rencontrer n'importe quel ensemble de chaînes dense fixé à l'avance.

Nous voyons à présent dans la section suivante que cette idée n'est pas nouvelle : le mathématicien René Baire avait déjà formalisé l'ensemble de ces mécanismes au début du XX^e siècle, notamment via le théorème suivant : « toute classe borélienne a la propriété de Baire. »

4.3. L'approche topologique : la propriété de Baire

Une technique importante pour l'étude d'objets complexes en mathématiques consiste à se ramener à des objets plus simples tout en contrôlant la marge d'erreur d'approximation. En particulier, dans l'étude des ensembles, que cela soit du point de vue de la théorie de la mesure ou de la théorie des catégories, il existe un certain nombre de théorèmes de la forme « tout ensemble complexe est équivalent à un ensemble simple modulo une quantité négligeable d'éléments ». En théorie de la mesure par exemple, nous avons les trois principes de Littlewood [145], qui énoncent que tout ensemble mesurable est « presque » une réunion finie d'intervalles, toute fonction est « presque » continue, et toute suite convergente est « presque » uniformément convergente. Dans ce contexte « presque » signifie : « sauf sur un ensemble de mesure inférieure à ε pour ε aussi petit que l'on souhaite. »

La propriété de Baire exprime le fait qu'une classe S est « presque » ouverte, où presque signifie dans ce contexte : « sauf sur une classe maigre. »

Définition 4.9. Une classe $\mathcal{B} \subseteq 2^{\mathbb{N}}$ a la propriété de Baire s'il existe un ouvert $\mathcal{U} \subseteq 2^{\mathbb{N}}$ tel que $\mathcal{B} \Delta \mathcal{U}$ est une classe maigre. Ici, $\mathcal{B} \Delta \mathcal{U}$ est la classe des éléments sur lesquels $\mathcal{B}$ et $\mathcal{U}$ ne coïncident pas, c'est-à-dire que

$$(\mathcal{B} \setminus \mathcal{U}) \cup (\mathcal{U} \setminus \mathcal{B}).$$

◇

Considérons un contrat $\mathcal{R}$, et soient $\mathcal{B}_{\mathcal{R}}$ et $\mathcal{B}_{\neg\mathcal{R}}$ les classes des éléments qui respectivement satisfont et ne satisfont pas ce contrat (en particulier, $\mathcal{B}_{\mathcal{R}} \cap \mathcal{B}_{\neg\mathcal{R}} = \emptyset$ et $\mathcal{B}_{\mathcal{R}} \cup \mathcal{B}_{\neg\mathcal{R}} = 2^{\mathbb{N}}$). Supposons que $\mathcal{B}_{\mathcal{R}}$ et $\mathcal{B}_{\neg\mathcal{R}}$ aient tous les deux la propriété de Baire, et fixons deux ouverts $\mathcal{U}_{\mathcal{R}}$ et $\mathcal{U}_{\neg\mathcal{R}}$ tels que $\mathcal{B}_{\mathcal{R}} \Delta \mathcal{U}_{\mathcal{R}}$ et $\mathcal{B}_{\neg\mathcal{R}} \Delta \mathcal{U}_{\neg\mathcal{R}}$ soient tous les deux maigres. Cela signifie qu'il existe une réunion dénombrable de fermés d'intérieur vide contenant $\mathcal{B}_{\mathcal{R}} \Delta \mathcal{U}_{\mathcal{R}}$ et $\mathcal{B}_{\neg\mathcal{R}} \Delta \mathcal{U}_{\neg\mathcal{R}}$. Par passage au complémentaire, il existe une

intersection dénombrable d'ouverts denses $\bigcap_n \mathcal{U}_n$ telle que pour tout X qui lui appartient, on a d'un côté $X \in \mathcal{B}_{\mathcal{R}}$ ssi $X \in \mathcal{U}_{\mathcal{R}}$, et de l'autre $X \in \mathcal{B}_{\neg\mathcal{R}}$ ssi $X \in \mathcal{U}_{\neg\mathcal{R}}$.

En particulier, $X \in \mathcal{B}_{\mathcal{R}}$ ssi il existe un préfixe $\sigma \prec X$ tel que $[\sigma] \subseteq \mathcal{U}_{\mathcal{R}}$. On dira alors que σ force le contrat $\mathcal{R}$. Notons que si $X \notin \mathcal{B}_{\mathcal{R}}$, alors $X \in \mathcal{B}_{\neg\mathcal{R}}$ et il existe à ce moment un préfixe $\sigma \prec X$ tel que $[\sigma] \subseteq \mathcal{U}_{\neg\mathcal{R}}$. À ce moment, σ force le contrat $\neg\mathcal{R}$. La définition suivante est simplement une reformulation de la définition 4.3 qui définit le forcing sémantique.

Définition 4.10. Soit $\mathcal{R}$ un contrat. On dit que σ *force sémantiquement* le contrat $\mathcal{R}$, et l'on écrit $\sigma \Vdash \mathcal{R}$ si $[\sigma] \setminus \mathcal{B}_{\mathcal{R}}$ est une classe maigre, où $\mathcal{B}_{\mathcal{R}}$ est la classe des éléments qui satisfont $\mathcal{R}$. ◇

Comme mentionné précédemment, la propriété fondamentale que l'on attend de la relation de forcing pour un contrat est la suivante.

(D) : L'ensemble des chaînes forçant $\mathcal{R}$ ou forçant $\neg\mathcal{R}$ est dense.

Nous commençons donc par une caractérisation des contrats pour lesquels cette propriété est vraie, à l'aide de la propriété de Baire.

Proposition 4.11. Soit $\mathcal{R}$ un contrat, et soit $\mathcal{B}_{\mathcal{R}}$ la classe des éléments qui satisfont $\mathcal{R}$. Les énoncés suivants sont équivalents :

- (1) $\mathcal{B}_{\mathcal{R}}$ a la propriété de Baire ;
- (2) $\{\sigma \in 2^{<\mathbb{N}} : \sigma \Vdash \mathcal{R} \text{ ou } \sigma \Vdash \neg\mathcal{R}\}$ est dense. ★

PREUVE. Soient $U_{\mathcal{R}} = \{\sigma \in 2^{<\mathbb{N}} : \sigma \Vdash \mathcal{R}\}$ et $U_{\neg\mathcal{R}} = \{\sigma \in 2^{<\mathbb{N}} : \sigma \Vdash \neg\mathcal{R}\}$.

(1) $\Rightarrow$ (2). Supposons que $\mathcal{B}_{\mathcal{R}}$ ait la propriété de Baire. Soit $\mathcal{U}$ un ouvert tel que $\mathcal{B}_{\mathcal{R}} \Delta \mathcal{U}$ est maigre. En particulier, $\mathcal{U} \setminus \mathcal{B}_{\mathcal{R}}$ est maigre, donc $\mathcal{U} \subseteq [U_{\mathcal{R}}]$. De plus, $\mathcal{B}_{\mathcal{R}} \setminus \mathcal{U} = (2^{\mathbb{N}} \setminus \mathcal{U}) \setminus \mathcal{B}_{\neg\mathcal{R}}$ est maigre, donc $\text{int}(2^{\mathbb{N}} \setminus \mathcal{U}) \setminus \mathcal{B}_{\neg\mathcal{R}}$ est maigre. Il s'ensuit que $\text{int}(2^{\mathbb{N}} \setminus \mathcal{U}) \subseteq [U_{\neg\mathcal{R}}]$. Comme $\mathcal{U} \cup \text{int}(2^{\mathbb{N}} \setminus \mathcal{U})$ est dense dans $2^{\mathbb{N}}$, il en est de même de $[U_{\mathcal{R}}] \cup [U_{\neg\mathcal{R}}]$, donc par l'exercice 2.8, $U_{\mathcal{R}} \cup U_{\neg\mathcal{R}}$ est dense dans $2^{<\mathbb{N}}$.

(2) $\Rightarrow$ (1). Supposons que $U_{\mathcal{R}} \cup U_{\neg\mathcal{R}}$ soit dense dans $2^{<\mathbb{N}}$. Par densité, le complémentaire de $[U_{\mathcal{R}}] \cup [U_{\neg\mathcal{R}}]$ est un fermé d'intérieur vide, donc maigre. Montrons que $\mathcal{B}_{\mathcal{R}} \Delta [U_{\mathcal{R}}]$ est maigre. Par définition, pour tout $\sigma \in U_{\mathcal{R}}$, la classe $[\sigma] \setminus \mathcal{B}_{\mathcal{R}}$ est maigre, donc $[U_{\mathcal{R}}] \setminus \mathcal{B}_{\mathcal{R}}$ est réunion dénombrable de classes maigres, donc est maigre. Par le même raisonnement, si $\mathcal{B}_{\neg\mathcal{R}}$ est la classe des éléments qui satisfont $\neg\mathcal{R}$, alors $[U_{\neg\mathcal{R}}] \setminus \mathcal{B}_{\neg\mathcal{R}}$ est maigre. Comme $[U_{\neg\mathcal{R}}] \setminus \mathcal{B}_{\neg\mathcal{R}}$ et $\mathcal{B}_{\mathcal{R}} \cap [U_{\neg\mathcal{R}}]$ sont égaux, et comme le complémentaire de $[U_{\mathcal{R}}] \cup [U_{\neg\mathcal{R}}]$ est maigre, alors $\mathcal{B}_{\mathcal{R}} \setminus [U_{\mathcal{R}}]$ est maigre. Ainsi, $\mathcal{B}_{\mathcal{R}} \Delta [U_{\mathcal{R}}]$ est maigre. ■

La relation de forcing syntaxique impliquant la relation sémantique, nous pouvons en déduire la densité de la relation de forcing sémantique pour les contrats arithmétiques.

Lemme 4.12. Soit $\mathcal{R}$ est un contrat arithmétique. L'ensemble

$$\{\sigma \in 2^{<\mathbb{N}} : \sigma \Vdash \mathcal{R} \vee \sigma \Vdash \neg \mathcal{R}\}$$

est dense. ★

PREUVE. Pour tout σ , il existe une extension $\tau \succeq \sigma$ telle que $\tau \Vdash^* \mathcal{R}$ ou telle que $\tau \Vdash^* \neg \mathcal{R}$. Si $\tau \Vdash^* \mathcal{R}$, alors $\tau \Vdash \mathcal{R}$, et si $\tau \Vdash^* \neg \mathcal{R}$, alors $\tau \Vdash \neg \mathcal{R}$. ■

Le corollaire suivant découle directement de la caractérisation des classes ayant la propriété de Baire.

Corollaire 4.13

Soit $\mathcal{R}$ est un contrat arithmétique. La classe $\mathcal{B}_{\mathcal{R}}$ des éléments qui satisfont $\mathcal{R}$, a la propriété de Baire.

PREUVE. Immédiat par le lemme 4.12 et la proposition 4.11. ■

Nous développerons dans le chapitre 17 et la section 27-3.1 la théorie des classes boréliennes, qui formalise et généralise la notion de contrats arithmétiques, et qui ont toutes la propriété de Baire.

Terminons cette section en clarifiant les liens entre la relation de forcing sémantique et la relation syntaxique $\Vdash^*$, que nous avons définies plus haut.

Théorème 4.14

Soit $\mathcal{R}$ un contrat arithmétique. Alors, $\sigma \Vdash \mathcal{R}$ si, et seulement si, l'ensemble $\{\tau \in 2^{<\mathbb{N}} : \tau \Vdash^* \mathcal{R}\}$ est dense sous σ .

Pour montrer le théorème, nous utiliserons deux lemmes. Le premier correspond à la proposition 4.6 pour la relation $\Vdash^*$.

Lemme 4.15. Supposons $\sigma \Vdash \mathcal{R}$ pour une chaîne σ et un contrat $\mathcal{R}$. Si $\tau \succeq \sigma$, alors $\tau \Vdash \mathcal{R}$. ★

PREUVE. Soit $\mathcal{B}_{\mathcal{R}}$ la classe des éléments qui satisfont $\mathcal{R}$. On a visiblement $[\tau] \setminus \mathcal{B}_{\mathcal{R}} \subseteq [\sigma] \setminus \mathcal{B}_{\mathcal{R}}$. Donc, si $[\sigma] \setminus \mathcal{B}_{\mathcal{R}}$ est maigre, alors $[\tau] \setminus \mathcal{B}_{\mathcal{R}}$ est maigre. ■

Lemme 4.16. Supposons $\sigma \Vdash \mathcal{R}$ pour une chaîne σ et un contrat $\mathcal{R}$. Alors, $\sigma \nVdash \neg \mathcal{R}$ ★

PREUVE. Soit $\mathcal{B}_{\mathcal{R}}$ la classe des éléments qui satisfont $\mathcal{R}$, et soit $\mathcal{B}_{\neg \mathcal{R}}$ la classe des éléments qui satisfont $\neg \mathcal{R}$. Supposons par l'absurde $\sigma \Vdash \mathcal{R}$ et $\sigma \Vdash \neg \mathcal{R}$. Alors, les classes $[\sigma] \setminus (\mathcal{B}_{\mathcal{R}} \cap [\sigma])$ et $[\sigma] \setminus (\mathcal{B}_{\neg \mathcal{R}} \cap [\sigma])$ sont toutes deux maigres. Par ailleurs, $([\sigma] \setminus (\mathcal{B}_{\mathcal{R}} \cap [\sigma])) \cup ([\sigma] \setminus (\mathcal{B}_{\neg \mathcal{R}} \cap [\sigma])) = [\sigma]$, ce qui contredit le fait que la réunion de deux classes maigres soit maigre et donc d'intérieur vide. ■

PREUVE DU THÉORÈME 4.14. Rappelons la proposition 4.8 : pour tout contrat $\mathcal{R}$ arithmétique et pour tout σ , si $\sigma \Vdash^* \mathcal{R}$, alors $\sigma \Vdash \mathcal{R}$.

Montrons que si $\{\tau \in 2^{<\mathbb{N}} : \tau \Vdash^* \mathcal{R}\}$ est dense sous σ , alors $\sigma \Vdash \mathcal{R}$. D'après la proposition 4.8, $A = \{\tau \in 2^{<\mathbb{N}} : \tau \Vdash \mathcal{R}\}$ est dense sous σ . La classe $\{X \in [\sigma] : \forall n \ X \restriction_n \Vdash \mathcal{R}\}$ est donc un fermé d'intérieur vide, que l'on notera $\mathcal{F}$. Soit $\mathcal{B}_{\mathcal{R}}$ la classe des éléments qui satisfont $\mathcal{R}$. Par définition de A , pour tout $\tau \in A$, la classe $\mathcal{M}_{\tau} = [\tau] \setminus (\mathcal{B}_{\mathcal{R}} \cap [\tau])$ est maigre. Il s'ensuit que la classe $[\sigma] \setminus (\mathcal{B}_{\mathcal{R}} \cap [\sigma])$ est incluse dans la réunion de $\mathcal{F}$ et de toutes les classes $\mathcal{M}_{\tau}$. Elle est donc maigre.

Montrons enfin que si $\sigma \Vdash \mathcal{R}$, alors $\{\tau \in 2^{<\mathbb{N}} : \tau \Vdash^* \mathcal{R}\}$ est dense sous σ . Par contraposée, supposons qu'il existe $\tau \succeq \sigma$ tel que pour tout $\rho \succeq \tau$ on a $\rho \nVdash^* \mathcal{R}$. Au vu de la proposition 4.7, il doit alors exister $\rho \succeq \tau$ tel que $\rho \Vdash^* \neg \mathcal{R}$. Et, par la proposition 4.8, on a alors $\rho \Vdash \neg \mathcal{R}$. D'après le lemme 4.16, $\rho \nVdash \mathcal{R}$, et l'on a donc d'après le lemme 4.15 $\sigma \nVdash \mathcal{R}$. ■

5. Ensembles arbitrairement génériques

La généricité peut être vue comme une notion de « typicité », au sens où tout ce qui peut arriver infiniment souvent finira par arriver. Dans le cas du forcing de Cohen, un ensemble dense a infiniment souvent la possibilité d'être rencontré, et cela arrivera si un ensemble est typique, ce qui correspond à la notion d'ensemble générique.

Nous avons vu les relativisations des ensembles faiblement 1-génériques et 1-génériques, et notamment les concepts de n -générique et faiblement n -générique. Nous étudions dans cette section les propriétés des ensembles typiques, c'est-à-dire les propriétés que vont avoir les ensembles suffisamment génériques.

Notons que si une propriété est vérifiée par tout ensemble suffisamment générique, elle est sans perte de généralité vérifiée pour tout ensemble faiblement 1-générique relativement à A pour un certain oracle A suffisamment puissant : il suffit que A encode l'intersection d'ouverts denses correspondant au niveau de généricité requis. On s'attachera donc à déterminer « le bon niveau » de généricité nécessaire pour satisfaire telle ou telle propriété. En pratique, cela correspondra souvent à être n -générique pour un certain n .

5.1. Propriétés des ensembles suffisamment génériques

Nous avons vu que la 1-généricité était le niveau de généricité correspondant au forcing des contrats Σ_1^0/Π_1^0 . Sans surprise, nous voyons à présent que la n -généricité est le niveau de généricité correspondant au forcing des contrats Σ_n^0/Π_n^0 , et nous allons montrer le théorème suivant.

Théorème 5.1

Soit G un ensemble n -générique. Soit $\mathcal{R}$ un contrat Σ_n^0 ou Π_n^0 . Alors, G satisfait $\mathcal{R}$ si, et seulement si, il existe un préfixe $\sigma \prec G$ tel que $\sigma \Vdash^* \mathcal{R}$.

Notons que cette itération n'est pas triviale : il faut faire appel pour cela aux développements de la relation de forcing des sections précédentes.

Nous allons maintenant passer à la preuve du théorème 5.1 en exploitant la simplicité définitionnelle de la relation de forcing syntaxique. Commençons par la proposition suivante.

Proposition 5.2. Soit $\mathcal{R}$ un contrat arithmétique.

- (1) Si $\mathcal{R}$ est Σ_n^0 , alors le prédicat $\sigma \Vdash^* \mathcal{R}$ est Σ_n^0 .
- (2) Si $\mathcal{R}$ est Π_n^0 , alors le prédicat $\sigma \Vdash^* \mathcal{R}$ est Π_n^0 . ★

PREUVE. Nous allons prouver (1) et (2) simultanément par induction sur la complexité arithmétique du contrat. Le cas Σ_1^0 et Π_1^0 a déjà été traité avec la proposition 3.17.

Si $\mathcal{R}$ est Σ_{m+1}^0 avec $m > 0$, alors il peut s'exprimer sous la forme $\exists x \mathcal{S}(x)$, où $\mathcal{S}$ est un contrat Π_m^0 . On a $\sigma \Vdash^* \mathcal{R}$ si, et seulement si, il existe un $n \in \mathbb{N}$ tel $\sigma \Vdash^* \mathcal{S}(n)$. Par hypothèse d'induction, le prédicat $\sigma \Vdash^* \mathcal{S}(n)$ est Π_m^0 , et le prédicat $\sigma \Vdash^* \mathcal{R}$ est donc Σ_{m+1}^0 .

Si $\mathcal{R}$ est Π_{m+1}^0 avec $m > 0$, alors il peut s'exprimer sous la forme $\forall x \mathcal{S}(x)$ où $\mathcal{S}$ est un contrat Σ_m^0 . On a $\sigma \Vdash^* \mathcal{R}$ ssi pour tout $\tau \succeq \sigma$ et tout n dans $\mathbb{N}$, $\tau \nVdash^* \neg \mathcal{S}(n)$. En particulier, $\neg \mathcal{S}(n)$ est Π_m^0 , donc par hypothèse d'induction, le prédicat $\tau \Vdash^* \neg \mathcal{S}(n)$ est Π_m^0 , donc $\tau \nVdash^* \neg \mathcal{S}(n)$ est Σ_m^0 , et le prédicat $\sigma \Vdash^* \mathcal{R}$ est Π_{m+1}^0 . ■

Nous voyons à présent le niveau de généricité requis pour rendre vraie la proposition 4.8.

Proposition 5.3. Soit $\mathcal{R}$ un contrat arithmétique tel que $\sigma \Vdash^* \mathcal{R}$ pour σ dans $2^{<\mathbb{N}}$.

- (1) Si $\mathcal{R}$ est Σ_n^0 , alors $\mathcal{R}$ est satisfait pour tout faiblement $(n-2)$ -générique qui étend σ .
- (2) Si $\mathcal{R}$ est Π_n^0 , alors $\mathcal{R}$ est satisfait pour tout faiblement $(n-1)$ -générique qui étend σ . ★

PREUVE. Par la définition de la relation de forcing, si $\mathcal{R}$ est Σ_1^0 ou Π_1^0 , alors il est satisfait pour tout ensemble qui étend σ . Supposons la proposition vraie pour n . Soit $\mathcal{R} = \exists x \mathcal{Q}(x)$ un contrat Σ_{n+1}^0 , avec $\mathcal{Q}(x)$ un contrat Π_n^0 . On a $\sigma \Vdash^* \mathcal{R}$ si, et seulement si, il existe $m \in \mathbb{N}$ tel que $\sigma \Vdash^* \mathcal{Q}(m)$. Par hypothèse d'induction, $\mathcal{Q}(m)$ est satisfait pour tout ensemble $(n-1)$ -générique qui étend σ , et il en est donc de même pour $\mathcal{R}$.

Supposons à présent que $\mathcal{R} = \forall x \mathcal{Q}(x)$ est un contrat Π_{n+1}^0 , avec $\mathcal{Q}(x)$ un contrat Σ_n^0 . On a $\sigma \Vdash^* \mathcal{R}$ si, et seulement si, pour tout $m \in \mathbb{N}$ et pour tout $\tau \succeq \sigma$,

$$\tau \nVdash^* \neg \mathcal{Q}(m).$$

D'après la proposition 4.7, cela signifie que pour m entier fixé l'ensemble $A_m = \{\tau : \tau \Vdash^* \mathcal{Q}(m)\}$ est dense sous σ . D'après la proposition 5.2, chaque ensemble A_m est Σ_n^0 . En particulier, si G est faiblement n -générique et étend σ , alors G rencontre A_m . Par hypothèse d'induction, si G rencontre A_m et est n -générique, alors $\mathcal{Q}(m)$ est vrai pour G . Comme c'est le cas pour tout m , alors $\mathcal{R}$ est vrai pour tout ensemble n -générique G qui étend σ . ■

Nous pouvons finalement montrer le théorème 5.1.

PREUVE DU THÉORÈME 5.1. Le cas Σ_1^0/Π_1^0 a déjà été traité au moyen du théorème 3.12. Soit $n > 1$. On peut supposer sans perte de généralité que $\mathcal{R}$ est Σ_n^0 . Dans le cas inverse, on reproduit l'argument suivant avec $\neg \mathcal{R}$ à la place de $\mathcal{R}$.

Soit $U = \{\sigma \in 2^{<\mathbb{N}} : \sigma \Vdash^* \mathcal{R}\}$. D'après le lemme 4.5,

$$U^\perp = \{\sigma \in 2^{<\mathbb{N}} : \sigma \Vdash^* \neg \mathcal{R}\}.$$

D'après la proposition 5.2, l'ensemble U est Σ_n^0 , et l'ensemble $U^\perp$ est Π_n^0 . Notons que comme G est n -générique, il rencontre $U \cup U^\perp$.

Supposons que $\mathcal{R}$ soit satisfait sur G . Alors, G ne peut pas rencontrer $U^\perp$, car on aurait dans ce cas un préfixe $\sigma \prec G$ tel que $\sigma \Vdash^* \neg \mathcal{R}$ et, par la proposition 5.3, $\neg \mathcal{R}$ serait donc satisfait sur G , ce qui contredit le fait que $\mathcal{R}$ soit satisfait sur G . Donc, G rencontre U , et l'on a un préfixe $\sigma \prec G$ tel que $\sigma \Vdash^* \mathcal{R}$.

Supposons que $\neg \mathcal{R}$ soit satisfait sur G . Symétriquement, G rencontre nécessairement $U^\perp$, et l'on a un préfixe $\sigma \prec G$ tel que $\sigma \Vdash^* \mathcal{R}$.

Réciproquement, si un préfixe $\sigma \prec G$ est tel que $\sigma \Vdash^* \mathcal{R}$ ou $\sigma \Vdash^* \neg \mathcal{R}$, alors respectivement $\mathcal{R}$ ou $\neg \mathcal{R}$ est satisfait sur G , d'après la proposition 5.3. ■

5.2. Degré Turing des ensembles suffisamment générique

Récapitulons d'abord un résultat que nous avons déjà établi : étant donné un ensemble A , si G est suffisamment générique, alors A ne calcule pas G et G ne calcule pas A .

Théorème 5.4

Soit $A \in 2^{\mathbb{N}}$ un ensemble non calculable. Si G est 1-générique relativement à A , alors A et G sont dans des degrés Turing incomparables.

PREUVE. D'après l'exercice 3.5 et l'exercice 3.6, si G est faiblement 1-générique relativement à A , il est de degré A -hyperimmune, et ne peut donc être calculé par A . D'après le théorème 3.28, si G est 1-générique relativement à A , il ne calcule pas A . ■

Notons que le théorème 5.4 peut être généralisé pour montrer que pour toute suite dénombrable d'ensembles fixée à l'avance $A_0, A_1, \dots$, tout ensemble G suffisamment générique pour le forcing de Cohen est incomparable avec les éléments de cette liste. En effet, si G est suffisamment générique, il sera 1-générique relativement à A_i pour tout i . Ce résultat, malgré sa simplicité, admet plusieurs conséquences intéressantes.

Corollaire 5.5

Soit G un ensemble suffisamment générique pour le forcing de Cohen. Alors, G n'est pas arithmétique et ne calcule aucun ensemble arithmétique non calculable.

Voyons à présent comment renforcer et itérer le théorème 5.4 : pour A non $\emptyset^{(n)}$ -calculable, si un ensemble G est suffisamment générique, non seulement il ne calculera pas A , mais aussi son n -ième saut Turing ne calculera pas A . Nous allons en fait voir quelque chose d'un peu plus précis : si A n'est pas Σ_n^0 et si G est suffisamment générique, alors A ne sera pas $\Sigma_n^0(G)$. Assurons-nous d'abord de la complexité du contrat $a \in G^{(n)}$, via le lemme suivant.

Lemme 5.6. Pour tout $a \in \mathbb{N}$, le contrat $a \in X^{(n)}$ est Σ_n^0 uniformément en a . ★

PREUVE. Pour le cas $n = 1$, on a $a \in X'$ ssi $\Phi_a(X, a) \downarrow$, ce qui est bien un contrat Σ_1^0 . Au cas n , supposons que $F_n(G, x)$ soit une formule Σ_n^0 de l'arithmétique du second ordre telle que pour tout $X \in 2^{\mathbb{N}}$ et pour tout $a \in \mathbb{N}$, la formule $F(X, a)$ est vraie ssi $a \in X^{(n)}$. On a alors :

$$\begin{aligned} a \in X^{(n+1)} &\leftrightarrow \exists \sigma \forall i < |\sigma| \left(\bigvee \begin{array}{l} (\sigma(i) = 0 \wedge i \notin X^{(n)}) \\ (\sigma(i) = 1 \wedge i \in X^{(n)}) \end{array} \right) \wedge \Phi_a(\sigma, a) \downarrow \\ &\leftrightarrow \exists \sigma \forall i < |\sigma| \left(\bigvee \begin{array}{l} (\sigma(i) = 0 \wedge \neg F_n(X, i)) \\ (\sigma(i) = 1 \wedge F_n(X, i)) \end{array} \right) \wedge \Phi_a(\sigma, a) \downarrow. \end{aligned}$$

En utilisant le fait que $\neg F_n(G, x)$ et $F_n(G, x)$ sont toutes les deux des formules Σ_{n+1}^0 , et en utilisant la clôture des formules Σ_{n+1}^0 par quantifi-

cation bornée, réunion finie et intersection finie, on peut définir une formule $\Sigma_{n+1}^0 H(G, \tau)$ telle que

$$H(X, \sigma) \leftrightarrow \forall i < |\sigma| ((\sigma(i) = 0 \wedge \neg F_n(X, i)) \vee (\sigma(i) = 1 \wedge F_n(X, i))).$$

Alors,

$$a \in X^{(n+1)} \leftrightarrow \exists \sigma H(X, \sigma) \wedge \Phi_a(\sigma, a) \downarrow,$$

ce qui est bien une formule Σ_{n+1}^0 .

Théorème 5.7

Soit A un ensemble non Σ_{n+1}^0 pour un entier $n \geq 0$. Alors, pour tout ensemble G 1-générique relativement à $A \oplus \emptyset^{(n)}$, A n'est pas $\Sigma_{n+1}^0(G)$.

PREUVE. Soit G un ensemble 1-générique relativement à $A \oplus \emptyset^{(n+1)}$. Il s'agit de montrer que pour tout e l'ensemble A est différent de $W_e^{G^{(n)}}$, l'ensemble $G^{(n)}$ -c. e. de code e . Soit

$$U = \{\sigma : \exists m \notin A \ \sigma \Vdash^* m \in W_e^{G^{(n)}}\}.$$

D'après le lemme 5.6, le contrat $m \in W_e^{G^{(n)}}$ est Σ_{n+1}^0 . Par suite, d'après la proposition 5.2, l'ensemble U est $\Sigma_1^0(A \oplus \emptyset^{(n)})$. Si G rencontre U , alors d'après la proposition 5.3 il existe $m \notin A$ tel que $m \in W_e^{G^{(n)}}$, et $A \neq W_e^{G^{(n)}}$.

Si G ne rencontre pas U , comme l'ensemble G est 1-générique relativement à $A \oplus \emptyset^{(n+1)}$, G rencontre $U^\perp = \{\sigma : \forall m \notin A \ \forall \tau \succeq \sigma \ \tau \nVdash^* m \in W_e^{G^{(n)}}\}$. D'après le lemme 4.5, on a

$$U^\perp = \{\sigma : \forall m \notin A \ \sigma \Vdash^* m \notin W_e^{G^{(n)}}\}.$$

Soit $\sigma \in U^\perp$ une chaîne fixée. Alors, l'ensemble

$$D_\sigma = \{m : \sigma \Vdash^* m \notin W_e^{G^{(n)}}\}$$

est un ensemble Π_{n+1}^0 , qui par hypothèse contient $\mathbb{N} \setminus A$. Comme $\mathbb{N} \setminus A$ n'est pas Π_{n+1}^0 , il y a forcément un élément $m \in A$ tel que $m \in D_\sigma$. Donc,

$$U^\perp \subseteq \{\sigma : \exists m \in A \ \sigma \Vdash^* m \notin W_e^{G^{(n)}}\}.$$

Alors, d'après la proposition 5.3, on aura $m \notin W_e^{G^{(n)}}$ pour $m \in A$, et donc $A \neq W_e^{G^{(n)}}$. ■

Corollaire 5.8

Soit A un ensemble non $\emptyset^{(n)}$ -calculable pour $n \geq 0$. Alors, si G est 1-générique relativement à $A \oplus \emptyset^{(n)}$, l'ensemble A n'est pas $G^{(n)}$ -calculable.

PREUVE. Comme A n'est pas $\emptyset^{(n)}$ -calculable, il n'est pas Δ_{n+1}^0 . Comme A n'est pas Δ_{n+1}^0 , soit A n'est pas Σ_{n+1}^0 , soit $\bar{A}$ n'est pas Σ_{n+1}^0 . Par le

théorème 5.7, pour tout ensemble G 1-générique relativement à $A \oplus \emptyset^{(n)}$, l'ensemble A n'est pas $\Sigma_{n+1}^0(G)$ dans le premier cas, et $\bar{A}$ n'est pas $\Sigma_{n+1}^0(G)$ dans le second cas. Dans tous les cas, A n'est pas $\Delta_{n+1}^0(G)$, et il n'est donc pas $G^{(n)}$ -calculable. ■

Nous avons vu avec le théorème 3.20 que le saut Turing était une fonction continue sur la classe des ensembles 1-génériques : les 1-génériques sont tous low généralisé. Ce résultat se relativise : le n -ième saut Turing est une fonction continue sur la classe des ensembles n -génériques.

Théorème 5.9

Soit G un ensemble n -générique. Alors, $G^{(n)} \leq_T G \oplus \emptyset^{(n)}$.

PREUVE. Soit $n > 0$, et soit G un ensemble n -générique. Soit

$$U_e = \{\sigma : \sigma \Vdash^* e \in G^{(n)}\}.$$

D'après le lemme 5.6, le contrat $e \in G^{(n)}$ est Σ_n^0 , et U_e est donc un ensemble Σ_n^0 . Considérons $U_e^\perp = \{\sigma : \forall \tau \succeq \sigma \ \tau \nVdash^* e \in G^{(n)}\}$. En particulier, $U_e^\perp$ est un ensemble Π_n^0 . D'après le lemme 4.5,

$$U_e^\perp = \{\sigma : \sigma \Vdash^* e \notin G^{(n)}\}.$$

Comme G est n -générique, il rencontre $U_e \cup U_e^\perp$. Il suffit à l'aide de $\emptyset^{(n)}$ de chercher un préfixe de G dans U_e ou $U_e^\perp$. D'après la proposition 5.3, on a dans le premier cas $e \in G^{(n)}$, et dans le second $e \notin G^{(n)}$. ■

Notons que le corollaire 5.8 se déduit aussi du théorème précédent et d'une version relativisée du théorème 3.28. Attention ! Dans la littérature, la notion de low_n -généralisé ne correspond pas à la propriété du théorème 5.9.

Définition 5.10. Un ensemble $G \in 2^{\mathbb{N}}$ est low_n généralisé si

$$G^{(n)} \leq_T (G \oplus \emptyset')^{(n-1)}.$$

◇

Notons que si X est low_n généralisé, il est aussi low_{n+1} généralisé. Chaque ensemble n -générique est bien low_n généralisé, mais le théorème 5.9 prouve quelque chose de plus fort.

Chapitre 11

Forcing effectif

Fort des intuitions créées avec l'étude du forcing de Cohen, nous pouvons alors introduire les notions abstraites de forcing sur un ordre partiel arbitraire, et développer toute la machinerie associée.

1. Fondements du forcing

Maintenant que nous nous sommes familiarisés avec les notions de densité et de généricité sur l'ordre partiel des chaînes binaires, muni de la relation d'extension, nous sommes prêts à aborder les concepts du forcing dans toute leur généralité, tout en conservant les intuitions de la méthode des extensions finies. Les bénéfices de cette abstraction n'apparaîtront qu'à partir de l'introduction de la relation de forcing, qui donne toute sa puissance au formalisme. Jusqu'ici, nous avons vu la notion de généricité dans l'ordre partiel des chaînes binaires comme une systématisation des constructions avec la méthode des extensions finies.

Ordre partiel. Dans toute sa généralité, une notion de forcing est tout simplement un ordre partiel $(\mathbb{P}, \leq)$, dont les éléments sont appelés *conditions*. Une condition représente intuitivement une approximation de l'objet que l'on est en train de construire. Une *extension* de $c \in \mathbb{P}$ est une condition $d \leq c$.

Remarque

Attention, pour des raisons historiques, la relation d'ordre du forcing est inversée. Une extension d'une condition c est donc une condition d plus petite. L'idée sous-jacente vient du fait que d est une approximation plus précise que c , et qu'ainsi l'ensemble des objets « candidats » que l'on

est en train de construire est un sous-ensemble des candidats de c , car plus l'on ajoute de contraintes, plus l'on exclut des candidats.

Filtre. Dans le cas de la méthode des extensions finies, l'objet construit est un élément de $2^{\mathbb{N}}$, à l'aide d'une suite infinie strictement croissante de chaînes. Dans le langage d'un ordre partiel arbitraire, nous allons construire une suite infinie décroissante de conditions. L'objet produit est un filtre maximal.

Définition 1.1. Deux conditions c_0, c_1 sont *compatibles* s'il existe une condition d qui étend à la fois c_0 et c_1 . Dans le cas contraire, c_0 et c_1 sont *incompatibles*. Un *filtre* est un ensemble $F \subseteq \mathbb{P}$ clos par le haut, tel que pour tous $c_0, c_1 \in F$, il existe une condition $d \in F$ telle que $d \leq c_0, c_1$. Un filtre est *maximal* s'il n'est pas inclus dans un filtre strictement plus grand. ◇

Si l'on considère l'ordre partiel sur les chaînes, muni de la relation de suffixe, les filtres maximaux sont en correspondance bijective avec l'espace de Cantor. En effet, pour tout $X \in 2^{\mathbb{N}}$, l'ensemble $\{X \upharpoonright_n : n \in \mathbb{N}\}$ est un filtre maximal, et tout filtre maximal est inversement de cette forme.

Dans le cadre d'un ordre partiel dénombrable, il peut être plus intuitif de se représenter un filtre comme la clôture par le haut d'une suite infinie décroissante de conditions. En particulier, pour toute chaîne infinie décroissante de conditions $c_0 \geq c_1 \geq c_2 \geq \dots$, l'ensemble

$$F = \{d \in \mathbb{P} : \exists n \ c_n \leq d\}$$

est un filtre. Cette intuition correspond plus à la construction de la méthode des extensions finies.

Notation

Comme expliqué, une condition $c \in \mathbb{P}$ peut être vue comme une approximation de l'objet que l'on construit, à savoir un filtre maximal. On peut donc associer à chaque condition l'ensemble $[c]_{\in}$ des filtres maximaux contenant c , représentant les objets candidats. En particulier, suivant l'intuition, si $d \leq c$, alors l'approximation d est plus précise que l'approximation c , réduisant ainsi le nombre de candidats. On a donc $[d]_{\in} \subseteq [c]_{\in}$.

Remarquons que pour tout filtre maximal F , $\bigcap_{c \in F} [c]_{\in} = \{F\}$. Autrement dit, F est l'unique candidat de toutes les conditions du filtre simultanément. Nous avons déjà rencontré plusieurs notions de forcing au cours des chapitres précédents. En voici quelques-uns. Nous allons voir que pour chacune de ces notions de forcing, les filtres maximaux peuvent être interprétés comme des ensembles d'entiers.

Exemple 1.2.

1. Le *forcing de Cohen* est l'ordre partiel des chaînes muni de sa relation de suffixe $(2^{<\mathbb{N}}, \succeq)$. Pour tout $\sigma \in 2^{<\mathbb{N}}$, $[\sigma]_\infty$ est en bijection avec l'ensemble $[\sigma] = \{X \in 2^\mathbb{N} : \sigma \prec X\}$, par la fonction qui à $F \in [\sigma]_\infty$ associe l'unique élément de $\bigcap_{\sigma \in F} [\sigma]$.
2. Le *forcing de Jockusch–Soare* est l'ordre partiel des classes Π_1^0 non vides, ordonnés par la relation d'inclusion. Pour toute classe $\Pi_1^0 \mathcal{P}$, l'ensemble $[\mathcal{P}]_\infty$ est en bijection avec $\mathcal{P}$. Un filtre maximal F contenant $\mathcal{P}$ peut donc être vu comme l'unique élément de $\bigcap_{Q \in F} Q$, qui est un membre de $\mathcal{P}$.
3. Le *forcing de Sacks* est l'ordre partiel des f -arbres calculables, ordonnées par la relation de sous- f -arbre. Pour tout f -arbre calculable T , l'ensemble $[T]_\infty$ est en bijection avec l'ensemble

$$[T] = \left\{ \bigcup_n T(X \upharpoonright_n) : X \in 2^\mathbb{N} \right\} = \{Y \in 2^\mathbb{N} : \exists^\infty n \ Y \upharpoonright_n \in \text{Im } T\}.$$

Dans chacun des exemples précédents, pour tout filtre maximal F , on notera $\dot{F}$ l'élément de $2^\mathbb{N}$ correspondant. On a donc pour chacune de ces notions de forcing l'égalité $[c] = \{\dot{F} : F \in [c]_\infty\}$.

Densité, généricité. Rappelons que dans la méthode des extensions finies, les contrats peuvent être représentés comme l'ensemble des chaînes qui les forcent. Cette représentation a l'avantage de s'abstraire de la notion de contrat et se généralise à tout ordre partiel.

Définition 1.3. Soit $(\mathbb{P}, \leq)$ un ordre partiel. Un ensemble $D \subseteq \mathbb{P}$ est *dense* dans $(\mathbb{P}, \leq)$ si, pour tout $c \in \mathbb{P}$, il existe un $d \leq c$ tel que $d \in D$. $\diamond$

L'intuition qu'il est utile de conserver avec la notion de densité est que si un ensemble D est dense, alors quel que soit le morceau fini de la suite décroissante de conditions que l'on a déjà construit, il n'est jamais trop tard pour intégrer un élément de D dans la suite.

Définition 1.4. Soit $\vec{D} = (D_n)_{n \in \mathbb{N}}$ une collection d'ensembles de conditions. Un filtre $F \subseteq \mathbb{P}$ est *$\vec{D}$ -générique* s'il intersecte l'ensemble D_n pour tout $n \in \mathbb{N}$. $\diamond$

La proposition suivante montre que si les ensembles de conditions sont denses, il existe un filtre générique pour ces ensembles. La preuve de cette proposition correspond à la construction par la méthode des extensions finies d'une suite infinie croissante de chaînes satisfaisant chaque contrat.

Proposition 1.5. Soit $\vec{D} = (D_n)_{n \in \mathbb{N}}$ une collection d'ensembles denses et soit $c \in \mathbb{P}$ une condition. Il existe un filtre $\vec{D}$ -générique contenant c . ★

PREUVE. Définissons inductivement une suite infinie décroissante de conditions $c_0 \geq c_1 \geq c_2 \geq \dots$ comme suit : $c_0 = c$. Si c_n est défini, $c_{n+1} \leq c_n$ est une condition appartenant à D_n . Une telle extension de c_n existe par densité de l'ensemble D_n . Soit $F = \{d \in \mathbb{P} : \exists n \, d \geq c_n\}$. L'ensemble F est un filtre contenant c et rencontrant D_n pour tout n . ■

Comme expliqué dans la section 10-2, il existe en général une quantité indénombrable d'ensembles denses, et un filtre F ne peut pas être générique pour tous ces ensembles simultanément. La notion de généricité est donc dépendante d'une collection $\vec{D}$ dénombrable d'ensembles denses. Nous dirons que tout filtre *suffisamment générique* pour une notion de forcing satisfait telle propriété s'il existe une collection dénombrable d'ensembles denses $\vec{D}$ telle que tout filtre $\vec{D}$ -générique satisfait cette propriété.

Définition 1.6. Soit $(\mathbb{P}, \leq)$ un ordre partiel, et soient $c \in \mathbb{P}$ et $D \subseteq \mathbb{P}$ un ensemble. On dit que D est *dense sous c* si pour tout $d \leq c$, il existe un $e \leq d$ tel que $e \in D$. ♦

L'exercice suivant sera utile dans la suite de ce développement.

Exercice 1.7. Supposons qu'un ensemble $D \subseteq \mathbb{P}$ est dense sous $c \in \mathbb{P}$. Montrer que tout filtre F suffisamment générique contenant la condition c intersecte D . ♦

2. Relation de forcing

Jusqu'ici, nous avons développé des notions s'exprimant purement en termes d'ordre partiel, à savoir, les notions de densité, de filtre et de généricité. Nous allons maintenant définir une généralisation de la relation de forcing introduite dans la section 10-4. Pour cela, nous allons nous restreindre à des notions de forcing produisant des ensembles d'entiers.

Définition 2.1. Un *forcing de Cantor* est un ordre partiel $(\mathbb{P}, \leq)$ muni d'une fonction $F \mapsto \hat{F}$ des filtres maximaux vers l'espace de Cantor $2^{\mathbb{N}}$, telle pour tout $c \in \mathbb{P}$, et tout $\sigma \in 2^{<\mathbb{N}}$ pour lequel $[c] \cap [\sigma] \neq \emptyset$, il existe une condition $d \leq c$ telle que $[d] \subseteq [\sigma]$. Ici, la notation $[c]$ désigne l'ensemble $\{\hat{F} : F \in [c]\}$. ♦

2.1. Relation sémantique de forcing

Par abus de langage, on dira qu'un ensemble $G \in 2^{\mathbb{N}}$ est *suffisamment générique* pour un forcing de Cantor s'il est de la forme $\dot{F}$ pour un filtre suffisamment générique.

Définition 2.2. Une condition c force sémantiquement un contrat $\mathcal{R}$, auquel cas l'on note $c \Vdash \mathcal{R}$ si $\dot{F}$ satisfait $\mathcal{R}$ pour tout filtre maximal F suffisamment générique et contenant c . $\diamond$

Cette définition sémantique nous donne « gratuitement » certaines propriétés de la relation.

Proposition 2.3. Soit $(\mathbb{P}, \leq)$ un forcing de Cantor, et soient $c, d \in \mathbb{P}$. Soit $\mathcal{R}$ un contrat.

(1) Si $c \Vdash \mathcal{R}$ et $d \leq c$, alors $d \Vdash \mathcal{R}$.

(2) Si $c \Vdash \mathcal{R}$, alors $c \not\Vdash \neg \mathcal{R}$. $\star$

PREUVE. (1) Soit F un filtre suffisamment générique contenant d . Par clôture par le haut des filtres, $c \in F$, et comme c force $\mathcal{R}$, $\dot{F}$ satisfait donc $\mathcal{R}$. Il s'ensuit que d force $\mathcal{R}$.

(2) Si c force $\mathcal{R}$ et $\neg \mathcal{R}$, alors pour tout filtre F suffisamment générique contenant c , $\dot{F}$ satisfait $\mathcal{R}$ et $\neg \mathcal{R}$, contradiction. $\blacksquare$

Exercice 2.4. Soit $(\mathbb{P}, \leq)$ un forcing de Cantor, et soit $c \in \mathbb{P}$. Soit $\mathcal{R}$ un contrat. Montrer que si $\{d \in \mathbb{P} : d \Vdash \mathcal{R}\}$ est dense sous c , alors $c \Vdash \mathcal{R}$. $\diamond$

En revanche, certaines propriétés sont beaucoup moins évidentes à prouver. Nous allons voir par exemple que pour tout contrat arithmétique $\mathcal{R}$, l'ensemble des conditions qui forcent $\mathcal{R}$ ou qui forcent $\neg \mathcal{R}$ est dense. Tout comme dans le cas du forcing de Cohen avec l'ordre partiel des chaînes $(2^{<\mathbb{N}}, \supseteq)$, nous allons définir une relation de forcing syntaxique qui nous permettra de rendre compte de ce phénomène.

Nous insistons auparavant sur les trois propriétés fondamentales que doit avoir une relation de forcing syntaxique.

Définition 2.5 (Relation de forcing). Étant donné $(\mathbb{P}, \leq)$ un forcing de Cantor, on appelle *relation de forcing* une relation $\Vdash^\circ$ satisfaisant les propriétés suivantes pour tous $c, d \in \mathbb{P}$ et tout contrat arithmétique $\mathcal{R}$.

(1) Si $c \Vdash^\circ \mathcal{R}$, alors $c \Vdash \mathcal{R}$.

- (2) Si $c \Vdash^\circ \mathcal{R}$ et $d \leq c$, alors $d \Vdash^\circ \mathcal{R}$.
- (3) L'ensemble $\{c \in \mathbb{P} : c \Vdash^\circ \mathcal{R} \text{ ou } c \Vdash^\circ \neg \mathcal{R}\}$ est dense. ◇

Les propriétés (1-3) correspondent aux propositions 10-4.8, 10-4.6 et 10-4.7 pour le forcing de Cohen. Il découle immédiatement de (1) que si $c \Vdash^\circ \mathcal{R}$, alors $c \not\Vdash^\circ \neg \mathcal{R}$. Rappelons qu'un ensemble $S \subseteq \mathbb{P}$ est dense sous $c \in \mathbb{P}$ si pour tout $d \leq c$, il existe un $e \leq d$ tel que $e \in S$.

Proposition 2.6. Soit $(\mathbb{P}, \leq)$ un forcing de Cantor, et soit $c \in \mathbb{P}$. Soient $\mathcal{R}$ un contrat arithmétique et $\Vdash^\circ$ une relation de forcing. Alors,

$$c \Vdash \mathcal{R} \quad \text{si, et seulement si,} \quad \{d \in \mathbb{P} : d \Vdash^\circ \mathcal{R}\}$$

est dense sous c . ★

PREUVE. Supposons que $c \Vdash^\circ \mathcal{R}$. Soit $d \leq c$. Par la définition 2.5(3), il existe un $e \leq c$ tel que $e \Vdash^\circ \mathcal{R}$ ou $e \Vdash^\circ \neg \mathcal{R}$. Si le second cas se présente, alors par la définition 2.5(1), $e \Vdash \neg \mathcal{R}$. On a donc $e \Vdash \mathcal{R}$ et $e \Vdash \neg \mathcal{R}$, ce qui contredit la proposition 2.3(2). Le premier cas se présente donc. Nous avons montré la densité de l'ensemble $\{d \in \mathbb{P} : d \Vdash^\circ \mathcal{R}\}$ sous c .

Réciproquement, supposons que l'ensemble $\{d \in \mathbb{P} : d \Vdash^\circ \mathcal{R}\}$ soit dense sous c . Soit F un filtre suffisamment générique contenant c . Par généralité, il existe $d \in F$ tel que $d \Vdash^\circ \mathcal{R}$, et par la définition 2.5(1), $d \Vdash \mathcal{R}$, donc $\dot{F}$ satisfait $\mathcal{R}$. Il s'ensuit que $c \Vdash \mathcal{R}$. ■

2.2. Relation syntaxique de forcing

Nous définissons à présent notre relation syntaxique de forcing $\Vdash^*$, qui généralise directement la relation $\Vdash^*$ définie dans la section 10-4 pour le forcing de Cohen. Tout comme pour le forcing de Cohen, l'intérêt de la relation $c \Vdash^* \mathcal{R}$ est qu'elle est simple à définir : relativement à $\mathbb{P}$, elle a la même complexité arithmétique que celle du contrat $\mathcal{R}$. Nous verrons dans les sections suivantes que $\mathbb{P}$ en tant qu'ordre partiel est malheureusement rarement calculable, ce qui amène à des complexités calculatoires supplémentaires qui doivent être traitées au cas par cas.

Définition 2.7. Soit $(\mathbb{P}, \leq)$ un forcing de Cantor. On définit la relation $\Vdash^*$ pour tout $c \in \mathbb{P}$ et tout contrat arithmétique $\mathcal{R}$:

- (1) $c \Vdash^* \mathcal{R}$ pour $\mathcal{R}$ un contrat Σ_1^0 ou Π_1^0 ssi tout $X \in [c]$ satisfait $\mathcal{R}$;
- (2) $c \Vdash^* \exists x \mathcal{R}(x)$ pour $\mathcal{R}(x)$ un contrat Π_k^0 avec $k \geq 1$ ssi il existe un $n \in \mathbb{N}$ tel que $c \Vdash^* \mathcal{R}(n)$;
- (3) $c \Vdash^* \forall x \mathcal{R}(x)$ pour $\mathcal{R}(x)$ un contrat Σ_k^0 avec $k \geq 1$ pour tout $d \leq c$ et tout $n \in \mathbb{N}$, $d \not\Vdash^* \neg \mathcal{R}(n)$. ◇

Notons que tout comme dans le cas du forcing de Cohen, on a $c \Vdash^* \forall x \mathcal{R}(x)$ ssi $\forall d \leq c \ d \Vdash^* \exists x \neg \mathcal{R}(x)$. Nous laissons en exercice la preuve que la relation $\Vdash^*$ respecte les points (1-3) d'une relation de forcing de la définition 2.5. Il s'agit à chaque fois de simples preuves par induction sur la complexité des contrats.

Exercice 2.8. (★) Soit $(\mathbb{P}, \leq)$ un forcing de Cantor, et soient $c, d \in \mathbb{P}$. Soit $\mathcal{R}$ un contrat arithmétique. Montrer que si $c \Vdash^* \mathcal{R}$ et $d \leq c$, alors $d \Vdash^* \mathcal{R}$. $\diamond$

Exercice 2.9. (★) Soit $(\mathbb{P}, \leq)$ un forcing de Cantor et soit $\mathcal{R}$ un contrat arithmétique. Montrer que $\{c \in \mathbb{P} : c \Vdash^* \mathcal{R} \text{ ou } c \Vdash^* \neg \mathcal{R}\}$ est dense. $\diamond$

Exercice 2.10. (★) Soit $(\mathbb{P}, \leq)$ un forcing de Cantor, et soit $c \in \mathbb{P}$. Soit $\mathcal{R}$ un contrat arithmétique. Montrer que si $c \Vdash^* \mathcal{R}$, alors $c \Vdash \mathcal{R}$. $\diamond$

— La complexité de $\Vdash$ —

La complexité de $\Vdash^*$ a des conséquences sur la complexité de la relation sémantique de forcing $\Vdash$. D'après les exercices précédents, $\Vdash^*$ respecte bien les points (1-3) de la définition 2.5. Donc, d'après la proposition 2.6, la relation $c \Vdash \mathcal{R}$ est équivalente à $\forall d \leq c \ \exists e \leq d \ e \Vdash^* \mathcal{R}$, ce qui par exemple pour un contrat Σ_n^0 sera un prédicat $\Pi_{n+1}^0(\mathbb{P})$. Il s'agit d'une simplification considérable de la relation sémantique, mais qui reste — relativement à $\mathbb{P}$ — de complexité arithmétique supérieure à celle des contrats qu'elle force, ce qui nous fera préférer la relation $\Vdash^*$.

3. Forcing avec des arbres

Le forcing à base d'arbres est une des grandes familles de forcing. Nous en détaillons ici deux exemples, déjà rencontrés dans les chapitres précédents : le forcing de Jockusch-Soare et le forcing de Sacks calculable. Ces notions ont été au départ créées pour contrôler le simple saut via le forcing de contrats Σ_1^0/Π_1^0 , ou le double saut via le forcing de contrats Σ_2^0/Π_2^0 . Nous en discuterons dans la section 4, et nous nous contentons pour le moment de voir en quoi ces forcings ont été utilisés implicitement à plusieurs reprises dans ce livre.

3.1. Forcing de Jockusch-Soare

Le forcing de Jockusch-Soare correspond à l'ordre partiel des classes Π_1^0 non vides, partiellement ordonnées par la relation d'inclusion. On peut associer à tout filtre maximal F sur cet ordre un ensemble d'entiers $\dot{F} \in 2^{\mathbb{N}}$ qui est

l'élément du singleton $\bigcap_{\mathcal{P} \in F} \mathcal{P}$. Muni de cette interprétation des filtres, le forcing de Jockusch-Soare est un forcing de Cantor (voir la définition 2.1).

Nous avons déjà rencontré plusieurs utilisations du forcing de Jockusch-Soare dans le chapitre 8 sur les classes Π_1^0 et les degrés PA. Voici une reformulation des théorèmes de base calculatoirement dominée et de base d'évitement de cône, via le vocabulaire du forcing.

Théorème (reformulation des th. 8-4.5 et 8-4.7)

Soit A un ensemble non calculable et soit G un ensemble suffisamment générique pour le forcing de Jockusch-Soare. Alors, G est calculatoirement dominé et ne calcule pas A .

PREUVE. Soit $(\mathbb{P}, \leq)$ le forcing Jockusch-Soare, c'est-à-dire l'ensemble des classes Π_1^0 non vides ordonnées par l'inclusion. La preuve du théorème 8-4.5 montre la chose suivante : pour toute fonctionnelle Φ_e et tout $c \in \mathbb{P}$, il existe $d \leq c$ tel que $d \Vdash^* \exists n \Phi_e(G, n) \uparrow$ ou il existe $d \leq c$ et une fonction calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $d \Vdash^* \Phi_e(G, n) \downarrow < f(n)$ pour tout n . Donc, l'ensemble C_e des conditions qui forcent Φ_e à être partielle ou bien bornée par une fonction calculable est dense. Tout filtre suffisamment générique contient une condition dans chacun des C_e . Si $G \in 2^{\mathbb{N}}$ est donc suffisamment générique, il est calculatoirement dominé.

La preuve du théorème 8-4.7 quant à elle montre la chose suivante : pour toute fonctionnelle Φ_e l'ensemble $D_e = \{c \in \mathbb{P} : c \Vdash^* \exists n \Phi_e(G, n) \uparrow \text{ ou } \exists n \ c \Vdash^* \Phi_e(G, n) \downarrow \neq A(n)\}$ est dense. Si G est donc suffisamment générique, il ne calcule pas A . ■

En ce qui concerne le théorème de la base low, les choses sont différentes : c'est bien le forcing de Jockusch-Soare que nous utilisons pour construire un ensemble low, mais il s'agit d'une utilisation effective de ce forcing, où l'on contrôle à l'aide de $\emptyset'$ le passage d'une étape n à une étape $n + 1$. Il ne s'agit donc pas à proprement parler d'un résultat de forcing, dans le sens où ce n'est pas une propriété satisfaite par tout ensemble suffisamment générique, mais au contraire par une petite classe dénombrable d'ensembles qui sont peu génériques.

Nous avons vu avec le forcing de Cohen que tout ensemble suffisamment générique différerait d'une quantité dénombrable d'ensembles fixés à l'avance (voir le théorème 10-5.4). En particulier, aucun ensemble suffisamment générique pour le forcing de Cohen n'est arithmétique. Ce n'est pas le cas du forcing de Jockusch-Soare. En effet, pour tout X calculable, le singleton $\{X\}$ est une classe Π_1^0 et l'unique filtre contenant $\{X\}$ — à savoir le filtre de toutes les classes Π_1^0 ayant X comme chemin infini — est maximal. L'ensemble X est donc aussi générique que l'on veut sous la condition $\{X\}$.

Il est toutefois possible de modifier légèrement le forcing de Jockusch-Soare afin d'éviter les éléments calculables.

Proposition 3.2. Soit $(\mathbb{P}, \leq)$ l'ordre partiel des classes Π_1^0 non vides sans élément calculable, ordonnées par l'inclusion. Soit $(A_n)_{n \in \mathbb{N}}$ une suite quelconque d'ensembles. Alors, si $G \in 2^{\mathbb{N}}$ est suffisamment générique pour $\mathbb{P}$, il est différent de chaque A_n . ★

PREUVE. Fixons un ensemble A quelconque et montrons que si G est suffisamment générique, il est différent de A . Le résultat suivra automatiquement pour la suite $(A_n)_{n \in \mathbb{N}}$.

Soit $D \subseteq \mathbb{P}$ l'ensemble des classes Π_1^0 de $\mathbb{P}$ ne contenant pas A . Montrons que D est dense dans $\mathbb{P}$. Soit $\mathcal{P} \in \mathbb{P}$. Par la proposition 8-3.6, la classe $\mathcal{P}$ contient au moins deux éléments. Soit $B \in \mathcal{P}$ tel que $B \neq A$, et soit $n \in \mathbb{N}$ tel que $A(n) \neq B(n)$. Alors, la classe $\mathcal{Q} = \{X \in \mathcal{P} : X(n) = B(n)\}$ est une classe Π_1^0 non vide incluse dans $\mathcal{P}$ et telle que $\mathcal{Q} \in D$. Donc, D est dense. ■

La modification du forcing de Jockusch-Soare de la proposition précédente peut se décliner de multiples manières pour obtenir différents résultats : on peut considérer l'ordre partiel des classes Π_1^0 non vides ne contenant que des degrés PA, ou encore celles des classes Π_1^0 non vides ne contenant que des ensembles aléatoires au sens de Martin-Löf (voir le chapitre 18).

3.2. Le forcing de Sacks calculable

Le forcing de Sacks désigne de manière générale l'ordre partiel des arbres parfaits de $2^{<\mathbb{N}}$, sans restriction particulière d'effectivité. Nous nous restreignons en calculabilité aux arbres parfaits calculables, qui ont déjà été abordés via la notion de *f-arbre*.

Rappelons qu'un *f-arbre* est une fonction totale $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ telle que pour tous $\sigma, \tau \in \text{dom } T$, $\sigma \preceq \tau$ si, et seulement si, $T(\sigma) \preceq T(\tau)$. Un *sous-f-arbre* d'un f-arbre T est un f-arbre S tel que $\text{Im } S \subseteq \text{Im } T$. Un f-arbre $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ s'étend en une fonction $\widehat{T} : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ définie par $\{\widehat{T}(X)\} = \bigcap_n [T(X \upharpoonright_n)]$. Un *chemin* de T est un élément de $\text{Im } \widehat{T}$. On note $[T]$ l'ensemble des chemins de T .

On appelle *forcing de Sacks calculable* l'ensemble des f-arbres calculables partiellement ordonnés par la relation de sous-f-arbre. Comme il n'y aura pas d'ambiguïté possible dans ce livre, on dira parfois plus simplement *forcing de Sacks*. On peut associer à tout filtre maximal F sur cet ordre un ensemble d'entiers $\dot{F} \in 2^{\mathbb{N}}$ qui est l'élément du singleton $\bigcap_{T \in F} [T]$. Muni de cette interprétation des filtres, le forcing de Sacks est un forcing de Cantor (voir la définition 2.1).

Remarque

Rappelons qu'une classe $\mathcal{P} \subseteq 2^{\mathbb{N}}$ est parfaite si $\mathcal{P} = [T]$ pour un f-arbre T . Le forcing de Sacks n'est cependant pas la restriction du forcing de Jockusch-Soare aux classes Π_1^0 parfaites : certaines classes Π_1^0 parfaites ne peuvent pas nécessairement être représentées par un f-arbre calculable.

En effet, pour tout f-arbre calculable T , $[T]$ contient une infinité d'éléments calculables, à savoir $T(X)$ pour tout ensemble calculable X , ce qui n'est par exemple pas le cas de la classe Π_1^0 des fonctions DNC_2 , qui est pourtant bien une classe parfaite.

Comme expliqué dans la remarque précédente, tout f-arbre calculable possède une infinité de chemins calculables. En revanche, tout ensemble suffisamment générique pour le forcing de Sacks sera non calculable.

Exercice 3.3. (★) Soit $(A_n)_{n \in \mathbb{N}}$ une suite quelconque d'ensembles. Montrer que tout G suffisamment générique pour le forcing de Sacks est différent de chaque A_n . $\diamond$

Un examen attentif de la première preuve que nous avons donnée de l'existence d'un degré calculatoirement dominé différent de $\mathbf{0}$, à l'aide de f-arbre, montre en fait que l'on a le résultat suivant.

Théorème (reformulation du théorème 7-5.6)

Soit G un ensemble suffisamment générique pour le forcing de Sacks. Alors, G est non calculable et calculatoirement dominé.

PREUVE. Soit $(\mathbb{P}, \leq)$ le forcing de Sacks. La preuve du théorème 7-5.6 montre la chose suivante : pour toute fonctionnelle Φ_e et tout $c \in \mathbb{P}$, il existe $d \leq c$ tel que $d \Vdash^* \exists n \Phi_e(G, n) \uparrow$ ou il existe $d \leq c$ et une fonction calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $d \Vdash^* \Phi_e(G, n) \downarrow < f(n)$ pour tout n . Donc, l'ensemble C_e des conditions qui forcent Φ_e à être partielle ou bien bornée par une fonction calculable est dense. Tout filtre suffisamment générique contient une condition de chacun des C_e . Donc si $G \in 2^{\mathbb{N}}$ est suffisamment générique il est calculatoirement dominé.

Par ailleurs, l'exercice 3.3 montre que si G est suffisamment générique, il est non calculable. $\blacksquare$

Exercice 3.5. (★) Soit A un ensemble non calculable. Montrer que si G est suffisamment générique pour le forcing de Sacks, il ne calcule pas A . $\diamond$

Exercice 3.6. (★★) Un ensemble X est *calculatoirement traçable* (Terwijn et Zambella [222]) s'il existe une borne calculable $h : \mathbb{N} \rightarrow \mathbb{N}$ telle

que pour toute fonction X -calculable $f : \mathbb{N} \rightarrow \mathbb{N}$, il existe une suite calculable $(T_n)_{n \in \mathbb{N}}$ d'ensembles finis tels que $|T_n| \leq h(n)$ et tels que $f(n) \in T_n$ pour tout n .

1. Montrer que si X est suffisamment générique pour le forcing de Sacks, il est calculatoirement traçable.
2. Montrer que si X est calculatoirement traçable via une borne calculable h , alors il est calculatoirement traçable pour n'importe quelle borne calculable h' telle que $h'(n) \leq h'(n+1)$ et $\lim_n h(n) = +\infty$ (Terwijn et Zambella [222]).
3. Soit $(2^n)^\mathbb{N}$ l'ensemble des fonctions $f : \mathbb{N} \rightarrow \mathbb{N}$ telles que $f(n) < 2^n$, et soit $(2^n)^{<\mathbb{N}}$ l'ensemble des préfixes de fonctions de $(2^n)^\mathbb{N}$. Soit $(\mathbb{P}, \leq)$ l'ensemble des conditions de forcing données par $T \in \mathbb{P}$ si $T \subseteq (2^n)^{<\mathbb{N}}$ est un arbre calculable qui vérifie la propriété suivante :
pour tout $\sigma \in T$, il existe $\tau \in T$ avec $\tau \succeq \sigma$ telle que pour tout $i < 2^{|\tau|}$ la chaîne τi est dans T ; autrement dit, chaque nœud a une extension de branchement maximal.
Montrer que tout ensemble suffisamment générique pour ce forcing est calculatoirement dominé et non calculatoirement traçable. $\diamond$

4. Complexité calculatoire et question de forcing

Le forcing de Cohen présente une particularité qui le distingue des autres forcings de la calculabilité : l'ordre partiel $(2^{<\mathbb{N}}, \succeq)$ est calculable : l'ensemble $2^{<\mathbb{N}}$ est calculable et étant donné $\sigma \in 2^{<\mathbb{N}}$, on peut calculer l'ensemble des chaînes $\tau \succeq \sigma$. Une autre de ses particularités est que la relation de forcing des contrats Σ_1^0 et Π_1^0 est respectivement Σ_1^0 et Π_1^0 . Ces deux propriétés permettent d'obtenir la proposition 10-5.2 : si $\mathcal{R}$ est un contrat Σ_n^0 (resp. Π_n^0), le prédicat $\sigma \Vdash^* \mathcal{R}$ est Σ_n^0 (resp. Π_n^0). Ce contrôle très fin de la complexité de la relation de forcing permet alors un contrôle fin des sauts itérés des ensembles génériques, afin d'obtenir par exemple les résultats suivants.

1. Le corollaire 10-5.8 : si X est non $\emptyset^{(n)}$ -calculable et si l'ensemble G est suffisamment générique, alors $G^{(n)}$ ne calcule pas X .
2. Le théorème 10-5.9 : si l'ensemble G est suffisamment générique, on a alors $G \oplus \emptyset^{(n)} \geq_T G^{(n)}$.

C'est en général le genre de propriété que l'on cherche à obtenir avec n'importe quelle notion de forcing : le contrôle de la valeur de vérité de contrats

arithmétiques, plus ce contrôle est fin, meilleurs sont les théorèmes que l'on peut obtenir. Malheureusement, les choses seront rarement aussi simples qu'avec le forcing de Cohen. Examinons la complexité calculatoire du forcing de Jockusch-Soare et celle du forcing de Sacks.

4.1. Complexité des ordres partiels

Pour parler de la calculabilité des ordres partiels d'objets abstraits, il est nécessaire de s'accorder d'abord sur leur représentation par des ensembles d'entiers. Pour le forcing de Cohen, l'ordre partiel des chaînes admet un codage bijectif naturel, alors que les notions de forcing de Jockusch-Soare et de Sacks font intervenir des objets plus complexes.

Forcing de Jockusch-Soare. Une idée naturelle est de confondre $\mathbb{P}$ avec l'ensemble des codes de classes Π_1^0 non vides. Notons que l'on a alors des répétitions, car plusieurs entiers codent pour la même classe, ce qui en pratique n'est pas un problème.

Proposition 4.1. L'ordre partiel du forcing de Jockusch-Soare est Π_2^0 . ★

PREUVE. Notons déjà que l'ensemble des codes de classes Π_1^0 non vides est Π_1^0 . En effet, si une classe Π_1^0 est vide, alors l'arbre calculable $T \subseteq 2^{<\mathbb{N}}$ qui le représente n'a aucun chemin infini, et d'après le lemme de König il existe un entier n tel qu'aucune chaîne de taille n n'appartient à T , ce qui est un événement Σ_1^0 .

À présent, étant donné deux classes Π_1^0 non vides $\mathcal{P}, \mathcal{Q}$, on a $\mathcal{P} \subseteq \mathcal{Q}$ ssi $\forall t \exists s \mathcal{P}[s] \subseteq \mathcal{Q}[t]$, ce qui est un prédicat Π_2^0 . ■

Notons qu'il est possible d'améliorer la complexité de l'ordre partiel du forcing de Jockusch-Soare, en ne travaillant que sur une partie bien choisie des codes de classes Π_1^0 non vides. Il existe en effet une fonction calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $f(e)$ code toujours pour une classe Π_1^0 non vide et telle que si e code pour une classe Π_1^0 non vide, alors e et $f(e)$ codent pour la même classe : il suffit étant donné e de stopper la co-énumération de la classe Π_1^0 correspondante si celle-ci s'apprête à rendre la classe vide.

On peut également dans la pratique améliorer la complexité de l'ordre partiel en restreignant là aussi l'ensemble des codes sur lesquels on travaille : étant donné le code e d'une classe Π_1^0 non vide $\mathcal{P}$, on peut considérer l'ensemble A des codes de toutes les classes Π_1^0 $\mathcal{Q}$ telles que $\mathcal{P} \cap \mathcal{Q}$ est non vide. L'ensemble A est Π_1^0 , et contient au moins un code correspondant à chaque classe Π_1^0 non vide incluse dans $\mathcal{P}$. Cela ne permet pas bien entendu de décider si deux codes représentent des conditions de forcing comparables, mais cela simplifie la complexité calculatoire de trouver la liste de toutes les conditions de $\mathbb{P}$ se trouvant sous une condition donnée.

Forcing de Sacks calculable. Ici, l'idée naturelle est de confondre dans ce cadre les conditions du forcing de Sacks calculable avec les codes de fonctions $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ correspondant à des f-arbres. Là encore, le codage n'est pas injectif, mais ce n'est en pratique pas un problème.

Proposition 4.2. L'ordre partiel du forcing de Sacks calculable est Π_2^0 . ★

PREUVE. L'ensemble des conditions est l'ensemble des codes e de fonctions $T_e : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ telles que $\forall \sigma \in 2^{<\mathbb{N}} \exists t \in \mathbb{N} T_e(\sigma)[t] \downarrow$ et telles que $\forall \sigma_0, \sigma_1 \in 2^{<\mathbb{N}} \sigma_0 \prec \sigma_1 \leftrightarrow T_e(\sigma_0) \prec T_e(\sigma_1)$. Ce sont bien des conditions Π_2^0 à vérifier.

Étant donné un f-arbre calculable T , l'ensemble des codes e de f-arbres calculables S_e tels que $[S_e] \subseteq [T]$ est l'ensemble des codes e de f-arbre (ce qui est une condition Π_2^0) qui vérifient que pour toute chaîne σ et pour tout entier n suffisamment grand tel que $|T(\tau)| \geq |S_e(\sigma)|$ pour toute chaîne τ de taille n , il existe une chaîne τ telle que $|\tau| \leq n$ pour laquelle $S_e(\sigma) = T(\tau)$. Il s'agit d'une condition Π_1^0 . ■

4.2. Forcer des contrats Σ_1^0/Π_1^0

La complexité de la relation de forcing pour les contrats Σ_1^0 et Π_1^0 n'est pas directement liée à la complexité de l'ordre partiel. Nous allons voir en particulier que la relation de forcing syntaxique du forcing de Jockusch-Soare, est plus complexe que celle du forcing de Sacks.

Forcing de Jockusch-Soare. La complexité de la relation de forcing ne suit pas celle de la complexité des contrats à forcer, y compris déjà pour les contrats Σ_1^0/Π_1^0 .

Proposition 4.3. Soit $(\mathbb{P}, \leq)$ le forcing de Jockusch-Soare et soit $\mathcal{R}$ un contrat. Soit $\mathcal{P}_e$ la classe Π_1^0 de code e , que l'on suppose non vide.

- (1) Si $\mathcal{R}$ est Σ_1^0 , alors le prédicat $\mathcal{P}_e \Vdash^* \mathcal{R}$ est Σ_1^0
- (2) Si $\mathcal{R}$ est Π_1^0 , alors le prédicat $\mathcal{P}_e \Vdash^* \mathcal{R}$ est Π_2^0 ★

PREUVE. Soit $T_e \subseteq 2^{<\mathbb{N}}$ un arbre calculable tel que $[T_e] = \mathcal{P}_e$.

(1) Soit $\mathcal{R}$ de la forme $\Phi(G, 0) \downarrow$ pour une fonctionnelle Φ . Alors, $\mathcal{P}_e \Vdash^* \mathcal{R}$ ssi $\Phi(X, 0) \downarrow$ pour tout $X \in \mathcal{P}_e$ ssi (par la propriété de l'usage et le lemme de König) $\exists n \forall \sigma \in T_e \cap 2^n$ on a $\Phi(\sigma, 0) \downarrow$.

(2) Soit $\mathcal{R}$ de la forme $\Phi(G, 0) \uparrow$ pour une fonctionnelle Φ . Alors, on peut obtenir $a \in \mathbb{N}$ le code de la classe Π_1^0 telle que $\mathcal{P}_a = \{X : \Phi(G, 0) \uparrow\}$. On a alors $\mathcal{P}_e \Vdash^* \mathcal{R}$ ssi $\mathcal{P}_e \subseteq \mathcal{P}_a$. Comme vue dans la preuve de la proposition 4.2, la relation d'inclusion sur les codes des classes Π_1^0 est Π_2^0 . ■

Nous verrons dans les deux sections à venir comment contourner le problème de complexité soulevé par la précédente proposition.

Exercice 4.4. (★) Soit $(\mathbb{P}, \leq)$ l'ordre partiel des arbres calculables $T \subseteq 2^{<\mathbb{N}}$ infinis.

Soit $\Vdash^\circ$ la relation définie par

- (1) $T \Vdash^\circ \exists n \Phi(G, n) \downarrow$ s'il existe $n, t \in \mathbb{N}$ tel que pour tout $\sigma \in T$ de longueur t , $\Phi(\sigma, n) \downarrow$
- (2) $T \Vdash^\circ \forall n \Phi(G, n) \uparrow$ si pour tout $\sigma \in T$ et tout $n < |\sigma|$, $\Phi(\sigma, n) \uparrow$

Montrer les faits suivants.

- (a) $(\mathbb{P}, \leq)$ est un forcing de Cantor, où $[T]$ est la classe des chemins de T .
- (b) Les ensembles suffisamment génériques pour le forcing de Jockusch-Soare et pour $(\mathbb{P}, \leq)$ coïncident.
- (c) L'ensemble $\{T \in \mathbb{P} : T \Vdash^\circ \exists n \Phi(G, n) \downarrow \text{ ou } T \Vdash^\circ \forall n \Phi(G, n) \uparrow\}$ est dense dans $(\mathbb{P}, \leq)$.
- (d) Les relations $T \Vdash^\circ \exists n \Phi(G, n) \downarrow$ et $T \Vdash^\circ \forall n \Phi(G, n) \uparrow$ sont respectivement Σ_1^0 et Π_1^0 . $\diamond$

Forcing de Sacks calculable. Le forcing de contrats Σ_1^0 et Π_1^0 est dans ce cas-ci de complexité minimale.

Proposition 4.5. Soit $(\mathbb{P}, \leq)$ le forcing de Sacks et soit $\mathcal{R}$ un contrat. Soit T_e le f-arbre calculable de code e .

- (1) Si $\mathcal{R}$ est Σ_1^0 , alors le prédicat $T_e \Vdash^* \mathcal{R}$ est Σ_1^0 .
- (2) Si $\mathcal{R}$ est Π_1^0 , alors le prédicat $T_e \Vdash^* \mathcal{R}$ est Π_1^0 . ★

PREUVE. (1) Soit $\mathcal{R}$ un contrat de la forme $\Phi(G, 0) \downarrow$ pour une fonctionnelle Φ . On a $T_e \Vdash^* \mathcal{R}$ ssi il existe n, t tels que $\Phi(\sigma, 0)[t] \downarrow$ pour toute chaîne $\sigma \in \text{Im } T_e$ de taille n .

- (2) Soit $\mathcal{R}$ de la forme $\Phi(G, 0) \uparrow$ pour une fonctionnelle Φ . On a $T_e \Vdash^* \mathcal{R}$ ssi pour tout $\sigma \in \text{Im } T_e$ et pour tout t on a $\Phi(\sigma, 0)[t] \uparrow$. ■

Continuité du saut Turing. Nous voyons à présent un théorème qui contraste avec le fait que tout suffisamment générique pour le forcing de Cohen est low généralisé. Ce n'est en général pas le cas pour les forcings à base d'arbres, et ce n'est en particulier pas le cas pour le forcing de Sacks calculable.

Théorème 4.6

Soit X un ensemble quelconque et soit G un ensemble suffisamment générique pour le forcing de Sacks calculable. Alors, $X \oplus G \not\leq_T G'$.

PREUVE. Soit $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ un f-arbre calculable et soit Φ_e une fonctionnelle Turing. On va construire un sous f-arbre S de T tel que pour chacun des chemins G de S on a $\Phi_e(X \oplus G, n) \neq G'(n)$ pour un certain n . On considère un sous f-arbre calculable S de T tel que pour tout $\sigma \in \text{Im } S$ il existe $\tau \succeq \sigma$ avec $\tau \in \text{Im } T$ et $\tau \notin \text{Im } S$.

On considère à présent le code a de la fonctionnelle partielle calculable telle que $\Phi_a(Y, a) \uparrow$ pour tout $Y \in [S]$ et telle que $\Phi_a(Y, a) \downarrow$ pour tout $Y \notin [S]$. Notons en particulier que $\Phi_a(Y, a) \downarrow$ pour tout $Y \in [T] \setminus [S]$. Supposons d'abord que $\Phi_e(X \oplus \sigma, a) \uparrow \notin \{0, 1\}$ pour tout $\sigma \in \text{Im } S$. Alors, on peut prendre S comme extension de forcing de T afin de forcer la partialité de Φ_e sur l'entrée a . Sinon, soit $\sigma \in \text{Im } S$ telle que $\Phi_e(X \oplus \sigma, a) \downarrow = i$ pour $i \in \{0, 1\}$. Si $i = 0$, alors on choisit une chaîne $\tau \succeq \sigma$ telle que $\tau \in \text{Im } T$ et $\tau \notin \text{Im } S$, et l'on prend comme extension de forcing un sous f-arbre de T dont l'image ne contient que des extensions de τ . Notons que l'on a alors $\Phi_a(Y, a) \downarrow$ pour tout chemin Y de notre extension de forcing. Si $i = 1$, alors on prend comme extension de forcing le sous f-arbre de S dont l'image ne contient que des extensions de σ . Notons que l'on a alors $\Phi_a(Y, a) \uparrow$ pour tout chemin Y de notre extension de forcing.

Dans les deux cas, on force $\Phi_e(X \oplus G, a)$ à être différent de $G'(a)$ pour tout ensemble G de notre condition de forcing. ■

Un théorème analogue pour le forcing de Jockusch-Soare ne sera pas nécessairement vrai, par exemple à cause de l'existence de classes Π_1^0 contenant un unique élément calculable. Même en se restreignant aux classes Π_1^0 ne contenant pas de points calculables, les choses ne sont pas aussi simples et dépendront de la classe Π_1^0 avec laquelle on commence le forcing.

Exercice 4.7. (★★) Soit $\mathcal{P}$ une classe Π_1^0 non vide. Alors, $\mathcal{P}$ est *fine* (définition due à Downey [49]) si pour toute sous-classe Π_1^0 non vide $\mathcal{Q} \subseteq \mathcal{P}$, il existe une suite finie de cylindres $[\sigma_0], \dots, [\sigma_n]$ telle que

$$\mathcal{Q} = \mathcal{P} \cap ([\sigma_0] \cup \dots \cup [\sigma_n]).$$

1. Montrer l'existence d'une classe Π_1^0 fine parfaite (il s'agit d'un algorithme du type méthode de priorité comme exposé dans le chapitre 13).
2. Montrer que si $\mathcal{P}$ est fine et si $X \in \mathcal{P}$, alors $X \oplus \emptyset'' \geq_T X'$. ◇

Exercice 4.8. (★★) Cet exercice anticipe sur la partie II au sein de laquelle le lemme 18-3.3 devrait être utile. On considère une variante du forcing de Jockusch-Soare avec des classes Π_1^0 ne contenant que des aléatoires au sens de Martin-Löf. Montrer que pour tout X et tout ensemble Z suffisamment générique pour ce forcing on a $Z \oplus X \not\geq_T Z'$. ◇

4.2.1. Forcer des contrats Σ_2^0/Π_2^0

Les forcing à base d'arbres sont en général adaptés, non pour forcer les contrats Σ_1^0 ou Π_1^0 (comme le montre le théorème 4.6, ils ne permettent pas d'obtenir des ensembles low généralisés), mais pour forcer les contrats Σ_2^0 ou Π_2^0 , sur lesquels ils fonctionnent particulièrement bien. Nous avons vu dans la section 10-4 que dans le cas du forcing de Cohen, si l'on définit la relation de forcing par « tout filtre maximal satisfait le contrat », alors l'ensemble des conditions forçant un contrat Π_2^0 ou sa négation n'est pas dense en général. Cela nous a conduit à définir la relation de forcing pour tout filtre maximal *suffisamment générique*.

Contrairement au forcing de Cohen, les forcings à base d'arbres, comme le forcing de Jockusch-Soare ou le forcing de Sacks permettent en général de trouver des conditions de forcing dont tous les membres satisfont des formules Σ_2^0 ou Π_2^0 . Le lecteur pourra constater que c'est bien ce mécanisme qui est à l'œuvre pour forcer un ensemble générique à être calculatoirement dominé. Nous introduisons pour voir cela la notion de *question de forcing*, notée $? \vdash$, qui sera développée et étudiée dans les sections suivantes pour des contrats arithmétiques arbitraires.

Définition 4.9. Soit $\mathcal{R}$ un contrat Σ_2^0 correspondant à $\exists n \Phi(G, n) \uparrow$ pour une fonctionnelle Φ .

(1) Soit $\mathcal{P}$ une condition du forcing de Jockusch-Soare. On définit

$$\mathcal{P} ? \vdash \exists n \Phi(G, n) \uparrow$$

s'il existe n tel que $\mathcal{P} \Vdash^* \Phi(G, n) \downarrow$.

(2) Soit T une condition du forcing de Sacks calculable. On définit

$$T ? \vdash \exists n \Phi(G, n) \uparrow$$

s'il existe n et σ tel que $T \restriction_\sigma \Vdash^* \Phi(G, n) \uparrow$, où $T \restriction_\sigma$ est le f-arbre S défini par $S(\tau) = T(\sigma\tau)$. ◇

Le premier intérêt de la relation de question de forcing que nous avons définie, est sa complexité qui est la même que celle du contrat concerné.

Proposition 4.10. Soit c une condition du forcing de Jockusch-Soare ou du forcing de Sacks. Soit $\mathcal{R}$ un contrat Σ_2^0 correspondant à $\exists n \Phi(G, n) \uparrow$ pour une fonctionnelle Φ . Le prédicat $c ? \vdash \exists n \Phi(G, n) \uparrow$ est Σ_2^0 . ★

PREUVE. Commençons par le forcing de Jockusch-Soare. D'après la proposition 4.3, étant donné n , le prédicat $\mathcal{P} \Vdash^* \Phi(G, n) \downarrow$ est Σ_1^0 , et le prédicat $\mathcal{P} \Vdash^* \Phi(G, n) \downarrow$ est donc Π_1^0 . Ainsi, le prédicat $\exists n \mathcal{P} \Vdash^* \Phi(G, n) \downarrow$ est Σ_2^0 , et le prédicat $\mathcal{P} ? \vdash \Phi(G, n) \uparrow$ est donc Σ_2^0 .

Passons au forcing de Sacks calculable. D'après la proposition 4.5, étant donné n , et un f-arbre S , le prédicat $S \Vdash^* \Phi(G, n) \uparrow$ est Π_1^0 . Ainsi, le prédicat $\exists n \exists \sigma T \restriction_\sigma \Vdash^* \Phi(G, n) \uparrow$ est Σ_2^0 , et le prédicat $T \Vdash \exists n \Phi(G, n) \uparrow$ est donc Σ_2^0 . ■

Le second intérêt de la relation de question de forcing, est qu'elle permet de *décider* si une condition c pourra être étendue en une condition d pour forcer un contrat Σ_2^0 , ou la négation de ce contrat. De plus, dans le cas de formules Σ_2^0/Π_2^0 , on pourra trouver une extension $d \leq c$ telle que le contrat sera satisfait *pour tous les éléments* de $[d]$ (comme pour les contrats Σ_1^0/Π_1^0).

Proposition 4.11. Soit c une condition du forcing de Jockusch-Soare ou du forcing de Sacks et soit $\mathcal{R}$ un contrat Σ_2^0 correspondant à $\exists n \Phi(G, n) \uparrow$ pour une fonctionnelle Φ .

1. Si $c \Vdash \exists n \Phi(G, n) \uparrow$, alors il existe $d \leq c$ tel que $\exists n \Phi(X, n) \uparrow$ est vrai pour tout $X \in [d]$.
2. Si $c \nVdash \exists n \Phi(G, n) \uparrow$, alors il existe $d \leq c$ tel que $\forall n \Phi(X, n) \downarrow$ est vrai pour tout $X \in [d]$.

Dans les deux cas, on peut trouver d uniformément en c et Φ à l'aide de $\emptyset''$.★

PREUVE. Commençons par le forcing de Jockusch-Soare. Soit $\mathcal{P} \subseteq 2^{\mathbb{N}}$ une classe Π_1^0 non vide.

1. Si $\mathcal{P} \Vdash \exists n \Phi(G, n) \uparrow$, alors il existe n tel que $\mathcal{P} \Vdash^* \Phi(G, n) \downarrow$. Cela signifie que la classe $\mathcal{Q} = \{X \in \mathcal{P} : \Phi(X, n) \uparrow\}$ est une classe Π_1^0 non vide. Il s'agit alors de notre extension de forcing.
2. Si $\mathcal{P} \nVdash \exists n \Phi(G, n) \uparrow$, alors $\mathcal{P} \Vdash^* \Phi(G, n) \downarrow$ pour tout n . Dans ce cas, pour tout $X \in \mathcal{P}$, on a déjà $\forall n \Phi(X, n) \downarrow$.

Notons que $\emptyset''$ peut décider entre les deux cas, et donc trouver l'extension de forcing appropriée. Passons au forcing de Sacks calculable. Soit à cet effet $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ un f-arbre calculable.

1. Si $T \Vdash \exists n \Phi(G, n) \uparrow$, alors $\exists \sigma \in 2^{<\mathbb{N}}$ et $\exists n \in \mathbb{N}$ tels que $T \restriction_\sigma \Vdash^* \Phi(G, n) \uparrow$. Dans ce cas, le f-arbre $T \restriction_\sigma$ est notre extension de forcing.
2. Si $T \nVdash \exists n \Phi(G, n) \uparrow$, alors pour tout $\sigma \in 2^{<\mathbb{N}}$ et pour tout $n \in \mathbb{N}$ on a $T \restriction_\sigma \nVdash^* \Phi(G, n) \uparrow$. Cela implique que pour tout $\sigma \in 2^{<\mathbb{N}}$ et pour tout $n \in \mathbb{N}$, il existe $\tau \succeq \sigma$ tel que $\Phi(T(\sigma\tau), n) \downarrow$. Alors, on procède comme dans la preuve du théorème 7-5.6 pour calculer un sous f-arbre S de T tel que $\forall n \Phi(X, n) \downarrow$ pour tout $X \in [S]$.

Notons que là encore $\emptyset''$ peut décider entre les deux cas, et donc trouver l'extension de forcing appropriée. ■

Avant de voir comment étendre la relation de question de forcing aux contrats de complexité arbitraires, voyons comment utiliser les développements obtenus jusqu'ici pour montrer que tout ensemble suffisamment générique pour le forcing de Sacks ou le forcing de Jockusch-Soare est low_2 généralisé.

Théorème 4.12

Si G est suffisamment générique pour le forcing de Sacks ou le forcing de Jockusch-Soare, il est low_2 généralisé dans un sens fort : $\emptyset'' \oplus G' \geq_T G''$.

PREUVE. Pour tout n , d'après le lemme 10-5.6 le contrat $n \in G''$ est Σ_2^0 . D'après la proposition 4.10 et la proposition 4.11, on peut énumérer à l'aide de $\emptyset''$ un ensemble D_1 de conditions c telles que $n \in X''$ pour tout $X \in [c]$, et un ensemble D_2 de conditions c telles que $n \notin X''$ pour tout $X \in [c]$, le tout tel que $D_1 \cup D_2$ soit un ensemble de conditions denses. Si G est suffisamment générique, il existe une condition de $c \in D_1 \cup D_2$ telle que $G \in [c]$. La condition c étant un arbre, on a besoin de G' pour savoir si $G \in [c]$. Une fois la condition c trouvée telle que $G \in [c]$, si $c \in D_1$ alors $n \in G''$, et si $c \in D_2$ alors $n \notin G''$. ■

4.2.2. Forcer des contrats Σ_n^0/Π_n^0

Pour les contrats plus complexes, il n'est pas possible de trouver des conditions dont tous les éléments satisfont le contrat, et il faut revenir à la définition inductive du forcing. Notre objectif est à présent de montrer un analogue du théorème 10-5.7 de préservation de la hiérarchie arithmétique. Comme la relation $\Vdash^*$ pour les forcings de Sacks et de Jockusch-Soare est trop complexe, nous allons étendre et utiliser à la place la relation de question de forcing définie précédemment pour les contrats Σ_2^0 .

Définition 4.13. Soit c une condition du forcing de Jockusch-Soare ou de Sacks et soit $\mathcal{R}$ un contrat arithmétique. On définit la relation $? \vdash$ entre c et $\mathcal{R}$ comme suit :

- (1) $c ? \vdash \mathcal{R}$ pour $\mathcal{R}$ un contrat Σ_1^0 ssi $c \Vdash^* \mathcal{R}$.
- (2) $c ? \vdash \exists x \mathcal{R}(x)$ où $\mathcal{R}(x)$ est un contrat Π_1^0 ssi $c ? \vdash \exists x \mathcal{R}(x)$ au sens de la définition 4.9.
- (3) $c ? \vdash \exists x \mathcal{R}(x)$ où $\mathcal{R}(x)$ est un contrat Π_k^0 pour $k \geq 2$ ssi il existe un $d \leq c$ et un $n \in \mathbb{N}$ tels que $d ? \vdash \mathcal{R}(n)$.
- (4) $c ? \vdash \mathcal{R}$ où $\mathcal{R}(x)$ est un contrat Π_k^0 ssi $c ? \nvdash \neg \mathcal{R}$. ◆

Nous étendons à présent la proposition 4.10 et la proposition 4.11, en commençant par montrer que la relation de question de forcing est Σ_n^0 pour les contrats Σ_n^0 .

Proposition 4.14. Soit c une condition du forcing de Jockusch-Soare ou du forcing de Sacks et soit $\mathcal{R}$ un contrat Σ_n^0 (resp. Π_n^0). La relation $c ?\vdash \mathcal{R}$ est Σ_n^0 (resp. Π_n^0). ★

PREUVE. Le cas où $\mathcal{R}$ est un contrat Σ_1^0 a été traité dans la proposition 4.3 et la proposition 4.5. Le cas où $\mathcal{R}$ est un contrat Σ_2^0 a été traité dans la proposition 4.10.

Supposons que $\mathcal{R}$ soit un contrat Σ_{n+1}^0 égal à $\exists x \mathcal{Q}(x)$, où $\mathcal{Q}(x)$ est un contrat Π_k^0 pour $k \geq 2$. Alors, $c ?\vdash \exists x \mathcal{Q}(x)$ ssi il existe une condition de forcing d et un entier n tels que $d \leq c$ et tels que $d ?\vdash \mathcal{Q}(n)$. Le prédicat $d \leq c$ est Π_2^0 (dans le cas du forcing de Jockusch-Soare et du forcing de Sacks) et le prédicat $d ?\vdash \mathcal{Q}(n)$ est par induction Π_k^0 pour $k \geq 2$. Il s'ensuit que le prédicat $c ?\vdash \exists x \mathcal{Q}(x)$ est Σ_{k+1}^0 .

Enfin, si $\mathcal{R}$ est un contrat Π_n^0 , alors $\mathcal{P} ?\vdash \mathcal{R}$ ssi $\mathcal{P} \not\vdash \neg \mathcal{R}$, ce qui est Π_n^0 par hypothèse d'induction. ■

On montre à présent l'extension de la proposition 4.11 : si $c ?\vdash \mathcal{R}$, alors on va pouvoir trouver une extension $d \leq c$ qui va forcer $\mathcal{R}$, mais attention, il s'agit ici du forcing sémantique et non du forcing syntaxique.

Proposition 4.15. Soit c une condition du forcing de Jockusch-Soare ou de Sacks et soit $\mathcal{R}$ un contrat arithmétique. Si $c ?\vdash \mathcal{R}$, alors il existe $d \leq c$ tel que $d \Vdash \mathcal{R}$. ★

PREUVE. Si $\mathcal{R}$ est un contrat Σ_1^0 ou Π_1^0 , alors le résultat est trivial par définition. Si $\mathcal{R}$ est un contrat Σ_2^0 ou Π_2^0 , alors il s'agit de la proposition 4.11.

Supposons $\mathcal{R} = \exists x \mathcal{S}(x)$ où $\mathcal{S}(x)$ est un contrat Π_k^0 pour $k \geq 2$. Supposons que $c ?\vdash \exists x \mathcal{S}(x)$. Par définition, il existe une extension $d \leq c$ et un entier $n \in \mathbb{N}$ tels que $d ?\vdash \mathcal{S}(n)$. Par hypothèse de récurrence, il existe un $e \leq d$ tel que $e \Vdash \mathcal{S}(n)$. En particulier, $e \Vdash \exists n \mathcal{S}(n)$.

Supposons $\mathcal{R} = \forall x \mathcal{S}(x)$, où $\mathcal{S}(x)$ est un contrat Σ_k^0 pour $k \geq 2$. Supposons que $c ?\vdash \forall x \mathcal{S}(x)$. Alors, $c \not\vdash \exists x \neg \mathcal{S}(x)$. Par définition, pour toute extension $d \leq c$ et tout $n \in \mathbb{N}$, $d \not\vdash \neg \mathcal{S}(n)$. Montrons que pour tout n , l'ensemble $D_n = \{d : d \Vdash \mathcal{S}(n)\}$ est dense sous c . Soient $n \in \mathbb{N}$ et $d \leq c$. Par définition, comme $d \not\vdash \neg \mathcal{S}(n)$, alors $d ?\vdash \mathcal{S}(n)$. Par hypothèse d'induction, comme $d ?\vdash \mathcal{S}(n)$, alors il existe une extension $e \leq d$ telle que $e \Vdash \mathcal{S}(n)$. L'ensemble D_n est donc dense sous c . Il s'ensuit par l'exercice 2.4 que l'on a $c \Vdash \mathcal{S}(n)$ pour tout n , donc $c \Vdash \forall x \mathcal{S}(x)$, autrement dit $c \Vdash \mathcal{R}$. ■

Cette fameuse question de forcing va nous permettre un contrôle fin de ce que calcule des itérations arbitraires du saut Turing. C'est l'objet de la section suivante.

4.3. Question de forcing

Essayons d'abstraire un peu ce qui a été fait jusqu'ici, afin d'étudier la notion de question de forcing, indépendamment du forcing sur lequel on travaille.

Définition 4.16. Soit $(\mathbb{P}, \leq)$ un forcing de Cantor. Une *question de forcing* est une relation $? \vdash$ entre les conditions et les contrats, telle que pour toute condition $c \in \mathbb{P}$ et tout contrat arithmétique $\mathcal{R}$

- (1) Si $c ? \vdash \mathcal{R}$, alors il existe une extension $d \leq c$ telle que d force $\mathcal{R}$
- (2) $c ? \vdash \mathcal{R}$ ou $c ? \vdash \neg \mathcal{R}$.

◇

Il s'ensuit de (1) et (2) que si $c ? \not\vdash \mathcal{R}$, alors il existe une extension $d \leq c$ telle que d force $\neg \mathcal{R}$. Toute relation de forcing $\Vdash$ induit une question de forcing $? \vdash$ en définissant $c ? \vdash \mathcal{R}$ pour un contrat $\Sigma_n^0 \mathcal{R}$ s'il existe une extension $d \leq c$ telle que $d \Vdash \mathcal{R}$ et $c ? \vdash \mathcal{R}$ pour un contrat Π_n^0 si $c ? \not\vdash \neg \mathcal{R}$. Si la notion de forcing est calculable, comme c'est le cas pour le forcing de Cohen, la complexité de la question de forcing pour les contrats Σ_n^0 hérite de la complexité de la relation de forcing pour les mêmes contrats.

Exercice 4.17. Soit $(\mathbb{P}, \leq)$ un forcing de Cantor, et $\Vdash$ une relation de forcing. Montrer que la relation définie par $c ? \vdash \mathcal{R}$ s'il existe une extension $d \leq c$ telle que $d \Vdash \mathcal{R}$ est une question de forcing. ◇

4.3.1. Préservation de la hiérarchie arithmétique

Un certain nombre de propriétés de faiblesse des ensembles suffisamment génériques pour les forcings de Cantor dépendent de l'existence d'une question de forcing avec de bonnes propriétés définitionnelles. Dans ce qui suit, nous fixons un forcing de Cantor $(\mathbb{P}, \leq)$ ainsi qu'une question de forcing $? \vdash$. La proposition 4.19 suivante en est le premier exemple, et généralise le théorème 10-5.7.

Définition 4.18. Une question de forcing $? \vdash$ *préserve la hiérarchie arithmétique* si pour toute condition c et tout contrat $\mathcal{R} \in \Sigma_n^0$, la relation $c ? \vdash \mathcal{R}$ est Σ_n^0 uniformément en c et $\mathcal{R}$. ◇

Proposition 4.19. Soit $? \vdash$ une question de forcing qui préserve la hiérarchie arithmétique. Alors, pour tout $n \geq 1$, pour tout ensemble A qui n'est pas Σ_n^0 , et pour tout ensemble G suffisamment générique, A n'est pas $\Sigma_n^0(G)$. ★

PREUVE. Pour tout $e \in \mathbb{N}$, soit

$$D_e = \{c \in \mathbb{P} : (\exists m \notin A \ c \Vdash m \in W_e^{G^{(n-1)}}) \text{ ou } (\exists m \in A \ c \Vdash m \notin W_e^{G^{(n-1)}})\}$$

Montrons que D_e est un ensemble dense dans $(\mathbb{P}, \leq)$. Soit $c \in \mathbb{P}$, et soit

$$U = \{m \in \mathbb{N} : c \text{ ?} \vdash m \in W_e^{G^{(n-1)}}\}$$

D'après le lemme 10-5.6, le contrat $m \in W^{G^{(n-1)}}$ est Σ_n^0 uniformément en m , donc comme la question de forcing préserve la hiérarchie arithmétique, l'ensemble U est Σ_n^0 . L'ensemble A n'étant pas Σ_n^0 , alors la différence symétrique $U \Delta A = (U \setminus A) \cup (A \setminus U)$ n'est pas vide.

Soit $m \in U \Delta A$.

Cas 1. On a $m \in U \setminus A$. Alors, par définition de la question de forcing, il existe une extension $d \leq c$ telle que $d \Vdash m \in W_e^{G^{(n-1)}}$. En particulier, $d \in D_e$.

Cas 2. On a $m \in A \setminus U$. Alors, toujours par la définition de la question de forcing, il existe une extension $d \leq c$ telle que $d \Vdash m \notin W_e^{G^{(n-1)}}$. Là encore, $d \in D_e$.

Dans tous les cas, il existe une extension de d dans D_e , et l'ensemble D_e est donc dense. Soit F un filtre suffisamment générique pour $(\mathbb{P}, \leq)$. Par densité de D_e , on peut supposer que F intersecte D_e pour tout $e \in \mathbb{N}$. Soit $G = \dot{F}$. Par définition de la relation de forcing, pour tout $e \in \mathbb{N}$, soit $m \in W_e^{G^{(n-1)}}$ pour un $m \notin A$, soit $m \notin W_e^{G^{(n-1)}}$ pour un $m \in A$. On en déduit donc que A n'est pas $\Sigma_n^0(G)$. ■

Corollaire 4.20

Soit ?- une question de forcing qui préserve la hiérarchie arithmétique.

Alors, pour tout $n \geq 0$ et tout ensemble A non $\emptyset^{(n)}$ -calculable, pour tout ensemble G suffisamment générique, A n'est pas $G^{(n)}$ -calculable.

PREUVE. Comme A n'est pas $\emptyset^{(n)}$ -calculable, il n'est pas Δ_{n+1}^0 . Soit A n'est pas Σ_{n+1}^0 , soit $\bar{A}$ n'est pas Σ_{n+1}^0 . Par la proposition 4.19, pour tout ensemble G suffisamment générique pour $(\mathbb{P}, \leq)$, A n'est pas $\Sigma_{n+1}^0(G)$ dans le premier cas, et $\bar{A}$ n'est pas $\Sigma_{n+1}^0(G)$ dans le second cas. Dans tous les cas, A n'est pas $\Delta_{n+1}^0(G)$, et donc pas $G^{(n)}$ -calculable. ■

Corollaire 4.21

Soit $n \geq 0$, et soit $A \in 2^{\mathbb{N}}$ un ensemble non $\emptyset^{(n)}$ -calculable. Si G est suffisamment générique pour le forcing de Jockusch-Soare ou pour le forcing de Sacks, alors A n'est pas $G^{(n)}$ -calculable.

PREUVE. D'après la proposition 4.14, ces deux notions de forcing préservent la hiérarchie arithmétique. ■

4.3.2. Préservation d'hyperimmunité

Rappelons qu'une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ est hyperimmune relativement à X si elle n'est dominée par aucune fonction X -calculable (voir la section 7-4). Aucune fonction n'est hyperimmune relativement à tous les degrés Turing, à commencer par le degré Turing de la fonction elle-même. Cependant, si une fonction est hyperimmune, elle est également hyperimmune relativement à tout degré calculatoirement dominé. On peut donc considérer que les degrés calculatoirement dominés « préservent » les hyperimmunités de toutes les fonctions simultanément. Nous allons maintenant étudier dans quelle mesure les ensembles suffisamment génériques pour des forcings de Cantor préservent des hyperimmunités.

Nous avons vu dans la section 10-3.1 que tout ensemble X faiblement 1-générique était de degré hyperimmune. Plus précisément, la fonction principale p_X de X qui à n associe le n -ième élément de X est hyperimmune. Les ensembles suffisamment génériques pour le forcing de Cantor ne préservent donc pas toutes les hyperimmunités simultanément. Cependant, il est possible de s'assurer qu'ils préservent l'hyperimmunité de toute fonction hyperimmune fixée à l'avance.

Définition 4.22. Une question de forcing $? \vdash$ est *compacte* si pour tout $c \in \mathbb{P}$, tout contrat arithmétique $\mathcal{R}(x)$, si $c ? \vdash \exists x \mathcal{R}(x)$, alors il existe un ensemble fini $U \subseteq \mathbb{N}$ tel que $c ? \vdash \exists x \in U \mathcal{R}(x)$. $\diamond$

La compacité d'une question de forcing est suffisante pour assurer la propriété de préservation d'une hyperimmunité.

Proposition 4.23. Soit $? \vdash$ une question de forcing compacte qui préserve la hiérarchie arithmétique. Alors, pour tout $n \geq 0$ et toute fonction hyperimmune $f : \mathbb{N} \rightarrow \mathbb{N}$ relativement à $\emptyset^{(n)}$, pour tout ensemble G suffisamment générique pour $(\mathbb{P}, \leq)$, la fonction f est hyperimmune relativement à $G^{(n)}$. $\star$

PREUVE. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ une fonction hyperimmune relativement à $\emptyset^{(n)}$. Pour tout $e \in \mathbb{N}$, soit

$$D_e = \{c \in \mathbb{P} : (\exists m \ c \Vdash \Phi_e(G^{(n)}, m) \uparrow) \text{ ou } (\exists m \ c \Vdash \Phi_e(G^{(n)}, m) \downarrow < f(m))\}.$$

Montrons alors que D_e est un ensemble dense dans $(\mathbb{P}, \leq)$. Soit $c \in \mathbb{P}$, et soit $g : \mathbb{N} \rightarrow \mathbb{N}$, la fonction partielle qui pour tout m , cherche un ensemble fini $U \subseteq \mathbb{N}$ tel que $c ? \vdash \Phi_e(G^{(n)}, m) \downarrow \in U$. Si un tel ensemble U est trouvé, $g(m) = 1 + \max U$, sinon $g(m)$ n'est pas défini. Sachant que la question de forcing préserve la hiérarchie arithmétique, g est partielle $\emptyset^{(n)}$ -calculable. Les deux cas suivants se présentent.

Cas 1. Il existe un m tel que $g(m)$ n'est pas défini. Alors, par compacité de la question de forcing, $c \not\vdash \Phi_e(G^{(n)}, m) \downarrow$, donc il existe un $d \leq c$ tel que $d \Vdash \Phi_e(G^{(n)}, m) \uparrow$. En particulier, $d \in D_e$.

Cas 2. La fonction g est totale $\emptyset^{(n)}$ -calculable. Par hyperimmunité de f relativement à $\emptyset^{(n)}$, il existe un $m \in \mathbb{N}$ tel que $g(m) \leq f(m)$. En particulier, $c \not\vdash \Phi_e(G^{(n)}, m) \downarrow \in U$ pour un ensemble fini U tel que $\max U < f(m)$. Par définition d'une question de forcing, il existe une extension $d \leq c$ telle que $d \Vdash \Phi_e(G^{(n)}, m) \downarrow \in U$, donc $\Phi_e(G^{(n)}, m) \downarrow < f(m)$. Il s'ensuit que $d \in D_e$.

Dans tous les cas, il existe une extension de d dans D_e , donc l'ensemble D_e est dense. Soit F un filtre suffisamment générique pour $(\mathbb{P}, \leq)$. Par densité de D_e , on peut supposer que F intersecte D_e pour tout $e \in \mathbb{N}$. Soit $G = \dot{F}$. Par définition de la relation de forcing, pour tout $e \in \mathbb{N}$, soit $\Phi_e(G^{(n)})$ est une fonction partielle, soit $\Phi_e(G^{(n)}, m) \downarrow < f(m)$ pour un $m \in \mathbb{N}$, donc f est hyperimmune relativement à $G^{(n)}$. ■

La question de forcing canonique du forcing de Cohen est compacte et préserve la hiérarchie arithmétique, ce qui implique que tout ensemble suffisamment générique pour le forcing de Cohen préserve l'hyperimmunité de toute fonction préalablement fixée. On peut montrer qu'il en va de même pour les forcings de Jockusch-Soare et de Sacks.

4.3.3. Préservation des degrés non PA

Nous allons terminer l'étude des propriétés des questions de forcing avec un critère pour ne pas calculer de degré PA.

Définition 4.24. Une question de forcing $\vdash$ est Π -fusionnable si pour tout $c \in \mathbb{P}$, toute paire de contrats $\Pi_n^0 \mathcal{R}_0, \mathcal{R}_1$ telle que $c \not\vdash \mathcal{R}_0$ et $c \not\vdash \mathcal{R}_1$, il existe une extension $d \leq c$ qui force simultanément $\mathcal{R}_0$ et $\mathcal{R}_1$. ◇

Proposition 4.25. Soit $\vdash$ une question de forcing Π -fusionnable qui préserve la hiérarchie arithmétique. Alors, pour tout $n \geq 0$ pour tout ensemble G suffisamment générique pour $(\mathbb{P}, \leq)$, $G^{(n)}$ n'est pas de degré PA relativement à $\emptyset^{(n)}$. ★

PREUVE. D'après le théorème 8-6.2, un degré Turing est PA ssi il calcule une fonction DNC à valeurs dans $\{0, 1\}$. Dans ce qui suit, nous supposons que $\Phi_0, \Phi_1, \dots$ est l'énumération des fonctionnelles de Turing à valeurs dans $\{0, 1\}$. Pour tout $e \in \mathbb{N}$, soit

$$D_e = \left\{ c \in \mathbb{P} : \begin{array}{ll} (\exists m \ c \Vdash \Phi_e(G^{(n)}, m) \uparrow) & \\ \text{ou} & (\exists m \ c \Vdash \Phi_e(G^{(n)}, m) \downarrow = \Phi_e(\emptyset^{(n)}, m)) \end{array} \right\}.$$

Montrons que D_e est un ensemble dense dans $(\mathbb{P}, \leq)$. Soit $c \in \mathbb{P}$, et soit la fonction partielle $g : \mathbb{N} \rightarrow \mathbb{N}$, qui pour tout m , cherche un entier $v \in \{0, 1\}$ tel que $c \Vdash \Phi_e(G^{(n)}, m) \downarrow = v$. Si un tel v est trouvé, $g(m) = v$, sinon $g(m)$ n'est pas défini. Sachant que la question de forcing préserve la hiérarchie arithmétique, g est partielle $\emptyset^{(n)}$ -calculable. Deux cas se présentent.

Cas 1. Il existe un m tel que $g(m)$ n'est pas défini. Alors, pour tout $v < 2$, on a $c \not\Vdash \Phi_e(G^{(n)}, m) \downarrow = v$, autrement dit $c \Vdash \neg(\Phi_e(G^{(n)}, m) \downarrow = v)$. Comme la relation est Π -fusionnable, il existe un $d \leq c$ tel que d force à la fois $\neg(\Phi_e(G^{(n)}, m) \downarrow = 0)$ et $\neg(\Phi_e(G^{(n)}, m) \downarrow = 1)$; or, la fonctionnelle étant à valeurs dans $\{0, 1\}$, d force donc $\Phi_e(G^{(n)}, m) \uparrow$. En particulier, $d \in D_e$.

Cas 2. La fonction g est totale $\emptyset^{(n)}$ -calculable. Sachant qu'un degré n'est pas PA relativement à lui-même, il existe $m \in \mathbb{N}$ tel que $g(m) = \Phi_m(\emptyset^{(n)}, m)$. En particulier, $c \Vdash \Phi_e(G^{(n)}, m) \downarrow = g(m)$, donc par définition d'une question de forcing, il existe une extension $d \leq c$ telle que

$$d \Vdash \Phi_e(G^{(n)}, m) \downarrow = g(m) = \Phi_m(\emptyset^{(n)}, m).$$

Il s'ensuit que $d \in D_e$.

Dans tous les cas, il existe une extension de d dans D_e , et l'ensemble D_e est donc dense. Soit F un filtre suffisamment générique pour $(\mathbb{P}, \leq)$. Par densité de D_e , on peut supposer que F intersecte D_e pour tout $e \in \mathbb{N}$. Soit alors $G = \dot{F}$.

Par définition de la relation de forcing, pour tout entier $e \in \mathbb{N}$, soit la fonction $m \mapsto \Phi_e(G^{(n)}, m)$ est partielle, soit $\Phi_e(G^{(n)}, m) \downarrow = \Phi_m(\emptyset^{(n)}, m)$ pour un entier $m \in \mathbb{N}$, donc $G^{(n)}$ n'est pas de degré PA relativement à $\emptyset^{(n)}$. ■

La question de forcing pour le forcing de Cohen est Π -fusionnable, tout comme la question de forcing pour le forcing de Sacks calculable. Il n'existe en revanche pas de question de forcing Π -fusionnable pour le forcing de Jockusch-Soare, car il existe des classes Π_1^0 non vides ne contenant que des ensembles de degré PA.

Exercice 4.26. $(\star)$ Une question de forcing $\Vdash$ est Π - ω -fusionnable si pour tout $c \in \mathbb{P}$, toute suite de contrats $\Pi_n^0 \mathcal{R}_0, \mathcal{R}_1, \dots$ telle que $c \Vdash \mathcal{R}_i$ pour tout $i \in \mathbb{N}$, il existe une extension $d \leq c$ qui force simultanément $\mathcal{R}_i$ pour tout i . Montrer que si $\Vdash$ est une question de forcing Π - ω -fusionnable qui préserve la hiérarchie arithmétique, alors pour tout ensemble G suffisamment générique et tout n , son saut itéré $G^{(n)}$ n'est pas de degré DNC relativement à $\emptyset^{(n)}$. ◇

Chapitre 12

La quête de degrés naturels

Les débuts de la calculabilité ont permis d'observer que tous les ensembles calculatoirement énumérables provenant de problèmes naturels étaient soit calculables, soit aussi puissants que le problème de l'arrêt, qui plus est via une réduction many-one (voir la section 5-4). Cela a conduit Post à se poser en 1944 la question suivante.

Question (Problème de Post [181]). Existe-t-il des degrés c. e. et non calculables qui soient strictement plus faibles que le problème de l'arrêt ?★

Le problème de Post est resté ouvert pendant près d'une décennie, avant d'être résolue par l'affirmative par Muchnik [163] et Friedberg [63] via la méthode des priorités, que nous verrons dans la section 13-3. Le problème de Post a depuis connu bien d'autres résolutions différentes, ne faisant pas nécessairement appel à la méthode de priorité. On peut citer par exemple la construction d'un ensemble K-trivial c. e. et non calculable, que nous verrons avec le théorème 16-4.5. Toutes ces constructions reposent toutefois sur une argumentation complexe permettant de construire tout exprès un ensemble « artificiel » ayant les propriétés voulues, et les seuls problèmes de décision « naturels » indécidables connus à ce jour se réduisent au problème de l'arrêt¹. À l'inverse, le problème de l'arrêt semble survenir naturellement un peu partout. La question s'est alors posée des propriétés qui lui confèrent cette naturalité.

1. Cette affirmation doit toutefois être reçue avec précaution, et sera atténuée dans la dernière section de ce chapitre.

1. Trois problèmes indécidables emblématiques

Avant d'aborder directement le problème de Post, voyons trois exemples emblématiques de problèmes de décision c. e. et indécidables, tous équivalents au problème de l'arrêt.

Problème de correspondance de Post

Nous commençons par un problème défini par Post lui-même en 1946, que l'on appelle problème de correspondance de Post, et qui ne doit pas être confondu avec le « Problème de Post », qui fait référence à la question ci-dessus.

Étant donné deux listes finies de chaînes

$$\sigma_0, \dots, \sigma_n \in 2^{<\mathbb{N}} \quad \text{et} \quad \tau_0, \dots, \tau_n \in 2^{<\mathbb{N}},$$

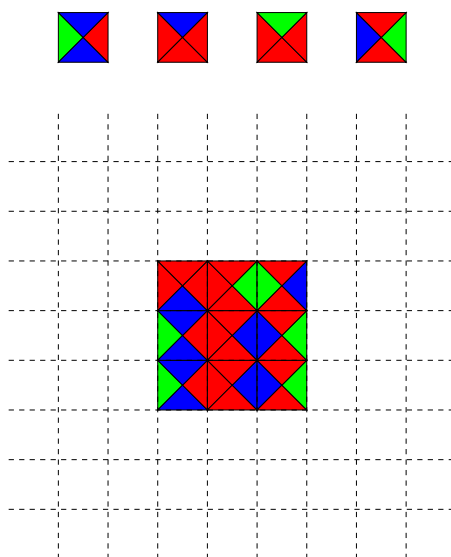
existe-t-il une suite d'indices $(i_k)_{k \leq K}$ — possiblement avec répétition — telle que les concaténations $\sigma_{i_0} \sigma_{i_1} \dots \sigma_{i_K}$ et $\tau_{i_0} \tau_{i_1} \dots \tau_{i_K}$ forment la même chaîne ?

La question peut sembler simple en apparence, et l'on pourrait même penser au premier abord qu'il est facile de créer un algorithme permettant de la résoudre. Après tout, le nombre de chaînes en question est fini. En y réfléchissant plus longuement, le problème ne devrait pas s'avérer si évident, et pour cause, même si cela peut paraître surprenant, il s'agit d'une question indécidable, et aussi difficile que celle de savoir si un programme informatique s'arrête ou pas.

Pour le montrer, Post trouve une manière astucieuse d'encoder le calcul d'une machine de Turing via des instances du problème de correspondance. Ainsi, une instance pour laquelle une correspondance existe correspondra à un calcul qui s'arrête, et une instance pour laquelle aucune correspondance n'existe correspondra à un calcul qui se déroule à l'infini.

Pavage du Plan

En 1961, Hao Wang imagine le problème suivant : étant donné un ensemble fini de *tuiles*, c'est-à-dire de carrés ayant une couleur sur chacun de leur côtés, l'objectif est de faire un pavage du plan en n'utilisant que des tuiles dont les couleurs correspondent à celles de l'une des tuiles de notre ensemble fini, en imposant que deux tuiles côte à côte partagent la même couleur sur leur côté en commun. Il y a bien entendu des ensembles de tuiles pour lesquels un tel pavage du plan est possible, et d'autres pour lesquels cela ne l'est pas.

FIGURE 1.1 – *Début de pavage du plan à l'aide des quatre tuiles ci-dessus*

Un lemme découlant de celui de König s'applique aux pavages du plan utilisant un nombre fini de tuiles : si pour tout n il existe un pavage de $n \times n$ tuiles, alors il existe un pavage infini du plan. On en déduit alors que si un ensemble fini de tuiles ne permet pas de paver le plan, il existe n tel qu'aucun pavage de taille $n \times n$ n'est possible. Si un pavage est impossible il suffit alors de chercher le plus petit n tel qu'aucune configuration de pavage de taille $n \times n$ ne fonctionne. Autrement dit, les ensembles finis de tuiles ne permettant pas de paver le plan peuvent être énumérés par un algorithme.

Wang conjecture alors qu'il en va de même pour les ensembles finis de tuiles permettant de paver le plan, ce qui rendrait décidable le problème de pavage du plan : il existerait un algorithme permettant de décider, étant donné un ensemble fini de tuiles, si ce dernier peut ou non paver le plan.

Mais en 1966, Robert Berger (un élève de Wang) montre que le problème de pavage du plan n'est pas décidable, en lui réduisant là encore le problème de l'arrêt des machines de Turing, à la manière de Post : Berger crée un système de pavage permettant de « simuler » le calcul s'effectuant sur une machine de Turing, un pavage impossible signifiant que la machine s'arrête, et un pavage possible signifiant que le calcul se poursuit indéfiniment.

Dixième problème de Hilbert

Une équation diophantienne est une équation polynomiale à une ou plusieurs inconnues dont les solutions sont à chercher parmi les nombres entiers — ou éventuellement rationnels —, les coefficients étant eux-mêmes également entiers. Par exemple, $a^2 + b^2 = c^2$ est une équation diophantienne ayant de nombreuses solutions comme $a = 3$, $b = 4$ et $c = 5$. Il y a en revanche des équations diophantiennes n'ayant aucune solution. Certaines d'entre elles ont demandé pour leur résolution — pour en trouver les solutions ou montrer qu'elles n'en ont pas — des efforts considérables de nombreux mathématiciens sur plusieurs siècles. L'exemple par excellence est certainement le fameux « dernier théorème de Fermat » qui stipule que pour tout entier $n > 2$, l'équation $a^n + b^n = c^n$ n'a pas de solutions entières (non triviales).

Fermat énonce son théorème en marge d'une traduction des « Arithmétiques de Diophante », dans laquelle il écrit : « *Au contraire, il est impossible de partager soit un cube en deux cubes, soit un bicarré en deux bicarrés, soit en général une puissance quelconque supérieure au carré en deux puissances de même degré : j'en ai découvert une démonstration véritablement merveilleuse que cette marge est trop étroite pour contenir.* »

De nombreux mathématiciens ont cherché cette démonstration merveilleuse des siècles durant et sans succès. C'est seulement après 357 ans d'efforts que le mathématicien Andrew Wiles, aidé de son ex-étudiant Richard Taylor apportera une preuve en utilisant des outils mathématiques évidemment bien plus complexes que ceux qui existaient à l'époque de Fermat.

En 1900, Hilbert place la question de la résolution des équations diophantiennes en dixième position dans sa fameuse liste de vingt-trois problèmes : « *On donne une équation diophantienne à un nombre quelconque d'inconnues et à coefficients entiers : on demande de trouver une méthode par laquelle, au moyen d'un nombre fini d'opérations, on pourra distinguer si l'équation est résoluble en nombres entiers.* »

Hilbert demande, avant que l'on en ait une définition formelle, l'existence d'un *algorithme* permettant de savoir si toute équation diophantienne admet ou non une solution.

Là encore, l'ensemble des équations diophantiennes ayant une solution est calculatoirement énumérable ; il suffit en effet de chercher parmi tous les candidats potentiels si l'un d'eux est une solution. Martin Davis, Hilary Putnam et Julia Robinson eurent l'idée de montrer que le dixième problème de Hilbert est indécidable en suivant une intuition audacieuse, et qui s'avérera juste : une équation diophantienne est la brique de base d'une formule de l'arithmétique, en l'occurrence l'égalité entre deux termes. Aussi

Gödel a-t-il montré que les ensembles c. e. sont exactement ceux qui peuvent se définir par des formules Σ_1^0 de l'arithmétique.

Ne serait-il pas possible de transformer une telle formule en une formule Σ_1^0 équivalente $\exists x_1 \dots \exists x_n F(x_1, \dots, x_n)$, mais où F ne soit plus qu'une grosse équation diophantienne ? Considérons par exemple le cas de la formule

$$t_1(a_1, \dots, a_i) = q_1(b_1, \dots, b_j) \vee t_2(x_1, \dots, x_k) = q_2(y_1, \dots, y_l),$$

où t_1, t_2 et q_1, q_2 sont des termes. Alors, cette formule est vraie dans $\mathbb{N}$ si, et seulement si, la formule

$$t_1(a_1, \dots, a_i) - q_1(b_1, \dots, b_j) = 0 \vee t_2(x_1, \dots, x_k) - q_2(y_1, \dots, y_l) = 0$$

est vraie dans $\mathbb{Z}$, ou ce qui revient au même si l'équation diophantienne

$$(t_1(a_1, \dots, a_i) - q_1(b_1, \dots, b_j)) \times (t_2(x_1, \dots, x_k) - q_2(y_1, \dots, y_l)) = 0$$

admet des solutions. On montre sans peine quelque chose de similaire pour le connecteur $\wedge$, la difficulté restante étant dans la suppression des quantifications existentielles et universelles bornées. Davis, Putnam et Robinson réussirent à supprimer les quantifications bornées au prix de l'utilisation de la fonction exponentielle dans les équations résultantes. Le travail sera alors achevé par Matiassevitch, qui parvint à encoder la fonction exponentielle en équations diophantiennes, conduisant au théorème suivant.

Théorème 1.2 (Théorème MRDP)

Soit $A \subseteq \mathbb{N}$ un ensemble calculatoirement énumérable. Alors, il existe une formule Σ_1^0 de l'arithmétique $F(x) = \exists y_1, \dots, \exists y_n G(x, y_1, \dots, y_n)$, où G n'a pas de quantificateur tel que $x \in A$ si, et seulement si, $\mathbb{N} \models F(x)$.

Le théorème MRDP est remarquable en ce sens qu'il illustre le fait que l'indécidabilité de l'arithmétique de Peano est dissimulée dans la structure même de l'addition et de la multiplication des entiers, et ce sans aucun besoin de recourir aux quantifications bornées. Il apporte bien sûr une réponse au dixième problème de Hilbert : si un algorithme permet de savoir si une équation diophantienne a des solutions entières, alors on peut créer un algorithme calculant le problème de l'arrêt.

2. Approche pour la naturalité des degrés Turing

Revenons à notre question d'origine : qu'y a-t-il de spécial à propos du problème de l'arrêt pour que tous les problèmes de décision c. e. lui soient équivalents ? Qu'est-ce qui fait au fond la naturalité d'un degré Turing ?

Steel [217] suggère une réponse : un degré Turing naturel devrait être définissable, et sa définition devrait se relativiser à n'importe quel degré. Par exemple, le problème de l'arrêt $\emptyset'$, initialement défini comme un ensemble particulier, à savoir $\{e : \Phi_e(e) \downarrow\}$, se relativise à tout ensemble X , en considérant l'ensemble $X' = \{e : \Phi_e(X, e) \downarrow\}$. Comme déjà vu dans la section 4-6, il s'agit d'une notion sur les degrés Turing, dans le sens où si $X \equiv_T Y$, alors $X' \equiv_T Y'$.

2.1. Question de Sacks

L'idée de Steel fait écho à une vieille question de Sacks [188] : existe-t-il une solution au problème de Post qui soit invariante sur les degrés Turing ? On dira que W est un *opérateur c. e.* si W correspond à une fonctionnelle Turing qui uniformément en un ensemble X énumère un ensemble, que l'on notera W^X .

Sacks demande s'il existe un opérateur c. e. W tel que $X <_T W^X <_T X'$ pour tout X et tel que $X_0 \equiv_T X_1$ implique $W^{X_0} \equiv_T W^{X_1}$ pour tout X_0, X_1 .

En y travaillant un peu, le lecteur pourra constater que la méthode de priorité utilisée dans le théorème 13-3.1 pour construire un ensemble c. e. Y tel que $0 <_T Y <_T \emptyset'$ peut se relativiser à n'importe quel oracle X pour obtenir un ensemble X -c. e. Y tel que $X <_T Y <_T X'$. En revanche, il est beaucoup plus incertain que cette relativisation soit invariante dans les degrés Turing, et de fait elle ne l'est pas. Le premier résultat dans cette direction fut obtenu par Lachlan, qui apporta une réponse négative à la question de Sacks, dans le cas particulier où l'on attend de l'invariance qu'elle soit uniforme, c'est-à-dire que l'on demande l'existence de fonctions h_1, h_2 telles que si $\Phi_{a_1}(X_1) = X_2$ et $\Phi_{a_2}(X_2) = X_1$, alors $\Phi_{h_1(a_1)}(W^{X_1}) = W^{X_2}$ et $\Phi_{h_2(a_2)}(W^{X_2}) = W^{X_1}$. Notons que Lachlan ne demande pas que les fonctions h_1, h_2 soient calculables, mais simplement qu'elles existent.

Dans le cas où l'opérateur est invariant dans les degrés Turing, la notation $W(\mathbf{a})$ pour un degré Turing $\mathbf{a}$ admet un sens : il s'agit du degré Turing obtenu en appliquant à W un ensemble quelconque dans le degré $\mathbf{a}$. Lachlan montre en fait la chose suivante : pour tout opérateur c. e. W uniformément invariant tel que $W(\mathbf{d}) \geq \mathbf{d}$ pour tout degré $\mathbf{d}$, il existe un degré $\mathbf{a}$ tel que pour tout degré $\mathbf{d} \geq \mathbf{a}$ on a $W(\mathbf{d}) = \mathbf{d}$, ou bien tel que pour tout degré $\mathbf{d} \geq \mathbf{a}$ on a $W(\mathbf{d}) = \mathbf{d}'$. Dans le premier cas, on dira que W coïncide avec l'identité *sur un cône*, et dans le second que W coïncide avec le saut Turing *sur un cône*. L'expression *sur un cône* signifiant alors sur un cône dans les degrés Turing, c'est-à-dire sur tous les degrés supérieurs à $\mathbf{a}$ pour un certain $\mathbf{a}$, le degré $\mathbf{a}$ étant *la base du cône*. Lachlan obtient donc le résultat suivant.

Théorème 2.1 (Lachlan [132])

Soit W un opérateur c. e. uniformément invariant tel que $W(\mathbf{d}) \geq \mathbf{d}$ pour tout degré $\mathbf{d}$. Alors, W est l'opérateur de saut sur un cône ou bien W est l'opérateur identité sur un cône.

2.2. Question de Sacks pour des degrés quelconques

Nous avons jusqu'à présent parlé des degrés c. e. naturels, en argumentant sur le fait que $\mathbf{0}$ et $\mathbf{0}'$ sont les seuls d'entre eux. Il existe toutefois beaucoup de problèmes de décision naturels strictement plus puissants que le problème de l'arrêt, et qui ne sont bien entendu pas c. e. On peut alors pousser la question de la naturalité pour des degrés quelconques. Voyons d'abord quelques exemples canoniques.

1. P : le problème de déterminer si un polynôme de $\mathbb{Z}[x, y_0, y_1, y_2, \dots]$ a des solutions pour tout élément x suffisamment grand.
2. T : le problème de savoir si un énoncé de l'arithmétique est vrai dans $\mathbb{N}$.
3. WF : le problème de savoir si un arbre calculable de $\mathbb{N}^{<\mathbb{N}}$ a des chemins infinis.

Le problème P est Σ_3^0 , et le triple arrêt s'y réduit de manière many-one. Le problème T permet de many-one calculer $\emptyset^{(n)}$ pour tout $n \in \mathbb{N}$ uniformément en l'entier n . Il est par ailleurs many-one calculable avec l'oracle $\emptyset^{(\omega)} = \bigoplus_n \emptyset^{(n)}$ et correspond donc au premier niveau transfini dans la hiérarchie des sauts Turing (nous verrons cela formellement dans la partie IV). Le problème WF est, quant à lui, plus complexe encore, et correspond à un saut Turing transfini plus élevé, que l'on peut noter $\emptyset^{(\omega_1^{ck})}$ et que l'on appelle communément *l'hypersaut* (nous verrons également cela formellement dans la partie IV).

Notons que les itérations du saut Turing, tout comme le saut Turing simple, se relativisent également à tout oracle de manière invariante sur les degrés Turing : par exemple, si $X \equiv_T Y$, alors $X^{(3)} \equiv_T Y^{(3)}$. Les opérateurs de saut itéré sont donc eux aussi naturels, et l'on peut trouver pour chacun d'entre eux des problèmes de décision qui leur correspondent. On peut parallèlement itérer la question de Sacks : les solutions du problème de Post ne sont pas les seules constructions de degrés exotiques en calculabilité, et cette question de l'invariance des opérateurs peut être étendue à toute construction. Considérons par exemple la construction d'un ensemble calculatoirement dominé du théorème 7-5.6 ou celle du théorème 8-4.5. Dans les deux cas, la construction nécessite $\emptyset''$. On vérifie sans peine avec l'une ou l'autre construction que l'on peut créer un opérateur $\emptyset''$ -calculable W

tel que pour tout X l'ensemble W^X vérifie $X <_T W^X$ et est tel que W^X est calculatoirement dominé relativement à X . Un tel opérateur peut-il être invariant dans les degrés Turing ? Si ce n'est pas le cas, peut-on en obtenir un calculable en $\emptyset'''$ ou en un oracle plus puissant ?

2.3. Conjecture de Martin

Inspiré par ces questions, et peut-être par le résultat de Lachlan sur les opérateurs c.e., Martin propose alors une conjecture pour le moins hardie qui dit essentiellement : l'opérateur de saut et ses itérations, sont les seuls opérateurs définissables et invariants dans les degrés Turing. Plus formellement, la conjecture comporte deux parties distinctes.

Conjecture 2.2 (Martin [1] p. 281). Soit $f : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ une fonction borélienne et invariante dans les degrés Turing. Alors, en voyant f comme une fonction sur les degrés Turing, on a ce qui suit :

- I. soit f est constante sur un cône, soit f est croissante sur un cône ;
- II. si f est croissante sur un cône, alors elle correspond à l'une des itérations (possiblement zéro ou transfinie) du saut Turing. ★

Que veut-on dire par f borélienne ? Avant d'y répondre, notons que si l'on sort complètement du cadre de la calculabilité, on peut parfaitement définir des fonctions f invariantes dans les degrés Turing telles que $\mathbf{a} < f(\mathbf{a}) < \mathbf{a}'$ pour tout $\mathbf{a}$, à l'aide simplement de l'axiome du choix de la théorie des ensembles. L'objectif de restreindre la conjecture aux fonctions f boréliennes a essentiellement pour but d'interdire l'utilisation de cet axiome, et la conjecture est généralement présentée sans la restriction pour f d'être borélienne, et avec des hypothèses supplémentaires sur les axiomes de la théorie des ensembles, signifiant essentiellement : « pour toute fonction f que l'on peut définir sans utiliser l'axiome du choix ».

La conjecture de Martin nous dit en substance que les seuls opérateurs naturels et non constants dans les degrés Turing sont l'opérateur identité, l'opérateur de saut et ses itérations. Si cette conjecture est à ce jour encore ouverte, de nombreux progrès ont été faits, en y ajoutant comme le fit Lachlan la condition d'uniformité dans l'invariance des fonctions f considérées. D'abord, Steel [217] a montré la partie II de la conjecture de Martin, pour les fonctions uniformément invariantes, généralisant donc considérablement le résultat de Lachlan. Ensuite, toujours pour les fonctions uniformément invariantes, Slaman et Steel [205] ont montré I, et ont aussi réussi à se passer, via une preuve très astucieuse, de l'uniformité pour le cas des fonctions f telles que $f(\mathbf{a}) < \mathbf{a}$ pour tout degré $\mathbf{a}$ sur un cône : ces fonctions là sont nécessairement constantes sur un cône.

On ne sait à ce jour pas grand-chose sur la conjecture dans le cas général, et c'est par ailleurs également valable pour la question de Sacks potentiellement bien plus simple, qui elle aussi reste ouverte dans le cas de la non-uniformité. Cette histoire de non-uniformité a toutefois de quoi instiller le doute : si l'on cherche au fond à construire des fonctions invariantes dans les degrés Turing, on arrive toujours à une invariance uniforme. Cette constatation a conduit Steel à faire la conjecture suivante :

Conjecture 2.3 (Steel [217]). Si une fonction borélienne est invariante dans les degrés Turing, alors elle est uniformément invariante. ★

Si la conjecture de Steel est vraie, cela montrera alors la conjecture de Martin.

3. Problèmes de masse

Face au problème de Post, nous avons vu une première réponse négative, à savoir que sous des hypothèses de naturalité, il n'existe pas de degrés c.e. non calculables intermédiaires. Il existe une autre approche, complémentaire de la première, qui consiste à dire que si les itérations du saut Turing sont les seuls degrés naturels, c'est à cause de la nature trop restrictive d'un degré Turing : il existe des puissances calculatoires qui ne correspondent pas à des degrés Turing pris individuellement, mais à des classes de degrés.

Considérons par exemple les complétions de l'arithmétique de Peano. Nous avons vu avec le théorème 9-3.10 une preuve que toute théorie complète et cohérente qui étend l'arithmétique de Peano est incalculable, mais nous ne l'avons pas fait en montrant que le problème de l'arrêt pouvait se réduire à une telle théorie, et pour cause ce n'est pas toujours le cas : il existe des degrés $\text{PA } \Delta_2^0$ ne calculant pas le problème de l'arrêt. Ce n'est pas incompatible avec la conjecture de Martin, dans le sens où si l'on cherche à définir une extension naturelle bien spécifique de l'arithmétique de Peano qui soit complète et cohérente, on en trouvera une qui calcule le problème de l'arrêt ou plus. C'est le cas par exemple de l'ensemble des formules vraies dans $\mathbb{N}$.

Les degrés PA forment une puissance calculatoire *naturelle* en tant que classe : pour tout arbre binaire infini calculable, il existe une procédure qui prend une complétion de l'arithmétique de Peano en entrée, et calcule un chemin de l'arbre en retour. On sort ici de la naturalité des degrés, pour considérer à la place la naturalité des classes de degrés. Comme nous l'avons vu dans ce livre, il existe une grande variété de classes de degrés toutes définies de manière très naturelle, et qui ne correspondent pas à des itérations de l'arrêt : les degrés high, hyperimmunes, PA , ... La notion de

classe de degrés est en général abordée via le principe des *problèmes de masse*. Ces derniers remontent à Kolmogorov [119], qui en parle informellement comme d'une formalisation des principes de logique intuitionniste de Brouwer, avant d'être définis de manière rigoureuse par Medvedev [154] et Muchnik [165].

Définition 3.1. Un *problème de masse* $\mathcal{P} \subseteq 2^{\mathbb{N}}$ est vu comme l'ensemble de ses solutions possibles, identifiées, via un codage approprié, à des éléments de $2^{\mathbb{N}}$. $\diamond$

Un problème sera par exemple résoluble s'il contient un élément calculable. Les problèmes sont étudiés en particulier via les rapports de force qu'ils entretiennent les uns par rapport aux autres. Muchnik propose l'approche suivante.

Définition 3.2 (Réduction de Muchnik). Un problème de masse $\mathcal{P}$ se réduit au sens de Muchnik à un problème de masse $\mathcal{Q}$, auquel cas on note $\mathcal{P} \leq_w \mathcal{Q}$, si toute solution de $\mathcal{Q}$ permet de calculer une solution de $\mathcal{P}$. $\diamond$

Par exemple, n'importe quelle classe Π_1^0 non vide se réduit au sens de Muchnik au problème constitué des extensions complètes et cohérentes de PA. Medvedev propose lui une définition plus restrictive en demandant l'unicité.

Définition 3.3 (Réduction de Medvedev). Un problème de masse $\mathcal{P}$ se réduit au sens de Medvedev à un problème de masse $\mathcal{Q}$, auquel cas on note $\mathcal{P} \leq_s \mathcal{Q}$, s'il existe une fonctionnelle Φ telle que $\Phi(X) \in \mathcal{P}$ pour tout $X \in \mathcal{Q}$. $\diamond$

Toute classe Π_1^0 non vide se réduit également au sens de Medvedev à la classe des ensembles PA. Les deux notions ne coïncident cependant pas dans le cas général. Jockusch [105] a par exemple montré que la classe des fonctions DNC_2 se réduisait au sens de Muchnik à celle des fonctions DNC_3 , mais pas au sens de Medvedev.

Les classes d'équivalence des relations $\equiv_w$ et $\equiv_s$ (dont les définitions découlent de $\leq_w$ et $\leq_s$) sont respectivement les *degrés Muchnik* et *degrés Medvedev*, dont la structure a été très étudiée. Il existe un plongement naturel des degrés Turing vers les degrés Muchnik et Medvedev, en associant à un degré Turing $\deg_T(X)$ le degré Muchnik ou Medvedev du problème $\{X\}$. Ce plongement respecte la structure de demi-treillis. On peut donc considérer les degrés Muchnik et Medvedev comme une extension des degrés Turing. En particulier, on peut définir $\mathbf{0}_w$ et $\mathbf{0}'_w$, les degrés Muchnik respectifs de $\{\emptyset\}$ et $\{\emptyset'\}$. La proposition suivante est donc d'une certaine manière liée à la question originale de Post.

Proposition 3.4. Soit PA le degré Muchnik des degrés PA . Alors,

$$\mathbf{0}_w <_w \text{PA} <_w \mathbf{0}'_w. \quad \star$$

Cette généralisation des degrés Turing a un coût : les degrés Muchnik et Medvedev sont bien plus nombreux. Plus précisément, les degrés Turing ont la puissance du continu ($|2^{\mathbb{N}}|$), tandis que les degrés Muchnik et Medvedev ont pour cardinalité $|2^{2^{\mathbb{N}}}|$ et possèdent des anti-chaînes de cette taille.

Chapitre 13

Méthode de priorité et degrés c. e.

Parmi les ensembles non calculables, les ensembles calculatoirement énumérables jouent un rôle particulièrement important. Ces ensembles sont « presque calculables » au sens où si un entier n appartient à un ensemble c. e. A , alors cette information sera connue en un temps fini. La classe des ensembles calculatoirement énumérables est assez naturelle, pour plusieurs raisons.

Tout d'abord, les ensembles c. e. possèdent plusieurs caractérisations très différentes, ce qui en fait une classe relativement *robuste*. Par définition, un ensemble est c. e. s'il est le domaine d'une fonction partielle calculable. Les ensembles c. e. non vides sont également précisément ceux qui sont l'image d'une fonction totale calculable, la fonction pouvant être injective si l'ensemble est infini (voir la proposition 3-7.2). De manière équivalente, un ensemble est c. e. si, et seulement si, il est réductible au problème de l'arrêt par une réduction many-one, ou encore s'il est Σ_1^0 .

Les ensembles calculatoirement énumérables forment une *classe syntaxique* contrairement aux ensembles calculables. En effet, les ensembles c. e. sont précisément les ensembles Σ_1^0 , tandis que les ensembles calculables sont les ensembles définissables à la fois par un prédicat Σ_1^0 et Π_1^0 . Cette nature syntaxique donne aux ensembles c. e. de meilleures propriétés d'uniformité. Ainsi, la classe des ensembles calculatoirement énumérables est uniformément c. e., car il suffit de lister toutes les fonctions partielles calculables, ou de manière équivalente toutes les formules Σ_1^0 . Les ensembles calculables ne peuvent en revanche pas être listés de manière calculable (d'après le théorème 7-6.2, les degrés high sont exactement ceux permettant de lister les ensembles calculables).

Enfin, et c'est peut-être un des arguments les plus importants en faveur de la naturalité des ensembles c. e., un certain nombre de problèmes non décidables en mathématiques se trouvent être calculatoirement énumérables. Parmi eux, on citera bien sûr le problème de l'arrêt, mais également l'ensemble des théorèmes de l'arithmétique (voir le théorème 9-3.7), ou même l'ensemble des solutions d'équations diophantiennes (voir, à cet effet, le théorème 12-1.2).

1. Degrés c. e.

Être calculatoirement énumérable est une propriété d'ensemble et non de degré Turing. En effet, nous avons vu que les degrés Turing sont clos par complémentaire, or par la proposition 3-7.4, si un ensemble et son complémentaire sont c. e., alors ils sont calculables. En particulier, le problème de l'arrêt $\emptyset'$ est c. e., mais est Turing-équivalent à son complémentaire, qui lui ne l'est pas. Nous avons cependant défini une notion de degré c. e. au début de la section 7-3, définition que nous répétons ci-après.

Définition 1.1. Un degré Turing est c. e. s'il contient un ensemble calculatoirement énumérable. $\diamond$

Nous avons vu que les degrés Turing n'étaient pas bornés, car pour tout degré $\mathbf{d}$, son saut Turing $\mathbf{d}'$ est strictement au-dessus. Les degrés c. e., en revanche, sont bornés par $\mathbf{0}'$.

Proposition 1.2. Les degrés c. e. ont pour degré maximum $\mathbf{0}'$. $\star$

PREUVE. Par le théorème de Post (voir la proposition 5-4.3), un ensemble est c. e. si, et seulement si, il est many-one réductible à $\emptyset'$. Les réductions many-one étant des cas particuliers de réductions Turing, tout degré c. e. est Turing-réductible à $\emptyset'$. $\blacksquare$

Les degrés c. e. formant un sous-ensemble des degrés Turing, aussi les questions concernant les degrés Turing se formulent-elles pour les degrés calculatoirement énumérables. Sont-ils linéairement ordonnés? Forment-ils un ordre bien fondé? Et, avant tout cela, existe-t-il des degrés c. e. autres que $\mathbf{0}$, le degré Turing des ensembles calculables, et $\mathbf{0}'$, le degré Turing du problème de l'arrêt?

Les degrés c. e. sont notoirement difficiles à manipuler, en raison de la contrainte de calculabilité de leur énumération. La méthode des extensions finies n'est plus adaptée, et il faudra faire appel à des techniques très élaborées pour prouver des résultats similaires à ceux obtenus dans les degrés Turing.

2. Méthode de permission

La méthode de permission permet de calculer un ensemble c. e. A à partir d'un autre ensemble c. e. B . Elle se base sur la notion de *fonction de calcul* vue à la section 4-7. Rappelons qu'étant donné une approximation c. e. $(A_s)_{s \in \mathbb{N}}$ d'un ensemble c. e. A (ou plus généralement pour toute approximation Δ_2^0), la fonction de calcul associée à cette approximation est la fonction $c_A : \mathbb{N} \rightarrow \mathbb{N}$ qui à n associe le plus petit temps s tel que $A_s \upharpoonright_n = A \upharpoonright_n$. En particulier, cette fonction est calculable en A . De plus, par la proposition 4-7.9, toute fonction dominant c_A recalcule A .

Proposition 2.1 (Méthode de permission). Soient A et B des ensembles c. e. d'approximations c. e. respectives $(A_s)_{s \in \mathbb{N}}$ et $(B_s)_{s \in \mathbb{N}}$. S'il existe une fonction B -calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que pour tout $n, s \in \mathbb{N}$

$$A_{s+1} \upharpoonright_n \neq A_s \upharpoonright_n \Rightarrow B_{s+1} \upharpoonright_{f(n)} \neq B_s \upharpoonright_{f(n)},$$

alors $A \leq_T B$. ★

PREUVE. Pour tout n , $c_A(n) \leq c_B(f(n))$, or $c_B \oplus f \leq_T B$, donc B calcule une fonction dominant c_A . Par la proposition 4-7.9, B calcule A . ■

La plupart du temps, la fonction f sera calculable et croissante, voire la fonction identité. Informellement, l'approximation de A n'est autorisée à ajouter un élément à A que si au même moment, B ajoute un élément plus petit que $f(x)$. Autrement dit, l'ensemble A attend la permission de B pour ajouter des éléments, ce qui donne son nom à cette méthode. La méthode de permission se combine souvent avec d'autres techniques comme la méthode de priorité que nous verrons dans la section suivante. La méthode de permission ne perd pas en généralité, au sens où l'on peut prouver la réciproque suivante dans le cas où l'ensemble B est infini.

Proposition 2.2. Soient A et B des ensembles c. e. tels que $A \leq_T B$ et B est infini. Alors, il existe des approximations c. e. $(A_s)_{s \in \mathbb{N}}$ et $(B_s)_{s \in \mathbb{N}}$ et une fonction B -calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que pour tout $n, s \in \mathbb{N}$

$$A_{s+1} \upharpoonright_n \neq A_s \upharpoonright_n \Rightarrow B_{s+1} \upharpoonright_{f(n)} \neq B_s \upharpoonright_{f(n)}. \quad \star$$

PREUVE. Soient $(A_s)_{s \in \mathbb{N}}$ et $(B_s)_{s \in \mathbb{N}}$ des approximations c. e. de respectivement A et B . Sachant que B est infini, en accélérant son approximation c. e., on peut supposer sans perte de généralité que $B_{s+1} \neq B_s$ pour tout s . Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ la fonction qui à n associe le plus petit entier m tel que $B_{s+1} \upharpoonright_m \neq B_s \upharpoonright_m$ pour tout $s \leq c_A(n)$. La fonction f est $c_A \oplus B$ -calculable, or $c_A \leq_T A \leq_T B$, donc $f \leq_T B$. Pour tout $n, s \in \mathbb{N}$, si $A_{s+1} \upharpoonright_n \neq A_s \upharpoonright_n$, alors $c_A(n) \geq s+1$, donc $B_{s+1} \upharpoonright_{f(n)} \neq B_s \upharpoonright_{f(n)}$. ■

3. Méthode de priorité Σ_1^0 (à blessure finie)

Post posa la question en 1944 [181] de savoir s'il existait des ensembles calculatoirement énumérables qui sont à la fois non calculables et Turing incomplets, c'est-à-dire qui ne permettent pas en tant qu'oracle de calculer le problème de l'arrêt. La question est restée ouverte pendant plus d'une décennie, avant d'être résolue par l'affirmative indépendamment par Muchnik [163] et Friedberg [63], qui ont introduit la fameuse *méthode de priorité*. Cette technique trouvera par la suite de très nombreuses applications pour l'étude des ensembles c. e. et Δ_2^0 , qui s'avèrent avoir une structure très riche, comme en témoigne le théorème de densité de Sacks : soient X, Y des ensembles c. e. tels que $X <_T Y$; il existe alors un ensemble c. e. Z tel que $X <_T Z <_T Y$.

De manière générale, la méthode de priorité sert le même but que la méthode des extensions finies (voir la section 4-8), c'est-à-dire construire des ensembles satisfaisant des propriétés de force et de faiblesse, mais cette fois-ci en contrôlant la complexité de ces ensembles dans la hiérarchie arithmétique. Cette contrainte supplémentaire est à l'origine d'une explosion de la complexité des constructions. En effet, comme dans la méthode des extensions finies, il s'agit de satisfaire une infinité de contrats simultanément, mais tandis que pour la méthode des extensions finies, la construction est omnisciente, l'argument de méthode de priorité doit s'effectuer avec une puissance calculatoire limitée. Il est donc nécessaire de poursuivre la construction de l'ensemble sans avoir une pleine connaissance de la situation. La construction se fait donc par essais-erreurs, avec des rétropédalages lorsque l'on s'aperçoit d'une erreur. Les stratégies pour satisfaire les contrats entrent en conflit, et toute la difficulté de la construction consiste à s'assurer que ces conflits et rétropédalages n'empêchent pas l'objectif global : construire un ensemble satisfaisant tous les contrats.

Nous commençons ici simplement par la plus facile des utilisations de la méthode de priorité, qui fut la première à être introduite, permettant de résoudre le problème de Post. Il s'agit d'une méthode de priorité *à blessure finie*, c'est-à-dire que chaque stratégie pour satisfaire un contrat ne devra rétropédaler qu'un nombre fini de fois avant de pouvoir réaliser son objectif.

Théorème 3.1 (Friedberg (1957), Muchnik (1956))

Il existe des ensembles c. e. incomparables pour la réduction Turing.

L'énoncé du théorème de Friedberg et Muchnick est similaire au théorème de Kleene et Post (voir la proposition 4-8.1), mais impose la contrainte supplémentaire aux ensembles d'être calculatoirement énumérables.

PREUVE. Comme pour le théorème de Kleene et Post (voir la proposition 4-8.1), nous allons construire deux ensembles, A et B , satisfaisant chacun une propriété de force et une propriété de faiblesse. Ces propriétés se déclinent chacune sous la forme de deux suites de contrats : $(\mathcal{R}_e)_{e \in \mathbb{N}}$ et $(\mathcal{S}_e)_{e \in \mathbb{N}}$:

$$\mathcal{R}_e : W_e^A \neq B \quad \mathcal{S}_e : W_e^B \neq A.$$

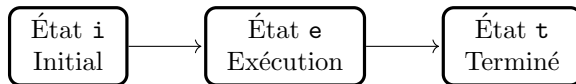
Rappelons le sens de la notation W_e^A , qui désigne l'ensemble c.e. relativement à A de code e (c'est-à-dire l'ensemble $\{n \in \mathbb{N} : \Phi_e(A, n) \downarrow\}$). Les contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$ assurent que $A \not\geq_T B$, car le complémentaire de B n'est pas c.e. en A . Symétriquement, si les contrats $(\mathcal{S}_e)_{e \in \mathbb{N}}$ sont simultanément satisfaits, alors $B \not\geq_T A$.

Remarque

Les ensembles A et B étant c.e., il est équivalent de dire qu'ils ne sont pas calculables, ou que leur complémentaire n'est pas c.e. Ainsi, les contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$ et $(\mathcal{S}_e)_{e \in \mathbb{N}}$ sont sans perte de généralité. Cette formulation des contrats simplifie les notations.

Les ensembles A et B devant être c.e., nous allons énumérer des éléments dans A et dans B au cours d'un processus calculable. Plus formellement, nous allons définir deux suites uniformément calculables d'ensembles finis $A_0 \subseteq A_1 \subseteq \dots$ et $B_0 \subseteq B_1 \subseteq \dots$ en commençant par $A_0 = B_0 = \emptyset$. Pour des raisons de présentation, nous omettrons l'indice s et parlerons de A et B comme des ensembles en cours de construction et évoluant au cours du temps. Nous utiliserons les indices lorsqu'il sera nécessaire de distinguer les ensembles à différentes étapes de temps. Pour chaque contrat $\mathcal{R}_e$ ou $\mathcal{S}_e$, nous allons décrire un processus responsable de sa satisfaction. Les différents processus s'exécuteront en parallèle. Un processus responsable de la satisfaction d'un contrat $\mathcal{R}_e$ (resp. $\mathcal{S}_e$) est appelé *stratégie pour $\mathcal{R}_e$* (resp. $\mathcal{S}_e$). Dans cette construction, une seule stratégie sera nécessaire par contrat. Nous verrons plus tard des arguments de priorité qui associeront plusieurs stratégies à chaque contrat.

Satisfaction d'un contrat $\mathcal{R}_e$. Voici une stratégie pour construire deux ensembles c.e. A et B satisfaisant un unique contrat $\mathcal{R}_e$. Pour gagner en généralité et préparer la satisfaction de plusieurs contrats, nous allons également supposer que d'autres processus ou stratégies tournent en parallèle, et ajoutent des éléments à A et B au cours du temps. À chaque instant t , la stratégie pour $\mathcal{R}_e$ se retrouve dans l'un des trois états suivants :



État i. C'est l'état initial du processus. Pour sortir de cet état, le processus passe par une phase d'initialisation qui consiste à choisir un entier $x_{\mathcal{R}_e} \notin B$ unique. Ce nombre existe, car, à tout instant, les ensembles A et B ont un nombre fini d'éléments. Une fois $x_{\mathcal{R}_e}$ choisi, notre processus pose une *restreinte* sur $x_{\mathcal{R}_e}$, c'est-à-dire qu'il interdit aux autres processus tournant en parallèle d'ajouter $x_{\mathcal{R}_e}$ dans B . Seul notre processus est décisionnaire de l'ajout de $x_{\mathcal{R}_e}$. Notons qu'une fois que $x_{\mathcal{R}_e}$ est ajouté à B , il ne pourra plus en ressortir, car l'ensemble B est c. e. Une fois la *restreinte* posée, le processus entre dans l'état **e**, dit d'exécution.

État e. Pendant cette phase, le processus exécute $\Phi_e^A(x_{\mathcal{R}_e})$ jusqu'à ce que l'exécution se termine. Si celle-ci ne se termine pas, le processus reste dans cet état. Notons qu'au cours de cette phase, l'ensemble A peut évoluer, car d'autres processus peuvent ajouter des éléments à A en parallèle. Le processus doit prendre en compte cette évolution, et calcule donc $\Phi_e^{A_t}(x_{\mathcal{R}_e})[t]$ à l'instant t . Si $\Phi_e^{A_t}(x_{\mathcal{R}_e}) \downarrow$ à un instant t , alors le processus pose une *restreinte* sur l'usage de ce calcul, c'est-à-dire sur les bits de l'oracle A_t ayant servi à cette exécution, empêchant ainsi les autres processus d'y effectuer une modification. On s'assure donc que $\Phi_e^A(x_{\mathcal{R}_e}) \downarrow$, autrement dit que $x_{\mathcal{R}_e} \in W_e^A$. Le processus ajoute alors $x_{\mathcal{R}_e}$ à B , de telle sorte que $W_e^A \neq \overline{B}$, et se retrouve dans l'état **t**.

État t. Dans cet état, le processus a terminé son exécution. Il ne sort pas de cet état.

Issues. Étudions les différentes issues de la stratégie pour $\mathcal{R}_e$. Le processus passe toujours de l'état **i** à l'état **e**, mais peut ne jamais arriver dans l'état **t**. Nous avons donc deux issues possibles. Issue **p** : il reste bloqué dans l'état d'exécution (état **e**). Dans ce cas, $\Phi_e^A(x_{\mathcal{R}_e}) \uparrow$ et $x_{\mathcal{R}_e} \notin B$, donc $W_e^A \neq \overline{B}$. On dit que le contrat $\mathcal{R}_e$ est *satisfait passivement*. Issue **a** : il entre dans l'état **t** de terminaison. Alors, par ses actions de *restreintes* et l'ajout de $x_{\mathcal{R}_e}$ dans B , il s'assure que le contrat $\mathcal{R}_e$ est satisfait par la seconde clause de la disjonction. On dit alors que le contrat $\mathcal{R}_e$ est *satisfait activement*. Dans les deux cas, le contrat $\mathcal{R}_e$ est satisfait.

État versus issue d'une stratégie

Il convient de bien distinguer l'état d'une stratégie, et son issue. L'état d'une stratégie dépend de chaque étape, et est une information connue à cette étape. L'issue de la stratégie est un comportement limite qui n'est pas connu en temps fini. Nous n'avons vu que des issues de la forme « La stratégie restera dans tel état au bout d'un moment », mais nous verrons d'autres issues dans la méthode de priorité à blessure infinie, comme « La stratégie passera successivement par tous ses états. »

Conflits. Des complications se posent lorsque l'on veut satisfaire plusieurs contrats simultanément. En effet, la stratégie d'un contrat pose des restrictions sur des bouts finis de A et B au cours du temps, ce qui peut entrer en conflit avec les besoins des autres stratégies. Nous allons donc analyser dans quelle mesure les stratégies peuvent entrer en conflit, que cela soit entre stratégies pour contrats de même type (par exemple $\mathcal{R}_a$ et $\mathcal{R}_b$), ou entre stratégies pour contrats de type différent (par exemple $\mathcal{R}_a$ et $\mathcal{S}_b$).

Satisfaction de tous les contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$. De manière générale, dans les arguments de priorité, les stratégies pour les contrats de même nature sont relativement faciles à satisfaire simultanément. Les seuls conflits possibles entre deux contrats $\mathcal{R}_e$ et $\mathcal{R}_d$ surviendraient si les deux stratégies avaient choisi le même entier x (autrement dit, $x_{\mathcal{R}_e} = x_{\mathcal{R}_d}$) lors de leur passage de l'état **i** à l'état **e**, et l'un des deux, disons $\mathcal{R}_e$, voit son calcul $\Phi_e^A(x)$ se terminer et cherche à ajouter x dans B pour passer à l'état **t** de terminaison, tandis que $\mathcal{R}_d$ est encore dans l'état **e** d'exécution. Pour éviter ces conflits, il suffit d'associer dans l'état **i** un entier x différent pour chaque contrat $\mathcal{R}_e$. Cela est possible, car, B étant fini au cours de la construction, il existe une infinité d'entiers hors de B .

Satisfaction d'un contrat $\mathcal{R}_e$ et $\mathcal{S}_d$. Supposons maintenant que l'on veuille satisfaire deux contrats de nature opposée. Par défaut, le contrat $\mathcal{S}_d$ jouant un rôle symétrique, la stratégie pour le satisfaire s'obtient à partir de celle pour $\mathcal{R}_e$ en substituant A à B et inversement.

- ▷ État **i**. Choisir un entier $x_{\mathcal{S}_d} \notin A$, et poser une restriction sur $x_{\mathcal{S}_d}$.
- ▷ État **e**. Exécuter $\Phi_d^B(x_{\mathcal{S}_d})$. Si l'exécution s'arrête à l'instant t , restreindre l'usage de $\Phi_d^{B_t}(x_{\mathcal{S}_d})$ et ajouter $x_{\mathcal{S}_d}$ à A , puis passer dans l'état **t**.
- ▷ État **t**. Le processus est terminé.

Un nouveau conflit peut se poser entre la stratégie pour $\mathcal{R}_e$ et celle pour $\mathcal{S}_d$. Dans l'état **e**, la stratégie pour $\mathcal{R}_e$ peut voir l'exécution de $\Phi_e^A(x)$ s'arrêter avec un usage de s bits d'oracle, pour $s > x_{\mathcal{S}_d}$. Il impose alors une restriction sur $A_t \upharpoonright_s$, interdisant aux autres processus de modifier ces valeurs. Si, plus tard, la stratégie pour $\mathcal{S}_d$ voit son exécution de $\Phi_d^B(x_{\mathcal{S}_d})$ se terminer, elle ne pourra pas ajouter $x_{\mathcal{S}_d}$ à A à cause de la restriction précédente. La stratégie pour $\mathcal{S}_d$ est *blessée* et va devoir revenir dans son état **i**, choisir un nouvel entier $x_{\mathcal{S}_d}$ suffisamment grand pour ne pas être restreint, et recommencer la procédure. La stratégie pour $\mathcal{R}_e$ étant à l'état **t** de terminaison, elle n'agira plus, et ne posera donc plus de nouvelle restriction risquant de blesser la stratégie pour $\mathcal{S}_d$.

De manière générale, on peut satisfaire un nombre fini de contrats $\mathcal{R}_e$ et $\mathcal{S}_d$ simultanément avec la même méthode. Dès lors qu'une stratégie pose

une restreinte sur l'usage de son calcul lorsque celle-ci passe à l'état $\mathbf{t}$ et se termine, elle blesse toutes les stratégies qui n'ont pas encore atteint l'état $\mathbf{t}$, et les fait revenir dans l'état $\mathbf{i}$. Les stratégies ainsi blessées vont alors choisir de nouveaux éléments en dehors de toute restreinte. Notons que lorsqu'une stratégie entre dans l'état $\mathbf{t}$, elle n'en sort plus. Les blessures étant causées uniquement par l'entrée d'une stratégie arrivant dans l'état $\mathbf{t}$, chaque stratégie n'est blessée qu'un nombre fini de fois, et finira par entrer dans l'état $\mathbf{t}$, ou restera bloquée dans l'état $\mathbf{e}$.

Satisfaction de tous les contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$ et $(\mathcal{S}_e)_{e \in \mathbb{N}}$. Un nouveau problème se pose lorsque l'on veut satisfaire une infinité de contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$ et $(\mathcal{S}_e)_{e \in \mathbb{N}}$. Supposons que la stratégie pour $\mathcal{R}_e$ choisisse dans son état $\mathbf{i}$ un entier $x_{\mathcal{R}_e}$ et se lance dans l'exécution de $\Phi_e^A(x_{\mathcal{R}_e})$, mais n'a pas le temps d'atteindre la fin de l'exécution, car la stratégie pour un contrat $\mathcal{S}_{d_0}$ voit sa propre exécution se terminer avant elle et pose une restreinte sur $x_{\mathcal{R}_e}$ pour préserver l'usage de son calcul. La stratégie pour $\mathcal{R}_e$ est alors blessée, et se retrouve à nouveau dans l'état $\mathbf{i}$ et choisit un nouvel entier $x_{\mathcal{R}_e}$ et recommence l'exécution de $\Phi_e^A(x_{\mathcal{R}_e})$ pour son nouvel $x_{\mathcal{R}_e}$. Avant d'atteindre la fin de son calcul, par manque de chance, une autre stratégie $\mathcal{S}_{d_1}$ atteint l'état $\mathbf{t}$, et blesse encore la stratégie pour $\mathcal{R}_e$, et ainsi de suite à l'infini. La stratégie pour $\mathcal{R}_e$ changera alors infiniment souvent d'entier $x_{\mathcal{R}_e}$, et le contrat $\mathcal{R}_e$ ne sera jamais satisfait.

Pour résoudre ce problème, nous allons ordonner les stratégies. Soient R_e la stratégie pour $\mathcal{R}_e$ et S_e la stratégie pour $\mathcal{S}_e$. On ordonne les stratégies comme suit :

$$R_0, S_0, R_1, S_1, R_2, S_2, \dots$$

en considérant qu'une stratégie est moins prioritaire que les stratégies à sa gauche, mais plus prioritaire que celles à sa droite. Par exemple, la stratégie pour $\mathcal{S}_1$ est moins prioritaire que les stratégies pour $\mathcal{R}_0$ et $\mathcal{S}_0$, mais plus prioritaire que les stratégies pour $\mathcal{R}_2, \mathcal{S}_2$, et ainsi de suite. De la sorte, chaque stratégie est « en dessous » d'un nombre fini de stratégies, et « au-dessus » du reste.

Remarque

Dans la preuve du théorème de Friedberg et Muchnik, chaque contrat possède exactement une stratégie, ce qui rend le distinguo entre stratégie R_e et contrat $\mathcal{R}_e$ inutile. L'ordre donné sur les stratégies induit ainsi un ordre sur les contrats. Cependant, dans les constructions suivantes, lorsque les contrats se verront attribuer plusieurs stratégies, il sera essentiel de donner un ordre total de priorité sur les stratégies et non sur les contrats.

La règle d'or que l'on établit est alors la suivante.

Les restreintes posées par une stratégie ne s'appliquent qu'aux stratégies de plus faible priorité. Ainsi, une stratégie ne peut être blessée que par les stratégies de plus forte priorité, et peut blesser toutes les stratégies de plus faible priorité.

En supposant qu'une stratégie ne pose qu'un nombre fini de restreintes avant d'atteindre son état terminal, et cesse d'agir au bout d'un moment si elle n'est pas blessée, une simple induction montre que chaque stratégie n'est blessée qu'un nombre fini de fois. Contrairement à la satisfaction d'un nombre fini de contrats simultanément, il se peut qu'un processus arrive à l'état $\mathbf{t}$, mais soit ensuite blessé par une stratégie de plus forte priorité qui aura ignoré la restreinte posée. Heureusement, au bout d'un moment, la stratégie de plus forte priorité cessera de blesser la plus faible, qui finira par se stabiliser.

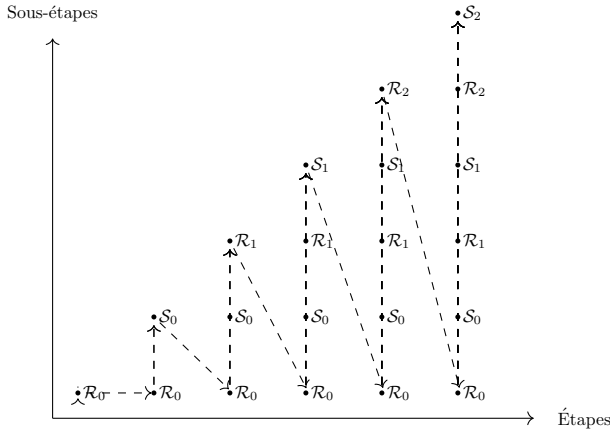
Construction. Formellement, la construction se fait par étapes $t=0,1,\dots$, et chaque étape est divisée en sous-étapes $s < t$. Initialement, toutes les stratégies sont dans l'état $\mathbf{i}$. À l'étape $t \geq 0$ et la sous-étape $s < t$, on considère le contrat $\mathcal{R}_e$ si $s = 2e$ et $\mathcal{S}_e$ si $s = 2e + 1$.

Au début de la sous-étape $s = 2e$, la stratégie pour $\mathcal{R}_e$ a les trois états possibles suivants.

- (i) La stratégie choisit un entier $x_{\mathcal{R}_e} \notin B_t$ qui n'a pas été restreint par une stratégie de plus grande priorité, et pose une restreinte dessus. Elle se retrouve alors dans l'état $\mathbf{e}$.
- (e) La stratégie exécute $\Phi_e^{A_t}(x_{\mathcal{R}_e})[t]$. Si $\Phi_e^{A_t}(x_{\mathcal{R}_e})[t] \downarrow$, alors elle pose une restreinte sur tous les bits ayant servi au calcul de $\Phi_s^{A_t}(x_{\mathcal{R}_e})[t]$, et blesse toutes les stratégies de plus faible priorité en les faisant retourner à l'état $\mathbf{i}$. Elle se retrouve alors dans l'état $\mathbf{t}$. Si $\Phi_e^{A_t}(x_{\mathcal{R}_e})[t] \uparrow$, la stratégie reste dans l'état $\mathbf{e}$.
- (t) La stratégie n'agit pas et reste dans cet état.

À la sous-étape $s = 2e + 1$, on applique la même procédure pour $\mathcal{S}_e$ *mutatis mutandis*, puis on passe à la sous-étape suivante, jusqu'à atteindre t , auquel cas on passe à l'étape $t + 1$, et ainsi de suite.

Cela conclut la construction.



Vérification. Tout d'abord, remarquons que la construction décrite plus haut est bien un processus calculable qui énumère deux ensembles A et B . En particulier, une fois un élément énuméré dans A ou B , on ne change pas d'avis et il y reste. Remarquons aussi que si une stratégie pour un contrat n'est plus blessée après une étape r , alors le contrat associé sera satisfait passivement ou activement à la fin de la construction. La difficulté réside dans le fait de montrer que pour tout contrat il existe bien un tel r .

Montrons par induction sur $e \in \mathbb{N}$ que les stratégies pour les contrats $\mathcal{R}_e$ et $\mathcal{S}_e$ blessent finiment souvent des stratégies de priorité inférieure. Supposons par hypothèse d'induction qu'il existe une étape t après laquelle aucune des stratégies pour les contrats $\mathcal{R}_d$ et $\mathcal{S}_d$, avec $d < e$, ne blessent des stratégies de priorité inférieure. En particulier, à partir de l'étape t , la stratégie pour $\mathcal{R}_e$ ne sera plus blessée, et soit reste désormais dans l'état d'exécution (état $\mathbf{e}$), soit passe au bout d'un moment dans l'état de terminaison (état $\mathbf{t}$) en ne blessant plus qu'une dernière fois les stratégies de priorité inférieure. Le même raisonnement s'applique à la stratégie pour le contrat $\mathcal{S}_e$. Ainsi, chaque stratégie blesse finiment souvent les stratégies de priorité inférieure, et chaque contrat finira par être satisfait, passivement ou activement. Cela conclut la preuve du théorème 3.1. ■

Remarque

La preuve précédente était très détaillée afin de donner les intuitions de la méthode de priorité. Nous irons désormais plus directement vers la construction finale dans les autres preuves. Il est cependant souvent utile, lorsque l'on cherche à prouver un nouveau résultat à l'aide de la méthode de priorité, de procéder étapes par étapes, en cherchant à satisfaire un contrat, puis plusieurs simultanément, pour finir par les satisfaire tous, afin de mieux comprendre leurs interactions.

Corollaire 3.2

Il existe un ensemble c. e. A incalculable tel que $A \not\leq_T \emptyset'$.

PREUVE. Soient A et B deux ensembles c. e. tels qu'aucun des deux ne calcule l'autre. Alors, ils sont en particulier tous les deux incalculables. Par ailleurs, si A pouvait calculer l'arrêt, il calculerait aussi B du fait que tout ensemble c. e. est many-one réductible à l'arrêt. ■

Avant de nous attaquer à des constructions plus élaborées, se basant sur la technique des méthodes de priorités à blessure finie de Friedberg et Muchnik, nous en voyons ici une autre application simple.

Théorème 3.3

Il existe un ensemble c. e. incalculable de degré low.

PREUVE. Nous allons construire un ensemble c. e. A par la méthode de priorité à blessure finie, ainsi qu'une fonction $\Gamma : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ totale calculable stable satisfaisant les contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$ et $(\mathcal{S}_e)_{e \in \mathbb{N}}$:

$$\mathcal{R}_e : W_e \neq \bar{A} \quad \mathcal{S}_e : A'(e) = \lim_t \Gamma(e, t).$$

Satisfaire tous les contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$ assure que l'ensemble A n'est pas calculable, tandis que les contrats $(\mathcal{S}_e)_{e \in \mathbb{N}}$ font en sorte que A' (le problème de l'arrêt relativement à A) admet une approximation Δ_2^0 , ce qui, par le lemme de limite de Shoenfield (voir le lemme 4-7.2), assure que $A' \leq_T \emptyset'$, et donc que A est de degré low.

Satisfaction d'un contrat $\mathcal{R}_e$. Voici une stratégie pour satisfaire un contrat $\mathcal{R}_e$, indépendamment des autres contrats. Notre stratégie a trois états : initial (**i**), en exécution (**e**) et terminée (**t**). La stratégie suit les étapes suivantes en fonction de son état.

- ▷ État **i**. Choisir un entier $x_{\mathcal{R}_e} \notin A$, et poser une restriction sur $x_{\mathcal{R}_e}$.
- ▷ État **e**. Exécuter $\Phi_e(x_{\mathcal{R}_e})$. Si l'exécution s'arrête à l'instant t , ajouter $x_{\mathcal{R}_e}$ à A , puis passer dans l'état **t**.
- ▷ État **t**. Le processus a terminé son exécution et reste dans cet état.

En supposant que la stratégie n'est blessée qu'un nombre fini de fois, elle pourra avoir deux issues possibles. Issue **p** : elle finira par rester dans l'état **e**, auquel cas, $\Phi_e(x_{\mathcal{R}_e}) \uparrow$ et $x_{\mathcal{R}_e} \notin A$, donc $W_e \neq \bar{A}$. Dans ce cas, le contrat $\mathcal{R}_e$ est dit satisfait passivement. Issue **a** : la stratégie atteindra l'état **t**, et s'arrêtera. Dans ce cas, $\Phi_e(x_{\mathcal{R}_e}) \downarrow$, $x_{\mathcal{R}_e} \in A$, et $\mathcal{R}_e$ est dit satisfait activement.

Satisfaction d'un contrat $\mathcal{S}_e$. À l'étape t , nous allons définir la valeur de $\Gamma(x, t)$ pour tout $x < t$. La stratégie comporte deux états : en exécution (**e**) et terminée (**t**). Voici la démarche à suivre en fonction de chaque état de la stratégie.

- ▷ État **e**. Exécuter $\Phi_e^A(e)$. Tant que $\Phi_e^A(e)[t] \uparrow$, maintenir $\Gamma(e, t) = 0$. Si l'exécution s'arrête à l'instant t , restreindre l'usage de $\Phi_e^A(e)$ puis passer dans l'état **t**.
- ▷ État **t**. Définir $\Gamma(e, t) = 1$ désormais pour toute nouvelle étape t .

Les deux issues possibles de la stratégie sont les suivantes. Issue **p** : elle finira par rester dans l'état **e** d'exécution, auquel cas $\Phi_e^A(e) \uparrow$ et $\lim_t \Gamma(e, t) = 0$. Ainsi, $A'(e) = 0 = \lim_t \Gamma(e, t)$. Dans ce cas, le contrat $\mathcal{S}_e$ est dit satisfait passivement. Issue **a** : la stratégie atteindra l'état **t**, et s'arrêtera. Dans ce cas, $\Phi_e^A(e) \downarrow$ et $\lim_t \Gamma(e, t) = 1$. Donc, $A'(e) = 1 = \lim_t \Gamma(e, t)$, et $\mathcal{S}_e$ est dit satisfait activement.

Construction. La construction est globalement la même que celle du théorème 3.1. Les stratégies sont ordonnées $R_0, S_0, R_1, S_1, \dots$, par priorité décroissante. La construction se divise en étapes $t = 0, 1, \dots$ et chaque étape est elle-même divisée en sous-étapes $s < t$. Initialement, toutes les stratégies sont dans l'état **i**. À l'étape $t \geq 0$ et la sous-étape $s < t$, on considère le contrat $\mathcal{R}_e$ si $s = 2e$ et $\mathcal{S}_e$ si $s = 2e + 1$. À la sous-étape $s = 2e$, on exécute la stratégie pour $\mathcal{R}_e$ comme décrit ci-dessus en fonction de son état. Lorsqu'elle atteint l'état **t**, toutes les stratégies de priorité inférieure sont blessées et reviennent soit dans l'état **i** dans le cas des stratégies pour des contrats de type $\mathcal{R}$, soit dans l'état **e** dans le cas des stratégies pour des contrats de type $\mathcal{S}$. De la même manière, à la sous-étape $s = 2e + 1$, on exécute la stratégie $\mathcal{S}_e$ comme décrit ci-dessus, en blessant toutes les stratégies de priorité inférieure si l'on arrive à l'état de terminaison **t**.

Vérification. L'ensemble A produit est bien c. e., car le processus est calculable, et ne retire aucun élément de A une fois ajouté. On prouve aisément par induction sur $e \in \mathbb{N}$ que les stratégies pour les contrats $\mathcal{R}_e$ et $\mathcal{S}_e$ blessent finiment souvent des stratégies de priorité inférieure. Ainsi, chaque stratégie n'est blessée que finiment souvent, et finira par avoir un comportement limite. Il s'ensuit également que $\lim_t \Gamma(e, t)$ existe et est, par construction, égale à $A'(e)$. Cela conclut la preuve du théorème 3.3. ■

4. Méthode de priorité Σ_2^0

Dans les constructions précédentes à l'aide de la méthode de priorité, les blessures finies sont assurées de manière structurelle, c'est-à-dire que dans la structure même de la construction, les stratégies ne posent qu'un nombre

fini de contraintes lorsqu'elles ne sont pas blessées, et sont assurées par construction de n'être blessées qu'un nombre fini de fois. Le nombre de blessures peut même être borné calculatoirement : dans la construction de Friedberg et Muchnik, le e -ième processus est blessé au plus $2^e - 1$ fois.

Nous allons maintenant voir une élaboration de la méthode précédente, qui pourrait structurellement résulter en un nombre infini de blessures sur des stratégies, mais des hypothèses sur la construction vont assurer que cette situation n'arrive jamais. Techniquement, il s'agit donc d'une méthode de priorité à blessure finie, car chaque stratégie ne sera blessée qu'un nombre fini de fois. Cependant, cette élaboration peut être vue comme une version dégénérée de la méthode de priorité à blessure infinie, présentée dans la section suivante.

Théorème 4.1 (Sacks)

Pour tout ensemble c. e. incalculable B , il existe ensemble c. e. incalculable A qui ne calcule pas B .

PREUVE. Nous allons construire un ensemble c. e. A satisfaisant les contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$ et $(\mathcal{S}_e)_{e \in \mathbb{N}}$ suivants :

$$\mathcal{R}_e : W_e \neq \overline{A} \quad \mathcal{S}_e : \Phi_e^A = B \Rightarrow B \text{ est calculable.}$$

Satisfaire tous les contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$ assure que l'ensemble A n'est pas calculable, tandis que les contrats $(\mathcal{S}_e)_{e \in \mathbb{N}}$ font en sorte que $B \not\leq_T A$. Le contrat $\mathcal{S}_e$ aurait également pu être noté $\mathcal{W}_e^A \neq \overline{B}$, mais sa formulation actuelle représente mieux la forme de l'argument utilisé pour le satisfaire.

Satisfaction d'un contrat $\mathcal{R}_e$. La satisfaction d'un contrat $\mathcal{R}_e$ est exactement la même que pour le théorème 3.3. Nous rappelons les actions de la stratégie en fonction de ses trois états.

- ▷ État i. Choisir un entier $x_{\mathcal{R}_e} \notin A$, et poser une contrainte sur $x_{\mathcal{R}_e}$.
- ▷ État e. Exécuter $\Phi_e(x_{\mathcal{R}_e})$. Si l'exécution s'arrête à l'instant t , ajouter $x_{\mathcal{R}_e}$ à A , puis passer dans l'état t.
- ▷ État t. Le processus est terminé.

Les deux issues de la stratégie sont toujours p et a.

Satisfaction d'un contrat $\mathcal{S}_e$. La stratégie pour satisfaire $\mathcal{S}_e$ est plus complexe et moins intuitive. Elle consiste à essayer de faire coïncider des segments initiaux de Φ^A et de B de plus en plus longs, de manière calculable, en posant chaque fois des contraintes de plus en plus grandes sur A pour préserver l'usage de Φ^A . À première vue, cette stratégie va donc causer une infinité de blessures aux stratégies de priorité inférieure.

Heureusement, l'ensemble B n'étant pas calculable, la procédure cessera de trouver de nouveaux segments initiaux de coïncidence, et satisfera ainsi le contrat $\mathcal{S}_e$. Plus précisément, la stratégie possède un état initial i , et une infinité d'états $(w_n)_{n \in \mathbb{N}}$. Pendant l'exécution du processus, nous allons définir une fonction calculable $\Delta : \mathbb{N} \rightarrow \{0, 1\}$ censée coïncider avec la fonction caractéristique de B . Soit $(B_t)_{t \in \mathbb{N}}$ une approximation c. e. de B .

- ▷ État i . Définir Δ comme la fonction de domaine vide, et passer à l'état w_0 .
- ▷ État w_n . Attendre une étape $t \geq n$, où $\Phi_e^{A_t}[t] \upharpoonright_{n+1} = B_t \upharpoonright_{n+1}$. Si cela arrive, poser alors une restriction sur l'usage de $\Phi_e^{A_t}[t] \upharpoonright_{n+1}$, définir ensuite $\Delta(n) = B_t(n)$, et passer enfin à l'état w_{n+1} .

Ici, $\Phi_e^{A_t}[t] \upharpoonright_{n+1} = B_t \upharpoonright_{n+1}$ signifie que pour tout $x \leq n$, $\Phi_e^{A_t}(x)[t] \downarrow \in \{0, 1\}$, et $\Phi_e^{A_t}(x)[t] = 1$ ssi $x \in B_t$ pour $x \leq n$. La stratégie ne retourne à l'état i que lorsqu'elle est blessée. À ce moment-là, la fonction Δ est également réinitialisée. Cependant, en supposant que la stratégie est blessée un nombre fini de fois, la fonction ne sera réinitialisée que finiment souvent, et sera donc définie de manière calculable.

La stratégie possède une infinité d'issues : pour tout n , l'issue p_n consiste à rester bloqué dans l'état w_n . Si la stratégie ne sort jamais de cet état, alors $\Phi_e^{A_t}[t] \upharpoonright_{n+1} \neq B_t \upharpoonright_{n+1}$ pour tout t , donc $\Phi_e^A \neq B$, et le contrat $\mathcal{S}_e$ est satisfait, car la prémisse de l'implication est fausse. La stratégie possède une dernière issue possible, d'ordre infinitaire, qui consiste à passer par tous les états w_n . Nommons cette issue ∞ . En utilisant l'hypothèse selon laquelle B n'est pas calculable, nous allons maintenant montrer que cette issue ne peut pas arriver.

Lemme 4.2. Si l'issue ∞ arrive, alors B est calculable. ★

PREUVE. Soit e le plus petit entier tel que la stratégie pour $\mathcal{S}_e$ passe par tous les états $(w_n)_{n \in \mathbb{N}}$. Par induction, toutes les stratégies de priorité supérieure sont finiment blessées, et atteindront un état limite où elles ne blesseront plus la stratégie pour $\mathcal{S}_e$. À partir de ce moment, la stratégie pour $\mathcal{S}_e$ passera par tous les états $(w_n)_{n \in \mathbb{N}}$ successivement. La fonction Δ définie par le processus est alors totale calculable. Montrons que Δ est la fonction caractéristique de B . Supposons par l'absurde que $\Delta(n) \neq B(n)$, pour un $n \in \mathbb{N}$. Par définition de Δ , $\Delta(n) = B_t(n) = \Phi_e^{A_t}(n)$ pour un $t \in \mathbb{N}$. Comme B est c. e., cette différence vient d'un élément qui apparaît dans B , car aucun élément ne peut en sortir : $\Delta(n) = B_t(n) = \Phi_e^{A_t}(n) = 0$ et $n \in B$. Soit m suffisamment grand tel que $n \in B_m$. Par conséquent, à l'état w_m , $\Phi_e^{A_t}(n)[t] = 0 \neq B_t(n)$ pour tout $t \geq m$, et $\Phi_e^{A_t}[t] \upharpoonright_{n+1}$ est donc différent de $B_t \upharpoonright_{n+1}$ pour tout $t \geq m$, et la stratégie n'atteindra jamais l'état w_{m+1} . Contradiction ! La fonction Δ est donc la fonction caractéristique de B , ce qui prouve que B est calculable. ■

Construction. La construction globale est celle d'une méthode de priorité à blessure finie standard. Les stratégies sont ordonnées par priorité décroissante $R_0, S_0, R_1, S_1, \dots$. La construction se divise alors en étapes $t = 0, 1, \dots$, et chaque étape est elle-même divisée en sous-étapes $s < t$. Initialement, toutes les stratégies sont dans l'état **i**. À l'étape $t \geq 0$ et la sous-étape $s < t$, on considère le contrat $\mathcal{R}_e$ si $s = 2e$, et $\mathcal{S}_e$ si $s = 2e + 1$. À la sous-étape $s = 2e$, on exécute la stratégie pour $\mathcal{R}_e$ comme décrite ci-dessus en fonction de son état. Lorsqu'elle atteint l'état **e**, toutes les stratégies de priorité inférieure sont blessées, et reviennent dans l'état **i**. De la même manière, à la sous-étape $s = 2e + 1$, on exécute la stratégie pour $\mathcal{S}_e$ suivant les étapes décrites ci-dessus. Chaque fois qu'elle passe à un état suivant $\mathbf{w}_{n+1}$, toutes les stratégies de priorité inférieure sont blessées, et retournent à l'état **i**.

Le lecteur fera la vérification. Cela conclut la preuve du théorème 4.1. ■

Stratégie de préservation de Sacks

La stratégie pour satisfaire $\mathcal{S}_e$ consiste à ne pas essayer de différencier deux ensembles activement, mais au contraire, de préserver des segments initiaux communs de plus en plus longs, pour rendre le processus effectif, puis utiliser l'hypothèse de non-effectivité de l'un des ensembles pour en déduire que ce processus devra échouer. Cette stratégie se retrouve fréquemment dans ce genre de constructions. Elle est parfois appelée *stratégie de préservation de Sacks*, en l'honneur de son auteur.

5. Méthode de priorité Π_2^0 (à blessure infinie)

Nous allons maintenant aborder une nouvelle élaboration de la méthode de priorité, dite à blessure infinie. Comme son nom l'indique, certaines stratégies vont agir infiniment souvent en posant des restrictions de plus en plus grandes, causant des blessures infinies à des stratégies de priorité inférieure. Nous allons donc commencer à avoir des stratégies conditionnelles, adoptant des comportements différents en fonction des issues des stratégies de priorité supérieure, tirant ainsi pleinement parti du formalisme de « l'arbre des stratégies » que nous verrons bientôt.

Notre illustration de la méthode de priorité à blessure infinie concerne l'existence de paires minimales de degrés c.e. Elle améliore le théorème de Friedberg-Muchnik en la combinant avec la stratégie de préservation de Sacks.

Définition 5.1. Deux degrés non calculables **a** et **b** forment une *paire minimale* si leur borne inférieure est **0**, autrement dit si pour tout ensemble A tel que $A \leq_T \mathbf{a}$ et $A \leq_T \mathbf{b}$, alors A est calculable. ◇

L'existence de paires minimales de degrés c. e. a été prouvée de manière indépendante par Lachlan [129] et Yates [234].

Théorème 5.2 (Lachlan 1966, Yates 1966)

Il existe une paire minimale de degrés c. e.

PREUVE. Nous allons construire deux ensembles c. e. A et B satisfaisant les contrats suivants $(\mathcal{R}_e, \mathcal{S}_e, \mathcal{N}_e)_{e \in \mathbb{N}}$:

$$\mathcal{R}_e : W_e \neq \bar{A} \quad \mathcal{S}_e : W_e \neq \bar{B} \quad \mathcal{N}_e : \Phi_e^A = \Phi_e^B \Rightarrow \Phi_e^A \text{ est calculable.}$$

Les contrats $(\mathcal{R}_e)_{e \in \mathbb{N}}$ et $(\mathcal{S}_e)_{e \in \mathbb{N}}$ s'assurent que A et B ne soient pas calculables. Les contrats $(\mathcal{N}_e)_{e \in \mathbb{N}}$ forcent la borne inférieure des degrés de A et B à être $\mathbf{0}$.

Astuce de Posner

À première vue, pour imposer que la borne inférieure $\deg_T A$ et $\deg_T B$ est $\mathbf{0}$, on s'attendrait à devoir satisfaire des contrats de la forme $(\mathcal{N}_{i,j})_{i,j \in \mathbb{N}}$ avec

$$\mathcal{N}_{i,j} : \Phi_i^A = \Phi_j^B \Rightarrow \Phi_i^A \text{ est calculable.}$$

Cependant, si $\Phi_i^A = \Phi_j^B$, alors il est possible de créer une nouvelle fonctionnelle Φ_e qui codera en dur un entier n tel que $A(n) \neq B(n)$, et exécutera Φ_i ou Φ_j en fonction de la valeur de son oracle à la position n . Ainsi, $\Phi_e^A = \Phi_i^A$ et $\Phi_e^B = \Phi_j^B$. Cette astuce, due à Posner, permet de simplifier les notations.

Satisfaction d'un contrat $\mathcal{R}_e$ ou $\mathcal{S}_e$. La satisfaction d'un contrat $\mathcal{R}_e$ ou $\mathcal{S}_e$ est exactement la même que pour le théorème 3.3. Nous rappelons dans le cas de $\mathcal{R}_e$ les actions de la stratégie en fonction de ses trois états :

- ▷ État **i**. Choisir un entier $x_{\mathcal{R}_e} \notin A$, et poser une restriction sur $x_{\mathcal{R}_e}$.
- ▷ État **e**. Exécuter $\Phi_e(x_{\mathcal{R}_e})$. Si l'exécution s'arrête à l'instant t , ajouter $x_{\mathcal{R}_e}$ à A , puis passer dans l'état **t**.
- ▷ État **t**. Le processus est terminé.

Jusqu'ici, nous avons considéré que cette stratégie avait deux issues, en fonction de l'état dans lequel elle se stabilise. Ces deux issues sont de même nature finitaire, en ce qu'elles ne posent qu'un nombre fini de restrictions lorsqu'elles sont blessées finiment souvent. Nous les considérerons donc comme une seule issue **f**.

Satisfaction d'un contrat $\mathcal{N}_e$. La satisfaction d'un contrat $\mathcal{N}_e$ va suivre la stratégie de préservation de Sacks pour préserver des segments initiaux communs à Φ_e^A et Φ_e^B de plus en plus longs.

Cependant, contrairement au théorème 4.1, le processus ne va pas nécessairement échouer au bout d'un nombre fini d'étapes, car rien dans les hypothèses n'empêche cette égalité. Nous sommes donc dans un cas où l'issue infinitaire va pouvoir se produire, avec des restreintes de plus en plus longues, résultant en une blessure infinie. Comme dans le cas du théorème 4.1, la stratégie possède un état initial i , et une infinité d'états $(w_n)_{n \in \mathbb{N}}$. Dans ce qui suit, nous appellerons *usage* de $\Phi_e^{A_t}[t] \upharpoonright_{n+1}$ le maximum des usages de $\{\Phi_e^{A_t}(x)[t] : x \leq n\}$. Pendant l'exécution de la stratégie, nous allons définir une fonction calculable $\Delta : \mathbb{N} \rightarrow \{0, 1\}$ telle que si Φ_e^A et Φ_e^B sont totaux et égaux, alors ils sont tous les deux égaux à Δ .

- ▷ État i . Définir Δ comme la fonction de domaine vide, et passer à l'état w_0 .
- ▷ État w_n . Attendre une étape $t \geq n$ où $\Phi_e^{A_t}[t] \upharpoonright_{n+1} = \Phi_e^{B_t}[t] \upharpoonright_{n+1}$. Si cela arrive, relâcher sa restreinte précédente, et poser une restreinte sur l'usage de $\Phi_e^{A_t}[t] \upharpoonright_{n+1}$ si n est pair, et sur l'usage de $\Phi_e^{B_t}[t] \upharpoonright_{n+1}$ si n est impair. Ensuite, définir $\Delta(n) = \Phi_e^{A_t}(n)[t]$, et passer à l'état w_{n+1} .

La stratégie a deux issues possibles. Issue f (finitaire) : elle se retrouve bloquée dans l'état w_n pour un n donné ; dans ce cas, soit Φ_e^A ou Φ_e^B est partiel, soit $\Phi_e^A \neq \Phi_e^B$. Issue ∞ (infinitaire) : la stratégie passe par tous les états $(w_n)_{n \in \mathbb{N}}$; dans ce cas, les deux ensembles coïncident, et infiniment souvent, la restreinte change de côté. Il nous reste encore à établir les égalités $\Delta = \Phi_e^A = \Phi_e^B$ pour en déduire que cet ensemble est calculable. L'idée sous-jacente de la preuve est très simple, mais elle est un peu lourde à formaliser. Nous allons donc l'illustrer par une figure (voir la figure 5.3) avant de prouver formellement le résultat à travers le lemme 5.4. Quelle que soit l'issue, le contrat $\mathcal{N}_e$ est donc satisfait.

Remarque

Le choix du côté de la restreinte (A si la stratégie passe d'un état w_n à w_{n+1} avec n pair, et B si n est impair) n'intervient pas dans la preuve de la validité d'une stratégie $\mathcal{N}_e$ indépendamment des autres. On aurait aussi bien pu garder toujours le même côté, ou bien même poser une restreinte des deux côtés, ce qui aurait considérablement simplifié la preuve de validité. Cependant, cette alternance de côtés devient nécessaire lorsque l'on cherche à satisfaire un contrat de type $\mathcal{R}$ ou $\mathcal{S}$ sous une stratégie pour $\mathcal{N}_e$, comme nous le verrons par la suite.

Lemme 5.4. Si l'issue ∞ arrive, alors $\Delta = \Phi_e^A = \Phi_e^B$. ★

PREUVE. Soit $P(n, s)$ la proposition « Soit (1) $\Delta \upharpoonright_{n+1} = \Phi_e^{A_s}[s] \upharpoonright_{n+1}$ avec une restreinte sur son usage, soit (2) $\Delta \upharpoonright_{n+1} = \Phi_e^{B_s}[s] \upharpoonright_{n+1}$ avec une restreinte sur son usage. »

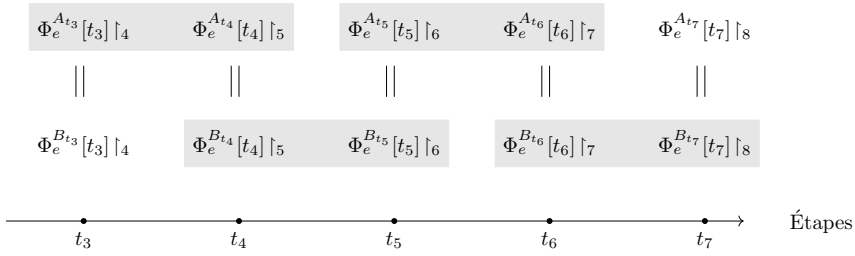


FIGURE 5.3 – Soit t_n l'étape de passage à l'état $\mathbf{w}_{n+1}$. Le rectangle gris entre l'étape t_3 et t_4 signifie que l'usage de $\Phi_e^{A_{t_3}}[t_3] \downarrow_4$ est restreint, et donc préservé jusqu'à l'étape t_4 . Ainsi, $\Phi_e^{A_{t_3}}[t_3] \downarrow_4 = \Phi_e^{A_{t_4}}[t_4] \downarrow_4$. À chaque étape t_n , les segments initiaux de longueur $n + 1$ des deux fonctionnelles sont égaux. Ainsi, $\Phi_e^{A_{t_4}}[t_4] \downarrow_5 = \Phi_e^{B_{t_4}}[t_4] \downarrow_5$. Nous avons donc

$$\Phi_e^{B_{t_3}}[t_3] \downarrow_4 = \Phi_e^{A_{t_3}}[t_3] \downarrow_4 = \Phi_e^{A_{t_4}}[t_4] \downarrow_4 = \Phi_e^{B_{t_4}}[t_4] \downarrow_4.$$

Même si l'usage de $\Phi_e^{B_{t_3}}[t_3] \downarrow_4$ peut être différent de celui de $\Phi_e^{B_{t_4}}[t_4] \downarrow_4$, car aucune restriction n'est posée sur B entre les étapes t_3 et t_4 , les quatre premières valeurs de sortie de la fonctionnelle sont préservées.

Pour tout n , soit t_n l'étape à laquelle la stratégie passe dans l'état $\mathbf{w}_{n+1}$. Nous allons montrer par induction sur les entiers n et s que pour tout $n \geq 0$ et $s \geq t_n$, la proposition $P(n, s)$ est vraie. Par convention, $t_{-1} = 0$, et pour tout $s \geq t_{-1}$, $P(-1, s)$ est vraie.

Soit $n \geq 0$. Montrons que si pour tout $s \geq t_{n-1}$, $P(n-1, s)$ est vraie, alors $P(n, t_n)$ est vraie. À l'étape t_n , $\Delta(n) = \Phi_e^{A_{t_n}}(n)[t_n] = \Phi_e^{B_{t_n}}(n)[t_n]$, et $\Phi_e^{A_{t_n}}[t_n] \downarrow_{n+1} = \Phi_e^{B_{t_n}}[t_n] \downarrow_{n+1}$. Comme $t_n \geq t_{n-1}$, par hypothèse d'induction, $P(n-1, t_n)$ est vraie, de sorte que l'on a soit $\Delta \downarrow_n = \Phi_e^{A_{t_n}}[t_n] \downarrow_n$, soit $\Delta \downarrow_n = \Phi_e^{B_{t_n}}[t_n] \downarrow_n$. Par suite, $\Delta \downarrow_{n+1} = \Phi_e^{A_{t_n}}[t_n] \downarrow_{n+1} = \Phi_e^{B_{t_n}}[t_n] \downarrow_{n+1}$. À cette étape-ci, la stratégie pose une restriction sur $\Phi_e^{A_{t_n}}[t_n] \downarrow_{n+1}$ ou bien sur $\Phi_e^{B_{t_n}}[t_n] \downarrow_{n+1}$, donc $P(n, t_n)$ est vraie.

Soit $s > t_n$. Montrons que si $P(n, s-1)$ est vraie, alors $P(n, s)$ est vraie. Si la stratégie ne change pas d'état à l'étape s , alors elle garde sa restriction, et par la propriété de l'usage, $P(n, s)$ reste vraie. Si la stratégie passe à un état $\mathbf{w}_{p+1}$, alors par définition, $\Phi_e^{A_s}[s] \downarrow_{p+1} = \Phi_e^{B_s}[s] \downarrow_{p+1}$, or $p \geq n$, donc $\Phi_e^{A_s}[s] \downarrow_{n+1} = \Phi_e^{B_s}[s] \downarrow_{n+1}$.

Par $P(n, s-1)$, soit $\Delta \downarrow_{n+1} = \Phi_e^{A_{s-1}}[s-1] \downarrow_{n+1}$ avec une restriction sur son usage, soit (2) $\Delta \downarrow_{n+1} = \Phi_e^{B_{s-1}}(n)[s-1] \downarrow_{n+1}$ avec une restriction sur son

usage. Par la restriction à l'étape $s - 1$ et la propriété de l'usage,

$$\text{soit } \Delta \upharpoonright_{n+1} = \Phi_e^{A_s}[s] \upharpoonright_{n+1}, \quad \text{soit } \Delta \upharpoonright_{n+1} = \Phi_e^{B_s}[s] \upharpoonright_{n+1}.$$

Il s'ensuit que $\Delta \upharpoonright_{n+1} = \Phi_e^{A_s}[s] \upharpoonright_{n+1} = \Phi_e^{B_s}[s] \upharpoonright_{n+1}$. La stratégie pose une restriction sur l'usage de $\Phi_e^{A_s}[s] \upharpoonright_{p+1}$ ou $\Phi_e^{B_s}[s] \upharpoonright_{p+1}$, donc $P(n, s)$ est vraie. Cela termine la preuve du lemme. ■

Notons que contrairement au théorème 4.1, l'issue infinitaire va vraiment se produire avec la stratégie pour $\mathcal{N}_e$. Elle ne se combine donc pas aussi bien avec les stratégies pour $\mathcal{R}_d$ et $\mathcal{S}_d$. En effet, lorsque cette issue arrive, la stratégie pour $\mathcal{N}_e$ va poser des restrictions de plus en plus longues, causant une blessure infinie. Nous allons donc devoir adapter la construction pour permettre aux autres contrats d'être satisfaits.

Étapes d'exécution

Il n'est pas nécessaire d'exécuter les stratégies pour $\mathcal{R}_e$, $\mathcal{S}_e$ et $\mathcal{N}_e$ à chaque étape pour satisfaire leurs contraintes respectives. Il suffit de les exécuter chacune pendant un nombre infini d'étapes, tout en maintenant leurs restrictions pendant les étapes intermédiaires.

Prenons l'exemple de la stratégie pour $\mathcal{R}_e$. Si elle n'est exécutée qu'aux temps $t_0 < t_1 < \dots$, il se peut qu'elle « manque » la première étape t entre t_0 et t_1 où $\Phi_e(x_{\mathcal{R}_e})[t]$ s'arrête, ce qui l'empêche de passer dans l'état $\mathfrak{t}$ à cette étape. Cependant, à l'étape t_1 , $\Phi_e(x_{\mathcal{R}_e})[t_1]$ s'arrêtera également, et le passage à l'état $\mathfrak{t}$ aura quand même lieu. Le comportement limite de la stratégie pour $\mathcal{R}_e$ ne dépend donc pas du choix des étapes.

Le cas de la stratégie pour $\mathcal{N}_e$ est un peu plus subtil. Il se peut que la stratégie par défaut pour $\mathcal{N}_e$ ait une issue infinitaire, mais que lorsqu'elle n'est exécutée qu'aux temps $t_0 < t_1 < \dots$, on ait

$$\Phi_e^{A_i}[t_i] \upharpoonright_{n+1} \neq \Phi_e^{B_i}[t_i] \upharpoonright_{n+1},$$

ce qui fait que la stratégie ne passera jamais à l'état $\mathfrak{w}_{n+1}$, ce qui correspond à l'issue finitaire. Heureusement, même dans ce cas, $\mathcal{N}_e$ sera satisfait, tant que l'énumération des étapes $(t_i)_{i \in \mathbb{N}}$ est calculable pour que la fonction Δ le soit également.

Satisfaction d'un contrat $\mathcal{R}_d$ ou $\mathcal{S}_d$ sous $\mathcal{N}_e$. Supposons que l'on veuille satisfaire un contrat $\mathcal{R}_d$ sous la stratégie pour $\mathcal{N}_e$, c'est-à-dire avec la stratégie pour $\mathcal{N}_e$ de priorité supérieure à celle pour $\mathcal{R}_d$. Plusieurs solutions se présentent, en fonction de l'issue de la stratégie pour $\mathcal{N}_e$.

- ▷ Issue $\mathfrak{f}$ (finitaire). Dans ce cas, il suffit d'utiliser la stratégie standard pour $\mathcal{R}_d$ présentée plus haut. En effet, la stratégie pour $\mathcal{N}_e$ posera un

nombre fini de contraintes, ce qui fait que la stratégie pour $\mathcal{R}_d$ sera blessée et réinitialisée un nombre fini de fois avant d'être satisfaite. Cette stratégie ne fonctionne pas si l'issue de la stratégie pour $\mathcal{N}_e$ est infinitaire (issue ∞). En effet, dans ce cas, la stratégie pour $\mathcal{R}_d$ sera blessée infiniment souvent, et pourrait ne jamais satisfaire $\mathcal{R}_d$.

- ▷ Issue ∞ (infinitaire). Remarquons que dans ce cas, la contrainte posée par la stratégie pour $\mathcal{N}_e$ alternera infiniment souvent de côté, et libérera donc l'autre côté, permettant à la stratégie pour $\mathcal{R}_d$ d'être satisfaite. Nous allons donc n'exécuter la stratégie pour $\mathcal{R}_d$ qu'aux étapes où la contrainte de la stratégie pour $\mathcal{N}_e$ est retirée du côté A . L'issue de la stratégie pour $\mathcal{N}_e$ étant infinitaire, la stratégie pour $\mathcal{R}_d$ va être exécutée pendant un nombre infini d'étapes, et comme expliqué plus haut, un sous-ensemble infini d'étapes est suffisant pour satisfaire $\mathcal{R}_d$. Cette stratégie pour $\mathcal{R}_d$ ne fonctionne cependant pas si l'issue de la stratégie pour $\mathcal{N}_e$ est finitaire, car il se peut qu'elle ne lève jamais sa contrainte et que la stratégie pour $\mathcal{R}_d$ attende donc pour toujours.

Nous avons donc deux stratégies différentes pour satisfaire $\mathcal{R}_d$ sous $\mathcal{N}_e$, en fonction de l'issue de la stratégie pour $\mathcal{N}_e$. Cette analyse de cas pose une difficulté : pour produire un ensemble c. e., il faut que la construction soit un processus calculable, or l'issue de $\mathcal{N}_e$ ne peut pas être décidée en un temps fini. Nous ne pouvons donc pas savoir quelle stratégie choisir pour $\mathcal{R}_d$. La solution consiste à lancer les deux stratégies pour $\mathcal{N}_e$ en parallèle, chacune faisant une supposition sur l'issue. Celle faisant la bonne supposition pourra alors satisfaire $\mathcal{R}_d$ sous $\mathcal{N}_e$. Nous allons donc nous retrouver avec un arbre de stratégies, induit par les analyses de cas successives sur les issues des stratégies de type $\mathcal{N}$. Nous détaillerons plus bas cette structure arborescente.

Satisfaction d'un contrat $\mathcal{N}_d$ sous $\mathcal{N}_e$. Les contrats de même nature sont souvent faciles à satisfaire simultanément, car leurs stratégies n'entrent généralement pas en conflit. Dans le cas de la satisfaction d'un contrat $\mathcal{N}_d$ sous $\mathcal{N}_e$, la difficulté n'est pas de satisfaire ces deux contrats simultanément, mais de laisser ensuite la place à un contrat de type $\mathcal{R}$ ou $\mathcal{S}$ d'être satisfait sous $\mathcal{N}_d$. En effet, dans le pire des cas, les stratégies pour $\mathcal{N}_e$ et $\mathcal{N}_d$ seront toutes les deux infinitaires, et à chaque fois que la stratégie pour $\mathcal{N}_e$ libérera ses contraintes du côté A , la stratégie pour $\mathcal{N}_d$ posera les siennes, ce qui fait que le côté A aura à tout moment des contraintes de plus en plus grandes, ne permettant pas aux stratégies de type $\mathcal{R}$ d'être satisfaites en dessous. La solution consiste à « synchroniser » les stratégies pour $\mathcal{N}_d$ et $\mathcal{N}_e$. Plus précisément, la stratégie pour $\mathcal{N}_d$ va être définie par analyse de cas en fonction de l'issue de $\mathcal{N}_e$.

- ▷ Issue **f** (finitaire). Dans ce cas, la stratégie pour $\mathcal{N}_d$ est la stratégie standard présentée plus haut. En effet, les contraintes de la stratégie pour $\mathcal{N}_e$

seront finitaires, et les autres contrats auront la possibilité d'être satisfaits sous $\mathcal{N}_d$ en étant blessés finiment souvent par la stratégie de $\mathcal{N}_e$.

- ▷ Issue ∞ (infinitaire). Modifions la stratégie pour $\mathcal{N}_d$, comme suit. Cette stratégie ne sera exécutée que pendant des étapes où la stratégie pour $\mathcal{N}_e$ change ses restrictions de côté, autrement dit passe de l'état $\mathbf{w}_n$ à $\mathbf{w}_{n+1}$. Cela permet de s'assurer que lorsque la stratégie pour $\mathcal{N}_d$ change ses restrictions de côté, à la même étape, la stratégie pour $\mathcal{N}_e$ changera les siennes de côté. Enfin, il faut s'assurer que ces changements aillent du même côté. Pour cela, si la stratégie pour $\mathcal{N}_d$ est dans un état $\mathbf{w}_n$ avec n un entier pair, autrement dit si ses restrictions sont du côté B , la stratégie pour $\mathcal{N}_d$ ne sera exécutée que pendant les étapes où la stratégie pour $\mathcal{N}_e$ passe d'un état $\mathbf{w}_m$ à $\mathbf{w}_{m+1}$ avec m pair, pour que les deux stratégies mettent leur restrictions sur le côté A simultanément. De la même manière, si la stratégie pour $\mathcal{N}_d$ est dans un état $\mathbf{w}_n$ avec n un entier impair, elle s'exécutera pendant les étapes où la stratégie pour $\mathcal{N}_e$ passe d'un état $\mathbf{w}_m$ à $\mathbf{w}_{m+1}$ avec m impair.

Comme dans le cas de la satisfaction d'un contrat $\mathcal{R}_d$ sous un contrat $\mathcal{N}_e$, nous avons donc des stratégies pour $\mathcal{N}_d$ différentes pour chaque issue de la stratégie pour $\mathcal{N}_e$. Nous allons donc lancer deux stratégies en parallèle, chacune supposant que son hypothèse est la bonne. Nous allons maintenant décrire l'arbre des stratégies.

Arbre des stratégies. Les contrats sont énumérés de la manière suivante.

$$\mathcal{N}_0, \mathcal{R}_0, \mathcal{S}_0, \mathcal{N}_1, \mathcal{R}_1, \mathcal{S}_1, \dots$$

Dans les constructions précédentes, chaque contrat se voyait attribuer une unique stratégie, et l'ordre de priorité des stratégies suivait l'énumération des contrats. Nous avons maintenant une structure arborescente de stratégies en fonction des issues des stratégies précédentes, comme ci-après : le contrat $\mathcal{N}_0$ possède une unique stratégie N_ϵ (où ϵ est la chaîne vide). Le contrat $\mathcal{R}_0$ admet deux stratégies R_∞ et $R_{\mathbf{f}}$, en fonction des issues ∞ et $\mathbf{f}$ de la stratégie $\mathcal{N}_0$. Le contrat $\mathcal{S}_0$ possède également deux stratégies $S_{\infty\mathbf{f}}$ et $S_{\mathbf{f}\mathbf{f}}$, en fonction des issues des stratégies plus hautes. Par exemple, $S_{\infty\mathbf{f}}$ doit satisfaire le contrat $\mathcal{S}_0$ sous l'hypothèse que l'issue de N_ϵ est ∞ et l'issue de R_∞ est $\mathbf{f}$.

De manière générale, nous pouvons définir un arbre $\mathcal{T} \subseteq \{\mathbf{f}, \infty\}^{<\mathbb{N}}$ de chaînes dans l'alphabet $\{\mathbf{f}, \infty\}$, induit par la relation de préfixe, tel que pour tout $\sigma \in \mathcal{T}$ et $i < |\sigma|$ tel que $i \not\equiv 0 \pmod 3$, $\sigma(i) = \mathbf{f}$. À chaque chaîne $\sigma \in \mathcal{T}$, si $|\sigma| = 3e$ on associe une stratégie N_σ pour $\mathcal{N}_e$, si $|\sigma| = 3e+1$ on associe une stratégie R_σ pour $\mathcal{R}_e$, et si $|\sigma| = 3e+2$ on associe une stratégie S_σ pour $\mathcal{S}_e$ (voir la figure 5.5). Pour simplifier les notations, nous noterons C_σ la stratégie dont l'indice est σ .

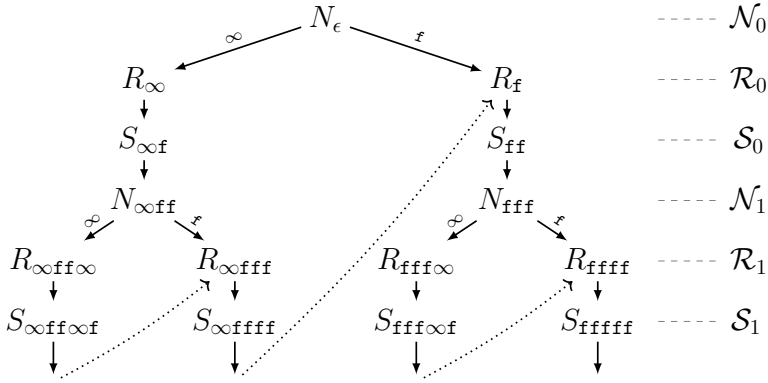


FIGURE 5.5 – Arbre des stratégies pour la construction d’une paire minimale. Les flèches pleines indiquent la structure d’arbre, mais également induisent un ordre partiel de priorité entre les stratégies. Les flèches en pointillés permettent de linéariser l’ordre partiel pour obtenir un ordre total entre les stratégies. Par exemple, toutes les stratégies du sous-arbre gauche de la stratégie $N_{\infty f f}$ sont plus faibles que N_{ϵ} , R_{∞} , $S_{\infty f}$ et $N_{\infty f f}$, mais sont plus fortes que toutes les autres stratégies.

Construction. Au début de la construction, chaque stratégie est dans l’état initial $\mathbf{i}$. À chaque étape t , nous allons définir un *nœud courant*, qui est une chaîne $\sigma_t \in \mathcal{T}$ de longueur t , représentant une approximation des issues à la fin de l’étape t . Plus précisément, $\sigma_t(i)$ représente l’approximation à l’étape t de l’issue de la stratégie $C_{\sigma_t \upharpoonright i}$. Le nœud courant à l’étape t est défini par induction sur les sous-étapes $s < t$ comme suit.

Au début de la sous-étape i , on suppose $\sigma_t \upharpoonright_i$ défini. On exécute alors la stratégie $C_{\sigma_t \upharpoonright_i}$ au temps t comme décrit ci-dessus. Rappelons qu’il se peut que l’on n’exécute pas la stratégie $C_{\sigma_t \upharpoonright_i}$ pour des raisons de synchronisation avec les stratégies pour les contrats de type $\mathcal{N}$. Dans ce cas, la stratégie reste dans son état et garde ses contraintes inchangées. À la fin de la sous-étape t , on définit $\sigma_t(i) = \infty$ si $i \equiv 0 \pmod{3}$, et la stratégie $N_{\sigma_t \upharpoonright_i}$ a changé d’état à l’étape t . Dans tous les autres cas, $\sigma_t(i) = \mathbf{f}$, car les stratégies de type $\mathcal{R}$ et $\mathcal{S}$ n’ont que l’issue finitaire possible, et si $i \equiv 0 \pmod{3}$, et $N_{\sigma_t \upharpoonright_i}$ ne vient pas de changer d’état, nous supposons que son issue est finitaire.

Vrai chemin. Nous pouvons maintenant définir le *vrai chemin* de $\mathcal{T}$, qui est le chemin le long duquel les issues sont les vraies. Plus précisément, le vrai chemin de $\mathcal{T}$ est la suite infinie $P \in \Lambda^{\mathbb{N}}$ (avec $\Lambda = \{\infty, \mathbf{f}\}$, où ∞ est plus petit que $\mathbf{f}$) définie inductivement sur i par

$$P(i) = \liminf \{o \in \Lambda : \exists^\infty t (P \upharpoonright_i) \frown o \preceq \sigma_t\}.$$

Intuitivement, les stratégies le long du vrai chemin vont être celles qui feront les bonnes hypothèses sur les issues des stratégies précédentes. Elles seront blessées finiment souvent par une stratégie de plus grande priorité, et réussiront à satisfaire leur contrat. Nous allons maintenant définir un ordre de priorité sur les stratégies pour que les stratégies le long du vrai chemin soient finiment blessées.

Ordre de priorité. Soit $\Lambda = \{\infty, \mathbf{f}\}$ l'ensemble des issues possibles. Munissons cet ensemble d'un ordre de priorité $<_p$ en considérant que $\infty <_p \mathbf{f}$, ce qui signifie que l'issue ∞ est prioritaire par rapport à $\mathbf{f}$. Cet ordre va induire un ordre total $(\mathcal{T}, <_p)$ (et donc un ordre total de priorité sur les stratégies) comme suit : $\sigma <_p \tau$ si $\sigma \preceq \tau$, ou si i est la première position où les deux chaînes diffèrent, et $\sigma(i) <_p \tau(i)$. Un exemple visuel de cet ordre est donné dans la figure 5.5. Notons que, contrairement aux méthodes de priorité à blessure finie, les stratégies sont généralement en dessous d'une infinité de stratégies de priorité supérieure. Par exemple, la stratégie $R_{\infty \mathbf{f} \mathbf{f}}$ est en dessous de N_ϵ , R_∞ , $S_{\infty \mathbf{f}}$, $N_{\infty \mathbf{f} \mathbf{f}}$, $R_{\infty \mathbf{f} \mathbf{f} \infty}$, $S_{\infty \mathbf{f} \mathbf{f} \infty \mathbf{f}}$, $\dots$

Remarque

On serait tenté de définir un ordre de priorité tel que chaque stratégie n'ait qu'un nombre fini de stratégies prioritaires, par exemple en définissant les priorités par un parcours en largeur des nœuds de l'arbre. Le problème suivant se pose cependant.

Supposons que la stratégie N_ϵ ait pour issue ∞ . La stratégie $R_{\mathbf{f}}$ faisant la mauvaise hypothèse selon laquelle l'issue de N_ϵ a une issue finitaire, elle sera blessée et réinitialisée à chaque changement d'état de N_ϵ . Elle posera donc potentiellement un nombre infini de contraintes, et empêchera donc toutes les stratégies de priorité inférieure de fonctionner normalement. Il est donc essentiel que toutes les stratégies le long du vrai chemin de l'arbre soient prioritaires par rapport à $R_{\mathbf{f}}$, afin d'ignorer ses contraintes. La stratégie pour $R_{\mathbf{f}}$ doit donc être sous une infinité de stratégies prioritaires.

Vérification. Notons tout d'abord que, par définition, les stratégies le long du vrai chemin P (stratégies C_σ pour $\sigma \prec P$) sont exécutées à une infinité d'étapes. Bien qu'une stratégie soit en général en dessous d'une infinité de stratégies, nous allons montrer que les stratégies le long du vrai chemin sont blessées finiment souvent, ce qui est la condition nécessaire pour les satisfaire. Le lemme suivant est une propriété que l'on attend généralement d'un argument de priorité Π_2^0 .

Lemme 5.6. Soit P le vrai chemin, et soit $\alpha \prec P$. Alors, $\alpha \leq_p \sigma_t$ pour un nombre co-fini d'étapes t . ★

PREUVE. Par induction sur la longueur n de α . Si $n = 0$, alors $\alpha = \epsilon$, et par définition, $\epsilon \leq_p \sigma_t$ pour tout t . Soit $n > 0$. Par hypothèse d'induction, il existe un seuil t_0 tel que $\alpha \upharpoonright_{n-1} \leq_p \sigma_t$ pour tout $t \geq t_0$. Soit

$$S = \{t \geq t_0 : \sigma_t <_p \alpha\}.$$

Supposons par l'absurde que S est infini.

Notons que pour tout $t \in S$, comme $\alpha \upharpoonright_{n-1} \leq_p \sigma_t$ et $\sigma_t <_p \alpha$, on a forcément $\alpha \upharpoonright_{n-1} \preceq \sigma_t$ avec $\sigma_t(n-1) <_p \alpha(n-1)$. Autrement dit, $\sigma_t(n-1) = \infty$ et $\alpha(n-1) = \mathbf{f}$ et $\forall t \in S$ $(\alpha \upharpoonright_{n-1})^\frown \infty \preceq \sigma_t$. Ainsi, $(\alpha \upharpoonright_{n-1})^\frown \infty \preceq P$, contredisant l'hypothèse $\alpha \preceq P$. Cela conclut la preuve du lemme. ■

Seules les stratégies le long de σ_t sont exécutées à l'étape t . Ainsi, pour toute stratégie C_α le long du vrai chemin P , il existe un seuil t_0 après lequel seules les stratégies de plus faible priorité ou des stratégies le long du vrai chemin vont être exécutées. Nous pouvons donc prouver par induction sur n que la stratégie $C_{P \upharpoonright_n}$ ne sera blessée que finiment souvent, et aura l'issue $P(n)$. Tout contrat étant représenté par une stratégie le long du vrai chemin, les contrats seront tous satisfaits. Cela conclut la preuve du théorème 5.2. ■

Degrés cappable

Nous avons vu deux manières de créer des degrés Turing incomparables. La première (voir la proposition 4-8.1) consiste à créer deux ensembles simultanément, tandis que la seconde (voir la proposition 4-8.2) part d'un ensemble non calculable, et crée un deuxième ensemble de degré incomparable avec le premier. Il est naturel de se demander s'il est possible, dans le cas des paires minimales de degrés c. e., de partir d'un ensemble c. e. non calculable arbitraire, et le compléter avec un autre ensemble c. e. pour former une paire minimale. Ce n'est pas le cas, comme l'ont prouvé Yates [234] et Lachlan [129].

Définition 5.7. Un degré c. e. $\mathbf{a} > \mathbf{0}$ est dit *cappable* s'il existe un degré $\mathbf{b} > \mathbf{0}$ tel que $\mathbf{a}$ et $\mathbf{b}$ forment une paire minimale. Sinon, $\mathbf{a}$ est dit *non cappable*. ◇

Notons que le terme « cappable » vient de l'anglais « cap » et « able », et n'a en particulier rien à avoir avec l'adjectif « capable ». Ambos-Pies et al. [5] ont obtenu une surprenante caractérisation des degrés non cappable, à l'aide des ensembles promptement simples.

Définition 5.8. Un ensemble co-infini c. e. A est *promptement simple* s'il existe une énumération c. e. calculable $A_0 \subseteq A_1 \subseteq \dots$ et une fonction calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telles que

$$W_e \text{ infini} \Rightarrow \exists^\infty x, s \ (x \in W_e[s] \setminus W_e[s-1] \wedge x \in A_{f(s)}). \quad \diamond$$

Autrement dit, un ensemble co-infini A est promptement simple s'il est non seulement co-immune, mais plus encore, cette co-immunité doit être réalisée en faisant infiniment souvent entrer des éléments dans A peu de temps après qu'ils apparaissent dans W_e .

Un degré c. e. est *promptement simple* s'il contient un ensemble promptement simple.

Théorème 5.9 (Ambos-Spies, Jockusch, Shore, et Soare [5])
Les degrés non cappables sont précisément les degrés promptement simples.

En particulier, la preuve montrant qu'un degré est cappable s'il n'est pas promptement simple est obtenue avec une variation du théorème 5.2.

Chapitre 14

Structure des degrés Turing

L'étude des degrés Turing a été menée conjointement avec celle de sa *structure*, en tant qu'ordre partiel. Gerald E. Sacks est sans aucun doute l'un des principaux protagonistes de cette aventure.

Sacks entame dans sa jeunesse des études d'ingénieur à l'université de Cornell, qu'il interrompt à mi-parcours pour s'engager trois ans durant dans l'armée. C'est à ce moment qu'il met la main sur une copie de *Introduction to Metamathematics* de Kleene, qui le passionne [35]. À son retour à la vie civile, il oriente alors la suite de ses études vers les mathématiques. Barkley Rosser, éminent logicien qui fut étudiant avec Kleene auprès de Church, accepte de le



Gerald Sacks, 1933–2019

prendre comme étudiant en thèse. Sacks deviendra dans les années soixante l'un des pionniers de la calculabilité moderne. Il participera notamment comme nous le verrons aux premières études sur la structure des degrés Turing, et outre son travail, il deviendra célèbre pour le grand nombre de ses étudiants qui deviendront des logiciens de premier plan, parmi lesquels on peut citer Harvey et Sy Friedman, Léo Harrington, Richard Shore, Théodore Slaman et Stephen Simpson. Nous verrons que les trois derniers

membres de cette liste ont pris une part très active dans l'étude de la structure des degrés Turing.

Posons sans plus tarder le vocabulaire que nous utiliserons.

Notation

On notera $(\mathcal{D}, \leq)$ la structure d'ordre partiel des degrés Turing.

On écrira $\mathbf{a} \leq \mathbf{b}$ pour deux degrés $\mathbf{a}, \mathbf{b} \in \mathcal{D}$ si $A \leq_T B$ pour tout élément $A \in \mathbf{a}$ et tout élément $B \in \mathbf{b}$, et l'on écrira $\mathbf{a} < \mathbf{b}$ si $\mathbf{a} \leq \mathbf{b}$ et $\mathbf{b} \not\leq \mathbf{a}$. Afin d'être tout à fait clairs, rappelons le vocabulaire d'usage des ordres partiels : étant donné un ensemble partiellement ordonné A et un sous-ensemble $B \subseteq A$, un *majorant* (resp. *minorant*) de B est un élément de A plus grand (resp. plus petit) que tous les éléments de B . Un majorant (resp. minorant) de B est *minimal* (resp. *maximal*) si aucun autre majorant de B n'est plus petit que lui (resp. aucun autre minorant de B n'est plus grand que lui). Enfin, un majorant (resp. minorant) de B est une *borne supérieure* (resp. *borne inférieure*) si elle est plus petite que tout majorant de B (resp. plus grande que tous les minorants de B). On montre sans peine qu'une borne supérieure (resp. inférieure) quand elle existe est unique.

La littérature sur le sujet étant exclusivement en anglais, nous mettons l'accent sur le fait que l'anglais *upper bound* et *lower bound* ne correspond pas au français « borne supérieure » et « borne inférieure », mais plutôt à majorant et minorant. L'anglais utilise à la place *least upper bound* et *greatest lower bound* pour borne supérieure et borne inférieure.

1. Degrés minimaux

Un des tout premiers résultats obtenus sur la structure des degrés Turing concerne les segments initiaux de $\mathcal{D}$, et notamment l'existence de degrés dits *minimaux*.

Définition 1.1. Un degré Turing $\mathbf{d}$ est *minimal* s'il est différent de $\mathbf{0}$ — le degré calculable — et s'il n'existe pas de degré $\mathbf{e}$ tel que $\mathbf{0} < \mathbf{e} < \mathbf{d}$. $\diamond$

Autrement dit, un ensemble $X \in 2^{\mathbb{N}}$ est de degré minimal s'il est non calculable et si pour toute fonctionnelle Φ telle que $\Phi(X, n) \downarrow \in \{0, 1\}$ pour tout n , soit l'ensemble $\{n : \Phi(X, n) \downarrow = 1\}$ est calculable, soit il permet de calculer X .

La preuve de l'existence d'un degré minimal est une des premières utilisations du forcing en calculabilité, via le forcing de Sacks, que nous avons vu dans la section 11-3.

1.1. Existence

L'existence de degrés minimaux, due à Spector [215], fut au départ faite par forcing avec des f-arbres $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ calculables *uniformes*, c'est-à-dire des f-arbres tels que pour toute taille n il existe une unique chaîne τ_n telle que, pour toute chaîne σ de taille n et tout $i \in \{0, 1\}$, on a

$$T(\sigma i) = T(\sigma) i \tau_n.$$

On peut aussi voir les chemins de ces f-arbres comme étant toutes les possibilités de complétion d'un ensemble X sur lequel une infinité de bits ne sont pas spécifiés.

La restriction de Spector présente son intérêt pour l'étude plus générale de segments initiaux dans les degrés Turing, mais pour ce qui est des degrés minimaux (c'est-à-dire des segments initiaux de taille 2), Shoenfield [198] a remarqué que le forcing de Sacks calculable, plus simple à manipuler, est suffisant.

Définition 1.2. Soit Γ une fonctionnelle Turing.

- (1) Deux chaînes $\sigma, \tau \in 2^{<\mathbb{N}}$ forment une Γ -*scission* s'il existe un entier n tel que $\Gamma^\sigma(n) \downarrow \neq \Gamma^\tau(n) \downarrow$.
- (2) Un f-arbre $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ est Γ -*scindé* si pour tout $\sigma \in 2^{<\mathbb{N}}$, les deux chaînes $T(\sigma 0)$ et $T(\sigma 1)$ forment une Γ -scission.
- (3) Un f-arbre T est Γ -*uni* si aucune paire de chaînes $\sigma, \tau \in \text{Im } T$ ne forme une Γ -scission. ◇

On considérera pour cette section les fonctionnelles Γ comme étant des fonctions partielles de $2^{\mathbb{N}}$ vers $2^{\mathbb{N}}$. Dans ce contexte particulier, on notera alors $\text{dom } \Gamma$ — le domaine de définition de Γ — comme étant la classe

$$\{X \in 2^{\mathbb{N}} : \forall n \ \Gamma(X, n) \downarrow \in \{0, 1\}\}.$$

Étant donné $X \in \text{dom } \Gamma$, on écrira $\Gamma(X)$ pour l'ensemble

$$\{n \in \mathbb{N} : \Gamma(X, n) \downarrow = 1\},$$

et l'on parlera de la totalité de Γ vis-à-vis de $2^{\mathbb{N}}$, et non pas vis-à-vis de ses entrées pour un oracle fixé.

Le lemme clef dans la construction d'un degré minimal dit que pour toute fonctionnelle Γ et tout f-arbre calculable, il existe un sous f-arbre calculable sur lequel Γ est partout définie et injective, ou sur lequel Γ est une fonction constante, restreinte à son domaine de définition dans le sous f-arbre.

Lemme 1.3. Pour tout f-arbre calculable $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ et toute fonctionnelle Turing Γ , il existe un sous f-arbre calculable S de T qui est soit Γ -scindé, soit Γ -uni. ★

PREUVE. Deux cas se présentent.

Cas 1. Il existe une chaîne $\sigma \in 2^{<\mathbb{N}}$ telle pour tous $\rho, \tau \succeq \sigma$, les chaînes $T(\rho)$ et $T(\tau)$ ne forment pas de Γ -scission. Soit S le sous-f-arbre calculable donné par $S(\mu) = T(\sigma\mu)$. Alors, S est un sous-f-arbre Γ -uni de T .

Cas 2. Pour toute chaîne $\sigma \in 2^{<\mathbb{N}}$, il existe des extensions $\rho, \tau \succeq \sigma$ telles que $T(\rho)$ et $T(\tau)$ forment une Γ -scission. On calcule alors S comme suit. On définit $S(\epsilon) = T(\epsilon)$. Supposons que l'on ait calculé $S(\sigma) = T(\mu)$ pour des chaînes $\sigma, \mu \in 2^{<\mathbb{N}}$. On cherche alors des chaînes $\rho_0, \rho_1 \succeq \mu$ telles que $T(\rho_0)$ et $T(\rho_1)$ forment une Γ -scission. Par hypothèse, la recherche aboutit nécessairement. On définit alors $S(\sigma 0) = T(\rho_0)$ et $S(\sigma 1) = T(\rho_1)$. Notons que comme $T(\rho_0)$ et $T(\rho_1)$ forment une Γ -scission, en particulier, ces deux chaînes sont incomparables.

Ainsi, S est bien un f-arbre, et $\text{Im } S \subseteq \text{Im } T$. Par construction, S est Γ -scindé. ■

L'intérêt d'obtenir des f-arbres Γ -unis ou Γ -scindés se trouve dans le lemme suivant.

Lemme 1.4. Soient T un f-arbre calculable et Γ une fonctionnelle.

- (1) Si T est Γ -uni, alors pour tout $X \in \text{dom } \Gamma \cap [T]$ l'élément $\Gamma(X)$ est calculable.
- (2) Si T est Γ -scindé, alors pour tout $X \in \text{dom } \Gamma \cap [T]$ on a $X \leq_T \Gamma(X)$. ★

PREUVE. (1) Supposons que T soit Γ -uni. Soit $X \in \text{dom } \Gamma \cap [T]$. Alors, pour connaître le n -ième bit de $\Gamma(X)$, il suffit de chercher $\sigma \in \text{Im } T$ tel que $\Gamma(\sigma, n) \downarrow = i$ pour $i \in \{0, 1\}$. Le n -ième bit de $\Gamma(X)$ est alors égal à i .

- (2) Supposons que T soit Γ -scindé, et soit $X \in \text{dom } \Gamma \cap [T]$. Soit $Y = \Gamma(X)$. Pour calculer X à partir de Y , on procède comme suit : comme $T(0)$ et $T(1)$ forment une Γ -scission, il existe $i \in \{0, 1\}$ tel que $\Gamma(T(i), n)$ soit différent de $Y(n)$, pour un certain n . On peut trouver i de manière calculable en Y . Le bon préfixe de X est alors nécessairement $T(1 - i)$. Supposons que l'on ait calculé un préfixe $\sigma \prec X$ et une chaîne τ telle que $\sigma = T(\tau)$. Comme $T(\tau 0)$ et $T(\tau 1)$ forment une Γ -scission, il existe $i \in \{0, 1\}$ tel que $\Gamma(T(\tau i), n)$ soit différent de $Y(n)$, pour un certain n . On peut trouver i de manière calculable en Y . Le bon préfixe de X est alors nécessairement $T(\tau(1 - i))$. En procédant de la sorte, on calcule alors des préfixes de X de plus en plus grands à partir de $Y = \Gamma(X)$. ■

Nous avons à présent tous les ingrédients nécessaires pour montrer l'existence de degrés minimaux.

Théorème 1.5 (Spector [215])

Tout ensemble suffisamment générique pour le forcing de Sacks est de degré minimal.

PREUVE. Soit $(\mathbb{P}, \leq)$ le forcing de Sacks calculable. Notons d'abord que, d'après l'exercice 11-3.3, si $G \in 2^{\mathbb{N}}$ est suffisamment générique pour ce forcing, alors il n'est pas calculable.

Soit Γ une fonctionnelle Turing. D'après le lemme 1.3, l'ensemble des conditions $c \in \mathbb{P}$ telles que c est Γ -scindé ou c est Γ -uni est dense. D'après le lemme 1.4, dans le premier cas, pour tout $G \in [c]$, la fonctionnelle Γ est définie sur G et $\Gamma(G) \geq_T G$. Dans le second cas, d'après le lemme 1.4, pour tout $G \in [c]$, si la fonctionnelle Γ est définie sur G , alors $\Gamma(G)$ est calculable.

Donc, si G est suffisamment générique pour le forcing de Sacks, il est de degré minimal. ■

Corollaire 1.6

Il existe une classe parfaite d'ensembles, tous dans des degrés minimaux distincts.

PREUVE. On peut d'abord montrer qu'il existe un arbre parfait de degrés minimaux. Il suffit de procéder comme dans la preuve du théorème 8-5.1, afin de « dupliquer » la construction d'un degré minimal. Avec le formalisme du forcing, soit $(D_n)_{n \in \mathbb{N}}$ une suite d'ensembles denses de conditions du forcing de Sacks, suffisante pour forcer un ensemble à être de degré minimal. On choisit $c_\epsilon \in D_0$, puis pour toute chaîne σ de taille n , en supposant $c_\sigma \in D_n$ défini, on définit $c_{\sigma 0} \leq c_\sigma$ et $c_{\sigma 1} \leq c_\sigma$ de telle manière à ce que $[c_{\sigma 0}] \cap [c_{\sigma 1}] = \emptyset$. On pourra également faire en sorte que le premier nœud branchant de $c_{\sigma i}$ étende strictement le premier nœud branchant de c_σ pour $i \in \{0, 1\}$. L'arbre final est donné par l'ensemble des chaînes τ telles qu'il existe $X \in 2^{\mathbb{N}}$ pour lequel $\tau \in \bigcap_{\sigma \prec X} c_\sigma$.

Une fois que l'on a un arbre parfait ne contenant que des degrés minimaux, on peut se reporter à l'exercice 8-5.4 pour en extraire un sous-arbre parfait ne contenant que des degrés deux à deux incomparables. ■

1.2. Calcul d'un degré minimal

Une analyse fine du niveau d'effectivité nécessaire pour mener à bien le théorème 1.5 montre qu'il existe un degré minimal $\emptyset''$ -calculable. Il est en fait possible d'améliorer considérablement ce résultat, en remarquant que l'utilisation de f-arbres calculables n'est pas absolument nécessaire.

Arbre c. e.

Un f -arbre c. e. est une fonction partielle $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ telle que pour tout σ tel que $T(\sigma) \downarrow$, soit $T(\sigma 0) \uparrow$ et $T(\sigma 1) \uparrow$, soit $T(\sigma 0) \downarrow \succeq T(\sigma)$ et $T(\sigma 1) \downarrow \succeq T(\sigma)$ avec $T(\sigma 0)$ et $T(\sigma 1)$ incomparables.

L'utilisation de f -arbres c. e. Γ -scindés ou Γ -unis permet de faire des constructions de degrés minimaux demandant moins de puissance de calcul. Il s'agit alors de constructions effectives qui ne relèvent plus à proprement parler du forcing. Nous listons sans les démontrer trois résultats importants qui utilisent ce nouveau type d'arbre pour calculer plus facilement des ensembles de degrés minimaux. Par « plus facilement », il faut comprendre « avec peu de puissance de calcul », les preuves étant elles au contraire, plus complexes...

Théorème 1.7 (Sacks [185])

Il existe un degré minimal sous $\emptyset'$.

Notons que, d'après la proposition 7-4.7, le résultat de Sacks implique l'existence d'un degré minimal de degré hyperimmune. Ce résultat a par la suite été amélioré par Yates [235], et indépendamment par Cooper [40].

Théorème 1.8 (Yates [235], Cooper [40])

Soit X un ensemble c. e. non calculable. Alors, X calcule un ensemble G de degré minimal.

Notons qu'un degré minimal ne peut en revanche jamais être c. e. : il s'agit d'une utilisation sophistiquée de la fameuse méthode des priorités, que nous avons vue dans le chapitre 13, et qui permet de montrer que tout ensemble c. e. non calculable A calcule un autre ensemble c. e. non calculable B qui ne le calcule pas A (voir le théorème 5.1 pour le résultat dans toute sa généralité).

Pour finir, signalons que Groszek et Slaman ont montré que tout degré PA pouvait calculer un degré minimal, via le remarquable résultat suivant.

Théorème 1.9 (Groszek et Slaman [78])

Tout degré PA calcule un degré minimal. Plus précisément, il existe une classe Π_1^0 non vide dont tous les membres sont soit de degré minimal, soit calculent un degré c. e. non calculable.

Exercice 1.10. (*) Montrer que toute classe Π_1^0 non vide contient un ensemble de même degré qu'un ensemble c. e. En déduire qu'il n'existe aucune classe Π_1^0 non vide ne contenant que des éléments de degrés minimaux. $\diamond$

La possibilité de construire des degrés minimaux « complexes » a également été très étudiée. Le principal résultat dans cette direction est le suivant.

Théorème 1.11 (Kumabe [125])

Il existe un degré minimal qui est aussi DNC.

Kumabe a d'abord montré l'existence d'un degré minimal et DNC, dans un article très complexe, qui ne fut jamais publié. La preuve fut ensuite retravaillée par Kumabe et Lewis [125], et la présentation en a par la suite été simplifiée par Khan et Miller [111], qui l'ont réécrite sous le formalisme du forcing, via le *forcing avec arbres touffus*. L'exercice suivant montre qu'un degré minimal ne peut en revanche jamais être DNC₂.

Exercice 1.12. (★) Montrer qu'il existe une classe Π_1^0 non vide d'ensembles $X \oplus Y$ tels que X est de degré PA et Y est de degré PA relativement à X . En déduire qu'aucun ensemble PA n'est de degré minimal (le lecteur pourra consulter la proposition 24-2.4 pour une itération de ce résultat). $\diamond$

1.3. Relativisation : couverture minimale

La construction d'un degré minimal avec le forcing sur les f-arbres se relativise à tout degré Turing dans le sens suivant.

Définition 1.13. Soient $\mathbf{a}$ et $\mathbf{b}$ des degrés Turing. On dit que $\mathbf{b}$ est une *couverture minimale* de $\mathbf{a}$ si $\mathbf{b} > \mathbf{a}$ et s'il n'existe pas de degré $\mathbf{c}$ tel que $\mathbf{a} < \mathbf{c} < \mathbf{b}$. $\diamond$

Autrement dit, $\mathbf{b}$ est une couverture minimale de $\mathbf{a}$ si c'est un élément minimal dans le cône des degrés Turing strictement au-dessus de $\mathbf{a}$. La relativisation du théorème 1.5 montre le théorème suivant.

Théorème 1.14

Tout degré Turing a une couverture minimale.

PREUVE. Soit $A \subseteq \mathbb{N}$ un ensemble quelconque. L'objectif est de construire un ensemble $B >_T A$ tel que tout ensemble C calculable par B est soit A -calculable, soit tel que $A \oplus C \geq_T B$. Ainsi, si $B \geq_T C >_T A$, on aura bien $C \geq_T B$.

Il suffit de considérer une variante du forcing de Sacks, pour lequel nos f-arbres sont cette fois-ci A -calculables, et tels que chacun de leurs chemins calcule A . On pourra se restreindre par exemple aux f-arbres A -calculables tels que A est encodé dans les bits pairs de chaque chemin du f-arbre.

Il suffit alors de répéter la preuve du théorème 1.5 avec ce nouvel ordre partiel, en remarquant la différence suivante : étant donné T un f -arbre Γ -scindé, pour tout $X \in [T]$, on a à présent besoin de A pour retrouver X à partir de $\Gamma(X)$: il nous faut en effet la connaissance de T . C'est la raison pour laquelle on aura $A \oplus \Gamma(X) \geq_T X$ et pas $\Gamma(X) \geq_T X$. ■

Notons qu'une couverture minimale $\mathbf{b}$ de $\mathbf{a}$ n'exclut pas l'existence de degrés $\mathbf{c} < \mathbf{b}$ incomparables avec $\mathbf{a}$. Cela nous amène à définir une notion de couverture plus forte.

Définition 1.15. Soient $\mathbf{a}$ et $\mathbf{b}$ deux degrés Turing. On dit que $\mathbf{b}$ est une *couverture minimale forte* de $\mathbf{a}$ si $\mathbf{b} > \mathbf{a}$ et, si pour tout degré Turing $\mathbf{c} < \mathbf{b}$, on a $\mathbf{c} \leq \mathbf{a}$. ◇

Comme déjà noté dans la preuve du théorème 1.14, la version relativisée du théorème 1.5 ne prouve pas l'existence d'une couverture minimale forte pour tout degré Turing, et pour cause : certains degrés n'admettent pas de couverture minimale forte, même si ce sera le cas pour beaucoup d'entre eux.

Ishmukhametov [95] a établi une élégante caractérisation des degrés c. e. admettant une couverture minimale forte.

Théorème 1.16 (Ishmukhametov [95])

Un ensemble c. e. $A \subseteq \mathbb{N}$ admet une couverture minimale forte si, et seulement si, toute fonction A -calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ est bornée pour n suffisamment grand par la fonction $n \mapsto \min \{s \in \mathbb{N} : \emptyset'[s] \upharpoonright_n = \emptyset' \upharpoonright_n\}$.

Une caractérisation générale des degrés admettant une couverture minimale forte est pour le moment inconnue, même si de nombreux résultats partiels ont été établis (voir Lewis [143]).

2. Nature de $\mathcal{D}$

À quoi ressemble l'ordre partiel $(\mathcal{D}, \leq)$? Concernant sa taille d'abord, nous avons vu dans le présent ouvrage plusieurs constructions d'arbres parfaits dont chaque chemin est dans un degré Turing différent (voir par exemple l'exercice 7-5.8, l'exercice 8-5.3 ou l'exercice 8-5.4). Cela nous donne une injection de $2^{\mathbb{N}}$ dans $\mathcal{D}$. À l'aide de l'axiome du choix, on peut choisir un représentant dans chaque degré Turing, ce qui donne une injection de $\mathcal{D}$ dans $2^{\mathbb{N}}$. La cardinalité de $\mathcal{D}$ est donc $|2^{\mathbb{N}}|$, celle de $2^{\mathbb{N}}$. Notons que l'on ne peut pas nécessairement choisir un représentant dans chaque degré Turing

si l'on ne dispose pas de l'axiome du choix, et il n'en reste pas moins que « moralement » $|2^{\mathbb{N}}|$ est la cardinalité de $\mathcal{D}$.

Étant donné un élément $\mathbf{a} \in \mathcal{D}$, l'ensemble des éléments sous $\mathbf{a}$ est au plus dénombrable puisqu'un ensemble ne peut calculer qu'une quantité dénombrable d'éléments. La cardinalité des éléments au-dessus de $\mathbf{a}$ est en revanche celle de $2^{\mathbb{N}}$: étant donné $A \in \mathbf{a}$, on peut facilement créer un arbre parfait dont les chemins sont tous de la forme $A \oplus X$ pour $X \in 2^{\mathbb{N}}$, et tous dans des degrés Turing différents.

L'utilisation de la jointure Turing nous amène à la considération suivante : étant donné deux ensembles A et B , l'ensemble $A \oplus B$ calcule à la fois A et B , et tout ensemble calculant à la fois A et B calcule $A \oplus B$. En termes de degrés, cela implique que toute paire de degrés $\mathbf{a}, \mathbf{b}$ possède une borne supérieure. Il y a donc un minimum de structure dans cet ordre partiel, pour lequel nous introduisons le concept suivant.

Définition 2.1. Un *treillis* est un ensemble partiellement ordonné dans lequel toute paire d'éléments a, b admet une borne supérieure, notée $a \cup b$, et une borne inférieure, notée $a \cap b$. Un *demi-treillis supérieur* (resp. inférieur) est un ensemble ordonné pour lequel toute paire d'éléments admet une borne supérieure (resp. inférieure). $\diamond$

Le paragraphe qui précède cette définition conduit au théorème ci-après, lequel figure dans l'article fondateur de l'étude de la structure des degrés Turing.

Théorème 2.2 (Kleene et Post [118])

L'ordre partiel $(\mathcal{D}, \leq)$ est un demi-treillis supérieur de cardinalité $|2^{\mathbb{N}}|$, avec un plus petit mais pas de plus grand élément, tel que chaque élément admet un ensemble au plus dénombrable d'éléments en dessous de lui et un ensemble de cardinalité $|2^{\mathbb{N}}|$ d'éléments au-dessus de lui.

On calque parfois le vocabulaire des degrés Turing sur celui des ensembles qu'ils contiennent : étant donné deux degrés $\mathbf{a}$ et $\mathbf{b}$, on dira que la borne supérieure $\mathbf{a} \cup \mathbf{b}$ de $\mathbf{a}$ et $\mathbf{b}$ est la *jointure* de $\mathbf{a}, \mathbf{b}$. Notons que dans un demi-treillis supérieur, toute suite finie d'éléments admet également une borne supérieure. En particulier, pour un ensemble fini de degrés $\mathbf{a}_1, \dots, \mathbf{a}_n$, on la notera $\mathbf{a}_1 \cup \dots \cup \mathbf{a}_n$, et elle sera le degré de la jointure $A_1 \oplus \dots \oplus A_n$, pour des représentants quelconques $A_i \in \mathbf{a}_i$.

Que se passe-t-il pour les ensembles dénombrables de degrés ? Le théorème suivant implique que si un tel ensemble est clos par jointure — c'est-à-dire que $\mathbf{a} \cup \mathbf{b}$ est dans notre ensemble pour tous $\mathbf{a}, \mathbf{b}$ dans notre ensemble — et n'a pas d'élément maximal, alors il n'a jamais de borne supérieure.

Théorème 2.3 (Sacks [188])

Tout ensemble dénombrable de degrés clos par jointure et sans élément maximal admet des majorants minimaux, en quantité $|2^{\mathbb{N}}|$.

IDÉE DE LA PREUVE. Notons d'abord qu'un majorant d'un ensemble dénombrable de degrés $(\mathbf{a}_n)_{n \in \mathbb{N}}$, est aussi majorant de

$$\mathbf{a}_0 \leq \mathbf{a}_0 \cup \mathbf{a}_1 \leq \mathbf{a}_0 \cup \mathbf{a}_1 \cup \mathbf{a}_2 \leq \dots$$

Comme $(\mathbf{a}_n)_{n \in \mathbb{N}}$ n'a pas d'élément maximal et qu'il est clos par jointure, on peut donc considérer sans perte de généralité que notre ensemble de degrés est tel que $\mathbf{a}_n < \mathbf{a}_{n+1}$. Pour tout n , soit A_n un représentant de $\mathbf{a}_n$.

Il suffit à présent d'élaborer sur le forcing de Sacks (voir la section 1) permettant de créer la couverture minimale d'un degré. On commence avec un f -arbre A_0 -calculable dont tous les chemins calculent A_0 . Une condition de forcing sera un f -arbre A_n -calculable dont tous les chemins calculent A_n (pour un certain n). On étend une telle condition de forcing $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ au f -arbre Q tel que $\text{Im } Q \subseteq \text{Im } T$ consiste en les chemins qui encodent $A_{n+1} \oplus X$ pour tout $X \in 2^{\mathbb{N}}$. Formellement, pour toute chaîne σ et tout $i \in \{0, 1\}$, $Q(\sigma i) = T(A_{n+1} \upharpoonright_{|\sigma i|} \oplus \sigma i)$. Comme $A_{n+1} \geq_T A_n$, alors A_{n+1} peut calculer T , et donc retrouver $A_{n+1} \oplus X$ à partir du chemin de Q qui encode $A_{n+1} \oplus X$ dans T .

La manière de faire une couverture minimale ne change pas, et la technique décrite dans la section 1 s'applique de la même manière. L'ensemble générique G obtenu calculera chaque ensemble A_n , et sera tel que tout ce qui est calculé par G et qui peut calculer chaque ensemble A_n peut aussi calculer G .

Pour obtenir des majorants minimaux en quantité $|2^{\mathbb{N}}|$, on peut construire un arbre parfait de majorants minimaux en subdivisant la construction en deux, puis chaque sous-construction en deux, etc., comme dans la preuve du théorème 8-5.1. ■

Un ensemble dénombrable de degrés clos par jointure et sans élément maximal a donc toujours deux majorants minimaux distincts, et l'on peut donc en déduire le corollaire suivant.

Corollaire 2.4

Un ensemble dénombrable de degrés clos par jointure et sans élément maximal n'a jamais de borne supérieure.

Notons que pour une suite d'ensembles $(A_n)_{n \in \mathbb{N}}$, l'ensemble $\bigoplus_{n \in \mathbb{N}} A_n$ (voir la définition 10-3.24) n'est en général par un majorant minimal, comme en témoigne l'élégant résultat suivant.

Théorème 2.5 (Enderton et Putnam [56], Sacks [190])

Il existe un majorant minimal de $(\emptyset^{(n)})_{n \in \mathbb{N}}$ dont le double saut est dans le même degré Turing que celui de $\bigoplus_n \emptyset^{(n)}$.

IDÉE DE LA PREUVE. Pour une direction, il suffit de remarquer que le double saut de tout majorant de $(\emptyset^{(n)})_{n \in \mathbb{N}}$ permet de calculer $\bigoplus_n \emptyset^{(n)}$. Soit B un majorant et soit f une fonction calculable telle que $f(X') = X$ pour toute X (voir l'exercice 4-6.4 pour plus de détails sur une telle fonction). À l'aide du double saut du majorant B , on cherche un code de fonctionnelle e_1 tel que $f(\Phi_{e_1}(B)) = \emptyset$, puis un code de fonctionnelle e_2 tel que $f(\Phi_{e_2}(B)) = \Phi_{e_1}(B)$, etc.

Pour l'autre direction, il suffit de voir que la construction d'une borne supérieure de $(\emptyset^{(n)})_{n \in \mathbb{N}}$ est effective à l'aide de $\bigoplus_n \emptyset^{(n)}$, et force à chaque étape une fonctionnelle Φ_e à être partielle ou bien totale sur tous les éléments de l'arbre considéré : on ne fait donc pas que calculer le générique résultant G , mais on peut aussi déterminer l'ensemble des codes de fonctionnelles totales sur G . Le double saut de G se réduit à cet ensemble (voir l'exercice 5-7.2). ■

Nous répondons pour finir cette section à la question qui taraude peut-être le lecteur depuis le début de ce chapitre : la structure $(\mathcal{D}, \leq)$ est-elle un treillis ? Nous allons voir que non, et nous introduisons pour cela la notion de paire exacte.

Définition 2.6. Les degrés $\mathbf{a}, \mathbf{b}$ forment une *paire exacte* pour un ensemble de degrés $C \subseteq \mathcal{D}$ si $\mathbf{a}$ et $\mathbf{b}$ bornent chacun tous les degrés de C , et si chaque degré à la fois sous $\mathbf{a}$ et $\mathbf{b}$ est aussi borné par un degré de C . ◇

Théorème 2.7

Tout ensemble dénombrable de degrés C clos par jointure admet une paire exacte.

PREUVE. Soit $(\mathbf{a}_n)_{n \in \mathbb{N}}$ un ensemble de degrés Turing clos par jointure. Pour tout n , soit A_n un représentant de $\mathbf{a}_n$. L'idée est de construire deux ensembles $G_0 = \bigoplus_n X_n^0$ et $G_1 = \bigoplus_n X_n^1$ tels que chaque colonne X_n^i pour $i \in \{0, 1\}$ est égale à l'ensemble A_n , sauf pour un nombre fini de bits. Il est clair que de tels ensembles G_0, G_1 permettent de calculer tous les $\mathbf{a}_n$. Il faut à présent les construire via un forcing adapté de telle manière à ce que si G_0 et G_1 calculent le même ensemble, alors cet ensemble est calculable par $A_0 \oplus A_1 \oplus \dots \oplus A_m$ pour un certain m .

Nos conditions de forcing sont constituées de triplets

$$(\sigma_0, \sigma_1, n), \text{ où } \sigma_0, \sigma_1 \in 2^{<\mathbb{N}} \text{ et } n \in \mathbb{N}.$$

Le paramètre n sert à contrôler les extensions possibles de nos conditions. On a $(\sigma_0, \sigma_1, n) \succeq (\tau_0, \tau_1, m)$ pour deux conditions de forcing si $\sigma_0 \preceq \tau_0$, si $\sigma_1 \preceq \tau_1$, et si $n \leq m$ avec la restriction que $\forall \langle k, a \rangle$ tel que $|\sigma_i| \leq \langle k, a \rangle < |\tau_i|$ pour $k \leq n$, on doit avoir $\tau_i(\langle k, a \rangle) = A_k(a)$. Étant donné un ensemble de conditions $(\sigma_0^0, \sigma_1^0, n_0) \succeq (\sigma_0^1, \sigma_1^1, n_1) \succeq (\sigma_0^2, \sigma_1^2, n_2) \succeq \dots$, le générique G_0 sera le point limite de $\sigma_0^0 \preceq \sigma_0^1 \preceq \sigma_0^2 \preceq \dots$ et de générique G_1 sera le point limite de $\sigma_1^0 \preceq \sigma_1^1 \preceq \sigma_1^2 \preceq \dots$. Notons que la restriction sur les extensions possibles garantit que tant que la suite $n_0 \leq n_1 \leq n_2 \leq \dots$ est non bornée, chaque n -ième colonne de G_i sera bien égale à A_n , sauf pour un nombre fini de bits.

Pour toute paire de fonctionnelles Φ_{e_0}, Φ_{e_1} , on va forcer

$$\Phi_{e_0}(G_0) = \Phi_{e_1}(G_1) = X \implies X \leq_T A_0 \oplus \dots \oplus A_n, \text{ pour un certain } n.$$

Notons que si G_0 et G_1 bornent le même degré il existe nécessairement deux fonctionnelles telles que G_0 et G_1 calculent le même ensemble dans ce degré. Une telle construction atteint donc bien nos objectifs.

On se donne une condition de forcing (σ_0, σ_1, n) et deux fonctionnelles Φ_{e_0} et Φ_{e_1} . On cherche deux chaînes τ_0, τ_1 telles que (τ_0, τ_1, n) est une extension valide de (σ_0, σ_1, n) , et telles que $\Phi_{e_0}(\tau_0, x) \downarrow \neq \Phi_{e_1}(\tau_1, x) \downarrow$ pour un certain x . Si cette recherche aboutit, on prend alors (τ_0, τ_1, n) comme extension de (σ_0, σ_1, n) . Sinon, cela signifie que sur tout x , les calculs $\Phi_{e_0}(\tau_0, x)$ et $\Phi_{e_1}(\tau_1, x)$, quand ils s'arrêtent, renvoient la même valeur pour toute extension valide $(\tau_0, \tau_1, n) \preceq (\sigma_0, \sigma_1, n)$.

Notons que par définition de ce qu'est une extension valide de (σ_0, σ_1, n) , il est possible de les énumérer à l'aide de $A_0 \oplus \dots \oplus A_n$. Si pour tout x il existe une extension valide (τ_0, τ_1, n) telle que $\Phi_{e_0}(\tau_0, x) \downarrow$ ou $\Phi_{e_1}(\tau_1, x) \downarrow$ alors on peut calculer via $A_0 \oplus \dots \oplus A_n$ l'unique élément Z ainsi potentiellement calculable par n'importe quel générique G_0 via Φ_{e_0} ou par n'importe quel générique G_1 via Φ_{e_1} . Sinon, au moins un des deux calculs $\Phi_{e_0}(G_0, x)$ ou $\Phi_{e_1}(G_1, x)$ sera partiel sur un certain x .

Si donc G_0, G_1 sont suffisamment génériques pour ce forcing, pour toutes fonctionnelles Φ_{e_0}, Φ_{e_1} , on aura

$$\Phi_{e_0}(G_0) = \Phi_{e_1}(G_1) = X \implies X \leq_T A_0 \oplus \dots \oplus A_n, \text{ pour un certain } n.$$

Il s'ensuit que G_0, G_1 est une paire exacte pour les degrés $(\mathbf{a}_n)_{n \in \mathbb{N}}$. ■

On peut à présent en déduire que $(\mathcal{D}, \leq)$ n'est pas un treillis.

Théorème 2.8 (Kleene et Post [118])

Le demi-treillis supérieur $\mathcal{D}$ n'est pas un treillis : il y a des degrés $\mathbf{a}, \mathbf{b}$ qui n'ont pas de borne inférieure.

PREUVE. Il suffit de considérer un ensemble de degrés $\mathbf{c}_0 < \mathbf{c}_1 < \mathbf{c}_2 < \dots$ nécessairement clos par jointure. Cet ensemble de degrés admet donc une paire exacte $\mathbf{a}, \mathbf{b}$. Une telle paire exacte n'a pas de borne inférieure puisque tout degré à la fois sous $\mathbf{a}$ et $\mathbf{b}$ est également sous un degré $\mathbf{c}_n$, pour un certain n . ■

3. Universalité de $\mathcal{D}$

Nous voyons dans cette section que $(\mathcal{D}, \leq)$ présente une certaine universalité, en ce sens que *tous les ordres partiels* peuvent *se plonger* dans $\mathcal{D}$, sauf ceux qui ne peuvent y prétendre pour des raisons de cardinalité.

Définition 3.1. Un ordre partiel $(A, \leq)$ se *plonge* dans $(\mathcal{D}, \leq)$ s'il existe une injection $f : A \rightarrow \mathcal{D}$ telle que $a \leq b$ ssi $f(a) \leq f(b)$. ◇

La structure $(\mathcal{D}, \leq)$ contient donc en elle tous les ordres partiels qui ne sont pas plus gros qu'elle. Cette affirmation, qui sera rendue précise, est en fait un peu fausse : il reste une question ouverte à ce sujet, que nous mentionnerons bientôt. Notons que cela ne nous renseigne pas nécessairement sur la complexité calculatoire de $\mathcal{D}$. Considérons par exemple l'ordre calculable partiel $\leq_R \subseteq \mathbb{Q}^2 \times \mathbb{Q}^2$ défini par $(p_1, p_2) \leq_R (q_1, q_2)$ si $p_1 \leq q_1$ et $p_2 \leq q_2$ pour $p_1, p_2, q_1, q_2 \in \mathbb{Q}$. Il n'est pas très difficile de montrer que tout ordre partiel dénombrable se plonge dans $(\mathbb{Q}^2, \leq_R)$. La construction d'un plongement se fait sans difficulté, en construisant l'injection de manière gloutonne élément par élément sans jamais violer à chaque étape finie les contraintes d'un plongement. La structure $(\mathbb{Q}^2, \leq_R)$ n'est pas calculatoirement complexe, mais elle est suffisamment riche en termes de possibilités pour contenir tous les ordres partiels.

Nous utiliserons à plusieurs reprises l'existence d'ensembles dit *calculatoirement indépendants*.

Définition 3.2. Des ensembles $(X_n)_{n \in \mathbb{N}}$ sont *calculatoirement indépendants* si $X_i \not\leq_T \bigoplus_{j \neq i} X_j$ pour tout $i \in \mathbb{N}$. ◇

L'existence d'ensembles calculatoirement indépendants ne présente pas de difficultés particulières et l'on peut se référer à l'exercice 10-3.25 pour voir que si $G = \bigoplus_n G_n$ est un ensemble 1-générique, alors les ensembles $(G_n)_{n \in \mathbb{N}}$ sont calculatoirement indépendants.

3.1. Plongements dans $\mathcal{D}$

Voyons tout de suite ce qui fut annoncé, sous la forme d'un premier théorème.

Théorème 3.3 (Sacks [188])

Tout ordre partiel dénombrable se plonge dans les degrés Turing.

PREUVE. Nous avons vu dans l'introduction de cette section que tout ordre partiel dénombrable peut se plonger dans la structure $(\mathbb{Q}^2, \leq_R)$ définie par $(p_1, p_2) \leq_R (q_1, q_2)$ si $p_1 \leq q_1$ et $p_2 \leq q_2$.

Il suffit alors de montrer que $(\mathbb{Q}^2, \leq_R)$ se plonge dans $(\mathcal{D}, \leq)$. Soit $(X_n)_{n \in \mathbb{N}}$ une suite d'ensembles calculatoirement indépendants, et soit $(a_n)_{n \in \mathbb{N}}$ une énumération calculable des éléments de $\mathbb{Q}^2$. Le plongement f assigne à l'élément a_n le degré Turing de l'ensemble $\bigoplus_{a_m \leq_R a_n} X_m$. On vérifie sans peine $a_n \leq_R a_m$ ssi $f(a_n) \leq f(a_m)$. ■

Sacks a par la suite cherché à étendre son résultat à de plus gros ordres partiels. Après tout, $(\mathcal{D}, \leq)$ admet pour cardinalité $2^{\aleph_1}$. On ne peut bien sûr pas attendre de tout ordre partiel de cardinalité $2^{\aleph_1}$ qu'il se plonge dans $(\mathcal{D}, \leq)$: si un élément dans un ordre partiel a une quantité indénombrable de prédécesseurs, il n'y a aucun espoir de construire un plongement de cet ordre vers $\mathcal{D}$ puisque chaque élément de $\mathcal{D}$ n'en a qu'une quantité dénombrable. On doit donc respecter cette restriction, mais y en a-t-il d'autres ? Nous avons besoin ici d'anticiper un peu sur les ordinaux qui seront introduits dans le chapitre 27, et en particulier sur l'ordinal ω_1 , le plus petit ordinal infini non dénombrable. Sacks a obtenu les résultats suivants.

Théorème 3.4 (Sacks [186])

N'importe quel ordre partiel avec une des propriétés suivantes peut se plonger dans les degrés Turing :

1. *l'ordre est de cardinalité $2^{\aleph_1}$, et chaque élément a une quantité finie de prédécesseurs ;*
2. *l'ordre est de cardinalité $|\omega_1|$, et chaque élément a une quantité au plus dénombrable de prédécesseurs ;*
3. *l'ordre est de cardinalité $2^{\aleph_1}$, et chaque élément a une quantité au plus dénombrable de prédécesseurs, ainsi qu'une quantité d'au plus ω_1 successeurs.*

En particulier, si l'on fait l'hypothèse du continu, à savoir $|\omega_1| = 2^{\aleph_1}$, le théorème de Sacks est optimal : tout ordre partiel de cardinalité $2^{\aleph_1}$, et où chaque élément a une quantité au plus dénombrable de prédécesseurs, peut se plonger dans les degrés Turing. Mais, si l'on n'utilise pas l'hypothèse du continu, la question est toujours ouverte.

Question 3.5. Peut-on plonger dans $(\mathcal{D}, \leq)$ tout ordre partiel de cardinalité $2^{\aleph_1}$, où chaque élément a une quantité dénombrable de prédécesseurs ? ★

Si nous ne présentons pas ici la preuve de Sacks, nous en voyons néanmoins deux ingrédients, via la notion de chaîne et d'anti-chaîne.

Définition 3.6. Un ensemble de degrés Turing linéairement ordonné est une *chaîne*. Un ensemble de degrés Turing deux à deux incomparables est une *anti-chaîne*. $\diamond$

Proposition 3.7 (Sacks [188]).

- (1) Chaque chaîne dénombrable peut être étendue dans les degrés Turing. En particulier, toute chaîne maximale est de cardinalité ω_1 .
- (2) Chaque anti-chaîne de cardinalité inférieure à $|2^{\mathbb{N}}|$ peut être étendue dans les degrés Turing. En particulier, toute anti-chaîne maximale est de cardinalité $|2^{\mathbb{N}}|$. $\star$

PREUVE.

- (1) Si la chaîne possède un plus grand élément, on peut considérer son saut Turing. Sinon, on peut considérer le degré de la jointure Turing d'un représentant de chacun de ses éléments.
- (2) Soit D l'ensemble des degrés minimaux. Par le corollaire 1.6, D est de cardinalité $|2^{\mathbb{N}}|$. Soit C une anti-chaîne de degrés Turing de cardinalité inférieure à $|2^{\mathbb{N}}|$. Chaque élément de C a une quantité au plus dénombrable d'éléments en dessous de lui. Ainsi, la clôture par le bas de C a la même cardinalité que C . Il doit donc exister un élément de $\mathbf{d} \in D$ qui n'est calculé par aucun élément de C . Comme $\mathbf{d}$ est un degré minimal, il ne peut borner aucun élément de C . On en déduit que $C \cup \{\mathbf{d}\}$ est une anti-chaîne. $\blacksquare$

3.2. Extension de plongements de $\mathcal{D}$

La notion de plongement peut être considérée comme faible, en particulier car elle ne dit rien sur les relations qu'entretiennent les degrés dans l'image d'un plongement, avec les degrés qui ne sont pas dans cette image. Une manière de pallier cette faiblesse est de considérer un plongement déjà existant d'une structure $(C, \leq)$ vers les degrés Turing, et d'essayer de voir dans quelle mesure ce plongement peut se prolonger à une extension de l'ordre partiel sur C . Une telle chose n'est bien entendu pas toujours possible : si des éléments $a_0, a_1, b \in C$, avec $a_0 < b, a_1 < b$ et a_0, a_1 incomparables, sont envoyés sur des degrés $\mathbf{a}_0, \mathbf{a}_1$ et $\mathbf{a}_0 \cup \mathbf{a}_1$, alors un tel plongement ne pourra se prolonger à aucun majorant $c < b$ de a_0, a_1 . On introduit pour cela la notion d'extension consistante.

Définition 3.8. Soit C un demi-treillis supérieur et soit $D \supseteq C$. Alors, D est une *extension consistante* de C si :

- (1) pour $a, b < d$ avec $a, b \in C$ et $d \in D \setminus C$, on a $a \cup b < d$;
- (2) aucun élément de $D \setminus C$ n'est sous un élément de C .

Notons que, comme C est un demi-treillis supérieur, $a \cup b \in C$ pour tous $a, b \in C$. ♦

Théorème 3.9 (Kleene et Post [118])

Soit C un demi-treillis supérieur fini, et soit D une extension consistante finie de C . Alors, tout plongement f de $(C, \leq)$ dans $(\mathcal{D}, \leq)$ peut être étendu en un plongement de $(D, \leq)$ dans $(\mathcal{D}, \leq)$.

PREUVE. Soit f un plongement de $(C, \leq)$ dans $(\mathcal{D}, \leq)$. On notera $\mathbf{a}$ l'image de $a \in C$ par f . Soit $a \in D \setminus C$ minimal dans $D \setminus C$. Comme D est une extension consistante, alors C peut être partitionné en une liste d'éléments $(b_i)_{i \leq n}$ et $(c_i)_{i \leq m}$ telle que $b_0 \cup \dots \cup b_n < a$, telle que a est incomparable avec chaque c_i et telle que $b_0 \cup \dots \cup b_n$ n'est au-dessus d'aucun c_i . Il suffit alors de construire un degré Turing $\mathbf{a}$ tel que $\mathbf{b}_0 \cup \dots \cup \mathbf{b}_n < \mathbf{a}$ et tel que $\mathbf{a}$ soit incomparable avec chaque $\mathbf{c}_i$.

En utilisant le fait que $\mathbf{b}_0 \cup \dots \cup \mathbf{b}_n$ n'est au-dessus d'aucun $\mathbf{c}_i$, on construit facilement par extensions finies un degré $\mathbf{d}$ tel que $\mathbf{d} \cup \mathbf{b}_0 \cup \dots \cup \mathbf{b}_n$ ne soit au-dessus d'aucun $\mathbf{c}_i$ et tel que $\mathbf{b}_0 \cup \dots \cup \mathbf{b}_n$ ne soit pas au-dessus de $\mathbf{d}$. Le plongement se prolonge alors en envoyant a sur $\mathbf{d} \cup \mathbf{b}_0 \cup \dots \cup \mathbf{b}_n$. Par minimalité du choix de a , l'ensemble $D - \{a\}$ est à présent une extension consistante de la clôture de $C \cup \{a\}$ en demi-treillis supérieur. On peut donc recommencer jusqu'à assignation de chaque élément de D . ■

Nous verrons avec le lemme 4.2 que la réciproque du théorème fonctionne : il est nécessaire d'être une extension consistante pour que tout plongement soit extensible. Le théorème précédent peut être étendu comme suit.

Théorème 3.10 (Sacks [186])

Soit C un demi-treillis supérieur dénombrable, et soit f un plongement de $(C, \leq)$ dans $(\mathcal{D}, \leq)$. Soit D une extension consistante et dénombrable de C . Alors, f peut être étendu en un plongement de $(D, \leq)$ dans $(\mathcal{D}, \leq)$.

3.3. Segments initiaux de $\mathcal{D}$

Une autre manière de renforcer l'étude des plongements possibles, est de considérer les plongements sur des *segments initiaux* de $\mathcal{D}$

Définition 3.11. Un *segment initial* de $\mathcal{D}$ est un ensemble de degrés clos par le bas. Un *segment final* est un ensemble de degrés clos par le haut. $\diamond$

Un plongement sur un segment initial nous donne une information complète sur tous les degrés qui se trouvent en dessous de ceux de l'image du plongement. Par exemple, la construction d'un degré minimal nous indique que l'ordre $a < b$ de deux éléments a, b peut se plonger sur un segment initial de $\mathcal{D}$. L'existence d'un degré minimal avec une couverture minimale forte (voir la définition 1.15) nous indique que l'ordre $a < b < c$ de trois éléments a, b, c peut se plonger sur un segment initial de $\mathcal{D}$. En élaborant sur la construction des degrés minimaux, Lachlan et Lebeuf ont obtenu le remarquable résultat suivant.

Théorème 3.12 (Lachlan et Lebeuf [133])

Un ordre partiel dénombrable se plonge sur un segment initial de $(\mathcal{D}, \leq)$ si, et seulement si, c'est un demi-treillis supérieur avec un plus petit élément.

La preuve de Lachlan et Lebeuf se fait petit à petit, l'étape la plus difficile étant de le montrer pour tout demi-treillis supérieur fini avec un plus petit élément. Il s'agit d'une modification non triviale de la construction de degrés minimaux. Considérons par exemple l'ordre partiel en diamant donné par $a \leq b_1$, $a \leq b_2$, $b_1, b_2 \leq c$ et b_1, b_2 incomparables. Il s'agit alors de construire deux degrés minimaux $\mathbf{b}_1, \mathbf{b}_2$, tels que $\mathbf{b}_1 \cup \mathbf{b}_2 = \mathbf{c}$, et tels que $\mathbf{b}_1, \mathbf{b}_2$ sont *les seuls degrés non calculables* se trouvant sous $\mathbf{c}$. Une telle construction repose sur un forcing avec des f-arbres calculables dits *uniformes*, comme expliqué dans la section 1.1. On peut par exemple construire avec un tel forcing un ensemble $X = X_0 \oplus X_1$ tel que X_0 et X_1 sont incomparables et minimaux, et tel que tout ce qui est calculé par X est soit sous X_0 , soit sous X_1 , soit peut recalculer X . La preuve détaillée peut être consultée dans [169] ou [138].

Notons que le théorème de Lachlan et Lebeuf donne une caractérisation complète des idéaux Turing de la forme $\mathcal{D}(\leq \mathbf{a})$ pour un certain degré $\mathbf{a}$ (i.e. l'ensemble des éléments qui se trouvent sous $\mathbf{a}$) : il s'agit des demi-treillis supérieurs au plus dénombrables ayant un plus petit et un plus grand élément. En fait, ce théorème peut être poussé un peu plus loin.

Théorème 3.13 (Abraham et Shore [2])

Un ordre partiel de cardinalité $|\omega_1|$ est isomorphe à un segment initial de $(\mathcal{D}, \leq)$ ssi c'est un demi-treillis supérieur avec un plus petit élément, et dans lequel chaque élément a une quantité au plus dénombrable de prédécesseurs.

En particulier, si l'on fait l'hypothèse du continu, cela caractérise complètement les segments initiaux possibles de $(\mathcal{D}, \leq)$. Si l'on ne fait pas l'hypothèse du continu, alors les choses se compliquent.

Théorème 3.14 (Groszek et Slaman [77])

Il existe un modèle de ZFC, dans lequel il existe un ordre partiel indénombrable avec un plus petit élément et pour lequel chaque élément a un nombre fini de prédécesseurs, qui ne peut se plonger comme segment initial de $(\mathcal{D}, \leq)$.

4. Théorie du premier ordre de $\mathcal{D}$

Nous abordons à présent la complexité de $\mathcal{D}$. La question qui nous intéresse est la suivante : étant donné un énoncé du premier ordre qui porte sur les degrés Turing, peut-on décider si ce dernier est vérifié ou non ? Le langage que nous pouvons utiliser est uniquement constitué des relations $\leq$, $<$ ou $=$, mais il sera possible d'étendre ce langage à tout ce qui est définissable avec $\leq$, $<$ ou $=$. Par exemple $\mathbf{0}$, le degré minimal, est définissable comme étant l'unique degré qui satisfait la formule $F(\mathbf{x}) = \forall \mathbf{y} \mathbf{x} \leq \mathbf{y}$. On peut alors par exemple exprimer l'existence d'un degré minimal de la manière suivante : $\exists \mathbf{x} (\mathbf{0} < \mathbf{x} \wedge \forall \mathbf{y} \leq \mathbf{x} (\mathbf{y} = \mathbf{0} \vee \mathbf{y} = \mathbf{x}))$. Le fait que $\mathbf{0}$ soit définissable par la formule $F(\mathbf{x})$ permet de s'en passer dans une formule équivalente :

$$\exists \mathbf{z} (F(\mathbf{z}) \wedge \exists \mathbf{x} (\mathbf{z} < \mathbf{x} \wedge \forall \mathbf{y} \leq \mathbf{x} (\mathbf{y} = \mathbf{z} \vee \mathbf{y} = \mathbf{x}))).$$

C'est bien entendu plus long, et l'on s'autorisera donc ces extensions de langage. On utilisera notamment la fonction à deux arguments $\cup$ qui nous donne la jointure de deux degrés, et qui est elle aussi définissable par la formule

$$F(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{x} \leq \mathbf{z} \wedge \mathbf{y} \leq \mathbf{z} \wedge \forall \mathbf{a} ((\mathbf{x} \leq \mathbf{a} \wedge \mathbf{y} \leq \mathbf{a}) \rightarrow \mathbf{z} \leq \mathbf{a}).$$

On vérifie sans peine que pour tout $\mathbf{a}, \mathbf{b} \in \mathcal{D}$, le degré $\mathbf{a} \cup \mathbf{b}$ est l'unique degré $\mathbf{z}$ tel que $F(\mathbf{a}, \mathbf{b}, \mathbf{z})$.

Revenons à présent à notre question : étant donné un énoncé sur les degrés Turing, peut-on décider si ce dernier est vrai ou faux ? La question est *a priori* d'une grande complexité si on l'aborde directement : un énoncé du type $\exists \mathbf{a}$ revient à un énoncé du type « existe-t-il un ensemble X tel que ». Il s'agit d'une quantification du second ordre, c'est-à-dire portant non pas sur les entiers, mais sur les ensembles d'entiers. Ce genre de quantification sera abordé en détail dans la partie IV. Les énoncés avec des quantifications sur les entiers étant déjà indécidables, il y a fort à parier que ce soit aussi le cas pour les énoncés sur les degrés Turing. On peut néanmoins montrer que si la formule est du type $\forall \exists$ ou du type $\exists \forall$, on peut alors décider si elle est vraie ou fausse. Cela constitue notre premier théorème sur le sujet.

Théorème 4.1 (Lerman [138] (théorème 4.4) et Shore [199])

La théorie Π_2^0 de $(\mathcal{D}, \leq)$, c'est-à-dire l'ensemble des énoncés Π_2^0 vrais dans $(\mathcal{D}, \leq)$, est décidable.

Pour montrer le théorème précédent, il suffit essentiellement de voir que la réciproque du théorème 3.9 est vraie.

Lemme 4.2. Soit C un demi-treillis supérieur fini, et soit D une extension finie de C . Alors, D est une extension consistante si, et seulement si, tout plongement f de $(C, \leq)$ dans $(\mathcal{D}, \leq)$ peut être étendu en un plongement de $(D, \leq)$ dans $(\mathcal{D}, \leq)$. ★

PREUVE. Le théorème 3.9 nous donne une direction du lemme. Supposons à présent que D ne soit pas une extension consistante de C . Si un élément de $d \in D$ est tel que $a, b \leq d$ mais $a \cup b \not\leq d$, il suffit de considérer un plongement qui associe $\mathbf{a} \cup \mathbf{b}$ à $a \cup b$. Il sera alors impossible d'étendre un tel plongement à D . Supposons à présent qu'un degré de D se trouve sous un degré de C . Par le théorème 3.12, il existe un plongement de C sur un segment initial de $\mathcal{D}$. Là encore, un tel plongement ne pourra pas être étendu à un degré inférieur à un degré de C . ■

Nous avons à présent l'ingrédient nécessaire pour montrer le théorème 4.1 :

PREUVE DU THÉORÈME 4.1. Soit un énoncé de la forme

$$\forall c_1, \dots, \forall c_n \exists d_1, \dots, d_m F(c_1, \dots, c_n, d_1, \dots, d_m),$$

où F est une combinaison booléenne de formules atomiques. Soit C l'ensemble fini des modèles possibles de demi-treillis supérieurs générés par les c_i et compatibles avec les conditions données par F . Si C est vide, alors la formule n'est pas satisfaite, sans même considérer la partie de F mentionnant les d_i . Sinon, pour tous les modèles de C , on vérifie si ce modèle peut être complété de manière compatible avec les conditions données par F , et de telle manière à ce que les d_i en soient une extension consistante. Si tel est le cas, la formule est vraie, et sinon elle est fausse. ■

Il s'agit là de la limite de ce qui est décidable. Lachlan [130] a montré que la théorie Π_3^0 ne l'était plus.

Théorème 4.3 (Schmerl — voir corollaire 4.6 de [138])

La théorie Π_3^0 de $(\mathcal{D}, \leq)$ est indécidable.

À quel point la théorie de $(\mathcal{D}, \leq)$ est-elle complexe ? Peut-on par exemple la décider à l'aide du saut Turing, ou encore à l'aide de la réunion disjointe

de toutes les itérations finies du saut Turing ? Nous allons voir que non : la théorie de $(\mathcal{D}, \leq)$ est de complexité *maximale*. Qu'entendons-nous par là ? Considérons T_2 , la théorie du second ordre de $(\mathbb{N}, \times, +, 0, 1)$, c'est-à-dire l'ensemble des formules du second ordre qui sont vraies dans $\mathbb{N}$. On rappelle qu'une formule du second ordre est de la forme $\forall X \exists Y \dots F(X, Y, \dots)$ où les variables $X, Y, \dots$ sont des ensembles d'entiers, et où F est une formule du premier ordre, paramétrée par ces ensembles.

La théorie T_2 est donc l'ensemble des formules du second ordre qui sont vraies dans $\mathbb{N}$. Si l'on a accès à T_2 , on peut savoir si une formule de la théorie du premier ordre de $(\mathcal{D}, \leq)$ est vraie : les quantifications $\exists a$ et $\forall a$ peuvent se remplacer par des quantifications sur les éléments de $2^{\mathbb{N}}$ et, à l'aide du théorème 9-3.4 relativisé aux paramètres du second ordre — lequel permet de transformer un prédicat $\Sigma_n^0(X, Y, \dots)$ en une formule Σ_n de l'arithmétique —, on peut transformer une formule du premier ordre F de $(\mathcal{D}, \leq)$ en une formule de second ordre équivalente F^* de $(\mathbb{N}, \times, +, 0, 1)$. Simpson a montré que l'inverse était vrai aussi : via un codage ingénieux, il est possible de transformer une formule de l'arithmétique du second ordre en une formule équivalente de $(\mathcal{D}, \leq)$.

Insistons avant d'aller plus loin sur la complexité *extrême* de T_2 . Nous étudierons dans la partie IV tous les détails de la complexité de T_2 restreinte aux formules Π_1^1 , c'est-à-dire restreinte aux formules au sein desquelles les quantifications du second ordre sont toutes universelles. Nous verrons que cette théorie a déjà un degré Turing considérablement élevé comparé à $\emptyset'$ ou même à toutes les itérations finies de $\emptyset'$. Ce degré Turing est néanmoins bien défini, et il est *absolu* dans le sens où la valeur de vérité d'une formule Π_1^1 sera la même dans les modèles transitifs de la théorie des ensembles partageant les mêmes ordinaux calculables (voir la partie IV pour une définition formelle). À partir du niveau de complexité Π_2^1 des formules, ce sens devient plus flou. La valeur de vérité de ce genre de formule restera toutefois inchangée dans tous les modèles transitifs de la théorie des ensembles qui partagent cette fois non pas les mêmes ordinaux calculables, mais les mêmes ordinaux dénombrables. Sous réserve d'accepter l'absoluité des ordinaux dénombrables, la vérité des formules Π_2^1 est elle aussi absolue. Le degré Turing du niveau Π_2^1 de T_2 est quant à lui plus élevé que tous les degrés Turing abordés dans le présent livre (il s'agit en quelque sorte du supremum de tous les singletons Π_1^1 , dont nous verrons la définition dans la section 30-4). La valeur de vérité d'une formule Π_3^1 pourra quant à elle différer entre deux modèles de ZFC qui partagent les mêmes ordinaux, et le sens qu'il y a à dire qu'une telle formule est *vraie* ou *fausse* s'évanouit ici encore un peu plus. Quant au degré Turing de la théorie Π_3^1 de T_2 , de là où nous sommes, c'est-à-dire le monde des choses calculables, depuis lequel

nous observons la structure de l'univers, même les meilleurs télescopes ne permettent pas de le voir : il est tout simplement trop éloigné de nous.

Voilà donc la complexité de la théorie des degrés Turing ! Nous livrons ci-après la preuve moderne du théorème de Simpson, qui diffère de celle qui fut originellement produite, et qui présente son intérêt propre. Le résultat de Simpson découle du théorème suivant, où $n \in \mathbb{N}$.

Théorème 4.4 (Slaman et Woodin [206])

Tout sous-ensemble dénombrable $R \subseteq \mathcal{D}^n$ de n -uplets de degrés Turing est uniformément définissable dans $(\mathcal{D}, \leq)$ avec un nombre fini de paramètres. Formellement, il existe une formule $F(x_1, \dots, x_n, y_1, \dots, y_m)$ telle que, pour tout $R \subseteq \mathcal{D}^n$, il existe des paramètres $\mathbf{p}_1, \dots, \mathbf{p}_m \in \mathcal{D}$ pour lesquels $(\mathbf{a}_1, \dots, \mathbf{a}_n) \in R$ si, et seulement si, $F(\mathbf{a}_1, \dots, \mathbf{a}_n, \mathbf{p}_1, \dots, \mathbf{p}_m)$ est vrai dans $(\mathcal{D}, \leq)$.

Voyons tout de suite comment utiliser le théorème 4.4 pour montrer le théorème de Simpson : il suffit de coder un modèle standard de l'arithmétique dans les degrés Turing.

Théorème 4.5 (Simpson [201])

La théorie du premier ordre des degrés Turing est many-one équivalente à celle de l'arithmétique du second ordre.

PREUVE. Nous avons déjà vu dans les paragraphes précédents comment transformer un énoncé de $(\mathcal{D}, \leq)$ en un énoncé équivalent de l'arithmétique du second ordre. Voyons à présent comment faire l'inverse.

L'idée est de coder un modèle standard de $(\mathbb{N}, +, \times, 0, 1)$ dans les degrés Turing. Un tel modèle sera codé par un ensemble fini de paramètres codant pour un ensemble N de degrés Turing qui représentent $\mathbb{N}$, avec un degré spécifique représentant 0 et un autre représentant 1. Les relations $+$ et $\times$ sont elles aussi codées par un ensemble fini de paramètres.

Il est possible de créer une formule de $\mathcal{D}$ qui vérifie si un ensemble fini de paramètres code bien pour le modèle standard de l'arithmétique : il s'agit simplement de vérifier les axiomes de l'arithmétique de Robinson (voir la section 9-2.3), lesquels sont en nombre fini, et de vérifier ensuite que tout sous-ensemble du modèle possède un plus petit élément. On peut se reporter au théorème 9-3.13 pour voir que ces conditions sont nécessaires et suffisantes pour vérifier que l'on a bien affaire au modèle standard des entiers. La quantification universelle « tout sous-ensemble du modèle possède un plus petit élément » peut être remplacée par une quantification universelle sur les degrés Turing utilisés comme paramètres pour coder des sous-ensembles de N (notre ensemble de degrés qui représente $\mathbb{N}$).

Étant donné une formule F de l'arithmétique du second ordre, on peut finalement la transformer en une formule équivalente F^* dans $\mathcal{D}$, en remplaçant les quantifications sur les ensembles par des quantifications sur les paramètres codant pour ces ensembles. La formule du second ordre de l'arithmétique sera donc interprétée dans $\mathcal{D}$ par la formule : il existe des paramètres codant pour un modèle standard de l'arithmétique, tel que F^* est vérifiée dans ce modèle. ■

Passons à présent au codage de Slaman et Woodin. Le lemme qui suit constitue la partie difficile de la preuve. Il repose sur un forcing qui peut sembler relativement simple dans son principe, mais dont l'exécution s'avère délicate et demande pas mal d'astuce pour être menée à bien.

Lemme 4.6 (Slaman et Woodin [206]). Toute anti-chaîne dénombrable dans les degrés Turing est uniformément définissable avec trois paramètres. ★

PREUVE. Soit $(\mathbf{a}_n)_{n \in \mathbb{N}}$ une anti-chaîne dans les degrés Turing, et soit $\mathbf{b}$ un majorant de cette anti-chaîne. Nous allons définir deux degrés $\mathbf{g}_0, \mathbf{g}_1$ tels que pour tout degré $\mathbf{y} \leq \mathbf{b}$ ne majorant aucun $\mathbf{a}_i$, alors $\mathbf{g}_0 \cup \mathbf{y}$ et $\mathbf{g}_1 \cup \mathbf{y}$ ont une borne inférieure, et cette borne inférieure est $\mathbf{y}$. En d'autres termes, tout degré à la fois sous $\mathbf{g}_0 \cup \mathbf{y}$ et sous $\mathbf{g}_1 \cup \mathbf{y}$ doit aussi être sous $\mathbf{y}$. À l'inverse, il existera pour tout i un degré à la fois sous $\mathbf{g}_0 \cup \mathbf{a}_i$ et sous $\mathbf{g}_1 \cup \mathbf{a}_i$ qui ne sera pas sous $\mathbf{a}_i$. Il s'ensuit que chaque $\mathbf{a}_i$ sera un élément minimal satisfaisant la formule

$$F(\mathbf{x}) = \mathbf{x} \leq \mathbf{b} \wedge \exists \mathbf{c} (\mathbf{c} \not\leq \mathbf{x} \wedge \mathbf{c} \leq \mathbf{g}_0 \cup \mathbf{x} \wedge \mathbf{c} \leq \mathbf{g}_1 \cup \mathbf{x}).$$

En particulier, les degrés $\mathbf{a}_i$ seront exactement les degrés $\mathbf{x}$ satisfaisant la formule $F(\mathbf{x}) \wedge \forall \mathbf{y} \leq \mathbf{x} \neg F(\mathbf{y})$. Le fait que les $\mathbf{a}_i$ forment une anti-chaîne est utilisé uniquement pour les définir comme solutions minimales de F , mais n'intervient plus par la suite.

Nous utiliserons pour la construction des degrés $\mathbf{g}_0$ et $\mathbf{g}_1$ le fait suivant : tout degré Turing contient un ensemble X calculable en n'importe quel sous-ensemble infini de X . On peut le voir de la manière suivante : étant donné un ensemble Y quelconque, on définit X comme étant l'ensemble des préfixes $\sigma \prec Y$, via un codage des chaînes finies par des entiers.

Soit B un représentant de $\mathbf{b}$ et, pour tout n , soit A_n un représentant de $\mathbf{a}_n$ calculable en n'importe lequel de ses sous-ensembles infinis. Nous allons définir deux ensembles G_0, G_1 tels que pour tout i il existe $C \not\leq_T A_i$ tel que $C \leq_T G_0 \oplus A_i$ et $C \leq_T G_1 \oplus A_i$, et tel que pour tout $Y \leq_T B$ et tout D tel que $D \leq_T G_0 \oplus Y$ et $D \leq_T G_1 \oplus Y$, alors $Y \geq_T D$ ou bien $Y \geq_T A_j$ pour un certain j .

On procède à cet effet via un forcing qui présente des similarités avec celui du théorème 2.7.

Soit $\mathbb{P}$ l'ensemble de conditions de la forme (σ_0, σ_1, n) pour $\sigma_0, \sigma_1 \in 2^{<\mathbb{N}}$, avec $|\sigma_0| = |\sigma_1|$ et $n \in \mathbb{N}$. L'entier n sert à restreindre les extensions possibles, la chaîne σ_0 est utilisée pour le premier générique G_0 et la chaîne σ_1 pour le deuxième générique G_1 . Tout comme dans la preuve du théorème 2.7, on peut voir G_0 et G_1 comme étant construits par colonne. L'entier n indique que la construction sera dorénavant restreinte sur les n premières colonnes : pour une colonne $k \leq n$, si $a \notin A_k$, il n'y a alors aucune restriction pour le bit $\langle k, a \rangle$ des deux génériques. Si en revanche $a \in A_k$, alors le bit $\langle k, a \rangle$ des deux génériques doit être identique (sans nécessairement être égal à $A_k(a)$).

Formellement, $(\sigma_0, \sigma_1, n) \succeq (\tau_0, \tau_1, m)$ si $\sigma_0 \succeq \tau_0$, si $\sigma_1 \succeq \tau_1$, si $n \leq m$, et si de plus la condition suivante est vérifiée : pour tout $k \leq n$, alors pour tout $a \in A_k$ tel que $|\sigma_i| \leq \langle k, a \rangle < |\tau_i|$, les valeurs $\tau_0(\langle k, a \rangle)$ et $\tau_1(\langle k, a \rangle)$ doivent être les mêmes.

Considérons G_0, G_1 deux ensembles suffisamment génériques pour ce forcing. Pour chaque A_n , et pour $a \in A_n$ suffisamment grand, on aura

$$G_0(\langle n, a \rangle) = G_1(\langle n, a \rangle),$$

par définition de ce qu'est une extension valide dans ce forcing. En particulier ; les ensembles $X_0^n, X_1^n \subseteq A_n$ définis par $X_i^n(a) = 0$ si $a \notin A_n$, et $X_i^n(a) = G_i(\langle n, a \rangle)$ sinon, sont les mêmes sauf pour un nombre fini de bits, et sont donc tous les deux calculés par $G_0 \oplus A_n$ et $G_1 \oplus A_n$.

Montrons que si G_0, G_1 sont suffisamment génériques, alors aucun A_n ne peut calculer les ensembles X_0^n, X_1^n ainsi définis. Étant donné une condition $\langle \sigma_0, \sigma_1, n \rangle$, on peut prendre n'importe quelle extension pour le côté σ_0 , ce qui force alors certains bits de l'extension pour l'autre côté. En considérant le fait qu'il y a nécessairement des sous-ensembles infinis de A_n non calculables en A_n , on peut nécessairement trouver une extension $\tau_0 \succeq \sigma_0$ telle que pour une fonctionnelle Φ_e donnée, $\Phi_e(A_n)$ ne produise jamais la restriction de τ_0 qui sera faite pour en faire un préfixe de X_0^n — soit parce que $\Phi_e(A_n)$ sera partielle, soit parce qu'elle produira une chaîne incompatible avec le préfixe de X_0^n ainsi forcé. Comme X_1^n coïncide avec X_0^n sauf sur un nombre fini de bits, alors A_n ne calculera pas non plus X_1^n . Cela nous donne la première partie de ce que l'on cherche à montrer : pour tout A_n , il existe un ensemble calculable en $G_0 \oplus A_n$ et en $G_1 \oplus A_n$, mais pas en A_n .

Il reste à montrer que pour tout $Y \leq_T B$ tel que Y ne calcule aucun A_n , si $G_0 \oplus Y$ et $G_1 \oplus Y$ calculent un même ensemble C , alors $Y \geq_T C$. Soit alors $p = (\sigma_0, \sigma_1, n)$ une condition et soient Φ_{e_0}, Φ_{e_1} une paire de fonctionnelles. On sépare dans un premier temps la chaîne σ_0 de notre condition p . S'il existe x et $\tau_0 \succeq \sigma_0$ tels que pour tout $\rho_0 \succeq \tau_0$ on a $\Phi_{e_0}(Y \oplus \rho_0, x) \uparrow$, on considère alors une chaîne τ_1 telle que la condition (τ_0, τ_1, n) forme une extension valide, pour laquelle on aura forcé la partialité de $\Phi_{e_0}(Y \oplus G_0)$.

Supposons à présent que pour tout x et pour tout $\tau_0 \succeq \sigma_0$ il existe $\rho_0 \succeq \tau_0$ tel que $\Phi_{e_0}(Y \oplus \rho_0, x) \downarrow$. Supposons dans un premier temps qu'il existe une extension $\tau \succeq \sigma_0$ telle que, pour toutes extensions $\rho_0, \rho_1 \succeq \tau$ et pour tout x , on ait $\Phi_{e_0}(Y \oplus \rho_0, x) \downarrow = a$ et $\Phi_{e_0}(Y \oplus \rho_1, x) \downarrow = b$ impliquent $a = b$. Alors, on ne peut produire qu'un unique ensemble via $\Phi_{e_0}(Y \oplus G_0)$ pour un générique G_0 quelconque, et cet ensemble est calculable alors en Y . On force donc $\Phi_{e_0}(Y \oplus G_0)$ à calculer quelque chose qui est déjà calculable en Y . Supposons finalement que, pour tout $\tau \succeq \sigma_0$, il existe $\rho_0, \rho_1 \succeq \tau$ et x tels que $\Phi_{e_0}(Y \oplus \rho_0, x) \downarrow \neq \Phi_{e_0}(Y \oplus \rho_1, x) \downarrow$. Le lemme suivant s'avère utile.

Lemme 4.7. Pour tout $\tau \succeq \sigma$, il existe x et deux extensions $\rho_0, \rho_1 \succeq \tau$ qui diffèrent sur seulement un bit et tels que

$$\Phi_{e_0}(Y \oplus \rho_0, x) \downarrow = a \neq \Phi_{e_0}(Y \oplus \rho_1, x) \downarrow = b. \quad \star$$

PREUVE. Il suffit de trouver deux extensions de τ de même taille et incompatibles sur un certain x . Soient $i_0, \dots, i_k$ les bits sur lesquelles ces extensions diffèrent. On inverse le bit i_0 dans la première extension, et on l'étend pour obtenir une valeur a_0 pour x . Si $a_0 \neq a$, on a terminé. Sinon, on inverse à son tour i_1 dans cette nouvelle extension, que l'on étend encore pour obtenir une valeur a_1 , et ainsi de suite. Si chaque valeur $a_1 = a_2 = \dots = a_{k-1}$, alors $a_{k-1} \neq b$, et notre chaîne diffère maintenant d'un seul bit de celle qui a produit b . ■

On se sert du lemme précédent pour calculer à l'aide de Y une suite de quadruplets $(\tau_{0,m}, \tau_{1,m}, i_m, x_m)_{m \in \mathbb{N}}$ avec $i_m < i_{m+1}$ telle que, pour tout m , les chaînes $\tau_{0,m}, \tau_{1,m}$ diffèrent sur exactement le bit i_m et soient incompatibles sur x_n . La suite de bits $(i_m)_{m \in \mathbb{N}}$ est une suite infinie et Y -calculable. Rappelons que notre condition de forcing est de la forme (σ_0, σ_1, n) . S'il existe i_m tel que $i_m = \langle k, a \rangle$ pour $k > m$, cela entraîne que pour toute extension τ' de σ_1 telle que $\langle \tau_{0,m}, \tau', n \rangle$ est une extension valide, alors $\langle \tau_{1,m}, \tau', n \rangle$ en est aussi une, car il n'y a aucune contrainte sur le bit i_m . On peut donc trouver une extension de τ' de σ_1 qui force une valeur pour x_n (à supposer que l'on ne puisse pas forcer la partialité de ce côté là), et l'on prend l'extension $\tau_{0,m}$ ou $\tau_{1,m}$ de σ_0 qui force une valeur différente. On force donc $\Phi_{e_0}(Y \oplus G_0)$ et $\Phi_{e_1}(Y \oplus G_1)$ à être différents.

Si à présent il n'existe pas $i_m = \langle k, a \rangle$ pour $k > m$, alors il doit exister par le principe des tiroirs un certain $k \leq m$ pour lequel une infinité de i_m est de la forme $\langle k, a \rangle$. Aussi n'est-il pas possible d'avoir $A_k(a) = 1$ pour chacun de ces i_m , car on aurait alors un sous-ensemble infini de A_k et Y -calculable, or par hypothèse tout sous-ensemble infini de A_k calcule A_k , et Y ne calcule pas A_k . Il doit donc exister $\tau_{0,m}, \tau_{1,m}$ et $i_m = \langle k, a \rangle$ tel que $A_k(a) = 0$. Là encore, toute extension τ' de σ_1 qui est compatible avec $\tau_{0,m}$ le sera aussi avec $\tau_{1,m}$, car il n'y a aucune contrainte sur le bit i_m . On peut alors trouver

une extension τ' de σ_1 qui force une valeur sur x_m — à moins que l'on ne puisse forcer la partialité de ce côté là — et choisir une extension parmi $\tau_{0,m}$ et $\tau_{1,m}$ qui force une autre valeur sur x_m . Cela conclut la preuve. ■

Et voilà. La preuve du lemme ne fut pas sans difficulté, mais nous sommes à présent presque au bout de nos peines. Montrons finalement que tout sous-ensemble dénombrable de $\mathcal{D}^n$ peut être codé dans les degrés Turing.

PREUVE DU THÉORÈME 4.4. Dans ce qui suit, les variables en majuscule dénotent des ensembles ou suites de degrés Turing. Soit un ensemble dénombrable de n -uplets $R \subseteq \mathcal{D}^n$. Soit $\mathbf{b}$ un majorant sur l'ensemble des degrés concernés par R (c'est-à-dire sur la réunion des projections de R sur chaque coordonnée), et soit $(\mathbf{x}_i)_{i \in \mathbb{N}}$ une liste de tous les degrés sous $\mathbf{b}$. Notons que $\mathbf{b}$ n'est qu'un majorant, et que certains $\mathbf{x}_i$ peuvent donc ne pas être des degrés du n -uplet R .

On trouve alors une anti-chaîne $(\mathbf{c}_i^k)_{k \leq n, i \in \mathbb{N}}$ telle que $B = (\mathbf{c}_i^k \cup \mathbf{x}_i)_{k \leq n, i \in \mathbb{N}}$ forme un ensemble de degrés calculatoirement indépendants (on montre sans peine qu'une telle anti-chaîne existe, par extensions finies). Pour $k \leq n$ fixé, soit $C_k = (\mathbf{c}_i^k)_{i \in \mathbb{N}}$. On définit finalement

$$S = \{\mathbf{c}_{i_1}^1 \cup \mathbf{x}_{i_1} \cup \dots \cup \mathbf{c}_{i_n}^n \cup \mathbf{x}_{i_n} : (\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_n}) \in R\}.$$

Comme B est calculatoirement indépendant, il existe pour tout $\mathbf{a} \in S$ un n -uplet de degrés $\mathbf{b}_1, \dots, \mathbf{b}_n \in B$, unique à l'ordre près, tel que

$$\mathbf{a} = \mathbf{b}_1 \cup \dots \cup \mathbf{b}_n.$$

Par ailleurs, l'indépendance calculatoire de B garantit également que pour le degré $\mathbf{b}_1$ il existe un unique $i \leq n$ tel que $\mathbf{b}_1 = \mathbf{c}_m^i \cup \mathbf{x}_m$ pour un certain m , et ce m est lui aussi unique. Il en va de même pour $\mathbf{b}_2, \mathbf{b}_3, \dots$, ce qui permet de définir R de la manière suivante : $(\mathbf{x}_1, \dots, \mathbf{x}_n) \in R$ si

$$\mathbf{x}_1 \leq \mathbf{b} \wedge \dots \wedge \mathbf{x}_n \leq \mathbf{b} \wedge \exists \mathbf{y}_1 \in C_1 \dots \exists \mathbf{y}_n \in C_n (\mathbf{x}_1 \cup \mathbf{y}_1) \cup \dots \cup (\mathbf{x}_n \cup \mathbf{y}_n) \in S.$$

Comme chaque C_i et comme S sont des anti-chaînes, elles sont définissables d'après le lemme 4.6. Une telle formule est donc définissable dans les degrés Turing. ■

Slaman et Woodin ont utilisé leur technique de codage comme point de départ à une étude complexe de la *rigidité* des degrés Turing. Une structure est dite *rigide* si elle n'admet pas d'automorphisme autre que l'identité, c'est-à-dire dans le cas des degrés, de bijection non triviale $f : \mathcal{D} \rightarrow \mathcal{D}$ telle que $\mathbf{a} \leq \mathbf{b} \leftrightarrow f(\mathbf{a}) \leq f(\mathbf{b})$. Par exemple, la structure $(\mathbb{R}, +, \times, \leq)$ des réels est une structure rigide. Étant donné un automorphisme $f : \mathbb{R} \rightarrow \mathbb{R}$, on doit avoir $f(0) + f(1) = f(1)$ et donc $f(0) = 0$, puis $f(1) = f(1) \times f(1)$

et donc $f(1) = 1$. En utilisant $f(n+1) = f(n) + 1$, on montre que f est nécessairement l'identité sur $\mathbb{N}$, et l'on montre la même chose sur $\mathbb{Q}$ en jouant de la multiplication. Comme f conserve les carrés, f est croissante. On utilise alors pour terminer le fait que tout nombre réel est limite d'une paire de suites adjacentes de nombres rationnels, si bien que f est forcément l'identité sur $\mathbb{R}$.

Peut-on faire quelque chose de similaire dans les degrés Turing ou bien peut-on « échanger » deux degrés **a** et **b**, et étendre cet échange de manière consistante en un automorphisme sur $\mathcal{D}$? La question reste pour le moment ouverte, même si Slaman et Woodin en ont, comme on va le voir derechef, considérablement réduit les possibilités.

Théorème 4.8 (Slaman et Woodin [207])

Tout automorphisme sur les degrés Turing est l'identité sur $\mathcal{D}(\geq 0'')$.

Slaman et Woodin utilisent ce résultat pour montrer dans le même article que la fonction de double saut est définissable dans les degrés Turing. Ce résultat sera étendu plus tard par Shore et Slaman.

Théorème 4.9 (Shore et Slaman [200])

Le saut Turing est définissable dans les degrés Turing, sans paramètres.

Ce résultat peut alors à son tour être utilisé pour montrer que tout automorphisme sur les degrés Turing est l'identité sur $\mathcal{D}(\geq 0')$.

La question suivante reste toutefois ouverte.

Question 4.10. Existe-t-il un automorphisme autre que l'identité sur les degrés Turing? ★

5. Structure des degrés c. e.

Un autre champ de recherche important dans les degrés Turing consiste à en restreindre l'étude à un sous-ensemble. Dans cet ordre d'idée, l'étude de la structure $(\mathcal{R}, \leq)$ des degrés c. e. est certainement la plus développée. Nous faisons ici un résumé rapide des principaux résultats les concernant.

Commençons tout de suite par l'impressionnant théorème de densité de Sacks, qui se montre au terme de ce qui est certainement l'une des méthodes de priorité à blessure infinie les plus complexes.

Théorème 5.1 (Sacks (1964))

L'ordre $(\mathcal{R}, \leq)$ est dense : étant donné des ensembles c. e. $A <_T B$, il existe un ensemble c. e. C tel que $A <_T C <_T B$.

La structure des degrés c. e. apparaît donc bien différente de celle des degrés Turing. Il y a tout de même des similarités : on vérifie ainsi sans problème que toute paire d'éléments admet une borne supérieure, l'ensemble $A \oplus B$ étant c. e. si A et B sont tous les deux c. e. Tout comme dans les degrés Turing, on peut montrer — en élaborant sur la construction de deux degrés incomparables — qu'il existe un ensemble dénombrable de degrés c. e. calculatoirement indépendants [164]. Tout comme pour les degrés Turing, on peut en déduire que tout ordre partiel dénombrable se plonge dans les degrés c. e., ce qui montre comme l'a remarqué Sacks que la théorie Π_1^0 des degrés c. e. est décidable.

Théorème 5.2 (Sacks [188])

La théorie Π_1^0 des degrés c. e. est décidable.

Étant donné une formule existentielle sur les degrés c. e., il suffit de vérifier si elle est compatible avec les axiomes d'un ordre partiel. Si tel est le cas elle est vraie, sinon elle est fausse.

Tout comme pour les degrés Turing, la question du plongement n'est pas vraiment satisfaisante en soi. Une question plus intéressante est celle du plongement de treillis, tel que les bornes supérieures sont envoyées vers les bornes supérieures, et les bornes inférieures sur les bornes inférieures. Les plongements de treillis mentionnés dorénavant seront considérés comme vérifiant cette contrainte. L'existence d'une paire minimale de degrés c. e. montre par exemple que le treillis en diamant généré par $\mathbf{c}_0, \mathbf{c}_1$ incomparables (c'est-à-dire avec $\mathbf{c}_0 \cap \mathbf{c}_1 < \mathbf{c}_0$, $\mathbf{c}_1 < \mathbf{c}_0 \cup \mathbf{c}_1$) peut se plonger de cette manière dans les degrés Turing, la borne inférieure étant le degré 0.

La construction effective d'une paire minimale de degrés c. e. (voir le théorème 13-5.2) a été exploitée pour montrer des résultats bien plus forts. Un treillis est dit *distributif* si $a \cap (b \cup c) = (a \cup b) \cap (a \cup c)$.

Théorème 5.3 (Thomason [223] Lachlan et Lerman)

Tout treillis distributif dénombrable peut se plonger dans les degrés c. e.

En ce qui concerne les treillis non distributifs, certains peuvent se plonger dans les degrés c. e. [131] et d'autres pas [134]. Aucune caractérisation n'est connue à ce jour.

Nous avons vu que la théorie des degrés Turing est de complexité maximale, il en va de même pour celle des degrés c. e. qui a la même complexité que la théorie constituée des formules vraies de l'arithmétique du premier ordre. Le premier résultat dans cette direction a été obtenu par Harrington et Shelah [83], qui ont montré que la théorie du premier ordre des degrés c. e. était indécidable. La preuve fut ensuite simplifiée et améliorée par Harrington et Slaman, puis par Nies, Shore et Slaman [168], qui ont montré le théorème suivant.

Théorème 5.4 (Nies, Shore et Slaman [168])

On peut transformer effectivement un énoncé F de l'arithmétique du premier ordre en énoncé F^* sur les degrés c. e., tel que :

F est vrai dans $(\mathbb{N}, +, \times, 0, 1)$ si, et seulement si, F^* est vrai dans $(\mathcal{R}, \leq)$.

Pour finir, Lempp, Nies et Slaman [137] ont montré que la théorie Π_3^0 des degrés c. e. était déjà indécidable, ce qui laisse ouverte la question suivante.

Question 5.5. La théorie Π_2^0 des degrés c. e. est-elle décidable ? ★

Deuxième partie

Aléatoire algorithmique

Chapitre 15

Introduction

Nous avons abordé dans la section 10-2 une notion de typicité des ensembles qui découle directement du travail de Baire : être générique. Il est peu dire que ces notions d'ensembles maigres et co-maigres se sont révélées être d'une grande utilité, et ce dans de multiples domaines : analyse, théorie des ensembles, relations d'équivalence, calculabilité, etc.

Et pourtant, si la fécondité des travaux de René Baire est exceptionnelle, celle des travaux de son collègue Henri Lebesgue est encore au-dessus. Les deux mathématiciens se connaissaient très bien et ont étudié auprès du même professeur — en l'occurrence Émile Borel — dont la carrière brillante fait écho à celle de ses deux élèves.



Henri Lebesgue, 1875–1941

Lebesgue travaille sur la théorie de la mesure, ce qui conduira à une notion de typicité orthogonale à celle de Baire : être aléatoire. La théorie de la mesure peut être utilisée de manière similaire à celle des catégories de Baire, pour des problématiques communes : par exemple, les points de discontinuité d'une fonction limite de fonctions continues forment une classe maigre. Dans l'espace de Cantor, il s'agit d'une conséquence de la preuve du théorème 10-3.20 : si G est 1-générique, alors le saut Turing est continu

en G . On montre quelque chose de similaire pour la mesure : soit f une fonction limite de fonctions continues, il existe des ensembles fermés de mesure arbitrairement grande, sur lesquels la restriction de f est continue. Ce sera une conséquence, dans l'espace de Cantor, de la preuve du théorème 19-3.8. La théorie de la mesure doit enfin sa renommée à l'utilisation qui en est faite pour axiomatiser la théorie des probabilités. Ainsi, un ensemble typique pour la théorie de la mesure peut être informellement vu comme un ensemble « aléatoire », c'est-à-dire qui satisfait toutes les propriétés qu'un ensemble vérifie avec probabilité 1.

L'étude des nombres aléatoires est aussi une manière d'étudier des nombres individuellement « inaccessibles », mais qui révèlent beaucoup de secrets quand on considère plutôt les groupes auxquels ils appartiennent. Cette intuition se trouve déjà chez Borel dans les années 1950. Aussi citons-nous pour illustrer notre propos un passage de son ouvrage « Les nombres inaccessibles » :

« Toutes les théories qui se rattachent à la mesure des ensembles peuvent donc être considérées comme une contribution à la théorie des nombres inaccessibles ; si nous ne pouvons étudier individuellement aucun de ces nombres, nous pouvons étudier des problèmes de probabilité qui sont relatifs, soit à l'ensemble de ces nombres, soit à certains sous-ensembles. La réponse à certaines questions se trouve être ainsi un coefficient de probabilité. Une telle réponse peut avoir souvent un grand intérêt dans bien des questions scientifiques. »

En calculabilité, cela nous amène à ce que l'on appelle aujourd'hui *l'aléatoire algorithmique*, un domaine qui est né d'une question d'ordre philosophique, en apparence simple : qu'est-ce qu'une suite de 0 et de 1 aléatoire ? Plus précisément, étant donné une suite de 0 et de 1, est-il « raisonnable » de penser qu'elle ait pu être obtenue par un tirage à pile ou face ?

Nous avons bien entendu une intuition sur la question : par exemple, la suite 0000000000000000 semble moins aléatoire que la suite 01101001010010. Mais comment donner un sens précis à cette intuition ? D'autant que quand on y réfléchit, chacune des deux suites ci-dessus a exactement la même probabilité d'apparition (de 2^{-14}). Pourtant, obtenir la deuxième avec un tirage à pile ou face apparaîtra parfaitement normal, alors qu'obtenir la première semblera extraordinaire. Nous avons en effet l'intuition qu'un nombre aléatoire — on entend ici par « nombre » un réel $r \in]0, 1[$, vu comme une suite infinie de 0 et de 1 via sa représentation binaire — devrait satisfaire la loi des grands nombres : la fréquence de 1 doit être la même que la fréquence de 0 c'est-à-dire $1/2$. En poussant plus loin, on peut aussi demander que la fréquence de 00, 01, 10 et 11 soit la même, à savoir $1/4$, et ainsi de suite pour toutes suites finies de bits. Émile Borel appelle de tels

nombre des *nombre normal*, et a montré qu'un nombre est normal avec probabilité 1. Mais est-ce suffisant pour considérer qu'un nombre puisse être le résultat d'un tirage aléatoire de ses bits ? La réponse est non, et l'on peut utiliser pour aller dans ce sens un résultat de Champernowne [32], qui a montré que la suite obtenue en concaténant tous les entiers écrits en binaire dans l'ordre, est un nombre normal : 0 1 10 11 100 101 110 111 ... Il est clair en revanche que ce nombre n'a rien d'aléatoire, car il est calculable par un procédé très simple, alors qu'un nombre aléatoire ne devrait pas être calculable : imaginez un algorithme prédisant les numéros du loto, ou la prochaine carte qui serait retournée au cours d'une partie de Blackjack ? Ce serait la fin des casinos. Notons au passage que les différentes machines électroniques de jeu de hasard ont recours à des processus physiques couplés à des algorithmes de génération aléatoire, afin justement qu'un joueur ne puisse pas « prédire l'avenir » avec un simple calcul.

La première tentative de définir formellement le caractère aléatoire d'une suite infinie de bits est en général attribuée à Richard von Mises, qui en 1919 propose la définition suivante : une suite est aléatoire si chacune de ses sous-suites infinies, obtenable par un « processus de sélection raisonnable », satisfait la loi des grands nombres. La calculabilité qui se développa peu après nous offre un cadre naturel pour définir formellement ce que doit être un processus de sélection raisonnable : simplement un processus calculable. En 1939, Jean Ville démontra dans sa thèse « Étude critique de la notion de collectif » [228] que la définition de von Mises était insuffisante : pour n'importe quel ensemble dénombrable de règles de sélection, il existe des suites qui sont aléatoires au sens de von Mises, mais dont chaque préfixe contient plus de 0 que de 1. Il est effectivement très improbable de tirer 1000 fois de suite à pile ou face, en ayant constamment plus de pile que de face. Quand le nombre de tirages s'étend vers l'infini, cette probabilité tombe à 0. On pourrait envisager d'étendre l'idée de von Mises et demander à ce que les sous-suites sélectionnées ne satisfassent pas seulement la loi des grands nombres, mais aussi d'autres lois qui sont satisfaites avec probabilité 1. Mais quelles lois choisir ? En fait, si l'on en fixe un nombre fini $L_1, \dots, L_n$, il n'y a aucune raison de croire que l'on ne pourra pas trouver une suite dont toutes les sous-suites calculables satisfont chacune de ces lois, tout en échappant à une $(n + 1)$ -ième.

C'est en 1966 que Martin-Löf proposa la définition qui fait encore aujourd'hui consensus comme étant une définition « satisfaisante » d'aléatoire : il considère toutes les lois Π_2^0 qui sont satisfaites avec probabilité 1, en remarquant qu'avec l'ajout d'une certaine condition sur les Π_2^0 en question, ceux-ci peuvent se combiner en une seule loi universelle. Un peu plus tard Chaitin [29], Gács [69] et Levin [142] ont tous les trois indépendamment

et presque simultanément introduit une définition de l'aléatoire algorithmique via un paradigme complètement différent, en l'occurrence celui de la compressibilité, en utilisant une version spéciale de la complexité de Kolmogorov. Par la suite, Levin [141] et Schnorr [193] ont démontré que la définition d'aléatoire de Martin-Löf coïncidait celle de Chaitin/Gács/Levin, révélant alors la robustesse de cette notion, pour laquelle on trouva par la suite bon nombre d'autres caractérisations.

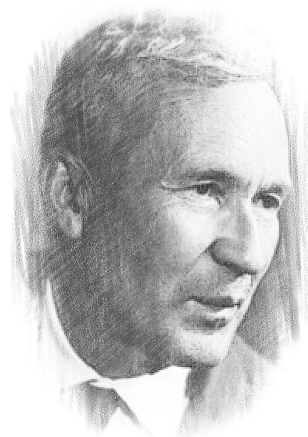
Les notions de compressibilité et de complexité de Kolmogorov permettent une transition en douceur de la calculabilité vers l'étude de l'aléatoire, et c'est donc par là que nous commencerons.

Chapitre 16

Complexité de Kolmogorov et nombres aléatoires

Andrei Nikolaïevitch Kolmogorov (1903 - 1987) est certainement l'un des mathématiciens les plus fameux et les plus prolifiques de *l'école de Moscou*, fondée par Dmitri Egorov et Nikolai Luzin au début du XX^e siècle, et dont nous aurons l'occasion de reparler dans la section 29-1. À l'issue de sa thèse en 1929, effectuée sous la direction de Luzin, Kolmogorov a déjà publié de nombreux articles et acquis une renommée internationale. Il devient en 1931 professeur à l'université de Moscou et y mènera une brillante carrière durant laquelle il participera à la fondation de pans entiers des mathématiques modernes.

Ses travaux les plus connus concernent sans aucun doute l'axiomatisation, en 1933, de la théorie des probabilités [10], dont nous reparlerons dans le chapitre 17. Trente ans plus tard, Kolmogorov a des contributions importantes en topologie, en théorie des systèmes dynamiques, et a participé à la résolution du treizième problème de Hilbert. Sa carrière n'est pas terminée pour autant. Il est alors sur le point de démarrer un autre champ d'étude mathématique



Andreï Nikolaïevitch Kolmogorov, 1903–1987

qui connaîtra une fois de plus un retentissement considérable : la théorie algorithmique de l'aléatoire, complémentaire de la théorie des probabilités, avec cette fois l'utilisation d'un outil nouveau, l'informatique. Il développe notamment sa notion de complexité éponyme, dont nous nous efforcerons au long de ce chapitre de montrer la richesse.

1. Complexité de Kolmogorov

Informellement, la complexité de Kolmogorov d'un objet fini est une mesure de la quantité d'information nécessaire pour calculer cet objet. Si Kolmogorov [121] fut le premier à publier sur le sujet, cette idée remonte en fait aux travaux de Solomonoff [209].

Définition 1.1. On appelle *Machine* une fonction partielle calculable de $2^{<\mathbb{N}}$ vers $2^{<\mathbb{N}}$. La *complexité de Kolmogorov de σ relativement à M* , notée $C_M(\sigma)$, est la longueur de la plus petite chaîne τ telle que $M(\tau) = \sigma$. Formellement, $C_M(\sigma) = \min\{|\tau| : M(\tau) = \sigma\}$. $\diamond$

La complexité de Kolmogorov d'une chaîne relativement à une machine M peut être vue comme une mesure de sa compression maximale possible via M . C'est donc une notion relative, qui dépend de la machine qui est utilisée, et si la machine en question ne fait rien, la notion n'est pas très intéressante. L'idée est bien sûr d'utiliser des machines qui compressent au maximum l'information.

1.1. Machine universelle

C'est Solomonoff le premier qui comprend la possibilité de définir une machine optimale, que l'on appelle aussi *universelle* : une machine dont le taux de compression sera au moins aussi bon que celui de n'importe quelle autre machine, à constante près.

Définition 1.2. Une machine U est dite *universelle* si, pour toute machine M , il existe une constante $c_M \in \mathbb{N}$ telle que $C_U(\sigma) \leq C_M(\sigma) + c_M$, pour toute chaîne σ . $\diamond$

Une machine universelle compresses donc aussi bien que n'importe quelle autre machine M , mais à une certaine constante additive près c_M . Évidemment, si la constante est grande par rapport à la chaîne que l'on veut compresser, la notion perd de sa force, mais le point important est que la constante ne dépend que de la machine M et non de la chaîne que l'on veut compresser. Le poids de cette constante s'amenuise donc à mesure que la taille des chaînes considérées augmente. En introduisant son concept de

machine universelle, Solomonoff a bien évidemment montré qu'une telle machine existait, ce qui a aussi été démontré indépendamment par Kolmogorov.

Théorème 1.3 (Solomonoff [210], Kolmogorov [122])

Il existe une machine universelle.

PREUVE. Soit $(M_e)_{e \in \mathbb{N}}$ une énumération des machines, c'est-à-dire que M_e est la machine de code e . il suffit de définir la fonction calculable

$$U : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$$

qui sur la chaîne $0^e 1 \sigma$ renvoie la valeur de $M_e(\sigma)$. Étant donné une machine M_e , il est clair que, pour toute chaîne σ , on a

$$C_U(\sigma) \leq C_{M_e}(\sigma) + (e + 1). \quad \blacksquare$$

Notation

On fixe dès à présent une machine universelle U , et l'on note $C(\sigma)$ la valeur $C_U(\sigma)$. Dès lors, $C(\sigma)$ sera la *complexité de Kolmogorov* de σ .

Remarquons que la machine universelle que nous avons fixée n'a pas d'importance : à constante additive près, toutes les machines universelles compressent les chaînes de manière optimale, et tous les théorèmes qui vont suivre sont indépendants du choix de celle-ci.

1.2. Les chaînes aléatoires

L'idée d'utilisation de la complexité de Kolmogorov comme mesure d'aléatoire est simple : moins une chaîne est compressible, plus elle est aléatoire. On montre facilement que des chaînes incompressibles de toutes les tailles existent.

Proposition 1.4. Pour tout n , il existe une chaîne σ de taille n telle que $C(\sigma) \geq n$. ★

PREUVE. Il s'agit d'un simple argument de comptage : U est une fonction, et associe donc à une chaîne au plus une autre chaîne. Aussi, pour tout n , le nombre de chaînes de taille strictement inférieure à n est de $\sum_{i=0}^{n-1} 2^i = 2^n - 1$. Il existe donc au moins une chaîne de taille n qui n'est calculée via U par aucune chaîne de taille strictement inférieure à n . ■

Nous nous intéresserons dans la suite aux chaînes incompressibles à constante près : si une chaîne de taille 10 000 est compressible par un programme de taille 9990, mais pas mieux, elle peut être moralement considérée comme

« fortement » aléatoire. L'importance de cette constante s'effacera complètement quand nous poursuivrons dans la section 2 notre étude de l'aléatoire sur les préfixes d'objets infinis.

1.3. Le degré Turing de la complexité de Kolmogorov

Avec l'utilisation d'une machine universelle, la complexité de Kolmogorov d'une chaîne s'apparente à la taille du plus petit programme informatique capable de calculer cette chaîne. Il s'agit en quelque sorte de sa meilleure compression possible, si l'on ne prend pas en considération le temps nécessaire à sa décompression, qui peut s'avérer particulièrement long. . . Quant au temps nécessaire à sa compression, la situation est encore pire : il ne s'agit même plus d'un processus calculable !

1.3.1. La complexité de Kolmogorov n'est pas calculable

On donne une première démonstration du fait que la complexité de Kolmogorov n'est pas calculable, via une formalisation mathématique du paradoxe de Berry : « soit n le plus petit entier que l'on ne peut pas définir en moins de cinquante mots ». Le paradoxe — qui devrait apparaître clairement au lecteur — vient de ce que le mot « définir » est lui-même mal défini. Il suffit de le remplacer par « calculable ».

Proposition 1.5. La fonction $\sigma \mapsto C(\sigma)$ n'est pas calculable. ★

PREUVE. Supposons que la fonction $\sigma \mapsto C(\sigma)$ soit calculable. Alors, on peut créer la fonction calculable $f : \mathbb{N} \rightarrow 2^{<\mathbb{N}}$ qui sur n renvoie la première chaîne σ — disons lexicographiquement — telle que $C(\sigma) > n$. En utilisant l'écriture en binaire des entiers, on peut définir la machine $M : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ qui sur la chaîne σ_n (qui est l'écriture binaire de n) renvoie donc $f(n)$.

Comme la taille nécessaire pour représenter n en base 2 est de $\log_2(n)$, on a ainsi $C_M(\sigma) \leq \log_2(n)$, pour toute chaîne $\sigma = f(n)$. Il y a donc une constante c_M telle que $C(\sigma) < \log_2(n) + c_M$, pour toute chaîne $\sigma = f(n)$. Dans le même temps, chacune de ces chaînes est choisie telle que $C(\sigma) > n$, ce qui donne $n < C(\sigma) < \log_2(n) + c_M$. Pour n suffisamment grand, tel que $n > \log_2(n) + c_M$, on a une contradiction. ■

La fonction $\sigma \mapsto C(\sigma)$ n'est donc pas calculable. En revanche, elle est *approchable par le dessus*.

Définition 1.6. Une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ est *approchable par le dessus* si elle est la limite d'une suite $(f_n)_{n \in \mathbb{N}}$ de fonctions calculables avec, pour tout n , l'inégalité $f_{n+1} \leq f_n$. ◇

On peut définir l'approximation $(f_n)_{n \in \mathbb{N}}$ de la complexité de Kolmogorov en assignant à $f_0(\sigma)$ la taille du premier programme τ trouvé tel que $U(\tau) \downarrow = \sigma$, et en assignant à $f_{n+1}(\sigma)$ la taille du plus petit programme τ tel que $U(\tau)[n+1] \downarrow = \sigma$ si cette taille est plus petite que $f_n(\sigma)$, et $f_n(\sigma)$ sinon.

On montre facilement que les fonctions approchables par le dessus sont calculables avec l'arrêt des programmes informatiques

Exercice 1.7. Montrer que toute fonction approchable par le dessus est $\emptyset'$ -calculable. $\diamond$

1.3.2. La complexité de Kolmogorov est Turing-complète

Il est possible de renforcer la proposition 1.5 et de montrer que la connaissance de la complexité de Kolmogorov permet en fait de calculer le problème de l'arrêt. Il s'agit d'un bon exercice, pour lequel nous préparons ci-après le lecteur avec une proposition plus simple, évidente si l'on s'en tient à la construction qui a été faite ci-dessus d'une machine universelle, mais qui demande un peu de travail si l'on considère les machines universelles de manière abstraite.

Proposition 1.8. Soit X une représentation d'une machine universelle

$$U : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$$

(par exemple, avec $X(\langle \sigma, \tau \rangle) = 1$ ssi $U(\sigma) \downarrow = \tau$ et $X(\langle \sigma, \epsilon \rangle) = 1$ ssi $U(\sigma) \uparrow$). Alors, $X \geq_T \emptyset'$. $\star$

PREUVE. On définit la machine M telle que $M(0^e) \downarrow = 0^s$ si s est le plus petit entier tel que $\Phi_e(e)[s] \downarrow$. Si un tel entier n'existe pas, le calcul $M(0^e)$ ne s'arrête pas. Soit une constante d telle que $C_U(\sigma) < C_M(\sigma) + d$, pour toute chaîne σ . Étant donné la connaissance de U , pour savoir si $\Phi_e(e) \downarrow$, il suffit de regarder $U(\sigma)$ pour toute chaîne σ de taille inférieure à $e + d + 1$, de récupérer la plus grande valeur s telle que $U(\sigma) \downarrow = 0^s$ pour l'une de ces chaînes σ , et de calculer $\Phi_e(e)[s]$. On a alors $\Phi_e(e) \downarrow \leftrightarrow \Phi_e(e)[s] \downarrow$. ■

L'un des exercices suivants consiste à montrer que la connaissance de la complexité de Kolmogorov permet de calculer la machine universelle U qui lui est associée, et donc le problème de l'arrêt. La considération suivante sera utile, et présente aussi son intérêt propre : la proposition 1.5 ne montre pas seulement que la complexité de Kolmogorov est non calculable, mais aussi que pour toute fonction de compression $I : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ telle que $C_I(\sigma) \leq C(\sigma)$, la fonction $\sigma \mapsto C_I(\sigma)$ est non calculable. En revanche, comme nous le montre l'exercice 1.11, une telle fonction ne permet pas nécessairement de calculer le problème de l'arrêt, et *a fortiori* la complexité de Kolmogorov qui lui est associée non plus. Il est nécessaire pour cela d'utiliser la connaissance *exacte* de la complexité de Kolmogorov.

Exercice 1.9. (★) (*Tibor Radó [182]*). Tibor Radó introduit dans un article en 1962 [182] la fonction $\Sigma : \mathbb{N} \rightarrow \mathbb{N}$ du *castor affairé*, dont on peut résumer l'idée, avec les notations de ce livre, comme suit : $\Sigma(n)$ est le plus grand temps de calcul t tel que $U(\sigma)[t] \downarrow$ pour une chaîne σ de taille inférieure ou égale à n . Les chaînes σ réalisant pour chaque n ces plus grands temps de calcul sont baptisées « busy beaver » par Tibor Radó — en français « castor affairé » — une expression anglaise imagée pour désigner des personnes très travailleuses. Montrer que toute fonction $f \geq \Sigma$ permet de calculer $\emptyset'$. $\diamond$

Exercice 1.10. (★★) (*créditée à P. Gács. Voir 2.7.7. de [144]*). Soit X une représentation de $C : 2^{<\mathbb{N}} \rightarrow \mathbb{N}$ (par ex. avec $X(\langle \sigma, n \rangle) = 1$ ssi $C(\sigma) = n$). Montrer que $X \geq_T \emptyset'$.

Indication.— Montrer que pour n suffisamment grand, il existe une chaîne de taille $2n$, de complexité inférieure à $2n$, mais telle qu'aucun programme de taille inférieure à $2n$ ne peut la calculer en un temps inférieur à $\Sigma(n)$. $\diamond$

Exercice 1.11. (★) Montrer qu'il existe une fonction de compression

$I : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ telle que $|I(\sigma)| \leq C(\sigma)$, $\forall \sigma \in 2^{<\mathbb{N}}$, et telle que $\forall I' \leq_T \emptyset'$. $\diamond$

1.4. Propriétés numériques de la complexité de Kolmogorov

Nous voyons ici quelques propriétés élémentaires de la complexité de Kolmogorov. Nous commençons par deux notations importantes, que nous réutiliserons tout au long de ce chapitre.

Notation

Étant donné un entier n , on écrira par abus de notation $C(n)$ pour signifier $C(\sigma)$, où σ est la représentation binaire standard de n . Notez que la taille de la représentation binaire de n est de l'ordre de $\log_2(n)$.

Notation

On écrira $\leq^+$ ou $=^+$ pour signifier inégalité ou égalité à constante additive près. Par exemple, l'inégalité $C(\sigma) \leq^+ |\sigma|$ de la proposition suivante signifie qu'il existe une constante d telle que, pour toute chaîne σ , on a $C(\sigma) \leq |\sigma| + d$.

1.4.1. Bornes pour la complexité de Kolmogorov

La complexité de Kolmogorov d'une chaîne admet comme borne inférieure la complexité de Kolmogorov de sa taille, et sa taille comme borne supérieure.

Proposition 1.12. Pour toute chaîne σ , on a $C(|\sigma|) \leq^+ C(\sigma) \leq^+ |\sigma|$. ★

PREUVE. Une chaîne σ fournit toujours au moins comme information la taille de σ , via la machine M qui sur σ renvoie $|\sigma|$. En utilisant M , il est clair que $C(|\sigma|) \leq^+ C(\sigma)$: en effet, si $U(\tau) = \sigma$, alors $M(U(\tau)) = |\sigma|$, et donc une description courte de σ fournit aussi la même description courte de $|\sigma|$.

Par ailleurs, en utilisant la machine identité, qui sur σ renvoie σ , on obtient $C(\sigma) \leq^+ |\sigma|$. ■

Les bornes supérieures et inférieures sont toutes les deux atteintes pour des chaînes arbitrairement grandes. Nous l'avons vu avec la proposition 1.4 pour la borne supérieure, et nous le verrons avec le théorème 1.17 pour la borne inférieure.

1.4.2. La complexité d'une paire d'éléments

Nous présentons ici un outil technique dont nous aurons besoin. Étant donné des chaînes σ et τ , quelle est la taille du plus petit programme permettant de retrouver à la fois σ et τ ? Pour formaliser cela, nous utiliserons la notation $\langle \sigma, \tau \rangle$ pour indiquer un encodage standard des paires de chaînes. Intuitivement, la complexité de $\langle \sigma, \tau \rangle$ peut être au moins aussi grande que la complexité de σ additionnée à celle de τ , en particulier si les chaînes σ et τ ne « partagent pas d'information ». Il est en fait possible de montrer que la complexité peut être encore plus grande que cela. Si σ^* et τ^* sont les plus petites descriptions respectivement pour σ et τ , la chaîne $\sigma^*\tau^*$ ne permettra pas nécessairement de retrouver à la fois σ et τ , car on ne sait pas où « découper » la chaîne $\sigma^*\tau^*$ pour retrouver la description de σ et celle de τ . Nous verrons avec l'exercice 1.15 qu'il n'existe pas de solutions pour contourner ce problème. Il est tout de même possible de ne pas « trop perdre », comme en témoigne la proposition suivante.

Proposition 1.13. Pour toutes chaînes σ, τ , on a

$$C(\langle \sigma, \tau \rangle) \leq^+ C(\sigma) + C(\tau) + 2 \log_2(C(\tau)). \quad \star$$

PREUVE. Soit σ^* et τ^* les plus petites descriptions pour σ et τ , respectivement. Une première idée est d'encoder $\langle \sigma, \tau \rangle$ par la chaîne $1^{|\tau^*|}0\tau^*\sigma^*$. Il suffit alors de concevoir la machine qui lit d'abord tous les bits 1 de son entrée jusqu'à arriver à 0, puis grâce à l'information du nombre n de 1 lus, sait ensuite où découper le reste de la chaîne pour produire τ et σ . On obtient alors $C(\langle \sigma, \tau \rangle) \leq^+ C(\sigma) + 2C(\tau)$.

On peut encore améliorer cela de la manière suivante : étant donné une chaîne ρ qui code pour $|\tau^*|$, et donc telle que $|\rho| = \log_2(C(\tau))$, il suffit

de considérer la chaîne $\bar{\rho}$ qui double tous les bits de ρ , c'est-à-dire qui remplace 0 par 00 et 1 par 11, puis de considérer la chaîne $\bar{\rho}01\tau^*\sigma^*$. Cette fois la machine lit les bits de son entrée jusqu'à arriver à 01, puis dédouble tous les bits pour retrouver un encodage de la taille de τ^* , ce qui lui permet ensuite de savoir où découper le reste de son entrée. ■

Les exercices suivants montrent que l'on perd nécessairement à encoder des paires de chaînes.

Exercice 1.14. (★★) (*Martin-Löf. Voir [120]*). Montrer que pour tout k , toute chaîne σ suffisamment longue — de taille $c+k+2^{c+k}$ pour une certaine constante c — possède un préfixe $\tau \preceq \sigma$ tel que $C(\tau) \leq |\tau| - k$.

Indication.— Utiliser le fait que toute chaîne ρ contient également $|\rho|$ comme information. ◇

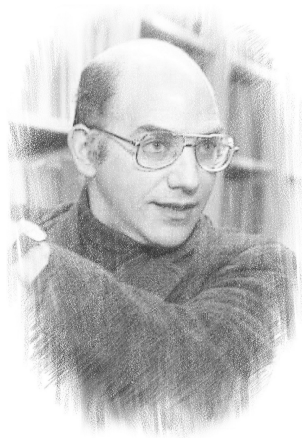
Exercice 1.15. (★★) Dédurre de l'exercice précédent que, pour tout k , il existe des chaînes τ, ρ telles que $C(\tau\rho) \geq C(\tau) + C(\rho) + k$. Dédurre que, pour tout k , il existe des chaînes τ, ρ telles que

$$C(\langle \tau, \rho \rangle) \geq C(\tau) + C(\rho) + k.$$

◇

1.5. Le théorème de Chaitin

Gregory Chaitin (1947 -) est un mathématicien qui contribua en même temps que Kolmogorov et Solomonov à l'essor de la théorie algorithmique de l'aléatoire. Il est à l'origine de plusieurs théorèmes remarquables et d'une très grande profondeur. Son travail le plus célèbre est sans doute la découverte du nombre Ω de Chaitin (voir la définition 2.9), et de ses premières propriétés. Si le résultat que nous présentons ici est moins connu, il n'en est pas moins riche d'enseignements tant sur le plan mathématique que philosophique.



Gregory Chaitin, 1947 -

Supposons $X \subseteq \mathbb{N}$ calculable. Alors, pour tout n , il devrait être à peu près clair que les n premiers bits de X contiennent autant d'information que leur taille, c'est-à-dire n . Formellement, $C(X \upharpoonright_n) =^+ C(n)$: d'après la proposition 1.12, on a $C(n) \leq^+ C(X \upharpoonright_n)$, puis comme X est calculable on

peut facilement créer une machine qui à partir de n produit $X \upharpoonright_n$, ce qui donne $C(X \upharpoonright_n) =^+ C(n)$.

Chaitin est parvenu à démontrer la réciproque de ce résultat : si chaque préfixe d'un ensemble X admet une compression maximale, alors X est un nombre calculable.

Théorème 1.17 (Chaitin [30], Zvonkin et Levin [237], Meyer)
Soit $X \in 2^{\mathbb{N}}$. Alors, X est calculable si, et seulement si, $C(X \upharpoonright_n) =^+ C(n)$ pour tout n .

La preuve repose sur deux lemmes surprenants. Le premier nous dit que pour toute machine M et toute chaîne σ , le nombre de M -compressions « maximales » de σ est borné par une constante qui ne dépend pas de σ . Le second découle du premier et nous dit que le nombre de chaînes de taille n qui admettent une compression « maximale » est bornée par une constante qui ne dépend pas de n .

Lemme 1.18. Pour toute machine M , il existe une constante c telle que, pour tout $\sigma \in 2^{<\mathbb{N}}$ et $d \in \mathbb{N}$, on a :

$$|\{\tau : |\tau| < C(\sigma) + d \wedge M(\tau) = \sigma\}| < 2^{c+d}. \quad \star$$

PREUVE. L'idée est la suivante : étant donné m, k , on peut calculatoirement énumérer la liste $L_{m,k}$ des chaînes qui ont au moins 2^m descriptions de tailles strictement inférieures à k via la machine M . Notez qu'il y a au plus 2^{k-m} éléments dans $L_{m,k}$. À présent, pour décrire un élément de $L_{m,k}$, il suffit d'avoir comme information m, k et une chaîne ρ de taille $k - m$ qui représente sa place dans la liste $L_{m,k}$. En fait, l'information donnée par m et ρ suffit puisque k peut se retrouver à partir de m et ρ . D'après la proposition 1.13, on a

$$C(\langle \rho, m \rangle) \leq^+ C(\rho) + C(m) + 2 \log_2(C(m)) \leq^+ k - m + 2 \log_2(m).$$

On peut donc construire une machine N — qui dépend de M — telle que chaque élément σ de $L_{m,k}$ admet une description de taille inférieure ou égale à $k - m + 2 \log_2(m) + c_0$ via N , pour une certaine constante c_0 indépendante de m, k . On a donc $C_N(\sigma) \leq k - m + 2 \log_2(m) + c_0$, pour tout $m, k \in \mathbb{N}$ et tout $\sigma \in L_{m,k}$.

Soit $c_1 \in \mathbb{N}$ tel que $C \leq C_N + c_1$. En posant

$$c = c_0 + c_1, \quad m = 2^{c+d} \quad \text{et} \quad k = C(\sigma) + d \quad \text{pour une chaîne } \sigma \text{ quelconque,}$$

il suffit de montrer que $\sigma \notin L_{m,k}$. Supposons par l'absurde que $\sigma \in L_{m,k}$. On a donc

$$C_N(\sigma) \leq C(\sigma) + d - 2^{c+d} + 2 \log_2(2^{c+d}) + c_0 \leq C_N(\sigma) + 3(c+d) - 2^{c+d},$$

ce qui donne $2^{c+d} \leq 3(c+d)$. On peut bien entendu supposer $c \geq 4$, ce qui amène à une contradiction. Donc, $\sigma \notin L_{m,k}$, et le lemme est vérifié. ■

Lemme 1.19. Il existe une constante c telle que, pour tous $n, d \in \mathbb{N}$, on a :

$$|\{\sigma : |\sigma| = n \text{ et } C(\sigma) < C(n) + d\}| < 2^{d+c}. \quad \star$$

PREUVE. Il suffit à présent d'appliquer le lemme précédent en remarquant qu'une description courte de la chaîne σ donne aussi une description courte de $|\sigma|$. Supposons qu'il y ait plus de 2^{d+c} chaînes σ de taille n telles que $C(\sigma) < C(n) + d$. Alors, il y a aussi une machine M telle qu'il y a plus de 2^{d+c} chaînes τ de taille inférieure à $C(n) + d$ pour lesquels $M(\tau) = n$, ce qui contredit le lemme précédent. ■

PREUVE DU THÉORÈME 1.17. L'idée est la suivante : étant donné d fixé, l'ensemble $T = \{\sigma : \forall \tau \preceq \sigma \ C(\tau) \leq C(|\tau|) + d\}$ est un arbre. D'après le lemme 1.19, il existe c tel que pour tout n cet arbre contient au plus 2^{c+d} chaînes de taille n . En particulier, $[T]$ est fini, et d'après la proposition 8-3.6, tous les chemins de T sont donc calculables relativement à la connaissance de T .

Le problème est que l'ensemble T lui-même n'est pas calculable, car C ne l'est pas. Nous allons donc le transformer de la manière suivante : nous remarquons d'abord que, d'après la proposition 1.4, il existe pour tout k un entier n entre 2^k et $2^{k+1} - 1$ tel que $\log_2(n) \leq C(n)$. Par ailleurs, d'après la proposition 1.12, il existe une constante e pour laquelle on a toujours $C(n) \leq \log_2(n) + e$.

Nous utilisons ces informations pour faire une première transformation de T :

$$T_1 = \{\sigma : \forall \tau \preceq \sigma \ C(\tau) \leq \log_2(|\tau|) + e + d\}.$$

Étant donné que $C(n) \leq \log_2(n) + e$, on a $T \subseteq T_1$. Étant donné que pour tout k il existe un entier n entre 2^k et $2^{k+1} - 1$ tel que $\log_2(n) \leq C(n)$, pour tout k on dispose aussi d'après le lemme 1.19 d'un entier n entre 2^k et $2^{k+1} - 1$ tel que $|\{\sigma \in T_1 : |\sigma| = n\}| < 2^{e+d+c}$.

En particulier, $|[T_1]| < 2^{e+d+c}$. L'arbre T_1 n'est toutefois toujours pas calculable, mais simplement c. e. Soit a_k défini pour tout k comme le minimum pour n compris entre 2^k et $2^{k+1} - 1$, du nombre de noeuds de taille n dans l'arbre T_1 . Notons que l'on a nécessairement $a_k < 2^{e+d+c}$. Soit alors $a = \limsup_{k \in \mathbb{N}} a_k$, et soit m le plus petit entier tel que $a_k \leq a$ pour tout $k \geq m$. On peut à présent chercher de manière effective un temps de calcul s_0 et un entier $k_0 > m$ tels que pour tout n compris entre 2^{k_0} et $2^{k_0+1} - 1$, au moins a noeuds de taille n sont énumérés dans $T_1[s_0]$. Soit $X \in [T_1]$. Nous prétendons que $X \upharpoonright_{2^{k_0}} \in T_1[s_0]$. En effet, dans le cas contraire, quand $X \upharpoonright_{2^{k_0+1}}$ et ses préfixes seront énumérés dans T_1 , cela rajoutera un noeud de taille n qui n'est pas dans $T_1[s_0]$, pour tout n compris entre 2^{k_0} et $2^{k_0+1} - 1$. Le nombre de noeuds de taille n dépasserait donc a , ce qui contredit le fait que $a_k \leq a$ pour tout $k \geq m$. Donc, $T_1[s_0]$ contient tous les préfixes de taille 2^{k_0} des éléments de T_1 .

Une fois s_i et k_i définis, on calcule de la même façon inductivement $s_{i+1} > s_i$ et $k_{i+1} > k_i$ tels que pour tout n compris entre $2^{k_{i+1}}$ et $2^{k_{i+1}+1} - 1$, au moins a noeuds de taille n sont énumérés dans $T_1[s_{i+1}]$. Par le même argument que dans le paragraphe précédent, tout $X \in [T_1]$ a son préfixe de taille $2^{k_{i+1}}$ énuméré dans $T_1[s_{i+1}]$.

Cela nous permet de définir un arbre calculable T_2 tel que $[T_2] = [T_1]$. En particulier, $|[T_2]| < 2^{e+d+c}$ et T_2 contient tous les ensembles X tels que $C(X \upharpoonright_n) \leq C(n) + d$, pour tout n . Tous ces ensembles sont donc calculables.

2. Nombres aléatoires à la Chaitin/Levin

La définition originelle de la complexité de Kolmogorov souffre de certains problèmes. D'après l'exercice 1.10, on n'a pas nécessairement

$$C(\langle \sigma, \tau \rangle) \leq^+ C(\sigma) + C(\tau),$$

pour toutes chaînes σ, τ . D'après l'exercice 1.14, pour tout entier k et toute chaîne σ suffisamment longue, on a l'inégalité $C(\tau) \leq |\tau| - k$, pour un certain préfixe $\tau \preceq \sigma$. Cela nous empêche d'utiliser C pour définir l'aléatoire sur les suites infinies. Il est nécessaire pour cela d'utiliser une variation de C : la complexité dite « sans préfixe ».

2.1. Complexité sans préfixe

La complexité sans préfixe fut introduite indépendamment par Levin [139] et Chaitin [29]. Elle est définie de la même manière que la complexité de Kolmogorov, mais en se restreignant aux machines dites « sans préfixe », que nous introduisons à présent.

Définition 2.1. Un ensemble $W \subseteq 2^{<\mathbb{N}}$ est *sans préfixe* si $\sigma \in W$ implique $\tau \notin W$, pour toute chaîne $\tau \succ \sigma$. Une machine M est *sans préfixe* si son domaine de définition est sans préfixe, c'est-à-dire que, pour toute chaîne σ , on a $M(\sigma) \downarrow$ implique $M(\tau) \uparrow$ pour toute chaîne $\tau \succ \sigma$. $\diamond$

Une machine M sans préfixe peut être vue comme une machine sur laquelle s'exécutent des programmes (les chaînes que M prend en paramètre) *délimités par une instruction de fin* : si σ est un programme valide, alors aucune extension de σ n'est valide.

Notons que dans la réalité, les programmes informatiques tendent à former des ensembles sans préfixe. Dans n'importe quel langage de programmation, les fonctions ont un symbole de début et de fin. Quand bien même ça ne serait pas le cas, les systèmes de fichiers utilisés pour enregistrer les

programmes en binaire sont sans préfixe (comment savoir sinon où s'arrête un fichier!). Voyons à présent un théorème analogue au théorème de la machine universelle pour les machines arbitraires.

Théorème 2.2

Il existe une machine sans préfixe qui est universelle pour les machines sans préfixe.

PREUVE. Soit $(M_e)_{e \in \mathbb{N}}$ une énumération des machines. Commençons par montrer qu'il existe une fonction calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que :

- ▷ pour tout $e \in \mathbb{N}$, la machine $M_{f(e)}$ est sans préfixe ;
- ▷ si M_e est sans préfixe, alors $M_{f(e)} = M_e$.

Étant donné un code e , la fonction f calcule le code de la machine qui sur une entrée σ cherche le plus petit $t \geq |\sigma|$ tel que $M_e(\sigma)[t] \downarrow$. Si un tel t est trouvé, alors s'il existe $s < t$ tel que $M_e(\tau)[s] \downarrow$, pour $\tau \neq \sigma$ comparable avec σ et tel que $|\tau| \leq s$, on décide que la machine ne s'arrête pas, et donc $M_{f(e)}(\sigma) \uparrow$. Dans le cas contraire, si $M_e(\tau)[t] \downarrow$ pour $\tau \prec \sigma$, là encore on décide $M_{f(e)}(\sigma) \uparrow$. Sinon, on décide $M_{f(e)}(\sigma) \downarrow = M_e(\sigma)[t] \downarrow$.

Il est clair que si M_e est sans préfixe, alors $M_{f(e)} = M_e$. Il est clair aussi que $M_{f(e)}$ est toujours sans préfixe : supposons par l'absurde que $M_{f(e)}(\sigma) \downarrow$ et $M_{f(e)}(\tau) \downarrow$ pour $\sigma \prec \tau$. Soit $t_\sigma \geq |\sigma|, t_\tau \geq |\tau|$ les plus petits temps de calcul tels que $M_e(\sigma)[t_\sigma] \downarrow$ et $M_e(\tau)[t_\tau] \downarrow$. Si $t_\sigma = t_\tau$, on a par définition $M_{f(e)}(\tau) \uparrow$. Si $t_\tau < t_\sigma$, on a par définition $M_{f(e)}(\sigma) \uparrow$, et symétriquement si $t_\sigma < t_\tau$.

On définit à présent la machine U suivante : $U(1^e 0 \sigma) = M_{f(e)}(\sigma)$. Notons que comme chaque machine $M_{f(e)}$ est sans préfixe, alors la machine U l'est également. ■

Notation

De manière générale si M est une machine sans préfixe on écrira $K_M(\sigma)$ à la place de $C_M(\sigma)$. On fixe une machine sans préfixe universelle U . On note $K(\sigma)$ pour la valeur $K_U(\sigma)$. On dira que $K(\sigma)$ est la *complexité sans préfixe* de σ , et par opposition que $C(\sigma)$ est sa *complexité pleine*.

Certains théorèmes restent vrais pour la complexité sans préfixe. On adapte par exemple sans mal la preuve de la proposition 1.8 pour montrer que la connaissance d'une machine universelle sans préfixe permet de calculer $\emptyset'$. En revanche, les différentes bornes varient un peu.

Exercice 2.3. (★) Montrer qu'aucune fonction calculable $f : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ — même partielle — ne peut augmenter la complexité de ses paramètres : montrer pour une telle fonction qu'il existe une constante c telle que $K(f(\sigma)) \leq K(\sigma) + c$, pour tout $\sigma \in 2^{<\mathbb{N}}$ sur laquelle f est définie. ◇

2.2. Propriétés numériques de la complexité sans préfixe

Une première spécificité, bien pratique, des machines sans préfixe est de pouvoir coder les paires de chaînes de manière optimale.

Proposition 2.4. Pour toutes chaînes σ, τ , on a

$$K(\langle \sigma, \tau \rangle) \leq^+ K(\sigma) + K(\tau). \quad \star$$

PREUVE. Soit U la machine sans préfixe universelle. On définit la machine sans préfixe M qui sur une chaîne σ cherche des chaînes τ, ρ telles que $\tau\rho = \sigma$, telles que $U(\tau) \downarrow$ et telles que $U(\rho) \downarrow$. la machine renvoie alors $\langle U(\tau), U(\rho) \rangle$. Notons que si un préfixe $\tau \preceq \sigma$ tel que $U(\tau) \downarrow$ existe, celui-ci est unique car U est sans préfixe.

À présent, si σ^* et τ^* sont les plus petits programmes tels que $U(\sigma^*) = \sigma$ et $U(\tau^*) = \tau$ pour des chaînes σ, τ arbitraires, alors $M(\sigma^*\tau^*) = \langle \sigma, \tau \rangle$, et donc $K(\langle \sigma, \tau \rangle) \leq^+ K_M(\langle \sigma, \tau \rangle) \leq K(\sigma) + K(\tau)$. ■

Tout comme on avait $C(|\sigma|) \leq^+ C(\sigma)$, on a ici toujours $K(|\sigma|) \leq^+ K(\sigma)$. En revanche, on ne peut plus forcément garantir $K(\sigma) \leq^+ |\sigma|$, la machine identité n'étant pas une machine sans préfixe. On a $K(\sigma) \leq^+ |\sigma| + K(|\sigma|)$, à la place.

Proposition 2.5. Pour toute chaîne σ , on a

$$K(|\sigma|) \leq^+ K(\sigma) \leq^+ |\sigma| + 2\log_2(|\sigma|). \quad \star$$

PREUVE. Soit U la machine universelle sans préfixe. Montrons la première inégalité. La machine M qui à τ associe $|U(\tau)|$ est elle aussi sans préfixe. Supposons $U(\tau) = \sigma$. Alors, $M(\tau) = |\sigma|$. On a donc $K(|\sigma|) \leq^+ K(\sigma)$.

Montrons à présent $K(\sigma) \leq^+ |\sigma| + 2\log_2(|\sigma|)$. On utilise la même technique que pour la preuve de la proposition 1.13 : étant donné une chaîne ρ , soit $\bar{\rho}$ la chaîne qui double tous les bits de ρ , c'est-à-dire qui remplace 0 par 00 et 1 par 11. On définit la machine M qui s'arrête sur les chaînes de la forme $\bar{\rho}01\sigma$ telles que ρ code pour la taille de σ , et renvoie alors σ . Notons que la machine M est sans préfixe : si $M(\bar{\rho}01\sigma) \downarrow$, cela signifie que ρ est un encodage de la taille de σ , et si à présent $M(\bar{\rho}01\sigma') \downarrow$ pour $\sigma' \prec \sigma$ ou $\sigma \prec \sigma'$, cela signifierait aussi que ρ encode $|\sigma'| \neq |\sigma|$, ce qui est impossible. On obtient enfin l'inégalité $K(\sigma) \leq^+ |\sigma| + 2\log_2(|\sigma|)$ via la machine M en utilisant le fait que la représentation binaire d'un entier et de l'ordre du $\log_2$ de cet entier. ■

Exercice 2.6. (*) (Chaitin [29]). Montrer que l'on a également

$$K(\sigma) \leq^+ |\sigma| + K(|\sigma|), \quad \text{pour toute chaîne } \sigma.$$

En quoi est-ce une amélioration de la proposition 2.5 ? ◇

Exercice 2.7. (★) Montrer que, pour tout c , il existe une chaîne σ telle que $K(\sigma) > |\sigma| + c$. $\diamond$

2.3. Nombres aléatoires au sens de Chaitin/Levin

Levin, le premier, et, indépendamment, Chaitin ont tous les deux utilisé la complexité sans préfixe pour définir l'aléatoire pour des objets infinis.

Définition 2.8 ([142], [31]). Un ensemble $X \in 2^{\mathbb{N}}$ est *aléatoire au sens de Chaitin/Levin* si $K(X \upharpoonright_n) \geq^+ n$, pour tout n . $\diamond$

En d'autres termes, une suite infinie de 0 et de 1 est aléatoire si aucun de ses préfixes n'est compressible, à constante additive près. Notons que pour une telle définition, la complexité sans préfixe est nécessaire : comme nous l'avons déjà mentionné, l'exercice 1.14 montre qu'aucun ensemble $X \in 2^{\mathbb{N}}$ ne satisfait $C(X \upharpoonright_n) \geq^+ n$ pour tout n . En revanche, nous pouvons montrer qu'un ensemble est aléatoire au sens de Chaitin/Levin avec probabilité 1. Nous reviendrons sur ce point dans le chapitre 17 après avoir défini formellement les notions de boréliens et de mesure.

Nous verrons que la définition de Chaitin et Levin est une « bonne » définition de l'aléatoire, et ce selon plusieurs critères imaginables. En particulier, un nombre aléatoire pour cette définition satisfait la loi des grands nombres. De plus, si un nombre présente une répétition, une incongruité, un « motif », il sera possible de créer une machine utilisant cela pour compresser même un peu les préfixes de ce nombre. Nous verrons en fait, pour aussi formelle que puisse être cette affirmation, qu'un nombre aléatoire pour cette définition satisfait toutes les lois de probabilité « naturelles » qui sont satisfaites avec probabilité 1.

Avant de voir cela plus en détail, nous présentons dès à présent le plus fameux résultat de Chaitin, que l'on peut résumer ainsi : la probabilité qu'un programme informatique s'arrête est un nombre aléatoire.

2.4. Le nombre Ω de Chaitin

Qu'entend-on précisément par « la probabilité qu'un programme informatique s'arrête » ? Considérons une machine sans préfixe universelle U . Supposons que l'on tire à pile ou face sans jamais s'arrêter pour déterminer un programme σ sur lequel U s'exécutera. Le tirage en question produit des chaînes $\sigma_1 \prec \sigma_2 \prec \sigma_3 \prec \dots$ avec $|\sigma_i| = i$. En procédant ainsi, la probabilité pour que l'on obtienne σ_i tel que $U(\sigma_i) \downarrow$, est un nombre aléatoire : écrit en base 10 par exemple, la fréquence d'apparition des chiffres 0, 1, 2, $\dots$, 9 dans les préfixes du développement décimal de ce nombre convergera vers 1/10. La fréquence d'apparition des nombres à deux chiffres convergera vers 1/100, etc. Définissons à présent formellement ce nombre.

Définition 2.9. Le nombre Ω de Chaitin est défini par :

$$\Omega = \sum_{\{\sigma : U(\sigma) \downarrow\}} 2^{-|\sigma|}.$$

◇

Etant donné que U est une machine sans préfixe, le nombre Ω est inférieur à 1. Pour voir cela, nous introduisons informellement le concept de mesure pour des intervalles $]a, b[\subseteq [0, 1]$ comme étant simplement ce que l'on mesurerait avec une règle graduée : la valeur $b - a$. Une chaîne σ peut se voir comme l'intervalle des réels compris entre $0.\sigma 0^\infty$ et $0.\sigma 1^\infty$. On vérifie facilement que la mesure de l'intervalle $]0.\sigma 0^\infty, 0.\sigma 1^\infty[$ est de $2^{-|\sigma|}$. Le fait que le domaine de définition de U est sans préfixe correspond au fait que les intervalles de la forme $]0.\sigma 0^\infty, 0.\sigma 1^\infty[$ tels que $U(\sigma) \downarrow$ sont deux à deux disjoints. La somme de leur mesure correspond donc à la mesure de leur réunion, qui est nécessairement inférieure à la mesure de $[0, 1]$, à savoir 1.

On considère dans la suite le nombre Ω comme la représentation en binaire de son développement décimal. Remarquons d'abord que Ω est approchable par la gauche.

Définition 2.10. Un ensemble $X \in 2^\mathbb{N}$ est *approchable par la gauche* s'il existe une suite calculable $(X_s)_{s \in \mathbb{N}}$ telle que X_s est lexicographiquement plus petit que X_{s+1} pour tout $s \in \mathbb{N}$, et pour laquelle $X = \lim_s X_s$. ◇

De manière équivalente, $X \in 2^\mathbb{N}$ est approchable par la gauche s'il existe une suite croissante calculable de réels dont la représentation binaire converge vers X . Concrètement, dans une approximation par la gauche de X , un bit peut passer de 0 à 1 — ce qui correspond à une augmentation de la valeur du réel — comme pour un ensemble c. e., et un bit peut aussi passer de 1 à 0, mais seulement si un bit de poids plus fort passe au même moment de 0 à 1 — ce qui correspond aussi à une augmentation de la valeur du réel. Par exemple, le nombre 0,010000 est plus grand que le nombre 0,000101.

Exercice 2.11. Montrer que toute classe Π_1^0 non vide $\mathcal{P}$ contient un élément approchable par la gauche. ◇

Il est clair que les ensembles approchables par la gauche sont tous $\emptyset'$ -calculable : il s'agit d'un cas particulier du lemme limite 4-7.2 de Shoenfield.

Proposition 2.12. Le nombre Ω est approchable par la gauche. En particulier, Ω est calculable en l'arrêt des programmes informatiques. ★

PREUVE. Définissons une suite calculable de réels calculables $(\Omega_s)_{s \in \mathbb{N}}$ tels que $\Omega_s \leq \Omega_{s+1}$ et $\Omega = \lim_s \Omega_s$. Chaque réel Ω_s est simplement

$$\sum_{\{\sigma : |\sigma| \leq s \wedge U(\sigma)[s] \downarrow\}} 2^{-|\sigma|}.$$

Il est clair que $\Omega_s \leq \Omega_{s+1}$, pour tout s , et que $\Omega = \lim_s \Omega_s$. ■

Théorème 2.13 (Chaitin [29])

Les n premiers bits de Ω suffisent pour déterminer uniformément quelles sont les chaînes σ de taille inférieure ou égale à n telles que $U(\sigma) \downarrow$. En particulier, $\Omega \equiv_T \emptyset'$.

PREUVE. Soit $(\Omega_s)_{s \in \mathbb{N}}$ l'approximation par la gauche de Ω décrite dans la preuve de la proposition 2.12. Supposons que l'on connaisse les n premiers bits de Ω . Alors, il suffit de chercher le plus petit s tel que $\Omega_s \upharpoonright_{n+1} = \Omega \upharpoonright_{n+1}$. Remarquons alors qu'un programme σ de taille $m \leq n$ tel que $U(\sigma)[s] \uparrow$ doit forcément être tel que $U(\sigma) \uparrow$. En effet, dans le cas inverse, on devrait ajouter la quantité 2^{-m} à Ω_s , ce qui ferait nécessairement passer un bit de position inférieure ou égale à m de 0 vers 1, et contredirait $\Omega_s \upharpoonright_{n+1} = \Omega \upharpoonright_{n+1}$. Pour savoir si $U(\sigma) \downarrow$ sur une chaîne σ de taille inférieure ou égale à n , il suffit de regarder si $U(\sigma)[s] \downarrow$.

En particulier, Ω peut calculer U , et d'après une adaptation de la proposition 1.8 à la complexité sans préfixe, Ω peut donc calculer $\emptyset'$. ■

Théorème 2.14 (Chaitin [29])

Le nombre Ω est aléatoire au sens de Chaitin/Levin.

PREUVE. Supposons par l'absurde que, pour tout c , il existe n tel que

$$K(\Omega \upharpoonright_n) < n - c.$$

Construisons la machine M qui sur la chaîne τ calcule $\sigma = U(\tau)$, et cherche ensuite à déterminer l'ensemble A_τ des chaînes de taille inférieure ou égale à $|\sigma|$ sur lesquelles la machine U s'arrête, en appliquant l'algorithme décrit dans le théorème 2.13, comme si $\sigma \prec \Omega$. Si jamais M pense avoir déterminé A_τ au bout d'un certain temps de calcul, alors M renvoie la première chaîne qui est différente de $U(\rho)$ pour toute chaîne $\rho \in A_\tau$.

Notons d'abord que si M est appliqué à une chaîne τ telle que $U(\tau) = \sigma \prec \Omega$, alors M s'arrêtera et sortira une chaîne ρ dont la complexité de Kolmogorov est plus grande que $|\sigma|$. Dans le même temps, on a $K_M(\rho) \leq |\tau|$. Soit c tel que $K \leq K_M + c$. Par hypothèse, il existe n tel que $K(\Omega \upharpoonright_n) < n - c$. Il existe donc τ telle que $|\tau| < n - c$ et tel que $U(\tau) = \Omega \upharpoonright_n$. On a alors par construction $M(\tau) = \rho$ pour une chaîne ρ telle que $n = |\Omega \upharpoonright_n| \leq K(\rho)$. Cela nous donne $n \leq K(\rho) \leq K_M(\rho) + c < n - c + c < n$, ce qui est une contradiction.

Donc, il existe c tel que pour tout n on a $K(\Omega \upharpoonright_n) \geq n - c$, ce qui signifie que Ω est aléatoire au sens de Chaitin/Levin. ■

Le nombre Ω de Chaitin a suscité un fort intérêt au sein de la communauté. Solovay dans une série de notes non publiées sur le travail de Chaitin [212] a introduit une notion clef qui devait en permettre une étude approfondie : pour deux réels X, Y approchables par la gauche, X est *Solovay réductible* à Y s'il existe une fonction calculable $f : \mathbb{Q} \rightarrow \mathbb{Q}$ et une constante c telles que pour un rationnel $q < Y$, on a $0 \leq X - f(q) < c(Y - q)$. En d'autres termes, une approximation rationnelle proche de Y permet de calculer une approximation rationnelle aussi proche de X , à constante multiplicative près. La réducibilité Solovay implique en particulier la réducibilité Turing. Solovay a ensuite montré que Ω était complet pour la réducibilité Solovay, c'est-à-dire que n'importe quel réel approchable par la gauche était Solovay réductible à Ω . Par la suite, Calude, Hertling, Khoussainov et Wang [26] ont montré qu'un réel approchable par la gauche et Solovay complet était nécessairement de la forme Ω pour une certaine machine sans préfixe universelle. Puis, Kučera et Slaman [127] ont finalement montré qu'un réel approchable par la gauche et aléatoire au sens de Chaitin/Levin était forcément Solovay complet, ce qui mis bout à bout nous donne le théorème suivant.

Théorème 2.15

Un ensemble $X \in 2^{\mathbb{N}}$ aléatoire au sens de Chaitin/Levin est approchable par la gauche si, et seulement si, il est la probabilité qu'une certaine machine universelle sans préfixe s'arrête sur son entrée.

3. Caractérisation de K

L'objectif de cette section est de montrer le théorème suivant, qui constitue une caractérisation élégante de la complexité sans préfixe.

Théorème 3.1 (Chaitin [31])

La fonction $\sigma \mapsto K(\sigma)$ est, à constante additive près, la plus petite fonction $f : 2^{\mathbb{N}} \rightarrow \mathbb{N}$ approchable par le dessus et telle que $\sum_{\sigma} 2^{-f(\sigma)} \leq 1$.

Le cœur de cette caractérisation réside dans le théorème KC, dit de Kraft-Chaitin, permettant de construire des machines sans préfixe à partir d'un ensemble de requêtes abstrait, dit « ensemble borné de requêtes ».

Définition 3.2

On appelle *ensemble borné de requêtes* tout ensemble $A \subseteq 2^{<\mathbb{N}} \times \mathbb{N}$ tel que

$$\sum_{(\sigma, l) \in A} 2^{-l} \leq 1,$$

et l'on note $\text{poids}(A)$ cette quantité.



Nous réutiliserons de manière intensive dans la suite le concept d'ensemble borné de requêtes et le théorème KC, démontré indépendamment par Levin [139], Schnorr [193] et Chaitin [29].

Théorème 3.3 (Théorème KC)

Soit A un ensemble borné de requêtes c. e. Alors, il existe une machine sans préfixe M telle que $(\sigma, l) \in A$ implique $M(\tau) = \sigma$ pour une chaîne τ telle que $|\tau| = l$. En particulier, $K_M(\sigma) \leq l$.

PREUVE. Le lecteur pourra s'aider de la figure 3.4 pour comprendre la construction qui suit. Supposons sans perte de généralité que A est un ensemble infini. Soit $(A_s)_{s \in \mathbb{N}}$ une approximation de A . Notons que l'on a donc forcément $\text{poids}(A_s) < 1$ pour tout s .

Nous allons décrire dans ce qui suit une machine M en tant qu'ensemble c. e. de couples (σ, τ) , signifiant alors $M(\sigma) = \tau$. On écrira M_s pour l'énumération de M à l'étape s . À chaque étape de calcul s , pour chaque taille $l \geq 1$, on définit une chaîne σ_s^l , soit de taille l , soit égale au mot vide ϵ , et un ensemble $r_s \in 2^{\mathbb{N}}$. Les chaînes σ_s^l différentes du mot vide correspondront aux chaînes disponibles pour une association à l'étape $s + 1$. Le rôle de la suite $(r_s)_{s \in \mathbb{N}}$ est double. D'abord, le réel représenté par r_s en base 2 sera égal au poids de A_s . Ensuite, si le $(n - 1)$ -ième bit de r_s est 0, cela signifie aussi que la chaîne σ_s^n est différente de ϵ , et est donc disponible pour une future association. On doit s'assurer à chaque étape s que les propriétés suivantes sont satisfaites.

- (1) L'ensemble des chaînes associées dans M_s forment avec l'ensemble des chaînes σ_s^l différentes du mot vide, un ensemble sans préfixe.
- (2) Le réel r_s vue comme la représentation binaire d'un réel est égale à $\text{poids}(A_s)$.
- (3) Si $r_s(n - 1) = 0$, alors la chaîne σ_s^n est une chaîne de longueur n . Sinon, c'est le mot vide.

À l'étape 0, on définit $\sigma_0^l = 1^{l-1}0$ et r_0 comme la suite infinie de 0. Les propriétés (1), (2) et (3) sont vérifiées à l'étape 0.

À l'étape $s + 1$, supposons que (τ, l) soit énuméré dans A_{s+1} . Si $r_s(l - 1) = 0$, on énumère (σ_s^l, τ) dans M à l'étape $s + 1$, on assigne σ_{s+1}^l au mot vide et l'on change $r_{s+1}(l - 1)$ à 1. Pour $i \neq l$, on garde $r_{s+1}(i - 1) = r_s(i - 1)$ et $\sigma_{s+1}^i = \sigma_s^i$. On vérifie facilement par induction que les propriétés (1), (2) et (3) sont vraies à l'étape $s + 1$.

PREUVE DU THÉORÈME 3.1. Soit $f : 2^{<\mathbb{N}} \rightarrow \mathbb{N}$ une fonction approchable par le dessus et telle que $\sum_{\sigma} 2^{-f(\sigma)} \leq 1$. Supposons dans un premier temps que f est calculable. Il nous suffit alors d'énumérer chaque couple $(\sigma, f(\sigma))$ dans un ensemble borné de requêtes. D'après le théorème KC, il existe une machine M telle que $K(\sigma) \leq^+ K_M(\sigma) \leq f(\sigma)$.

À présent, si la fonction f est approchable par le dessus via une approximation $(f_s)_{s \in \mathbb{N}}$, on énumère dans un ensemble borné de requêtes L les couples $(\sigma, f_0(\sigma) + 1)$, puis les couples $(\sigma, f_{s+1}(\sigma) + 1)$ pour chaque σ et chaque s tel que $f_{s+1}(\sigma) < f_s(\sigma)$. Soit A_0 l'ensemble des paires $\langle s, \sigma \rangle$ telles que s est le dernier temps de calcul pour lequel $f_s(\sigma) > f_{s+1}(\sigma) = f(\sigma)$, puis soit récursivement A_{n+1} l'ensemble des paires $\langle s, \sigma \rangle$ telles que s est le dernier temps de calcul pour lequel $f_s(\sigma) > f_{s+1}(\sigma) = \dots = f_{s+t}(\sigma)$ pour $\langle s+t, \sigma \rangle \in A_n$. On a

$$\text{poids}(L) = \sum_{\sigma} 2^{-f(\sigma)-1} + \sum_n \sum_{\langle s, \sigma \rangle \in A_n} 2^{-f_s(\sigma)-1}.$$

Par hypothèse sur f ,

$$\delta_0 = \sum_{\langle s, \sigma \rangle \in A_0} 2^{-f_s(\sigma)-1} \leq \sum_{\sigma} 2^{-f(\sigma)-1-1} \leq 1/4.$$

Puis, pour tout n ,

$$\delta_{n+1} = \sum_{\langle s, \sigma \rangle \in A_{n+1}} 2^{-f_s(\sigma)-1} \leq \sum_{\langle s, \sigma \rangle \in A_n} 2^{-f_s(\sigma)-1-1} = \frac{1}{2} \sum_{s \in A_n, \sigma} 2^{-f_s(\sigma)-1} = \frac{1}{2} \delta_n.$$

Le poids de notre ensemble de requêtes est donc borné par le nombre réel $\sum_{\sigma} 2^{-f(\sigma)-1} + 1/4 + 1/8 + \dots \leq 1$. On peut donc construire une machine M via le théorème KC telle que $K(\sigma) \leq^+ K_M(\sigma) \leq f(\sigma) + 1$. ■

On utilisera parfois le théorème 3.1 sous la forme suivante, connue sous le nom de « coding theorem » en anglais : pour toute machine sans préfixe M , la probabilité d'obtenir σ via M est toujours bornée, à constante multiplicative près, par $2^{-K(\sigma)}$.

Théorème 3.5 (Chaitin [29], Levin [139, 142])

Soit M une machine sans préfixe. On dénote par $P_M(\sigma)$ la probabilité que M écrive σ , c'est-à-dire

$$P_M(\sigma) = \sum_{\{\tau : M(\tau) \downarrow = \sigma\}} 2^{-|\tau|}.$$

Alors, il existe une constante c_M telle que, pour tout σ , on ait

$$P_M(\sigma) \leq 2^{-K(\sigma)} \times 2^{c_M}.$$

PREUVE. Il suffit de remarquer que la fonction $\sigma \mapsto -\log_2(P_M(\sigma))$ est approchable par le dessus et vérifie $\sum_{\sigma} 2^{\log_2(P_M(\sigma))} \leq 1$. On a donc une constante c_M telle que $K(\sigma) \leq -\log_2(P_M(\sigma)) + c_M$, pour tout σ . ■

4. Ensembles K-triviaux

Souvenons-nous à présent de la remarquable caractérisation de Chaitin des ensembles calculables comme étant ceux dont la complexité de Kolmogorov des préfixes est toujours minimale. Nous introduisons ici la même notion, mais pour la complexité préfixe.

Définition 4.1. Soit $A \subseteq \mathbb{N}$. L'ensemble A est *K-trivial* si, pour tout n ,

$$K(A \upharpoonright_n) \leq^+ K(n).$$

◇

Chaitin essaya d'adapter sa preuve pour montrer que les K-triviaux sont tous calculables, mais sans succès. Il réussit tout de même à démontrer que les K-triviaux sont en quantité dénombrable et en particulier tous Δ_2^0 . Solovay, dans sa série de notes non publiées consacrées à l'aléatoire algorithmique, démontra l'existence d'ensembles K-triviaux non calculables. Nous utiliserons pour le montrer une méthode pratique permettant de construire facilement des degrés c. e. incalculables : les ensembles dits *simples*.

4.1. Ensembles simples

Les ensembles simples furent introduits par Emil Post dans sa quête de degrés c. e. incomplets (voir le chapitre 12). En particulier, Post a montré que tous les ensembles simples étaient incomplets pour la réduction many-one.

Définition 4.2. Un ensemble A est *simple* s'il est c. e., de complémentaire infini, et si $A \cap W_e \neq \emptyset$ pour tout ensemble c. e. infini W_e . ◇

Autrement dit, un ensemble A est simple s'il est c. e. et son complémentaire est infini et immune.

Proposition 4.3. Un ensemble simple n'est pas calculable. ★

PREUVE. Soit A un ensemble simple. Comme le complémentaire de A est infini, et comme A intersecte tout ensemble c. e. infini, le complémentaire de A ne peut pas être c. e. D'après la proposition 3-7.4, l'ensemble simple A ne peut donc pas être calculable. ■

La proposition suivante donne une technique pour construire un ensemble simple, qui sera utilisée à plusieurs reprises dans l'étude de différentes notions d'aléatoire, et en particulier pour construire un ensemble K-trivial non calculable.

Proposition 4.4. Il existe un ensemble simple. ★

PREUVE. Soit $(W_e)_{e \in \mathbb{N}}$ une énumération des ensembles c. e. On énumère un ensemble simple A . À l'étape de calcul t , pour tout $e \leq t$ tel que A n'intersecte pas encore W_e à l'étape t , on regarde s'il existe un élément x dans $W_e[t]$ tel que $x > 2e$. Si c'est le cas, on énumère x dans A .

Il est clair que si W_e est infini, alors $A \cap W_e \neq \emptyset$. Montrons à présent que $\mathbb{N} \setminus A$ est infini. Pour tout W_e , on énumère au plus un élément dans A . De plus, pour tout entier $n = 2e$, seuls les ensembles W_a pour $a < e$ peuvent énumérer un élément plus petit que n dans A . Il y a donc au plus $n/2$ éléments plus petits que n dans A , et ce, pour tout n . Le complémentaire de A est donc infini. ■

4.2. Le théorème de Solovay

Nous sommes maintenant prêts à montrer que les K-triviaux ne coïncident pas avec les C-triviaux, autrement dit qu'il existe des K-triviaux non calculables.

Théorème 4.5 (Solovay [212])

Il existe des K-triviaux c. e. non calculables.

PREUVE. Le lecteur peut s'aider de la figure 4.6 pour suivre la preuve. Soit $(W_e)_{e \in \mathbb{N}}$ une énumération des ensembles c. e. Nous allons construire un ensemble A simple — avec la même technique que celle de la preuve de la proposition 4.4 — et K-trivial. Afin de garantir que A soit K-trivial, on va créer nous même notre machine M telle que $K_M(A \upharpoonright_n) \leq^+ K(n)$ pour tout n . On utilisera pour cela un ensemble borné de requêtes L .

On notera A_t et L_t les résultats des énumérations de A et L à l'étape t . À l'étape de calcul 0, les ensembles A_0 et L_0 sont vides. À l'étape de calcul $t + 1$, on procède comme suit.

- (1) Pour tout $\langle n, \tau \rangle \leq t + 1$, si $t + 1$ est le plus petit tel que $U(\tau)[t + 1] = n$, on énumère alors $(A_t \upharpoonright_{n+1}, |\tau| + 1)$ dans L à l'étape $t + 1$.
- (2) Puis, pour tout $e \leq t + 1$ tel que A_t n'intersecte pas $W_e[t + 1]$, on regarde s'il existe $n \geq 2e$ tel que $n \in W_e[t + 1]$ et tel que

$$\sum_{\substack{U(\tau)[t+1] \downarrow = m, \\ \text{avec } m \geq n}} 2^{-|\tau|} < 2^{-e-2}.$$

Si l'on trouve un tel entier n , on énumère alors n dans A à l'étape $t+1$. Puis, pour toute chaîne τ telle que $U(\tau)[t+1] = m$ pour $m \geq n$, on énumère $(A_{t+1} \upharpoonright_{m+1}, |\tau|)$ dans L à l'étape $t+1$.

Cela termine la construction.

Assurons-nous d'abord que L est bien un ensemble borné de requêtes. Le poids de L qui vient de (1) est forcément inférieur à $1/2$, car dans L on augmente par rapport à U la longueur de chaque description de 1. Enfin, pour chaque e , le poids de L qui provient de (2), c'est-à-dire de ce que l'on rajoute après avoir énuméré dans A un élément de W_e , est forcément inférieur à 2^{-e-2} , par construction. Le poids total pour tous les entiers e est donc inférieur à $\sum_{e \in \mathbb{N}} 2^{-e-2} = 1/2$. Le poids total de L est donc borné par 1, ce qui implique que L est un ensemble borné de requêtes.

D'après le théorème KC (3.3), on peut à présent définir une machine M telle que $(\sigma, l) \in L$ implique $K_M(\sigma) \leq l$. Il est clair par construction que $K(n) \leq l$ implique $K_M(A \upharpoonright_n) \leq l+1$. En effet, à chaque fois qu'une description de taille l est découverte pour n à l'étape t , on s'assure que l'on aura une description de taille $l+1$ pour $A_t \upharpoonright_n$. Par ailleurs, à chaque fois que $A_{t+1} \neq A_t$ à cause de l'énumération d'un élément n , on met à jour l'ensemble L en s'assurant que chaque chaîne $A_t \upharpoonright_m$ pour $m \geq n$ aura des descriptions de même taille que toutes les descriptions de m dans la version courante de U .

Il reste à montrer que A est simple. Par l'argument usuel (voir la proposition 4.4), A est co-infini, car on énumère au plus un élément dans A , pour chaque W_e , et cet élément est supérieur à $2e$. Soit à présent e tel que W_e est infini. On a $\sum_{U(\tau) \downarrow} 2^{-|\tau|} < 1$, ce qui implique qu'il existe n suffisamment grand tel que $\sum_{U(\tau) \downarrow \geq n} 2^{-|\tau|} < 2^{-e-2}$. Donc, si W_e est infini, on finira par trouver un tel entier n et par l'énumérer dans A . ■

Il est clair que si l'ensemble A est K-trivial, $\bar{A}$ l'est également. Il existe donc des K-triviaux Δ_2^0 non c.e.

Exercice 4.7. (★) Montrer que la classe des ensembles K-triviaux est close par jointure Turing : si les ensembles A_0 et A_1 sont K-triviaux, alors $A_0 \oplus A_1$ est également K-trivial. ◇

4.3. Le théorème de Chaitin

On peut être tenté au premier abord de voir le résultat de Solovay comme un échec de la complexité préfixe. La complexité pleine permet d'échafauder une caractérisation propre et soignée des ensembles C -triviaux, mais qui semble perdre de son élégance avec K . Cette nouvelle classe d'ensembles mérite-t-elle que l'on s'y intéresse ? Nous allons voir que oui.

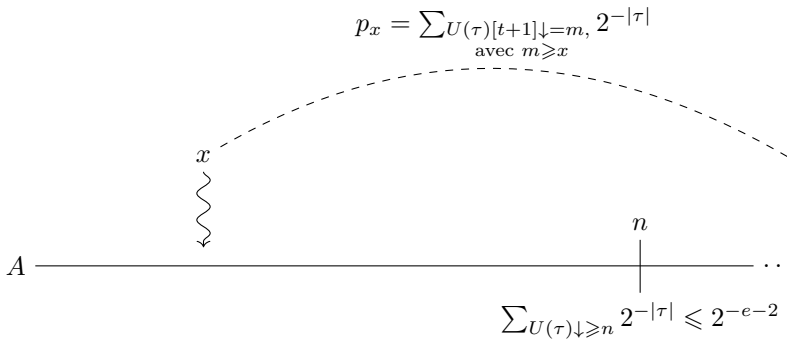


FIGURE 4.6 – Illustration de la preuve du théorème 4.5. L’ajout de x dans A à l’étape $t + 1$ provoque une perte potentielle de p_x : l’alignement sur $K(m)$ qui a été fait pour tous les préfixes de A de taille m plus grande que x . Il existe nécessairement un entier n suffisamment grand tel que cette perte sera bornée par 2^{-e-2} à n’importe quel temps de calcul t .

La classe des K -triviaux est sans aucun doute la plus étudiée et la plus riche de l’aléatoire algorithmique. Les différents travaux de nombreux auteurs ont conduit à en donner pas moins d’une dizaine de caractérisations différentes. Nous y consacrerons donc le chapitre 20 entièrement.

Pour le moment, nous montrons simplement le résultat de Chaitin, central pour le chapitre 20 à venir : les K -triviaux sont tous Δ_2^0 . L’idée est la même que pour montrer que les ensembles C -triviaux sont calculables.

Théorème 4.8 (Chaitin [29])

Il existe une constante c telle que, pour tout $\sigma \in 2^{<\mathbb{N}}$ et $d \in \mathbb{N}$, on a

$$|\{\sigma : |\sigma| = n \wedge K(\sigma) \leq K(n) + d\}| < 2^c \times 2^d.$$

PREUVE. Soit U notre machine universelle sans préfixe, et soit M la machine sans préfixe qui sur τ renvoie $|U(\tau)|$. D’après le théorème 3.5, soit c une constante telle que $P_M(\sigma) < 2^{-K(\sigma)} \times 2^c$ pour toute chaîne σ .

Supposons à présent que, pour n fixé, on ait plus de $2^c \times 2^d$ chaînes σ de taille n telles que $K(\sigma) \leq K(n) + d$. Alors, via M , on a également plus de $2^c \times 2^d$ descriptions distinctes de n qui sont toutes de taille inférieure ou égale à $K(n) + d$. Cela nous donne l’inégalité

$$P_M(n) \geq 2^c \times 2^d \times 2^{-K(n)-d} = 2^c \times 2^{-K(n)},$$

ce qui contredit le choix de c . ■

Corollaire 4.9 (Chaitin [29])

Les ensembles K-triviaux sont tous Δ_2^0 .

PREUVE. Étant donné que $K \leq_T \emptyset'$, pour tout d , la classe

$$\{X : \forall n \ K(X \upharpoonright_n) \leq K(n) + d\}$$

est une classe $\Pi_1^0(\emptyset')$. Par le théorème 4.8, cette classe contient un nombre fini d'éléments. Ils sont donc tous $\emptyset'$ -calculables, par le corollaire 8-3.8 relativisé à $\emptyset'$. ■

Chapitre 17

Boréliens, mesure et calculabilité

Le logicien, philosophe et mathématicien suédois Per Martin-Löf propose une notion d'aléatoire suivant un paradigme bien différent de celui de Chaitin/Levin : un ensemble aléatoire est un ensemble qui n'a aucune propriété « atypique », c'est-à-dire aucune propriété qui est vraie avec probabilité 0, étant donné des tirages indépendants d'une suite infinie de bits, via un processus aléatoire qui produit à chaque coup 0 ou 1 avec probabilité $1/2$.

La formalisation mathématique de la théorie des probabilités a vu le jour au XX^e siècle, avec les travaux de Kolmogorov, qui publia en 1933 son manuel *Fondements de la théorie des probabilités*. Kolmogorov s'appuie pour cela sur les travaux du début du XX^e siècle de Borel et Lebesgue sur la théorie de la mesure et de l'intégration. La théorie de la mesure appliquée à ce que l'on appelle « les boréliens effectifs » est le concept central que nous utiliserons dans notre étude de l'aléatoire algorithmique.

1. Un peu d'histoire

Nous sommes au mois d'août de l'année 1900. Des mathématiciens du monde entier convergent vers la Sorbonne, où se tient leur deuxième congrès international. La belle époque bat son plein au cœur d'un Paris paisible. La tour Eiffel y fut érigée un an plus tôt, Sarah Bernhardt au sommet de sa gloire se produit au théâtre de la Renaissance, la première ligne de métro est en service, les frères Lumière projettent leurs premiers films sur le champ de Mars, et la grande roue de Paris vient d'y être construite, à l'occasion de

la cinquième exposition universelle, qui a accueilli plus de cinquante millions de visiteurs. Les bouleversements culturels et techniques qui marquent ce début de XX^e siècle font écho aux bouleversements mathématiques du moment.

1.1. Le congrès international de 1900

Émile Borel, alors jeune maître de conférence à l'ENS assiste au congrès international. Il y écoute Henri Poincaré, qui disserte avec talent de cette discipline émergente qu'est la logique mathématique et des questions philosophiques qu'elle pose. Contrairement à certains de ses contemporains, Poincaré n'est pas foncièrement opposé à la logique, mais il défend face à elle une attitude prudente et au fond empreinte d'une certaine lucidité [177] : « *En devenant rigoureuse, la science mathématique prend un caractère artificiel qui frappera tout le monde ; elle oublie ses origines historiques ; on voit comment les questions peuvent se résoudre, on ne voit plus comment et pourquoi elles se posent.* » Poincaré, également physicien, considère que les mathématiques ne doivent pas s'éloigner de la physique, par laquelle elles gardent une connexion nécessaire avec le réel. Ce point de vue apparaît plus clairement quelques années plus tard dans son ouvrage « Science et Méthode », en 1908 [178] : « *La logique parfois engendre des monstres. On vit surgir toute une foule de fonctions bizarres qui semblaient s'efforcer de ressembler aussi peu que possible aux honnêtes fonctions qui servent à quelque chose. Plus de continuité, ou bien de la continuité, mais pas de dérivées [...]* Autrefois, quand on inventait une fonction nouvelle, c'était en vue de quelque but pratique ; aujourd'hui, on les invente tout exprès pour mettre en défaut les raisonnements de nos pères, et l'on n'en tirera jamais que cela. »

Le point de vue de Poincaré est opposé à celui de l'autre géant mathématique de l'époque : le mathématicien David Hilbert, pour qui les mathématiques se suffisent très bien à elles-mêmes en tant qu'ensemble de concepts abstraits. Et si ce deuxième congrès international de mathématiques est resté dans l'Histoire, c'est sans aucun doute dû à la conférence qu'y donne Hilbert, durant laquelle il énonce une liste de vingt-trois problèmes mathématiques non résolus, et qui marqueront la recherche du siècle à venir. Hilbert connaît bien les travaux de Cantor sur l'infini, et est sans doute un des plus fervents partisans de cette nouvelle mathématique, à laquelle la logique seule et non le réel donne sa réalité. Aussi est-ce tout naturellement qu'Hilbert place comme problème numéro 1 celui de l'hypothèse du continu : « *Existe-t-il un infini strictement compris entre celui des nombres entiers et celui des nombres réels ?* » C'est dans ce contexte que Borel commence sa carrière.

1.2. Le trio français

Fils d'un pasteur protestant, Émile Borel est né à Saint-Affrique en 1871. Élève brillant et polyvalent, il est reçu à dix-huit ans premier au concours de l'École polytechnique ainsi qu'à celui de l'École normale supérieure, dans laquelle il rentre en 1889. Il est ensuite reçu premier à l'agrégation de mathématiques en 1893, puis devient maître de conférence à Lille, puis à Paris quelques années plus tard. Il s'intéresse très vite aux travaux de Cantor, et montre rapidement son talent et sa créativité, en l'utilisant avec succès pour l'étude des fonctions. Il y initiera également ses élèves, parmi lesquels figurent deux étudiants exceptionnels : Henri Lebesgue et René Baire.



Émile Borel, 1871–1956

Tous deux issus de familles très pauvres, ils sortiront de leur condition sociale par leur travail et leur talent mathématique. Borel et ses deux élèves, le trio français, marqueront tous les trois une avancée majeure dans les mathématiques, à travers une utilisation ingénieuse des notions de « dénombrable » et de « nombres transfinis » développées par Cantor, et qui déboucheront sur la découverte des catégories de Baire, dont nous avons déjà parlé dans la section 10-2.2, sur celle des ensembles dit boréliens, et sur la théorie de la mesure et de l'intégration. C'est de ces deux dernières découvertes dont il est question dans ce chapitre.

1.3. Le point de vue de Borel

Les recherches remarquables de notre trio français connaîtront leurs limites. Borel tout d'abord au fil de sa carrière fait de moins en moins confiance à la théorie des ensembles, encore mal formalisée, et sujette à de nombreux paradoxes. Borel écrira vers la fin de sa vie [107, p.59] « *J'ai été extrêmement séduit dès l'âge de 20 ans, par la lecture des travaux de Cantor [...] Georg Cantor a apporté, dans l'étude des mathématiques, cet esprit romantique qui est l'un des côtés les plus séduisants de l'âme allemande* ». Cette attitude déclinera avec le temps, et Borel adoptera petit à petit un point de vue proche de celui de Poincaré vis-à-vis de la théorie des ensembles, tout en lui ajoutant une touche constructiviste, une idée naturelle chez lui depuis ses

premiers travaux mathématiques, en substance : les objets que l'on manipule doivent être descriptibles explicitement. De ces objets-là seulement on peut garantir une certaine forme d'existence, le reste n'étant qu'abstraction inaccessible. Borel utilise de fait, dès 1898 dans sa « Leçon sur la théorie des fonctions » le mot *effectif* dans un sens proche de ce que l'on appelle aujourd'hui « calculable », concept encore non formalisé à l'époque, mais dans lequel Borel — précurseur — décelait déjà une importance fondamentale. Il fut ainsi le premier à définir les ensembles calculatoirement énumérables, afin de résoudre le paradoxe de Berry (que nous avons déjà présenté dans la section 16-1.3) : « Soit n le plus petit nombre non définissable en moins de cinquante mots. » Borel propose une nouvelle approche pour la résolution de ce paradoxe [22] : si l'on considère la liste des phrases écrites en français qui définissent un nombre de manière unique, il n'existe pour autant aucun algorithme permettant de trouver *effectivement* quel nombre est défini par une phrase. Borel fait alors la première distinction entre ensembles calculatoirement énumérables, qu'un algorithme peut mettre en bijection avec $\mathbb{N}$, et ensemble simplement dénombrable, pour lesquels on ne peut pas nécessairement calculer une bijection avec $\mathbb{N}$. Cet exemple montre le goût prononcé de Borel pour le constructivisme mathématique. Arnaud Denjoy raconte encore à propos de Borel [107, p.56] : « *Le nombre avait pour lui une réalité presque charnelle. Il déniait la qualité d'existence à la plupart des nombres incommensurables, dont l'intervention est pourtant indispensable pour fonder nos théories générales. Il exigeait des mathématiques une clarté, une évidence cartésienne, c'est-à-dire qu'elles soient aussi proches du sensuel que du rationnel* ».

1.4. La fin du trio

Borel perdra petit à petit son intérêt pour la théorie des ensembles, qu'il juge sans doute trop abstraite. Il faut dire que les mathématiques sont loin d'être son seul domaine d'activité. Il est curieux de tout, et n'a aucun mal à remplacer sa passion de jeunesse. Il crée en 1906 avec sa femme la « Revue du Mois », journal scientifique et littéraire, il devient directeur en 1910 de l'École normale supérieure, s'engage volontairement en 1914 durant la Première Guerre mondiale et prend rapidement de nombreuses responsabilités politiques, d'abord député, puis chef de cabinet du ministère de la guerre et responsable des services techniques de l'armement. Il sera enfin un court moment ministre de la Marine après la guerre. Il participe en 1936 à la création de l'organisation d'État de la Recherche, qui deviendra plus tard le CNRS. Il sera également un membre actif de la résistance durant la Seconde Guerre mondiale, et publiera toute sa vie de nombreux ouvrages de mathématiques et de vulgarisation. Borel ne s'est donc pas ennuyé, loin de là, et n'a sans doute pas seulement arrêté son activité de

recherche mathématique par manque de temps, mais aussi car il voulait prendre ses distances avec ce qu'il appelait « les hautes mathématiques ». Il déclare ainsi à son ami Paul Valéry en 1914 [107, p.83] qu'il était effrayé des conséquences sur son esprit de la recherche en théorie des ensembles « à cause de la fatigue qu'elle impose et qui fait craindre des troubles sérieux à ceux qui s'imposent ce travail. » Les deux autres membres de notre trio eurent leurs propres problèmes, et stoppèrent eux aussi leurs travaux.

Baire n'a pas une très bonne santé et souffre de jalousie envers son collègue Henri Lebesgue, dont les travaux — contrairement aux siens — eurent presque instantanément un retentissement international. Baire s'estime lésé par rapport à Lebesgue ; qui occupe rapidement des postes prestigieux à la Sorbonne et au Collège de France, alors que lui doit se contenter de la faculté des sciences de Dijon¹. Les problèmes de santé physique et psychologique de Baire l'empêchent de travailler durant de longues périodes, conduisant de surcroît à des difficultés financières. À cela s'ajoute la frustration de ce qu'il considère comme une non-reconnaissance de ses mérites scientifiques, et qui le plonge dans une grave dépression. Il obtient en 1914 un congé maladie, qui se prolongera jusqu'en 1932, date à laquelle il se suicide dans un hôtel genevois près du lac Léman.

Lebesgue n'a pas les problèmes physiques et psychologiques de Baire, mais lui aussi souffre de jalousie : envers son mentor Émile Borel ! Lebesgue conteste à Borel la paternité de certaines idées sur la théorie de la mesure et la relation entre les deux hommes se dégrade, comme en témoigne cette lettre de 1917 adressée à Borel : « *Je vous l'ai dit... je n'ai plus en vous la même confiance qu'autrefois... Pour le moment toute relation qui sortirait de la banalité de la camaraderie serait de ma part, une hypocrisie. Ce ne serait pas avec vous que je déjeunerais, ce serait avec de vieux souvenirs... Le genre de relations que nous avons eues depuis un an vous a assez fait sentir mon état d'esprit pour que cette lettre ne vous surprenne pas trop. Je crois cependant qu'elle vous fera quelques peines et j'ai conservé trop d'amitié cachée pour vous pour ne pas en être peiné moi-même* ». Ainsi s'éteint petit à petit l'élan créateur de l'école française de la théorie des ensembles. C'est à quelques milliers de kilomètres de Paris, que sera repris le flambeau, avec la naissance de la très fameuse école de Moscou. Il s'agit là d'une autre histoire, que nous aborderons dans la section 29-1.

2. Premières intuitions sur la mesure

Essayons d'expliquer un peu ce que l'on entend par *mesure* en mathématiques. Un des buts de la théorie de la mesure est de formaliser la notion de « taille » des parties d'un ensemble.

1. Nous retranscrivons ici l'état d'esprit de Baire, sans préjuger aucunement des mérites — dont nous ne doutons pas — de l'université de Bourgogne.

Considérons par exemple la ligne des réels. Nous avons une bonne intuition de ce qu'est la mesure d'un intervalle $[a, b]$, car celle-ci correspond à ce que l'on mesurerait effectivement en pratique avec une règle graduée : il s'agit du réel $b - a$. Nous écrirons alors $\lambda([a, b]) = b - a$, la fonction λ désignant la mesure dite « standard », que l'on appelle généralement *mesure de Lebesgue*. Étant donné une réunion $[a, b] \cup [c, d]$ de deux intervalles disjoints, il paraît tout aussi naturel de définir

$$\lambda([a, b] \cup [c, d]) = \lambda([a, b]) + \lambda([c, d]) = (b - a) + (d - c)$$

Ce que l'on entend par mesure d'un ensemble arbitraire de réels $I \subseteq \mathbb{R}$ est en revanche beaucoup moins clair. Nous verrons que quelle que soit la notion de mesure que l'on prend, dès lors qu'elle satisfait quelques propriétés raisonnables, par exemple la mesure d'une réunion disjointe est la somme des mesures, il existe nécessairement des ensembles non mesurables.

Dans le cas qui nous occupe, nous allons nous employer à définir une mesure sur des sous-parties de l'espace de Cantor. Comme nous l'avons vu dans la section 2-6, l'espace de Cantor peut être vu comme l'intervalle $[0, 1]$ en voyant toute suite binaire infinie $X \in 2^{\mathbb{N}}$ comme le réel $0.X$ en développement binaire. En particulier, nous définirons $\lambda(2^{\mathbb{N}}) = \lambda([0, 1]) = 1$. La règle qui veut que la mesure des réunions disjointes est égale à la somme de la mesure des réunions fonctionne aussi pour les réunions dénombrables : par exemple,

$$\begin{aligned} \lambda\left(\bigcup_{n \in \mathbb{N}} [2^{-2n-1}, 2^{-2n}]\right) &= \sum_{n \in \mathbb{N}} (2^{-2n} - 2^{-2n-1}) = \sum_{n \in \mathbb{N}} 2^{-2n} (1 - 2^{-1}) \\ &= 2^{-1} \sum_{n \in \mathbb{N}} 2^{-2n} = \frac{1}{2} \times \frac{1}{1 - \frac{1}{4}} = \frac{2}{3}. \end{aligned}$$

Chaque ouvert de l'espace de Cantor possède donc une mesure : un ouvert $\mathcal{U} \subseteq 2^{\omega}$ est de la forme $\mathcal{U} = \bigcup_{\sigma \in W} [\sigma]$. Notons que l'on peut toujours choisir W de telle manière à ce que celui-ci soit sans préfixe : si $\sigma \preceq \tau$ et $\sigma, \tau \in W$, alors on peut toujours enlever τ de W , car $[\tau] \subseteq [\sigma]$. Pour un tel ensemble W sans préfixe, on a donc $\lambda(\mathcal{U}) = \sum_{\sigma \in W} 2^{-|\sigma|}$.

La règle de la mesure des réunions disjointes nous donne dès lors une mesure des ensembles fermés : si $\mathcal{C} \subseteq 2^{\mathbb{N}}$ est fermé, alors $\lambda(\mathcal{C}) + \lambda(2^{\mathbb{N}} \setminus \mathcal{C}) = \lambda(2^{\mathbb{N}})$. Comme $\lambda(2^{\mathbb{N}}) = 1$, nous avons $\lambda(\mathcal{C}) = 1 - \lambda(2^{\mathbb{N}} \setminus \mathcal{C})$, où $2^{\mathbb{N}} \setminus \mathcal{C}$ est un ouvert.

2.1. Classes Π_2^0

Pour pouvoir définir l'aléatoire de Martin-Löf, il est nécessaire d'étendre la mesure à des classes plus complexes encore que les fermés et les ouverts. À titre d'exemple, considérons la loi des grands nombres : une suite aléatoire

devrait avoir comme propriété que la limite de ses fréquences de 0 et de 1 parmi ses préfixes, existe et tende vers $1/2$ quand la taille des préfixes tend vers $+\infty$. Essayons de décrire la classe $\mathcal{A}$ des suites qui ne satisfont pas cette propriété. Il s'agit de la classe des suites X telles que pour un $\varepsilon > 0$ donné, pour tout $n \in \mathbb{N}$ il existe $m \geq n$ tel que la fréquence de 0 dans le préfixe de X de taille m dépasse $1/2 + \varepsilon$ ou est en dessous de $1/2 - \varepsilon$. Pour ε fixé, on peut décrire ces classes de la manière suivante : $\mathcal{A}_\varepsilon = \bigcap_n \mathcal{U}_{\varepsilon,n}$, où

$$\mathcal{U}_{\varepsilon,n} = \bigcup_{m \geq n} [C_{\varepsilon,m}] \quad \text{et} \quad C_{\varepsilon,m} = \left\{ \sigma \in 2^m : \left| \frac{\#\{i \leq m : \sigma(i) = 0\}}{m} - \frac{1}{2} \right| > \varepsilon \right\},$$

où $[C_{\varepsilon,m}]$ dénote l'ouvert décrit par l'ensemble $C_{\varepsilon,m} \subseteq 2^{<\mathbb{N}}$, où $||$ dénote la valeur absolue et $\#$ la taille d'un ensemble. Il est clair que chaque ensemble $\mathcal{U}_{\varepsilon,n}$ est une classe Σ_1^0 : un ouvert effectif. Chaque $\mathcal{A}_\varepsilon$ est donc une intersection dénombrable d'ouverts effectifs.

Nous appellerons *classes* Π_2^0 les intersections effectives d'ouverts effectifs. Quelle est la mesure de la classe Π_2^0 $\mathcal{A}_\varepsilon = \bigcap_n \mathcal{U}_{\varepsilon,n}$? Il est possible de montrer que la classe $\mathcal{A}_\varepsilon$ n'est ni ouverte ni fermée : elle est d'une complexité supérieure. En revanche, d'après le fait 10-2.2 chaque étape de l'intersection dénombrable est un ouvert, c'est-à-dire que, pour tout n , la classe $\mathcal{A}_{\varepsilon,n} = \bigcap_{m \leq n} \mathcal{U}_{\varepsilon,m}$ reste une classe ouverte et même une classe Σ_1^0 (voir le lemme 3.5). Il s'ensuit que chaque classe $\mathcal{A}_{\varepsilon,n}$ est mesurable : en tant qu'ouvert, la mesure $\lambda(\mathcal{A}_{\varepsilon,n})$ est bien définie. Par ailleurs, $\mathcal{A}_{\varepsilon,n+1} \subseteq \mathcal{A}_{\varepsilon,n}$. À chaque étape, la mesure ne peut donc que diminuer et comme elle ne peut descendre en dessous de 0, elle a nécessairement une limite. On peut donc définir en toute logique

$$\lambda\left(\bigcap_n \mathcal{U}_{\varepsilon,n}\right) = \lim_{n \rightarrow +\infty} \lambda(\mathcal{A}_{\varepsilon,n}) = \lim_{n \rightarrow +\infty} \lambda\left(\bigcap_{m \leq n} \mathcal{U}_{\varepsilon,m}\right).$$

2.2. Lien avec les probabilités

Les notions de mesure, de classe Π_2^0 et d'autres classes de complexité supérieure, ont été formellement définies et étudiées par Borel et Lebesgue au début du XX^e siècle. Si Lebesgue utilise surtout la mesure pour sa théorie de l'intégration, Borel a compris très tôt les liens entre mesure et probabilité, qu'il utilise par exemple en 1909 pour montrer que la loi des grands nombres est satisfaite avec probabilité 1 par les ensembles d'entiers [23]. Il faudra toutefois attendre les travaux de Kolmogorov en 1933 pour que la théorie de la mesure soit utilisée en toute généralité pour une axiomatique formelle de la théorie des probabilités [10].

L'analogie entre mesure et probabilité est simple : supposons que l'on tire des bits aléatoirement à pile ou face, en utilisant un processus qui donne 0

avec probabilité $1/2$ et 1 avec probabilité $1/2$. Ce processus génère petit à petit une suite $Z \in 2^{\mathbb{N}}$. La mesure d'une classe $\mathcal{B} \subseteq 2^{\mathbb{N}}$ correspond simplement à la probabilité que Z appartienne à $\mathcal{B}$. L'idée de Martin-Löf sera donc de dire qu'une suite « aléatoire » ne devrait appartenir à aucune classe de mesure 0 , c'est-à-dire à aucun ensemble « atypique ». Évidemment, chaque ensemble Z appartient à la classe $\{Z\}$ qui est de mesure 0 — on vérifie en effet facilement que le complémentaire de $\{Z\}$ est un ouvert de mesure 1 —, ce qui est problématique pour cette intuition : aucun ensemble n'est alors aléatoire. Martin-Löf propose alors de ne sélectionner qu'une partie de ces classes : les Π_2^0 dont la mesure converge vers 0 de manière effective.

3. Classes boréliennes

Nous commençons par présenter les classes sur lesquelles la mesure se définit naturellement, à la manière dont nous l'avons décrite dans la section précédente. Il s'agit de la hiérarchie borélienne, nommée d'après Émile Borel. Cette hiérarchie est écrite en gras (boldface) pour la distinguer de sa version effective (lightface) que nous définirons plus tard. Un « $\sim$ » est parfois ajouté sous la notation de la hiérarchie borélienne pour éviter toute confusion — c'est ce qui est fait dans ce texte.

Définition 3.1 (Hiérarchie borélienne). Une classe $\mathcal{B} \subseteq 2^{\mathbb{N}}$ est Σ_1^0 si elle est ouverte et Π_1^0 si elle est fermée. On définit ensuite inductivement sur $n > 1$:

1. une classe $\mathcal{B} \subseteq 2^{\mathbb{N}}$ est Σ_{n+1}^0 si elle est une réunion dénombrable de classes Π_n^0 ;
2. une classe $\mathcal{B} \subseteq 2^{\mathbb{N}}$ est Π_{n+1}^0 si elle est une intersection dénombrable de classes Σ_n^0 .

Une classe est Δ_n^0 si elle est à la fois Σ_n^0 et Π_n^0 .



Notons que les classes Σ_n^0 sont les complémentaires dans $2^{\mathbb{N}}$ des classes Π_n^0 (cela vient de l'égalité $2^{\mathbb{N}} \setminus \bigcup_n \mathcal{A}_n = \bigcap_n (2^{\mathbb{N}} \setminus \mathcal{A}_n)$). La hiérarchie borélienne est en quelque sorte une hiérarchie de complexité sur « la forme » des ensembles. Il est clair que les classes Σ_n^0 (respectivement Π_n^0) sont aussi Π_{n+1}^0 (respectivement Σ_{n+1}^0), car on peut toujours écrire $\mathcal{A} = \mathcal{A} \cup \mathcal{A} \cup \mathcal{A} \dots$ ou $\mathcal{A} = \mathcal{A} \cap \mathcal{A} \cap \mathcal{A} \dots$. Nous allons voir avec le lemme 3.12 que les classes Σ_n^0 sont aussi Σ_{n+1}^0 , et que les classes Π_n^0 sont aussi Π_{n+1}^0 . Nous verrons également que la hiérarchie borélienne est stricte : de la même manière qu'un ouvert n'est pas forcément un fermé, une intersection dénombrable

d'ouverts ne pourra pas forcément être décrite comme une réunion dénombrable de fermés, et ainsi de suite.

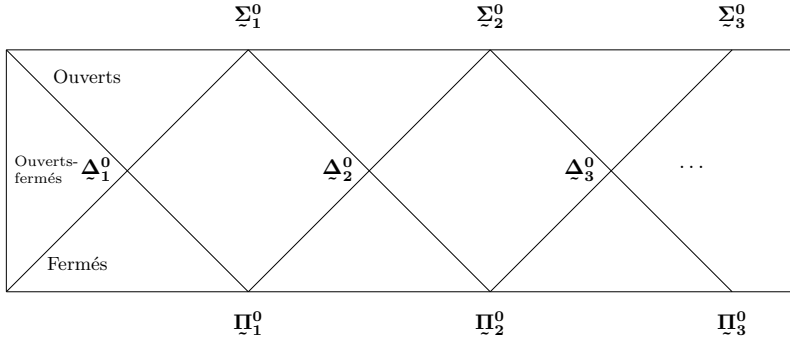


FIGURE 3.2 – *Hiérarchie borélienne*

Souvenons-nous que les ouverts et fermés possèdent un analogue effectif, que l'on note respectivement Σ_1^0 et Π_1^0 . Nous allons maintenant voir comment étendre cette correspondance en définissant les Σ_n^0 et Π_n^0 , correspondant aux Σ_n^0 et Π_n^0 effectifs : on demande simplement à ce que les réunions/intersections soient uniformes.

Définition 3.3 (Hiérarchie borélienne effective). Un *code* Σ_1^0 pour une classe $\Sigma_1^0 \mathcal{U} \subseteq 2^{\mathbb{N}}$ est un entier de la forme $\langle 1, 0, e \rangle$, où $W_e \subseteq 2^{<\mathbb{N}}$ décrit $\mathcal{U}$, c'est-à-dire $\mathcal{U} = \bigcup_{\sigma \in W_e} [\sigma]$. Un *code* Π_1^0 pour son complémentaire est donné par $\langle 1, 1, e \rangle$. On définit par induction pour $n > 0$, comme suit.

- (1) Une classe $\mathcal{B}$ est Σ_{n+1}^0 s'il existe un ensemble c. e. $W_e \subseteq \mathbb{N}$ énumérant des codes Π_n^0 de classes telles que $\mathcal{B} = \bigcup_n \mathcal{B}_n$, où $\mathcal{B}_n$ est la classe correspondant au n -ième code énuméré dans W_e . Un *code* Σ_{n+1}^0 pour $\mathcal{B}$ est donné par $\langle n+1, 0, e \rangle$.
- (2) Une classe $\mathcal{B}$ est Π_{n+1}^0 s'il existe un ensemble c. e. $W_e \subseteq \mathbb{N}$ énumérant des codes Σ_n^0 de classes telles que $\mathcal{B} = \bigcap_n \mathcal{B}_n$, où $\mathcal{B}_n$ est la classe correspondant au n -ième code énuméré dans W_e . Un *code* Π_{n+1}^0 pour $\mathcal{B}$ est donné par $\langle n+1, 1, e \rangle$.

Une classe $\mathcal{B}$ est Δ_n^0 si elle est à la fois Σ_n^0 et Π_n^0 . ◇

Remarquons la similarité des classes Σ_{n+1}^0 de l'espace de Cantor avec les ensembles d'entiers Σ_{n+1}^0 . Ces derniers peuvent également être définis comme des réunions dénombrables de classes Π_n^0 , via la correspondance usuelle réunion/quantification existentielle et intersection/quantification universelle.

Exercice 3.4. Montrer qu'à partir du code Σ_n^0 (resp. Π_n^0) d'une classe $\mathcal{A}$, on peut obtenir de manière calculable le code Π_n^0 (resp. Σ_n^0) de $2^{\mathbb{N}} \setminus \mathcal{A}$. $\diamond$

Voyons un petit lemme similaire aux propositions 5-1.7 et 5-1.6 de la hiérarchie arithmétique pour les entiers, que nous utiliserons implicitement dans la suite.

Lemme 3.5. Les classes Σ_n^0 sont closes par intersections finies et réunions dénombrables effectives. Les classes Π_n^0 sont closes par réunions finies et intersections dénombrables effectives. $\star$

PREUVE. Si une réunion dénombrable de classes Σ_n^0 est effective, on peut aisément calculer un code Σ_n^0 pour cette réunion. Par passage au complémentaire (exercice 3.4), les intersections dénombrables de classes Π_n^0 sont des classes Π_n^0 .

Montrons que les classes Σ_1^0 sont closes par intersections finies. Soient donc $\mathcal{U}_0 = \bigcup_{\sigma \in W_0} [\sigma]$ et $\mathcal{U}_1 = \bigcup_{\sigma \in W_1} [\sigma]$ deux classes Σ_1^0 . Alors,

$$\mathcal{U}_0 \cap \mathcal{U}_1 = \bigcup_{\sigma_0 \in W_0, \sigma_1 \in W_1} [\sigma_0] \cap [\sigma_1].$$

On peut également décrire l'ensemble c.e. des chaînes constituant l'ouvert $\mathcal{U}_0 \cap \mathcal{U}_1$ de la manière suivante : on énumère une chaîne σ quand on la trouve dans W_i pour $i < 2$ et quand un préfixe de σ est énuméré dans W_{i-1} . Par passage au complémentaire, les classes Π_1^0 sont closes par réunions finies. Notons que l'on peut uniformément produire un code de $\mathcal{U}_0 \cap \mathcal{U}_1$ à partir de codes de $\mathcal{U}_0$ et de $\mathcal{U}_1$, et que l'on peut faire de même pour les réunions de deux fermés.

Supposons à présent que les classes Σ_n^0 sont closes par intersections finies et que les classes Π_n^0 sont closes par réunions finies. Soient $\mathcal{B}_0 = \bigcup_{n \in W_0} \mathcal{A}_{0,n}$ et $\mathcal{B}_1 = \bigcup_{n \in W_1} \mathcal{A}_{1,n}$ des classes Σ_{n+1}^0 , avec chaque $\mathcal{A}_{i,n}$ une classe Π_n^0 pour $i < 2$. Alors, $\mathcal{B}_0 \cap \mathcal{B}_1 = \bigcup_{n_0 \in W_0, n_1 \in W_1} \mathcal{A}_{0,n_0} \cap \mathcal{A}_{1,n_1}$. Par hypothèse d'induction, chaque classe $\mathcal{A}_{0,n_0} \cap \mathcal{A}_{1,n_1}$ est Π_n^0 , et l'on peut uniformément en obtenir un code Π_n^0 à partir d'un code Π_n^0 de $\mathcal{A}_{0,n_0}$ et de $\mathcal{A}_{1,n_1}$. On peut donc facilement trouver un code pour la classe $\mathcal{B}_0 \cap \mathcal{B}_1$. Par passage au complémentaire (exercice 3.4), les classes Π_{n+1}^0 sont closes par réunions finies. $\blacksquare$

Exercice 3.6. S'assurer que la preuve du lemme 3.5 fournit aussi l'uniformité : étant donné des codes de $\mathcal{B}_1$ et $\mathcal{B}_2$ deux classes Σ_n^0 (resp. Π_n^0), on peut calculer un code Σ_n^0 (resp. Π_n^0) de la classe $\mathcal{B}_1 \cap \mathcal{B}_2$ (resp. $\mathcal{B}_1 \cup \mathcal{B}_2$). $\diamond$

Le lemme précédent permet de généraliser aux boréliens quelque chose qui a été vu dans la section 8-2.1 pour les ouverts et les fermés : les classes Π_{n+1}^0 peuvent toujours être considérées décroissantes.

Étant donné $\bigcap_n \mathcal{B}_n$, on peut considérer de manière équivalente la classe $\bigcap_n (\bigcup_{m \leq n} \mathcal{B}_m)$. Notons que les classes usuelles — en général définies à l'aide de formules — tendent à être effectives. Voyons quelques exemples.

Exemple 3.7. (1) La classe des ensembles infinis est Π_2^0 :

$$\{X : \forall n \exists m > n \ m \in X\} = \bigcap_n \{X : X(m) = 1 \text{ pour } m > n\}.$$

En utilisant les catégories de Baire, on montre facilement que cette classe n'est pas Σ_2^0 . En effet, un fermé ne contenant que des ensembles infinis est forcément d'intérieur vide; et, dans le complémentaire d'une réunion de fermés d'intérieur vide, on trouve facilement en utilisant le lemme 10-2.14 un ensemble infini.

- (2) La classe des ensembles qui sont des sauts Turing d'autres ensembles est Π_2^0 . On montre facilement (voir l'exercice 4-6.4) l'existence d'une fonctionnelle Φ_e telle que $\Phi_e(X)$ est totale pour tout X et telle que $\Phi_e(X') = X$ pour tout X . Dès lors, on définit l'ensemble des sauts comme :

$$\begin{aligned} & \left\{ X : \forall n \begin{array}{l} (X(n) = 1 \wedge \Phi_n(\Phi_e(X), n) \downarrow) \\ \vee (X(n) = 0 \wedge \Phi_n(\Phi_e(X), n) \uparrow) \end{array} \right\} \\ &= \\ & \bigcap_n \left(\{X : X(n) = 1 \wedge \Phi_n(\Phi_e(X), n) \downarrow\} \cup \{X : X(n) = 0 \wedge \Phi_n(\Phi_e(X), n) \uparrow\} \right), \end{aligned}$$

ce qui est une intersection dénombrable de réunions d'un ouvert et d'un fermé. D'après le lemme 3.12, à venir, une intersection d'un ouvert et d'un fermé peut toujours être présentée comme une classe Π_2^0 . En utilisant le théorème de la base low, on voit facilement que cet ensemble n'est pas Σ_2^0 puisqu'un ensemble low n'est pas le saut d'un autre ensemble. On peut consulter la correction de l'exercice 3.10 pour une preuve que cet ensemble n'est pas Σ_2^0 .

- (3) La classe des ensembles de *densité supérieure positive* — dont le ratio d'éléments parmi leurs préfixes est infiniment souvent plus grand qu'un certain ε — est Σ_3^0 :

$$\left\{ X : \exists \varepsilon > 0 \forall n \exists m > n \frac{\#\{i \leq m : X(i) = 1\}}{m} > \varepsilon \right\}.$$

Là encore, on peut consulter la correction de l'exercice 3.11 pour une preuve que cet ensemble n'est pas Π_3^0 .

- (4) Toute classe dénombrable $\mathcal{B} = \{X_0, X_1, \dots\}$ est Σ_2^0 . En effet, on a l'égalité $\mathcal{B} = \bigcup_n \{X_n\}$, où $\{X_n\}$ est un fermé.

Le lecteur pourra se rendre compte avec les exercices 3.10 et 3.11 de la difficulté, en général, de séparer les complexités boréliennes. Les techniques utilisées sont souvent des utilisations sophistiquées des catégories de Baire, via des forcings spécifiques.

Les exemples précédents illustrent le fait que la complexité borélienne des ensembles n'est pas liée à une puissance de calcul, mais réellement à une différence de *forme* : une intersection effective d'ouverts effectifs peut ne pas être *de la forme* d'une réunion de fermés, indépendamment de la puissance de calcul dont on dispose pour définir cette réunion de fermés. Cela nous conduit aussi à la considération suivante.

— Relativisation à un oracle —

La hiérarchie borélienne effective se relativise à un oracle $X \in 2^{\mathbb{N}}$, en remplaçant c. e. par X -c. e. dans la définition 3.3. Par ailleurs, on montre sans mal qu'un ensemble Σ_n^0 (resp. Π_n^0) est $\Sigma_n^0(X)$ (resp. $\Pi_n^0(X)$) pour un certain oracle X , qui par exemple encodera les différentes réunions/intersections des différents ouverts/fermés constituant le borélien.

Voyons à présent d'autres exemples illustrant la relativisation à un oracle.

Exemple 3.8. (1) La classe des ensembles low est $\Sigma_2^0(\emptyset'')$: chaque ensemble low est l'unique point d'un $\Pi_1^0(\emptyset')$. En revanche, $\emptyset''$ est nécessaire pour que la réunion de tous ces fermés soit uniforme. Le lecteur pourra consulter l'exercice 3.9 pour s'en assurer.

(2) La classe des ensembles high est $\Sigma_4^0(\emptyset'')$. Soit $T_{e,n} \subseteq 2^{<\mathbb{N}}$ l'ensemble $\emptyset''$ -calculable des chaînes σ telles que $\Phi_e(\sigma, n) \downarrow = \emptyset''(n)$. La classe des high peut se décrire comme :

$$\bigcup_e \bigcap_n \bigcup_{\sigma \in T_{e,n}} \mathcal{A}_{e,n,\sigma},$$

où $\mathcal{A}_{e,n,\sigma}$ est l'ensemble

$$\{X : \forall i < |\sigma|, [(\sigma(i) = 1 \wedge \Phi_i(X, i) \downarrow) \vee (\sigma(i) = 0 \wedge \Phi_i(X, i) \uparrow)]\}.$$

Pour voir qu'il s'agit bien d'une description des ensembles high, il faut utiliser le fait que $\sigma \preceq \tau$ implique $\mathcal{A}_{e,n,\tau} \subseteq \mathcal{A}_{e,n,\sigma}$ et σ, τ incomparables impliquent $\mathcal{A}_{e,n,\tau} \cap \mathcal{A}_{e,n,\sigma} = \emptyset$. D'après le lemme 3.12, à venir, chaque classe $\mathcal{A}_{e,n,\sigma}$ est Σ_2^0 (uniformément en e, n et σ) en tant que réunion d'un ouvert et d'un fermé, ce qui rend la classe des ensembles high $\Sigma_4^0(\emptyset'')$.

Exercice 3.9. (★★) Établir que la classe des ensembles low n'est pas $\Sigma_2^0(\emptyset')$. ◇

Exercice 3.10. (★★) Montrer que la classe des ensembles qui sont des sauts Turing n'est pas Σ_2^0 . $\diamond$

Exercice 3.11. (★★) Montrer que la classe des ensembles de *densité supérieure positive* — dont le ratio d'éléments parmi ses préfixes est infiniment souvent plus grand qu'un certain ε — n'est pas Π_3^0 . $\diamond$

Nous terminons par un dernier lemme dont la contrepartie pour la hiérarchie arithmétique est évidente, mais qui requiert un peu de travail dans le cas de la hiérarchie borélienne.

Lemme 3.12. Les classes Σ_n^0 ou Π_n^0 sont aussi Δ_{n+1}^0 . $\star$

PREUVE. Il est clair par définition que les classes Σ_n^0 sont Π_{n+1}^0 et que les classes Π_n^0 sont Σ_{n+1}^0 . Les classes Σ_1^0 sont Σ_2^0 en tant que réunions effectives de cylindres, chaque cylindre étant une classe Π_1^0 dégénérée. Notons qu'à partir du code d'une classe Σ_1^0 , on peut uniformément calculer un code Σ_2^0 de la même classe. Par passage au complémentaire (exercice 3.4), les classes Π_1^0 peuvent être uniformément transformées en classes Π_2^0 . Supposons à présent que les classes Σ_n^0 sont uniformément Σ_{n+1}^0 et que les classes Π_n^0 sont uniformément Π_{n+1}^0 . Soit $\bigcup_n \mathcal{A}_n$ une classe Σ_{n+1}^0 , où chaque $\mathcal{A}_n$ est Π_n^0 . Par hypothèse d'induction, chaque $\mathcal{A}_n$ peut être uniformément transformé en classe Π_{n+1}^0 , et donc $\bigcup_n \mathcal{A}_n$ est Σ_{n+2}^0 . Par passage au complémentaire (exercice 3.4), les classes Π_{n+1}^0 sont Π_{n+2}^0 . ■

4. Mesure de Lebesgue

Nous allons maintenant voir comment étendre la mesure de Lebesgue — jusque-là définie pour les ouverts, les fermés et les classes Π_2^0 — à tous les boréliens de l'espace de Cantor, de manière à ce qu'elle possède des bonnes propriétés d'additivité.

Notation

On dénote par λ la mesure de Lebesgue sur les boréliens de l'espace de Cantor, c'est-à-dire telle que $\lambda([\sigma]) = 2^{-|\sigma|}$ pour tout cylindre σ .

4.1. Définition de la mesure

Le mathématicien Caratheodory a montré que la mesure de Lebesgue, définie simplement sur les cylindres $[\sigma]$ comme étant $2^{-|\sigma|}$, se prolongeait de manière unique sur les boréliens en utilisant l'intuition donnée au début de ce chapitre pour mesurer les Π_2^0 .

Théorème 4.1 (Caratheodory)

La mesure λ telle que $\lambda([\sigma]) = 2^{-|\sigma|}$ pour tout cylindre $[\sigma]$, se prolonge de manière unique sur les boréliens en une fonction qui conserve la propriété d'additivité dénombrable : si $(\mathcal{B}_n)_{n \in \mathbb{N}}$ est une suite de boréliens deux à deux disjoints, alors :

$$\lambda\left(\bigcup_n \mathcal{B}_n\right) = \sum_n \lambda(\mathcal{B}_n).$$

Le prolongement se définit par induction de la manière suivante : pour une suite croissante (resp. décroissante) de boréliens $\mathcal{B}_n \subseteq \mathcal{B}_{n+1}$ (respectivement $\mathcal{B}_{n+1} \subseteq \mathcal{B}_n$), on a $\lambda(\bigcup_n \mathcal{B}_n) = \sup_n \lambda(\mathcal{B}_n)$ (respectivement $\lambda(\bigcap_n \mathcal{B}_n) = \inf_n \lambda(\mathcal{B}_n)$).

Le concept de mesure défini par Lebesgue en 1901 est au départ différent de celui présenté dans le théorème 4.1 [136] :

« Considérons un ensemble de points de $[a, b]$; on peut d'une infinité de manières enfermer ces points dans une infinité dénombrable d'intervalles ; la limite inférieure de la somme des longueurs de ces intervalles est la mesure de l'ensemble. Un ensemble E est dit mesurable si sa mesure augmentée de celle de l'ensemble des points ne faisant pas partie de E donne la mesure de $[a, b]$. Voici deux propriétés de ces ensembles : une infinité d'ensembles mesurables E_i étant donnée, l'ensemble des points qui font partie de l'un au moins d'entre eux est mesurable ; si les E_i n'ont deux à deux aucun point commun, la mesure de l'ensemble obtenu est la somme des mesures des E_i . L'ensemble des points communs à tous les E_i est mesurable. »

Constatons que Lebesgue ne parle pas de classes boréliennes, mais à la place de classes *mesurables* (ou d'ensembles mesurables dans la terminologie ci-dessus) : il prétend que si chaque classe $(E_i)_{i \in \mathbb{N}}$ est mesurable, alors $\bigcup_i E_i$ est mesurable, tout comme $\bigcap_i E_i$. Il donne aussi la propriété d'additivité dénombrable. Mais pour définir la mesure d'une classe $\mathcal{B}$ quelconque, il utilise le fait que la mesure des ouverts — des réunions disjointes d'intervalles — est définie sans ambiguïté pour l'esprit humain, pour définir alors la mesure de n'importe quelle classe $\mathcal{B}$ comme étant

$$\inf_{\mathcal{U} \text{ classe } \Sigma_1^0, \text{ avec } \mathcal{B} \subseteq \mathcal{U}} \lambda(\mathcal{U}).$$

Notons que Lebesgue laisse entendre la possibilité que *toutes les classes ne sont pas forcément mesurables*. Il pose pour la mesurabilité d'une classe la condition que le calcul de sa mesure soit cohérent avec l'intuition que la mesure de $\mathcal{B} \subseteq [a, b]$ plus la mesure de $[a, b] \setminus \mathcal{B}$ doit correspondre à la mesure de $[a, b]$, c'est-à-dire $b - a$.

Caratheodory a formalisé l'intuition de Lebesgue sous le nom de *mesure extérieure*. Il est en effet possible de construire — à l'aide de l'axiome

du choix — des classes *non mesurables* de $[0, 1]$ dans le sens imaginé par Lebesgue, c'est-à-dire telles que la mesure extérieure de l'une quelconque d'entre elles ajoutée à la mesure de son complémentaire dans $[0, 1]$ ne soit pas égale à 1. Nous nous apprêtons à voir toutefois que la définition de Lebesgue rejoint celle que nous avons donnée pour les classes boréliennes, qui sont elles toutes mesurables (voir le théorème 4.4).

4.2. Propriétés de la mesure

Résumons ici les propriétés de la mesure de Lebesgue qui seront utilisées régulièrement tout au long de cette partie.

Propriétés de la mesure

▷ Mesure des classes de bases : $\lambda([\sigma]) = 2^{-|\sigma|}$.

▷ Mesure sur les boréliens (définie par induction) :

$$\lambda\left(\bigcup_n \mathcal{B}_n\right) = \sup_n \lambda\left(\bigcup_{m \leq n} \mathcal{B}_m\right) \quad \text{et} \quad \lambda\left(\bigcap_n \mathcal{B}_n\right) = \inf_n \lambda\left(\bigcap_{m \leq n} \mathcal{B}_m\right).$$

▷ Mesure du complémentaire :

$$\lambda(\mathcal{A} \setminus \mathcal{B}) = \lambda(\mathcal{A}) - \lambda(\mathcal{A} \cap \mathcal{B}).$$

▷ Additivité : pour $(\mathcal{A}_n)_{n \in \mathbb{N}}$ une suite de classes mesurables deux à deux disjointes, on a

$$\lambda\left(\bigcup_n \mathcal{A}_n\right) = \sum_n \lambda(\mathcal{A}_n).$$

▷ Sous-additivité : pour $(\mathcal{A}_n)_{n \in \mathbb{N}}$ une suite de classes mesurables, on a

$$\lambda\left(\bigcup_n \mathcal{A}_n\right) \leq \sum_n \lambda(\mathcal{A}_n).$$

Notons que la sous-additivité se déduit de l'additivité : si $\mathcal{A}_0$ et $\mathcal{A}_1$ ne sont pas disjoints, on a en revanche $\mathcal{A}_0 \cup \mathcal{A}_1 = \mathcal{A}_0 \cup (\mathcal{A}_1 \setminus \mathcal{A}_0)$, avec $\mathcal{A}_0$ disjoint de $\mathcal{A}_1 \setminus \mathcal{A}_0$. Par additivité de la mesure, on a donc

$$\lambda(\mathcal{A}_0 \cup \mathcal{A}_1) = \lambda(\mathcal{A}_0) + \lambda(\mathcal{A}_1 \setminus \mathcal{A}_0) \leq \lambda(\mathcal{A}_0) + \lambda(\mathcal{A}_1).$$

Nous allons à présent montrer que tout borélien $\mathcal{B}$ est mesurable dans le sens donné par Lebesgue : l'infimum de la mesure des ouverts contenant $\mathcal{B}$ correspond à la mesure de $\mathcal{B}$ comme définie inductivement dans le théorème 4.1. Si l'on y ajoute l'infimum de la mesure des ouverts contenant le complémentaire de $\mathcal{B}$, on obtient nécessairement 1. Nous montrerons aussi une version effective de cela, et nous avons besoin pour l'effectivité d'étudier le niveau de complexité calculatoire de la mesure d'un borélien.

4.3. Calcul de la mesure

Nous allons montrer que la mesure d'une classe Σ_n^0 est un réel approchable par la gauche relativement à $\emptyset^{(n-1)}$, et en particulier que la mesure d'une classe Σ_1^0 est un réel approchable par la gauche. Nous avons besoin pour cela d'un lemme sur les classes Σ_1^0 qui sera souvent réutilisé par la suite. Souvenons-nous de la définition 16-2.1 d'ensembles sans préfixe.

Lemme 4.2. Soit $\mathcal{U} = [W]$ une classe Σ_1^0 pour un ensemble c. e. $W \subseteq 2^{<\mathbb{N}}$. On peut calculer uniformément en un code de W le code d'un ensemble c. e. W' sans préfixe tel que $\mathcal{U} = [W']$. ★

PREUVE. À l'étape de calcul t , si une nouvelle chaîne σ entre dans W , et si un préfixe de σ appartient déjà à W' , on n'énumère pas σ dans W' . Sinon, si des extensions $\tau \succ \sigma$ sont déjà dans W' , alors on calcule n , la plus petite taille telle que les extensions de σ dans W' à cette étape soient toutes de taille inférieure à n (ce qui est possible car, à chaque étape, W' ne contient qu'un nombre fini d'éléments). On énumère alors dans W' toutes les extensions de σ de taille n qui n'ont pas de préfixe dans W' .

Il est clair que W' est sans préfixe, et que l'on a $[W] = [W']$. ■

Ensemble sans préfixe minimal

Un ensemble sans préfixe W est *minimal* si pour toute chaîne σ et tout entier n , on a $\sigma\tau \notin W$ pour au moins une chaîne τ de taille n . Notons que le lemme précédent ne nous garantit pas que l'ensemble sans préfixe que l'on énumère est minimal : il se peut que l'énumération sans préfixe contienne par exemple la chaîne $\sigma 0$ et $\sigma 1$, qui pourraient alors être remplacées par la chaîne σ . Tout ouvert $\mathcal{U}$ est généré par un ensemble sans préfixe minimal W , mais il ne sera pas toujours possible d'obtenir W de manière c. e.

Nous pouvons à présent passer à la proposition annoncée.

Proposition 4.3. Soit $n > 0$, et soit $\mathcal{B} \subseteq 2^{\mathbb{N}}$ une classe Σ_n^0 . Le réel $\lambda(\mathcal{B})$ est approchable par la gauche relativement à $\emptyset^{(n-1)}$ et uniformément en un code de $\mathcal{B}$. En particulier, $\lambda(\mathcal{B})$ est $\emptyset^{(n)}$ -calculable uniformément en un code de $\mathcal{B}$. ★

PREUVE. Soit $\mathcal{B} = \bigcup_{\sigma \in W} [\sigma]$ une classe Σ_1^0 avec W un ensemble c. e. Par le lemme 4.2, on peut supposer que W est un ensemble sans préfixe. Alors, $\lambda(\mathcal{B}) = \sum_{\sigma \in W} 2^{-|\sigma|}$. Il est donc clair que $\lambda(\mathcal{B})$ est un réel approchable par la gauche uniformément en un code de $\mathcal{B}$.

Supposons que la mesure de toute classe Σ_n^0 soit $\emptyset^{(n)}$ -calculable uniformément en son code. Soit $\mathcal{B} = \bigcup_m \mathcal{A}_m$ une classe Σ_{n+1}^0 , avec chaque $\mathcal{A}_m$

une classe Π_n^0 . Posons alors $\mathcal{C}_m = \bigcup_{k \leq m} \mathcal{A}_k$. Comme $\mathcal{C}_m \subseteq \mathcal{C}_{m+1}$, il vient $\lambda(\mathcal{C}_m) \leq \lambda(\mathcal{C}_{m+1})$. Aussi, par définition, on a $\lambda(\bigcup_m \mathcal{A}_m) = \sup_m \lambda(\mathcal{C}_m)$.

Par le lemme 3.5 et l'exercice 3.6, chaque classe $\mathcal{C}_m$ est Π_n^0 , et l'on peut lui trouver un code Π_n^0 uniformément en m et en un code de $\mathcal{B}$. Comme la mesure du complémentaire de chaque classe $\mathcal{C}_m$ est $\emptyset^{(n)}$ -calculable, alors la mesure de $\mathcal{C}_m$ l'est également — comme 1 moins la mesure du complémentaire de $\mathcal{C}_m$ dans $2^{\mathbb{N}}$ —, et elle l'est de plus uniformément en un code de $\mathcal{C}_m$. La mesure $\mathcal{B}$ est donc le supremum de la suite $r_1 \leq r_2 \leq r_3 \leq \dots$, où chaque $r_m = \lambda(\mathcal{C}_m)$. Comme chaque r_m est un réel $\emptyset^{(n)}$ -calculable uniformément en m , alors $\lambda(\mathcal{B}) = r = \sup_m r_m$ est un réel approchable par la gauche relativement à $\emptyset^{(n)}$ et uniformément en un code de $\mathcal{B}$. ■

Nous sommes à présent prêts à montrer le théorème voulu, dans sa version effective.

Théorème 4.4

Pour tout $\mathcal{B} \subseteq 2^{\mathbb{N}}$ classe Σ_n^0 et pour tout rationnel $\varepsilon > 0$, il existe :

- (1) $\mathcal{U}$ une classe $\Sigma_1^0(\emptyset^{(n-1)})$ telle que $\mathcal{B} \subseteq \mathcal{U}$ et telle que $\lambda(\mathcal{U} \setminus \mathcal{B}) \leq \varepsilon$;
- (2) $\mathcal{F}$ une classe $\Pi_1^0(\emptyset^{(n-1)})$ telle que $\mathcal{F} \subseteq \mathcal{B}$ et telle que $\lambda(\mathcal{B} \setminus \mathcal{F}) \leq \varepsilon$.

De plus, un code de $\mathcal{U}$ peut être calculé uniformément en ε et en un code de $\mathcal{B}$, et un code de $\mathcal{F}$ peut être calculé à l'aide de $\emptyset^{(n)}$ uniformément en ε et en un code de $\mathcal{B}$.

PREUVE. Soit $\mathcal{B}$ une classe Σ_1^0 . Alors, le point (1) est évident. Pour le point (2), il suffit de considérer les approximations ouvertes/fermées de $\mathcal{U}$ à chaque étape de calcul t , données par $\mathcal{U}[t]$. On peut à l'aide de l'arrêt calculer pour tout ε le plus petit t tel que $\lambda(\mathcal{U} \setminus \mathcal{U}[t]) < \varepsilon$. Supposons le théorème vrai pour des classes Σ_n^0 et montrons qu'il est vrai pour des classes Σ_{n+1}^0 . Soit $\mathcal{B} = \bigcup_m \mathcal{A}_m$ une classe Σ_{n+1}^0 avec chaque $\mathcal{A}_m$ une classe Π_n^0 .

Montrons (1). Le lecteur peut s'aider de la figure 4.5 pour suivre la preuve. Le complémentaire $\overline{\mathcal{A}_m}$ de chaque $\mathcal{A}_m$ est Σ_n^0 . Par hypothèse d'induction, d'après (2) on peut uniformément trouver pour tout m à l'aide de $\emptyset^{(n)}$ une classe $\Pi_1^0(\emptyset^{(n-1)})$ $\mathcal{F}_m \subseteq \overline{\mathcal{A}_m}$ telle que $\lambda(\overline{\mathcal{A}_m} \setminus \mathcal{F}_m) \leq \varepsilon 2^{-m-1}$, pour tout m . Soit $\mathcal{U}_m$ le complémentaire du fermé $\mathcal{F}_m$. En particulier, pour tout m , on a $\lambda(\mathcal{U}_m \setminus \mathcal{A}_m) \leq \varepsilon 2^{-m-1}$. Cela nous donne :

$$\begin{aligned} \lambda(\bigcup_m \mathcal{U}_m \setminus \bigcup_m \mathcal{A}_m) &\leq \lambda(\bigcup_m (\mathcal{U}_m \setminus \mathcal{A}_m)) \\ &\leq \sum_m \lambda(\mathcal{U}_m \setminus \mathcal{A}_m) \\ &\leq \sum_m \varepsilon 2^{-m-1} \\ &\leq \varepsilon. \end{aligned}$$

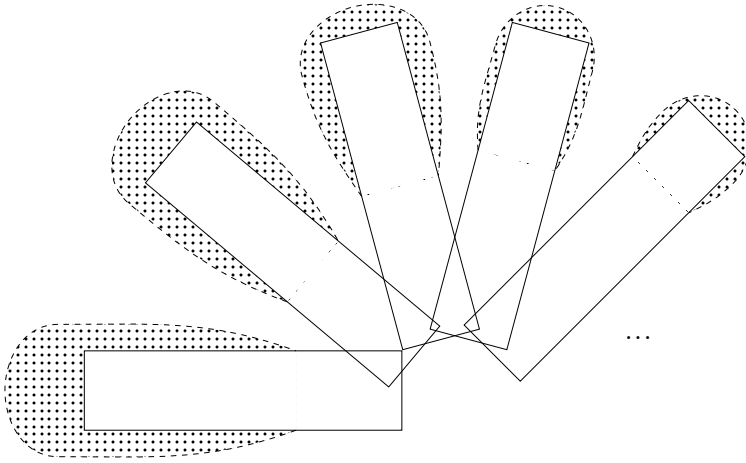


FIGURE 4.5 – *Illustration. Les rectangles sont les composantes Π_n^0 d'une classe Σ_{n+1}^0 croissante. Les contours arrondis en pointillés sont des ouverts contenant chaque composante. La partie grisée est la partie en trop, dont la mesure doit décroître suffisamment rapidement.*

Par ailleurs chaque $\mathcal{U}_m$ est une classe $\Sigma_1^0(\emptyset^{(n-1)})$ dont le code est calculable en $\emptyset^{(n)}$ uniformément. Donc, $\mathcal{U} = \bigcup_m \mathcal{U}_m$ est une classe $\Sigma_1^0(\emptyset^{(n)})$.

Montrons (2). Le lecteur peut s'aider de la figure 4.6 pour suivre la preuve. Par hypothèse d'induction, il existe d'après (1) une classe $\Sigma_1^0(\emptyset^{n-1})$ $\mathcal{U}_m$ telle que $\overline{\mathcal{A}_m} \subseteq \mathcal{U}_m$ et $\lambda(\mathcal{U}_m \setminus \overline{\mathcal{A}_m}) \leq \varepsilon 2^{-m-2}$ pour tout m . Soit $\mathcal{F}_m$ le complémentaire de l'ouvert $\mathcal{U}_m$. On a alors $\lambda(\mathcal{A}_m \setminus \mathcal{F}_m) \leq \varepsilon 2^{-m-2}$. D'après le lemme 3.5, chaque classe $\bigcup_{m \leq k} \mathcal{A}_m$ pour $k \in \mathbb{N}$ est Π_n^0 , et donc chaque classe $\bigcup_m \mathcal{A}_m \setminus \bigcup_{m \leq k} \mathcal{A}_m$ pour $k \in \mathbb{N}$ est Σ_{n+1}^0 . À l'aide de $\emptyset^{(n+1)}$, on cherche en utilisant la proposition 4.3 le plus petit k tel que $\lambda(C) < \varepsilon/2$ pour $C = \bigcup_m \mathcal{A}_m \setminus \bigcup_{m \leq k} \mathcal{A}_m$. On a alors :

$$\begin{aligned}
 \lambda(\bigcup_m \mathcal{A}_m \setminus \bigcup_{m \leq k} \mathcal{F}_m) &\leq \lambda(C) + \lambda(\bigcup_{m \leq k} \mathcal{A}_m \setminus \bigcup_{m \leq k} \mathcal{F}_m) \\
 &\leq \lambda(C) + \lambda(\bigcup_{m \leq k} (\mathcal{A}_m \setminus \mathcal{F}_m)) \\
 &\leq \frac{1}{2} \varepsilon + \frac{1}{2} \sum_{m \leq k} \varepsilon 2^{-m-1} \\
 &\leq \varepsilon.
 \end{aligned}$$

Comme chaque $\mathcal{F}_m$ est une classe $\Pi_1^0(\emptyset^{n-1})$ dont le code peut être trouvé uniformément en m , alors $\mathcal{F} = \bigcup_{m \leq k} \mathcal{F}_m$ est aussi une classe $\Pi_1^0(\emptyset^{n-1})$ et en particulier $\Pi_1^0(\emptyset^n)$, dont le code est uniforme en la borne k , qui elle est calculable en à l'aide de $\emptyset^{(n+1)}$. ■

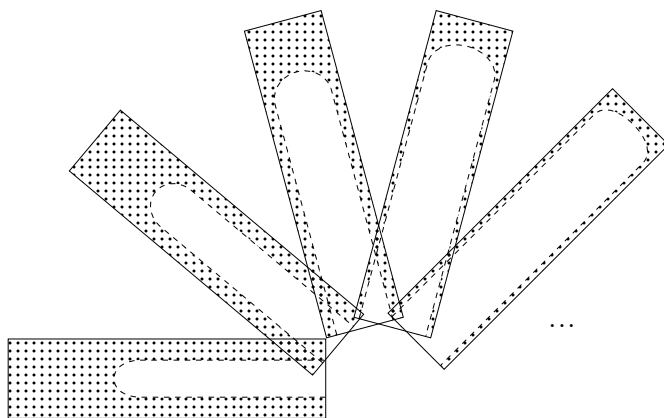


FIGURE 4.6 – *Illustration. Les rectangles sont les composantes Π_n^0 d'une classe Σ_{n+1}^0 croissante. Les contours arrondis en pointillés sont des fermés contenus dans chaque composante. La partie grisée est la partie qui manque, dont la mesure doit décroître suffisamment rapidement. De plus il ne sera pas nécessaire de prendre des classes fermées dans chaque composante : on peut s'arrêter quand les composantes de la classe Σ_{n+1}^0 n'ajoutent presque plus de mesure. Ainsi, on a bien une réunion finie de fermés qui reste fermée.*

Exercice 4.7. (★) Montrer que la borne calculatoire de la proposition 4.3 ne peut pas être simplifiée : pour tout n , il existe $\mathcal{B}$ une classe Σ_n^0 telle que le réel $\lambda(\mathcal{B})$ permet de calculer $\emptyset^{(n)}$. $\diamond$

Chapitre 18

Aléatoire au sens de Martin-Löf

Nous avons à présent les outils nécessaires pour définir l'aléatoire de Martin-Löf. L'idée est de définir un ensemble X comme étant aléatoire s'il n'a aucune propriété « atypique », une propriété étant atypique si la classe des ensembles qui la partagent est de mesure 0.

1. Intuitions et définitions

Reprenons notre exemple de la loi des grands nombres de la section 17-2.1. Nous avons défini pour un certain $\varepsilon > 0$ la classe $\mathcal{A}_\varepsilon = \bigcap_n \mathcal{U}_{\varepsilon,n}$ où

$$\mathcal{U}_{\varepsilon,n} = \bigcup_{m \geq n} [C_{\varepsilon,m}] \text{ et } C_{\varepsilon,m} = \left\{ \sigma \in 2^m : \left| \frac{\#\{i \leq m : \sigma(i)=0\}}{m} - \frac{1}{2} \right| > \varepsilon \right\},$$

où $[C_{\varepsilon,m}]$ dénote l'ouvert décrit par l'ensemble $C_{\varepsilon,m} \subseteq 2^{<\mathbb{N}}$. La classe $\mathcal{A}_\varepsilon$ contient tous les X dont la fréquence de 1 parmi ses préfixes est infiniment souvent au-dessus de $1/2 + \varepsilon$ ou alors infiniment souvent en dessous de $1/2 - \varepsilon$. Il est possible de montrer, en utilisant les inégalités de Hoeffding [93], que l'on a $\lambda([C_{\varepsilon,m}]) \leq 2e^{-2m\varepsilon^2}$, pour tout m . Notons que les classes $([C_{\varepsilon,m}])_{m \in \mathbb{N}}$ ne sont pas forcément disjointes, et rappelons ici la propriété de *sous-additivité de la mesure* : si $(\mathcal{B}_n)_{n \in \mathbb{N}}$ est une suite quelconque de classes mesurables, alors $\lambda(\bigcup_n \mathcal{B}_n) \leq \sum_n \lambda(\mathcal{B}_n)$.

Posons $a_m = e^{-2m\varepsilon^2}$. Alors, pour tout n , en utilisant la sous-additivité de la mesure, on a $\lambda(\mathcal{U}_{\varepsilon,n}) \leq \sum_{m \geq n} 2a_m = 2a_n \times (1 + a_1^1 + a_1^2 + \dots)$. La formule de convergence des séries géométriques donne $\lambda(\mathcal{U}_{\varepsilon,n}) \leq 2a_n/(1 - a_1)$. Quand n tends vers l'infini, la suite $2a_n/(1 - a_1)$ converge clairement vers 0. Il s'ensuit que $\lambda(\mathcal{U}_{\varepsilon,n})$ se rapproche de plus en plus de 0 quand n augmente, et donc que $\lambda(\bigcap_n \mathcal{U}_{\varepsilon,n}) = 0$.

Pour tout ε , chaque classe $\bigcap_n \mathcal{U}_{\varepsilon,n}$ sera donc comme un *test* à passer pour être accepté dans le club des nombres aléatoires : le test en question est une classe de mesure 0 spécifique dans laquelle aucun aléatoire digne de ce nom ne peut figurer. La sous-additivité de la mesure implique que $\bigcup_{\varepsilon \in \mathbb{Q}^+} \bigcap_n \mathcal{U}_{\varepsilon,n}$ est toujours une classe de mesure 0. Un nombre aléatoire digne de ce nom n'y figure donc pas. L'intuition de Martin-Löf est alors la suivante : les critères importants pour nous les mathématiciens, qui recherchent une définition formelle de l'aléatoire rejoignant notre intuition, sont capturables par des classes de mesure 0 du même ordre de complexité que chaque $\bigcap_n \mathcal{U}_{\varepsilon,n}$: des classes Π_2^0 . Enfin, l'idée clef de Martin-Löf est la suivante : si l'on considère tous les Π_2^0 de mesure 0, on a affaire à une infinité dénombrable de tests que chaque suite aléatoire doit réussir. En revanche, avec une simple restriction sur les Π_2^0 que l'on considère, il est possible de tous les regrouper en un seul et même test, les contenant tous (nous le verrons avec le corollaire 2.2). Il suffit de considérer les classes Π_2^0 décroissantes de la forme $\bigcap_n \mathcal{U}_n$ telles que $\lambda(\mathcal{U}_n) \leq 2^{-n}$ pour tout n : on ne cherche pas simplement à ce que $\lambda(\bigcap_n \mathcal{U}_n)$ soit égal à 0, mais aussi à ce que la convergence vers 0 se fasse suffisamment rapidement.

Définition 1.1 (Martin-Löf [150]). Un *test de Martin-Löf* est donné par $\bigcap_n \mathcal{U}_n$, une classe Π_2^0 décroissante telle que $\lambda(\mathcal{U}_n) \leq 2^{-n}$ pour tout n . Un ensemble $Z \in 2^{\mathbb{N}}$ *passse* le test si $Z \notin \bigcap_n \mathcal{U}_n$. Si Z ne passe pas le test, il est *capturé* par le test. $\diamond$

Notons que la condition $\lambda(\mathcal{U}_n) \leq 2^{-n}$ pour un test $\bigcap_n \mathcal{U}_n$ peut être relaxée : il suffit qu'il existe une fonction calculable $f : \mathbb{N} \rightarrow \mathbb{Q}$ telle que $\lim_n f(n) = 0$ et telle que $\lambda(\mathcal{U}_n) \leq f(n)$ pour tout n . On peut alors à partir de cela définir une autre classe Π_2^0 satisfaisant la définition donnée plus haut, en cherchant simplement pour tout n de trouver la n -ième composante $\mathcal{U}_{m_n}$ telle que $\lambda(\mathcal{U}_{m_n}) \leq 2^{-n}$.

Définition 1.2 (Martin-Löf [150]). Un ensemble Z est *aléatoire au sens de Martin-Löf* s'il passe tous les tests de Martin-Löf. $\diamond$

Nous avons montré ci-dessus que la classe $\bigcap_n \mathcal{U}_{\varepsilon,n}$, capturant les ensembles dont la fréquence de 1 parmi ses préfixes est infiniment souvent au-dessus de $1/2 + \varepsilon$, ou infiniment souvent en dessous de $1/2 - \varepsilon$, est telle que

$$\lambda(\mathcal{U}_{\varepsilon,n}) \leq 2a_n/(1 - a_1), \quad \text{où } a_n = e^{-2n\varepsilon^2}.$$

Chaque $\mathcal{U}_n$ est donc bien borné par une fonction calculable en n , et qui tend vers 0. Donc, si Z est un aléatoire de Martin-Löf, il ne peut appartenir à aucune classe $\bigcap_n \mathcal{U}_{\varepsilon,n}$, et la fréquence de ses 1 parmi ses préfixes va converger vers $1/2$. On peut généraliser ce résultat comme suit.

Exercice 1.3. (*) Généraliser l'exemple donné en début de section pour montrer que pour tout ensemble X aléatoire au sens de Martin-Löf, pour toute chaîne σ de taille n , si $\tau_0\tau_1\tau_2\dots = X$ est le découpage de X en chaînes de taille n , alors la fréquence des $i \leq m$ tels que $\tau_i = \sigma$ converge vers $2^{-|\sigma|}$ quand m tend vers $+\infty$.

Indication. – On pourra utiliser l'inégalité de Hoeffding pour X écrit en base 2^n . Pour toute base q et tout $a < q$, l'inégalité de Hoeffding donne

$$\lambda([C_{\varepsilon,m}^{q,a}]) \leq 2e^{-2m\varepsilon^2},$$

où

$$C_{\varepsilon,m}^{q,a} = \left\{ \sigma \in q^m : \left| \frac{\#\{i \leq m : \sigma(i) = a\}}{m} - \frac{1}{q} \right| > \varepsilon \right\}. \quad \diamond$$

Nous donnons tout de suite un autre type de test, moins restrictif mais qui donne la même notion d'aléatoire, et qui nous sera utile de temps à autre.

Définition 1.4

Un *test de Solovay* est donné par une suite calculable $(\mathcal{U}_n)_{n \in \mathbb{N}}$ de classe Σ_1^0 telle que $\sum_n \lambda(\mathcal{U}_n)$ est finie. Un ensemble Z passe le test de Solovay si Z n'appartient qu'à un nombre fini de classes $\mathcal{U}_n$. Sinon, Z est capturé par le test. $\diamond$

Théorème 1.5

Un ensemble est aléatoire au sens de Martin-Löf si, et seulement si, il passe tous les tests de Solovay.

PREUVE. Une direction est triviale : tout test de Martin-Löf est un test de Solovay. Pour l'autre direction, supposons Z capturé par un test de Solovay $(\mathcal{U}_n)_{n \in \mathbb{N}}$. Il doit exister m suffisamment grand tel que $\sum_{n \geq m} \lambda(\mathcal{U}_n) < 1$. Notons que Z est alors toujours capturé par $(\mathcal{U}_n)_{n \geq m}$, et l'on peut donc considérer sans perte de généralité $\sum_n \lambda(\mathcal{U}_n) < 1$.

On définit $\mathcal{V}_m$ comme étant la classe Σ_1^0 des ensembles X appartenant au moins à 2^m classes $\mathcal{U}_{i_1}, \mathcal{U}_{i_2}, \dots, \mathcal{U}_{i_{2^m}}$ distinctes. Soit W_m l'ensemble sans préfixe minimal qui décrit la classe $\mathcal{V}_m$. Pour chaque chaîne $\sigma \in W_m$, il vient $[\sigma] \subseteq \mathcal{U}_{i_j}$ pour au moins 2^m classes $\mathcal{U}_{i_1}, \mathcal{U}_{i_2}, \dots, \mathcal{U}_{i_{2^m}}$ distinctes. En particulier, $2^m \times 2^{-|\sigma|} = \lambda(\mathcal{U}_{i_1} \cap [\sigma]) + \dots + \lambda(\mathcal{U}_{i_{2^m}} \cap [\sigma])$. Comme $\sigma_1, \sigma_2 \in W_m$ implique $[\sigma_1] \cap [\sigma_2] = \emptyset$, alors $\sigma_1, \sigma_2 \in W_m$ implique

$$(\mathcal{U}_i \cap [\sigma_1]) \cap (\mathcal{U}_i \cap [\sigma_2]) = \emptyset, \text{ pour tout } i.$$

On a donc $2^m \sum_{\sigma \in W_m} 2^{-|\sigma|} \leq \sum_n \lambda(\mathcal{U}_n) < 1$. Cela donne $2^m \lambda(\mathcal{V}_m) \leq 1$, et donc $\lambda(\mathcal{V}_m) < 2^{-m}$. Ainsi, $\bigcap_m \mathcal{V}_m$ est un test de Martin-Löf, qui par hypothèse sur Z le capture. $\blacksquare$

2. Les aléatoires de Martin-Löf et de Chaitin/Levin coïncident

Martin-Löf dans son article [150] a montré comment construire un *test universel* : un test de Martin-Löf les contenant tous. Quelques années plus tard, Levin et Schnorr ont montré indépendamment que ce test universel pouvait en fait être défini en utilisant la complexité de Kolmogorov sans préfixe : les nombres aléatoires au sens de Chaitin/Levin sont exactement ceux qui sont aléatoires au sens de Martin-Löf. Ce résultat fait de l'aléatoire de Martin-Löf une notion d'aléatoire *robuste*, dans le sens où elle possède plusieurs caractérisations qui n'ont *a priori* rien à voir entre elles.

Théorème 2.1 (Levin [140], Schnorr [193])

Un ensemble Z est aléatoire au sens de Martin-Löf ssi il est aléatoire au sens de Chaitin/Levin.

PREUVE. Supposons que Z ne soit pas aléatoire au sens de Chaitin/Levin. Alors, pour tout c , il existe n tel que $K(Z \upharpoonright_n) < n - c$. Donc, Z appartient à l'ensemble $\bigcap_c \mathcal{U}_c$, où $\mathcal{U}_c = \{X : \exists n \ K(X \upharpoonright_n) < n - c\}$. Notons que chaque $\mathcal{U}_c$ est une classe Σ_1^0 uniformément en c , et donc que $\bigcap_c \mathcal{U}_c$ est une classe Π_2^0 . Pour chaque $c \in \mathbb{N}$, soit W_c un ensemble sans préfixe tel que $[W_c] = \mathcal{U}_c$ et tel que pour tout $\sigma \in W_c$, $K(\sigma) < |\sigma| - c$. Notons que W_c n'est pas nécessairement c. e. On a donc

$$\sum_{\sigma \in W_c} 2^{-|\sigma|+c} \leq \sum_{\sigma \in W_c} 2^{-K(\sigma)}.$$

Comme K est la complexité sans préfixe, on a $\sum_{\sigma \in W_c} 2^{-K(\sigma)} \leq 1$. Il en découle que $\lambda(\mathcal{U}_c) = \sum_{\sigma \in W_c} 2^{-|\sigma|} \leq 2^{-c}$. Par suite, $\bigcap_c \mathcal{U}_c$ est un test de Martin-Löf, et donc que Z n'est pas aléatoire au sens de Martin-Löf.

Supposons à présent que Z ne soit pas aléatoire au sens de Martin-Löf. Il existe alors $\bigcap_n \mathcal{U}_n$ une classe Π_2^0 telle que $\mathcal{U}_{n+1} \subseteq \mathcal{U}_n$, telle que $\lambda(\mathcal{U}_n) \leq 2^{-n}$ et pour laquelle $Z \in \bigcap_n \mathcal{U}_n$. Soit $W_n \subseteq 2^{<\mathbb{N}}$ un ensemble Σ_1^0 sans préfixe tel que $\mathcal{U}_n = \bigcup_{\sigma \in W_n} [\sigma]$. On définit pour tout c l'ensemble borné de requêtes L_c en énumérant $(\sigma, |\sigma| - c + 1)$ dans L_c , pour tout $\sigma \in W_{2c}$. Le poids de L_c est alors égal à $\sum_{\sigma \in W_{2c}} 2^{-|\sigma|+c-1} = 2^{c-1} \lambda(\mathcal{U}_{2c}) \leq 2^{c-1} 2^{-2c} = 2^{-c-1}$. En particulier la réunion des L_c est un ensemble de requêtes dont le poids est borné par $\sum_{c \in \mathbb{N}} 2^{-c-1} = 1$. D'après le théorème KC (théorème 16-3.3), on a donc une machine M sans préfixe telle que $K_M(\sigma) \leq |\sigma| - c + 1$, pour toute chaîne $\sigma \in W_{2c}$. Il s'ensuit que pour tout $Z \in \bigcap_n \mathcal{U}_n$, on a pour tout c un entier n tel que $K_M(Z \upharpoonright_n) \leq n - c$. Aucun $Z \in \bigcap_n \mathcal{U}_n$ n'est donc aléatoire au sens de Chaitin/Levin. ■

Corollaire 2.2

Il existe un test de Martin-Löf universel, c'est-à-dire un test de Martin-Löf qui les contient tous.

PREUVE. D'après la preuve du théorème précédent, l'ensemble

$$\bigcap_c \mathcal{U}_c, \text{ où } \mathcal{U}_c = \{X : \exists n \ K(X \upharpoonright_n) < n - c\}$$

est un test de Martin-Löf contenant tous les tests de Martin-Löf. ■

Notation

On appellera *MLR* les ensembles aléatoires au sens de Martin-Löf, de l'anglais « Martin-Löf random ».

Corollaire 2.3

Il existe un ensemble Z low et MLR. Il existe un ensemble Z calculatoirement dominé et MLR.

PREUVE. Comme la classe des MLR est Σ_2^0 , c'est-à-dire une réunion effective de Π_1^0 , chacune de ces classes Π_1^0 contient d'après le théorème 8-4.3 de la base low un ensemble low, et d'après le théorème 8-4.5 de la base calculatoirement dominé, un ensemble calculatoirement dominé. ■

3. Aléatoire et degré Turing

Voyons à présent un théorème qui fait écho au théorème 10-3.37 qui nous disait que pour X non calculable, la classe des ensembles qui calculent X est maigre. De la même manière, Sacks a montré que la classe des ensembles qui calculent une suite $X \in 2^{\mathbb{N}}$ non calculable est de mesure nulle.

Nous utilisons pour cela un remarquable lemme de Lebesgue qui découle du théorème 17-4.4 : le fait que tout borélien $\mathcal{B}$ de mesure positive « concentre toute sa mesure dans des intervalles » : il n'est pas possible de construire $\mathcal{B}$ de mesure disons $1/4$, de telle manière à ce que pour tout σ , la classe $\mathcal{B} \cap [\sigma]$ ne contienne qu'un quart de $[\sigma]$ en termes de mesure. Il y aura forcément un intervalle $[\sigma]$ au sein duquel $\mathcal{B}$ occupe « presque toute la place ». On introduit pour cela la notation suivante.

Notation

Soit $\mathcal{B} \subseteq 2^{\mathbb{N}}$ un borélien et soit $[\sigma]$ un cylindre. On note $\lambda(\mathcal{B} \mid [\sigma])$ pour la mesure de $\mathcal{B}$ relativement à $[\sigma]$, c'est-à-dire :

$$\lambda(\mathcal{B} \mid [\sigma]) = \frac{\lambda(\mathcal{B} \cap [\sigma])}{\lambda([\sigma])}.$$

Par exemple, si $\lambda(\mathcal{B} \mid [\sigma]) = 1/2$, cela signifie que $\mathcal{B}$ occupe « la moitié de la place » dans le cylindre $[\sigma]$. Voyons à présent le lemme de densité Lebesgue.

Lemme 3.1 (Lemme de densité de Lebesgue)

Soit $\mathcal{B}$ un borélien de mesure positive. Alors, pour tout $\varepsilon > 0$ il existe un cylindre $[\sigma]$ tel que $\lambda(\mathcal{B} \mid [\sigma]) > 1 - \varepsilon$. ★

PREUVE. L'idée est simple. D'après le théorème 17-4.4, on peut approximer $\mathcal{B}$ par un ouvert $\mathcal{U}$ le contenant et dont la mesure est aussi proche de $\mathcal{B}$ que l'on le souhaite. Pour un ouvert dont la mesure est suffisamment proche de celle de $\mathcal{B}$, il n'est pas possible que la mesure de $\mathcal{B}$ à l'intérieur de chaque cylindre $[\sigma]$ de l'ouvert soit trop petite, sinon la mesure totale de $\mathcal{B}$ est trop petite par rapport à celle de $\mathcal{U}$. Nous donnons à présent la preuve formelle.

Fixons $\varepsilon > 0$. D'après le théorème 17-4.4, il existe un ouvert $\mathcal{U} \supseteq \mathcal{B}$ tel que $\lambda(\mathcal{U} \setminus \mathcal{B}) < \varepsilon \lambda(\mathcal{B})$. Soit $W \subseteq 2^{<\mathbb{N}}$ sans préfixe tel que $\mathcal{U} = \bigcup_{\sigma \in W} [\sigma]$. Notons que l'on a $\lambda(\mathcal{B}) = \sum_{\sigma \in W} \lambda(\mathcal{B} \cap [\sigma])$. Supposons par l'absurde que, pour tout $\sigma \in W$, on ait $\lambda(\mathcal{B} \mid [\sigma]) \leq 1 - \varepsilon$. Alors,

$$\sum_{\sigma \in W} \lambda(\mathcal{B} \cap [\sigma]) \leq \sum_{\sigma \in W} (1 - \varepsilon) \lambda([\sigma]) = (1 - \varepsilon) \sum_{\sigma \in W} \lambda([\sigma]) = (1 - \varepsilon) \lambda(\mathcal{U}).$$

On a donc $\lambda(\mathcal{B}) \leq (1 - \varepsilon) \lambda(\mathcal{U})$. Mais, par hypothèse,

$$\lambda(\mathcal{U}) - \lambda(\mathcal{B}) = \lambda(\mathcal{U} \setminus \mathcal{B}) < \varepsilon \lambda(\mathcal{B}) \leq \varepsilon \lambda(\mathcal{U}).$$

Cela donne $\lambda(\mathcal{B}) > \lambda(\mathcal{U}) - \varepsilon \lambda(\mathcal{U}) = (1 - \varepsilon) \lambda(\mathcal{U})$, et une contradiction. ■

Notons que Lebesgue a montré quelque chose de plus fort encore, que nous verrons avec le théorème 19-4.6 : les intervalles σ du lemme précédent sont en fait très nombreux : pour presque tous les $X \in \mathcal{B}$, la quantité $\lambda(\mathcal{B} \mid [X \upharpoonright_n])$ tend vers 1 quand n tend vers $+\infty$. Voyons à présent le théorème de Sacks.

Théorème 3.2 (Sacks [188])

Soit $Y \in 2^{\mathbb{N}}$ non calculable. Alors, $\lambda(\{X \in 2^{\mathbb{N}} : X \geq_T Y\}) = 0$.

PREUVE. Soit Φ une fonctionnelle Turing. Supposons par l'absurde que $\lambda(\{X \in 2^{\mathbb{N}} : \Phi(X) = Y\}) > 0$. D'après le lemme 3.1 de densité de Lebesgue, il existe une chaîne σ telle que $\lambda(\{X \in 2^{\mathbb{N}} : \Phi(X) = Y\} \mid [\sigma]) > 1/2$.

On décrit à présent un algorithme pour calculer Y , dont le principe se résume à un « vote à la majorité » : pour tout n et pour tout $i \in \{0, 1\}$, on énumère petit à petit l'ouvert $\mathcal{U}_{n,i}$ décrit par les chaînes $\tau \succeq \sigma$ telles que $\Phi(\tau, n) \downarrow = i$. D'après notre hypothèse sur σ , pour tout n , on doit avoir $\lambda(\mathcal{U}_{n,i} \mid [\sigma]) > 1/2$, pour $i = Y(n)$. Notons que pour des raisons

évidentes de « manque de place », on ne peut pas avoir $\lambda(\mathcal{U}_{n,i} \mid [\sigma]) > 1/2$, pour $i \neq Y(n)$. Il suffit donc d'attendre d'avoir $\lambda(\mathcal{U}_{n,i} \mid [\sigma]) > 1/2$, pour un certain i . À ce moment, on est sûr que $Y(n) = i$.

L'algorithme contredit le fait que Y est non calculable. Il s'ensuit que

$$\lambda(\{X \in 2^{\mathbb{N}} : \Phi(X) = Y\}) = 0.$$

Comme c'est le cas pour toutes les fonctionnelles, et qu'une réunion dénombrable d'ensembles de mesure 0 est encore de mesure 0, on en déduit donc l'égalité $\lambda(\{Y \in 2^{\mathbb{N}} : Y \geq_T X\}) = 0$. ■

Le théorème précédent implique qu'un ensemble Z suffisamment aléatoire, ne peut pas calculer un ensemble non calculable Y . Cela montre en particulier que les algorithmes probabilistes sont impuissants à fournir plus de puissance de calcul : à supposer que l'on dispose d'un moyen de produire des bits aléatoires (via un processus physique par exemple), la probabilité pour que la suite produite permette par exemple de calculer l'arrêt est nulle.

Nous voyons à présent un théorème dual : étant donné un ensemble arbitraire, on peut toujours trouver un aléatoire de Martin-Löf le calculant. L'idée générale est la suivante : étant donné un arbre calculable T où chaque nœud a au moins deux extensions incomparables, il est trivial de calculer une bijection entre $2^{\mathbb{N}}$ et $[T]$: si $f(\sigma)$ est défini, alors $f(\sigma 0)$ et $f(\sigma 1)$ sont envoyés chacun vers les premières extensions incomparables de $f(\sigma)$ dans T . Comme la bijection est calculable, on en déduit que $[T]$ contient un élément Turing équivalent à n'importe quel ensemble.

Si à présent l'arbre T contient en plus des feuilles — comme c'est le cas pour les arbres qui représentent des classes Π_1^0 —, l'algorithme ne fonctionne plus du tout. Le problème est de savoir quel nœud est « réellement » branchant : il se peut qu'un nœud soit branchant, mais que l'une de ses extensions n'aboutisse finalement qu'à des feuilles. Il est en fait possible de construire des classes Π_1^0 indénombrables dont aucun membre ne calcule l'arrêt, ou même dont aucun membre ne calcule d'ensemble Martin-Löf aléatoire [38, Lemme 5.1]. En revanche, quand la classe Π_1^0 en question est de mesure positive, la situation change : la mesure positive nous donne une certaine garantie sur le fait que « beaucoup » de nœuds sont branchants, ce qui nous permet de développer un encodage — non calculable — de n'importe quel élément de telle manière à ce qu'un algorithme de décodage soit possible. Nous avons pour cela besoin d'un lemme garantissant la rapidité avec laquelle on peut trouver des nœuds branchants dans un arbre de mesure positive.

Lemme 3.3 (Kučera [128])

Soit $\mathcal{B}$ un borélien. Supposons $\lambda(\mathcal{B} \mid [\sigma]) \geq 2^{-n}$. Alors, il existe deux extensions distinctes $\tau_0, \tau_1 \succeq \sigma$ de taille $|\sigma| + n + 1$ telles que $\lambda(\mathcal{B} \mid [\tau_i]) \geq 2^{-n-1}$. ★

PREUVE. Il s'agit d'un simple argument de comptage : supposons que pour toutes les chaînes τ de taille $|\sigma| + n + 1$ qui étendent σ , sauf une, on ait $\lambda(\mathcal{B} \mid [\tau]) < 2^{-n-1}$. Alors, en considérant τ tel que $|\tau| = |\sigma| + n + 1$, on a :

$$\begin{aligned} \lambda(\mathcal{B} \cap [\sigma]) &\leq 2^{-|\sigma|-n-1} + (2^{n+1} - 1)2^{-|\tau|-n-1} \\ &\leq 2^{-|\sigma|-n-1} + (2^{n+1} - 1)2^{-|\sigma|-2n-2} \\ &\leq 2^{-|\sigma|-n-1} + 2^{n+1}2^{-|\sigma|-2n-2} - 2^{-|\sigma|-2n-2} \\ &\leq 2^{-|\sigma|-n-1} + 2^{-|\sigma|-n-1} - 2^{-|\sigma|-2n-2} \\ &< 2^{-|\sigma|-n}, \end{aligned}$$

ce qui contredit $\lambda(\mathcal{B} \mid [\sigma]) \geq 2^{-n}$. ■

D'après le lemme p, notons que si $\mathcal{B}$ est une classe Π_1^0 de mesure positive et que T soit l'arbre calculable qui le représente, alors $\lambda([T] \mid [\sigma]) \geq 2^{-n}$ implique que la prochaine extension branchante de σ arrivera avant $n + 1$ bits.

Théorème 3.4 (Kučera [128], Gács [70])

Soit $X \in 2^{\mathbb{N}}$. Alors, il existe $Z \in 2^{\mathbb{N}}$ un ensemble MLR tel que $Z \geq_T X$.

PREUVE. On montre le théorème suivant. Soit $\mathcal{P}$ une classe Π_1^0 de mesure positive; alors, pour tout X il existe $Z \in \mathcal{P}$ tel que $Z \geq_T X$. Le lecteur peut consulter la figure 3.5 pour une illustration de la preuve. Fixons X . Construisons notre élément $Z \in \mathcal{P}$ tel que $Z \geq_T X$. Fixons d'abord c tel que $\lambda(\mathcal{P}) > 2^{-c}$. On définit, en rapport avec le lemme 3.3, les constantes $m_0 = 0$ et $m_{n+1} = m_n + c + n + 1$.

On définit $\sigma_0 = \epsilon$. Supposons que σ_n de taille m_n est défini tel que

$$\lambda(\mathcal{P} \mid [\sigma_n]) \geq 2^{-c-n}.$$

On définit une extension σ_{n+1} avec la même propriété. D'après le lemme 3.3, il existe deux extensions τ_0, τ_1 distinctes de taille $m_n + c + n + 1$ telles que $\lambda(\mathcal{P} \mid [\tau_i]) \geq 2^{-c-(n+1)}$. Si $X(n) = 0$, on définit alors σ_{n+1} comme l'extension la plus à gauche vérifiant l'inégalité. Si $X(n) = 1$, on définit alors σ_{n+1} comme l'extension la plus à droite vérifiant l'inégalité. Quant à Z , il est défini comme le point limite de $\sigma_0 \prec \sigma_1 \prec \dots \prec \sigma_n \prec \sigma_{n+1} \prec \dots$.

Il est clair que $Z \in \mathcal{P}$. On doit maintenant détailler la manière dont on peut utiliser Z pour calculer X . Pour trouver le bit $X(n)$, soit $\sigma = Z \upharpoonright_{m_n}$. On co-énumère l'ensemble A des chaînes $\tau \succeq \sigma$ de taille m_{n+1} telles

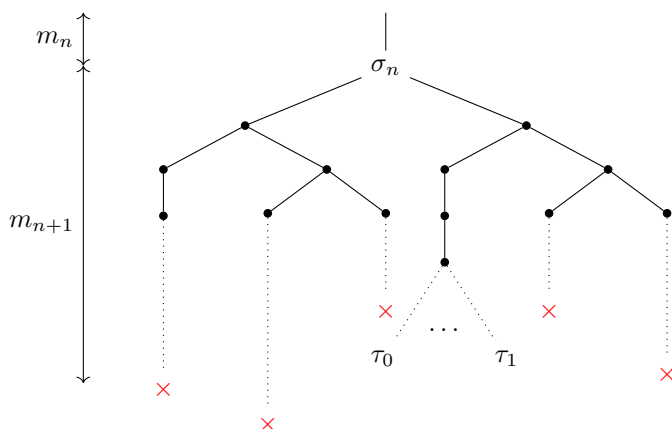


FIGURE 3.5 – *Illustration de la preuve du théorème de Kučera/Gács. On a défini notre chaîne σ_n de taille m_n . Nous avons la garantie qu'il existe deux chaînes distinctes τ_0, τ_1 de taille m_{n+1} et telles que l'on puisse continuer la construction à partir de n'importe laquelle des deux. Il suffit de choisir τ_0 comme la plus à gauche de ces chaînes et τ_1 comme étant la plus à droite. On choisit $\sigma_{n+1} = \tau_0$ pour encoder un 0, et $\sigma_{n+1} = \tau_1$ pour encoder un 1. Pour le décodage, en ayant la connaissance de σ_n et σ_{n+1} , il suffit alors d'attendre que suffisamment de nœuds de l'arbre étendant σ_n se terminent en feuille — plus précisément que la condition de mesure les concernant descende sous un certain seuil — jusqu'à ce que σ_{n+1} devienne le nœud le plus à gauche ou le plus à droite de ce qu'il reste. À ce moment, on sait quel bit a été encodé.*

que $\lambda(\mathcal{P} \mid [\tau]) \geq 2^{-c-(n+1)}$. Si jamais $Z \upharpoonright_{m_{n+1}}$ apparaît à une certaine étape de calcul t comme étant la chaîne la plus à gauche de $A[t]$, alors $X(n) = 0$. Si jamais $Z \upharpoonright_{m_{n+1}}$ apparaît à une certaine étape de calcul t comme étant la chaîne la plus à droite de $A[t]$, alors $X(n) = 1$. Par construction, un de ces deux événements doit forcément arriver. ■

4. Aléatoire et degré DNC

Le théorème 3.4 de Kučera/Gács nous dit qu'il existe un aléatoire de Martin-Löf *au-dessus* de chaque degré Turing, mais pas nécessairement *dans* chaque degré Turing. Nous allons en fait voir que si par exemple tout ensemble 1-générique peut être calculé par un ensemble Martin-Löf aléatoire, aucun ensemble Martin-Löf aléatoire ne peut être en revanche calculé par un ensemble 1-générique. L'idée est simplement que tout Martin-Löf aléatoire est de degré DNC et, comme nous l'avons vu avec le théo-

rème 10-3.21, aucun ensemble 1-générique ne borne de degré DNC. Précisément, nous allons montrer le théorème suivant, qui donne deux autres caractérisations des degrés DNC. L'équivalence (1) $\leftrightarrow$ (2) a été montrée par Kjos-Hanssen, Merkle et Stephan [112], et l'équivalence (1) $\leftrightarrow$ (3) par Greenberg et Miller [76].

Théorème 4.1

Soit $X \in 2^{\mathbb{N}}$. Les énoncés suivants sont équivalents :

- (1) X est de degré DNC;
- (2) X calcule une fonction $f : n \rightarrow 2^{<\mathbb{N}}$ telle que $K(f(n)) \geq n$;
- (3) X calcule un sous-ensemble infini d'un ensemble MLR.

Afin de montrer (1) $\rightarrow$ (3), nous utiliserons un autre lemme de Kučera.

Lemme 4.2 (Kučera [128]). Soit $\mathcal{P}$ une classe Π_1^0 de mesure positive. Il existe $\mathcal{Q} \subseteq \mathcal{P}$ une sous-classe Π_1^0 de mesure positive et un entier c tels que, pour $\mathcal{P}_e$ la classe Π_1^0 de code e , on a $\mathcal{Q} \cap \mathcal{P}_e \neq \emptyset \rightarrow \lambda(\mathcal{Q} \cap \mathcal{P}_e) > 2^{-e-c}$. ★

PREUVE. Soit c tel que $\lambda(\mathcal{P}) > 2^{-c+1}$. On définit $\mathcal{Q}$ comme étant la classe qui co-énumère $\mathcal{P}$ et dans le même temps, à chaque étape s , pour tout $e \leq s$, si la mesure de $\mathcal{Q}[s] \cap \mathcal{P}_e[s]$ descend en dessous 2^{-e-c} , enlève $\mathcal{P}_e[s]$ de $\mathcal{Q}[s]$.

Pour chaque entier e , on enlève un morceau d'une mesure d'au plus 2^{-e-c} . La mesure totale enlevée est dès lors bornée par $\sum_{e \in \mathbb{N}} 2^{-e-c} = 2^{-c+1}$. Comme $\lambda(\mathcal{P}) > 2^{-c+1}$, on a $\lambda(\mathcal{Q}) > 0$. ■

Nous pouvons à présent montrer le théorème annoncé.

PREUVE DU THÉORÈME 4.1. Montrons (1) $\rightarrow$ (2). Supposons X de degré DNC. Pour tout n , on peut calculer le code a_n de l'ensemble W_{a_n} qui énumère toutes les chaînes σ telles que $K(\sigma) < n$. Notons que

$$|W_{a_n}| < \sum_{m < n} 2^m = 2^n.$$

D'après le théorème 7-2.6, l'ensemble X calcule donc uniformément en n une chaîne $\sigma \notin W_{a_n}$, et donc telle que $K(\sigma) \geq n$.

Montrons à présent (2) $\rightarrow$ (1). Supposons que X ne soit pas de degré DNC. Alors, pour toute fonction $f \leq_T X$, on a $f(n) = \Phi_n(n) \downarrow$ pour une infinité de n . Or, $\Phi_n(n) \downarrow$ implique $K(\Phi_n(n)) <^+ K(n)$ (voir l'exercice 16-2.3 si cette étape n'est pas claire), et l'on a $K(n) \leq^+ \log_2(n) + 2 \log_2(\log_2(n))$, d'après la proposition 16-2.5, ce qui donne

$$K(\Phi_n(n)) \leq^+ \log_2(n) + 2 \log_2(\log_2(n)).$$

Or, $\log_2(n) + 2 \log_2(\log_2(n)) < n$, pour n suffisamment grand.

Montrons à présent (1) $\rightarrow$ (3). Soit X de degré DNC. Soit $\mathcal{P}$ une classe Π_1^0 de mesure positive ne contenant que des ensembles aléatoires au sens de Martin-Löf. D'après le lemme 4.2, on peut supposer qu'il existe $c \in \mathbb{N}$ tel que l'on a on a $\mathcal{P} \cap \mathcal{P}_e \neq \emptyset$ implique $\lambda(\mathcal{P} \cap \mathcal{P}_e) > 2^{-c-e}$ pour $\mathcal{P}_e$ la classe Π_1^0 de code e . Pour un ensemble $S \subseteq \mathbb{N}$ quelconque, on définit

$$\mathcal{Q}_S = \{Y : S \subseteq Y\}.$$

Notons que $\mathcal{Q}_S$ est Π_1^0 pour tout $S \subseteq \mathbb{N}$ fini. Supposons que l'on ait défini $S_n \subseteq \mathbb{N}$ de taille n tel que $\mathcal{Q}_{S_n}$ est de code e_n et tel que $\mathcal{P} \cap \mathcal{Q}_{S_n} \neq \emptyset$. Soit W_n l'ensemble c.e. des entiers a tels que $\mathcal{P} \cap \mathcal{Q}_{S_n \cup \{a\}} = \emptyset$. Par hypothèse, on a

$$\lambda(\mathcal{P} \cap \mathcal{Q}_{S_n}) > 2^{-c-e_n}.$$

Aussi, si jamais un entier a est dans W_n , cela signifie que pour tout élément Z de $\mathcal{P} \cap \mathcal{Q}_{S_n}$ on a $Z(a) = 0$. Pour une suite d'entiers $a_1, a_2, \dots, a_m$ fixée, l'ensemble des Z tels que $\forall i < m \ Z(a_i) = 0$ est un ensemble de mesure 2^{-m} puisque la moitié des éléments Z sont tels que $Z(a_1) = 0$, puis la moitié de cette moitié sont tels que $Z(a_2) = 0$, etc.

Comme $\lambda(\mathcal{P} \cap \mathcal{Q}_{S_n}) > 2^{-c-e_n}$, on a donc forcément au plus $c+e_n$ éléments dans W_n . On peut donc appliquer le théorème 7-2.6 pour calculer uniformément à partir de X un entier $a \notin W_n$, et donc tel que $\mathcal{P} \cap \mathcal{Q}_{S_n \cup \{a\}} \neq \emptyset$. On définit $S_{n+1} = S_n \cup \{a\}$, et l'on calcule un code e_{n+1} pour $\mathcal{Q}_{S_{n+1}}$. On calcule de cette manière à l'aide de X l'ensemble infini $S = \bigcup_n S_n$ pour lequel $\mathcal{P} \cap \mathcal{Q}_S = \bigcap_n \mathcal{P} \cap \mathcal{Q}_{S_n}$. Comme $\lambda(\mathcal{P} \cap \mathcal{Q}_{S_n}) > 0$ pour tout n , alors $\mathcal{P} \cap \mathcal{Q}_S$ est non vide pour tout n . En tant qu'intersection décroissante de fermés non vides, la classe $\mathcal{P} \cap \mathcal{Q}_S$ est non vide. En particulier, il existe un élément $Z \in \mathcal{P}$ tel que $S \subseteq Z$.

Montrons à présent (3) $\rightarrow$ (1). Soit $X \subseteq Z$ un sous-ensemble infini d'un MLR. Pour tout n , soit $f(n)$ le n -ième élément de X . Nous allons établir que

$$K(X \upharpoonright_{f(n)}) >^+ n.$$

Supposons le contraire, c'est-à-dire $\forall c \ \exists n \ K(X \upharpoonright_{f(n)}) < n - c$. Alors, on peut aussi compresser $Z \upharpoonright_{f(n)}$ de c bits, en utilisant machine M qui sur $\tau\sigma$ tel que $U(\tau) \downarrow = \rho$ renvoie la chaîne σ dans laquelle on intercale les 1 de ρ aux positions auxquels ils se trouvent dans ρ (si jamais σ est suffisamment grande). La chaîne σ représente ici le préfixe de $Z \upharpoonright_{f(n)}$ auquel on a « enlevé » les n bits à 1 de $X \upharpoonright_{f(n)}$. Elle est donc de taille $f(n) - n$. La chaîne τ est quant à elle une compression de $X \upharpoonright_{f(n)}$ de taille inférieure à $n - c$. La chaîne $\tau\sigma$ est donc une compression de taille inférieure à $f(n) - c$ d'un préfixe de Z de taille $f(n)$, ce qui aboutit à une contradiction pour c suffisamment grand. On a donc $K(X \upharpoonright_{f(n)}) >^+ n$ et, d'après (2) $\rightarrow$ (1), on a bien (3) $\rightarrow$ (1). ■

Corollaire 4.3

La classe des ensembles de degré DNC est de mesure 1.

PREUVE. On déduit aisément du précédent théorème que tous les ensembles MLR sont de degré DNC. ■

Nous avons vu avec l'exercice 8-7.6 que pour tout ensemble X , toute fonction $f : \mathbb{N} \rightarrow \{0, 1\}$ DNC relativement à X calculait X , autrement dit tout degré PA relativement à X calcule X . La situation est différente lorsque l'on considère des fonctions DNC à valeurs arbitraires.

Corollaire 4.4

Pour tout ensemble X , il existe une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ DNC relativement à X qui ne calcule pas X .

PREUVE. Par le corollaire 4.3 relativisé, la mesure de la classe des ensembles calculant une fonction DNC relativement à X est 1, tandis que par le théorème 3.2, la mesure de la classe des ensembles calculant X est 0. Il s'ensuit qu'il existe un ensemble calculant une fonction DNC relativement à X , mais ne calculant pas X .

Chapitre 19

Autres notions d'aléatoire

La définition d'aléatoire de Martin-Löf est-elle *la* bonne définition ? La concordance entre le point de vue « incompressibilité » et le point de vue « typicité » va dans ce sens. Il y a toutefois peu d'espoir d'obtenir de certitudes sur le fait que cette définition mathématique — ou une autre — vienne à épouser un jour avec perfection les contours des aspects épistémologiques dont elle relève.

Nous rejoignons ici le point de vue de Christopher Porter, qui dans une thèse pluridisciplinaire mathématiques/philosophie [179] défend la « no thesis thesis », selon laquelle il n'y a pas de définition absolue de ce qu'est un nombre aléatoire. À titre d'exemple, on peut légitimement considérer que la capacité à calculer $\emptyset'$ est une propriété atypique. C'est pourtant le cas du nombre aléatoire Ω , qui ne l'est donc pas tant que cela.

Nous voyons donc ici d'autres notions d'aléatoires, plus ou moins fortes et dignes d'intérêt, notamment à travers les liens que l'on peut tisser entre elles et la calculabilité. Notons qu'un très grand nombre de classes de nombres aléatoires ont été étudiées, certaines plus faibles et d'autres plus fortes ou même incomparables à l'aléatoire au sens de Martin-Löf. Notre objectif n'est pas ici d'en présenter une liste exhaustive, mais plutôt de donner une photographie de quelques-unes des plus emblématiques d'entre elles.

1. Les fortement MLR

Commençons d'abord par examiner ce qu'il se passe si l'on enlève la condition d'être effectivement de mesure 0 pour un test.

Définition 1.1. Un ensemble Z est *fortement MLR*^a s'il n'appartient à aucun ensemble Π_2^0 de mesure 0. ◇

a. Pour des raisons historiques dans l'avancée de la compréhension des différentes notions d'aléatoire, le concept est connu en anglais sous le nom de « weak-2-random ». Nous l'avons ici rebaptisé d'une manière que nous jugeons plus conforme à la réalité.

La notion d'ensemble fortement MLR est naturelle. Après tout, pourquoi s'embêter avec la condition de convergence rapide de la mesure des Π_2^0 vers 0? On obtient une notion plus forte, mais peut-être moins élégante, comme en témoigne la proposition et les corollaires suivants.

Proposition 1.2. Aucun ensemble Δ_2^0 n'est fortement MLR. ★

PREUVE. Soit A un ensemble Δ_2^0 , et soit $(A_s)_{s \in \mathbb{N}}$ une suite d'ensembles calculables telle que $\lim_s A_s = A$. On définit $\mathcal{U}_{\langle n, t \rangle} = \bigcup_{s > t} [A_s \upharpoonright n]$. Il est clair que $\bigcap_{\langle n, t \rangle \in \mathbb{N}} \mathcal{U}_{\langle n, t \rangle} = \{A\}$. En particulier, $\bigcap_{\langle n, t \rangle \in \mathbb{N}} \mathcal{U}_{\langle n, t \rangle}$ est un Π_2^0 de mesure 0 qui contient A . ■

Corollaire 1.3

L'ensemble des fortement MLR est strictement inclus dans celui des MLR.

PREUVE. En tant qu'ensemble approchable par la gauche, Ω est Δ_2^0 , et donc non fortement MLR. ■

Corollaire 1.4

Il n'existe aucun test universel pour la classe des fortement MLR.

PREUVE. Soit $\bigcap_n \mathcal{U}_n$ un Π_2^0 de mesure 0. Pour un certain n , l'ensemble $2^{\mathbb{N}} \setminus \mathcal{U}_n$ est donc un Π_1^0 de mesure positive. Soit T l'arbre calculable qui représente $2^{\mathbb{N}} \setminus \mathcal{U}_n$. Le chemin le plus à gauche de T est un ensemble approchable par la gauche, et donc Δ_2^0 , et donc aussi non fortement MLR. On en conclut qu'il n'existe aucune classe Π_2^0 de mesure 0 qui contienne tous les ensembles non fortement MLR. ■

Il est possible d'améliorer considérablement le corollaire 1.4. Yu Liang a en effet montré [236] que la classe des éléments qui ne sont pas fortement MLR n'est pas Π_2^0 (même relativement à un oracle quelconque, il n'existe pas de test capturant exactement les non fortement MLR) et même n'est pas Π_3^0 . La complexité borélienne de la classe des non fortement MLR est donc strictement Σ_3^0 .

On constate donc qu'en enlevant la condition de convergence rapide vers 0 dans la mesure des Π_2^0 , on enlève certains ensembles MLR — au moins ceux qui sont Δ_2^0 . Il paraît probable que ces derniers ne soient pas les seuls à se faire exclure. On peut en fait en donner une caractérisation élégante découverte par Downey, Nies, Weber et Yu.

Théorème 1.5 (Downey, Nies, Weber et Yu [48])

Soit Z un ensemble MLR. Les énoncés suivants sont équivalents :

- (1) *Z n'est pas fortement Martin-Löf aléatoire ;*
- (2) *Z calcule un ensemble Δ_2^0 non calculable ;*
- (3) *Z calcule un ensemble c. e. non calculable.*

PREUVE. Montrons d'abord (2) $\rightarrow$ (1). Soit A un ensemble Δ_2^0 non calculable avec $A = \lim_s A_s$. Soit Φ telle que $\Phi(Z) = A$. On définit $\mathcal{U}_{\langle n, t \rangle}$ la classe Σ_1^0 égale à $\bigcup_{s > t} \{X : \Phi(X) \succeq A_s \upharpoonright_n\}$. Il est clair que $\bigcap_{\langle n, t \rangle} \mathcal{U}_{\langle n, t \rangle}$ contient exactement l'ensemble des X tels que $\Phi(X) = A$. Par le théorème 18-3.2 de Sacks, cet ensemble est de mesure nulle. Donc, Z n'est pas fortement Martin-Löf aléatoire.

Comme (3) $\rightarrow$ (2) est trivial, il nous reste à montrer (1) $\rightarrow$ (3). Supposons Z MLR tel que $Z \in \bigcap_n \mathcal{U}_n$, où $\bigcap_n \mathcal{U}_n$ est une classe Π_2^0 de mesure nulle. On peut supposer $\mathcal{U}_{n+1} \subseteq \mathcal{U}_n$. Nous allons construire un ensemble c. e. simple A (voir la définition 16-4.2) tel que le « temps d'entrée » de Z dans $\mathcal{U}_n$ borne le temps nécessaire à l'énumération de n dans A . En même temps que l'on construit A , on construit pour tout e des classes $\mathcal{V}_e$ qui sont Σ_1^0 et telles que $\lambda(\mathcal{V}_e) < 2^{-e}$. Les classes $\mathcal{V}_e$ seront utilisées pour créer un test de Solovay qui nous aidera à conclure. On fixe une énumération $(W_e)_{e \in \mathbb{N}}$ des ensembles c. e. Au départ, chaque $\mathcal{V}_e$ est l'ensemble vide. À l'étape de calcul t , pour tout $e \leq t$, si jamais W_e n'intersecte pas encore A , alors on cherche $n \in W_e[t]$, avec $n > 2e$ tel que $\lambda(\mathcal{U}_n[t]) < 2^{-e}$. Si l'on trouve un tel n , on l'énumère dans A à l'étape t , et l'on fixe $\mathcal{V}_e = \mathcal{U}_n[t]$. Cela conclut la construction.

Notons que si W_e est infini, on finira forcément par trouver $n \in W_e[t]$ avec $n > 2e$ tel que $\lambda(\mathcal{U}_n[t]) < 2^{-e}$, car à partir d'un certain n on a de toute façon $\lambda(\mathcal{U}_n) < 2^{-e}$. Donc, par l'argument usuel (voir la proposition 16-4.4), A est bien un ensemble simple, et donc non calculable. Il reste à montrer que Z permet de calculer A . On utilise pour cela nos ensembles $\mathcal{V}_e$ qui forment un test de Solovay, car $\lambda(\mathcal{V}_e) \leq 2^{-e}$, et donc $\sum_e \lambda(\mathcal{V}_e)$ est fini. En particulier, si Z est MLR, il ne peut appartenir qu'à un nombre fini de $\mathcal{V}_e$. Il existe donc m tel que, pour tout $n \geq m$, si l'entier n est énuméré dans A à l'étape t , alors $Z \notin \mathcal{U}_n[t]$. Il suffit donc pour savoir si $n \geq m$ appartient à A de chercher le plus petit temps t tel que $Z \in \mathcal{U}_n[t]$, et d'énumérer A jusqu'à l'étape t . On a alors $n \in A$ ssi $n \in A[t]$. ■

Parmi les ensembles MLR qui sont capturés par des Π_2^0 de mesure nulle, il y a donc bien sûr tous les MLR qui calculent $\emptyset'$, mais ce ne sont pas les seuls (c'est en particulier une conséquence du corollaire 20-3.10 à venir). Nous verrons en revanche avec le corollaire 20-3.2 que si un MLR est incomplet — ne calcule pas $\emptyset'$ — et calcule un ensemble c. e. non calculable, alors cet ensemble est nécessairement K-trivial. Il s'agit en fait d'une caractérisation des ensembles c. e. et K-triviaux, comme nous le verrons avec le corollaire 20-3.10.

Voyons à présent un corollaire intéressant du théorème 1.5. Nous avons vu avec le corollaire 18-2.3 qu'il existait des ensembles MLR et calculatoirement dominés. Nous verrons avec le théorème 3.4 qu'être calculatoirement dominé est toutefois une propriété atypique : la classe des ensembles calculatoirement dominés est de mesure nulle. Le corollaire suivant indique que la notion d'être fortement Martin-Löf aléatoire n'est pas suffisante pour en rendre compte.

Corollaire 1.6

Soit Z un ensemble MLR, mais non fortement Martin-Löf aléatoire. Alors, Z n'est pas calculatoirement dominé.

PREUVE. Si Z est un ensemble MLR, mais non fortement Martin-Löf aléatoire, alors il calcule un ensemble c. e. non calculable, qui est donc par la proposition 7-4.7 non calculatoirement dominé. ■

On déduit du corollaire précédent que tous les MLR qui sont calculatoirement dominés sont aussi fortement MLR. Il nous faudra donc une notion plus forte pour capturer ces éléments. La notion d'être fortement Martin-Löf aléatoire est en revanche suffisante pour montrer qu'il est atypique d'être de degré PA, via une preuve due à Frank Stephan, et dont nous espérons que le lecteur appréciera la finesse. Stephan lui-même résume son résultat d'une manière très eastwoodienne : « *Il y a deux types d'ensembles Martin-Löf aléatoires dans ce monde : ceux qui sont calculatoirement suffisamment puissants pour résoudre le problème de l'arrêt, et ceux qui sont calculatoirement trop faibles pour être PA.* »

Théorème 1.7 (Stephan [218])

Soit Z Martin-Löf aléatoire. Alors, Z est de degré PA si, et seulement si, $Z \geq_T \emptyset'$.

PREUVE. Supposons que $\Phi(Z)$ soit un ensemble DNC_2 , dans le but de montrer $Z \geq_T \emptyset'$. Durant la construction, nous allons définir de manière effective une suite de codes de fonctions partielles calculables

$$a_0 < a_1 < a_2 < \dots$$

Aussi, d'après le théorème du point fixe, on peut supposer que chacun des a_k code pour une fonction qui a accès à cette suite, y compris au code a_k lui-même. Nous partitionnons la suite $(a_k)_{k \in \mathbb{N}}$ en une série d'intervalles consécutifs I_n de telle manière à ce que I_n contiennent 2^n éléments de la suite. Nous nommerons dorénavant a_n^k le k -ième élément de I_n . Au départ, chaque a_n^k est le code d'une fonction définie nulle part, qui attend certains évènements avant de peut-être s'arrêter sur certaines entrées.

Pour tout $n \in \mathbb{N}$, on définit $\mathcal{U}_{n,i}^0$ la classe Σ_1^0 des $\{X : \Phi(X, a_n^0) \downarrow = i\}$. On cherche ensuite $i \in \{0, 1\}$ et un temps de calcul t tels que $\lambda(\mathcal{U}_{n,i}^0[t]) > 2^{-n}$. Si l'on trouve pour n un tel élément $i \in \{0, 1\}$ et un tel temps de calcul t , alors on définit $\mathcal{U}_n^0 = \mathcal{U}_{n,i}^0[t]$, on décide que a_n^0 est le code de la fonction qui sur a_n^0 renvoie i , et l'on définit $\mathcal{U}_{n,i}^1 = \{X \notin \mathcal{U}_n^0 : \Phi(X, a_n^1) \downarrow = i\}$. On continue ensuite inductivement : si $\mathcal{U}_{n,i}^{k-1}$ est défini ainsi que $\mathcal{U}_{n,i}^k$ pour $i \in \{0, 1\}$, alors on cherche $i \in \{0, 1\}$ et un temps de calcul t tels que $\lambda(\mathcal{U}_{n,i}^k[t]) > 2^{-n}$. Si l'on trouve un tel élément $i \in \{0, 1\}$ et un tel temps de calcul t , on définit $\mathcal{U}_n^k = \mathcal{U}_n^{k-1} \cup \mathcal{U}_{n,i}^k[t]$, on décide que a_n^k est le code de la fonction qui sur a_n^k renvoie i , et l'on définit $\mathcal{U}_{n,i}^{k+1} = \{X \notin \mathcal{U}_n^k : \Phi(X, a_n^{k+1}) \downarrow = i\}$.

Dans la suite, nous appellerons les différents $\mathcal{U}_{n,i}^0, \mathcal{U}_{n,i}^1, \mathcal{U}_{n,i}^2, \dots$ des *versions* de $\mathcal{U}_{n,i}$. Les classes $\mathcal{U}_n^0, \mathcal{U}_n^1, \mathcal{U}_n^2, \dots$ seront quant à elles des *versions tronquées*. Lors de ce procédé, remarquons deux choses : d'abord, pour tout n , on arrivera à des versions finales et non tronquées $\mathcal{U}_{n,0}^k$ et $\mathcal{U}_{n,1}^k$ de $\mathcal{U}_{n,0}$ et $\mathcal{U}_{n,1}$, telles que $\lambda(\mathcal{U}_{n,0}^k \cup \mathcal{U}_{n,1}^k) < 2^{-n+1}$. En effet, par construction, chaque nouvelle version est disjointe de sa version tronquée précédente, qui a une mesure strictement plus grande que 2^{-n} . Comme il y a 2^n versions possibles, chacune des deux dernières doit forcément avoir une mesure inférieure à 2^{-n} .

Ensuite, aucun ensemble appartenant à une version tronquée $\mathcal{U}_n^k$ ne peut calculer un ensemble DNC₂ via Φ : on s'assure de cela en définissant le code a_n^k comme étant tel que $\Phi_{a_n^k}(a_n^k) \downarrow = i$, pour i tel que tout élément de $\mathcal{U}_n^k$ renvoie aussi i sur a_n^k . Comme $\Phi(Z)$ est un ensemble DNC₂, alors Z est forcément dans chaque dernière version non tronquée. En revanche, si pour les dernières versions $\mathcal{U}_{n,0}^k$ et $\mathcal{U}_{n,1}^k$, on a bien que $\mathcal{U}_{n,0}^k \cup \mathcal{U}_{n,1}^k$ est une classe Σ_1^0 de mesure inférieure à 2^{-n+1} , ces classes ne sont pas obtenues uniformément en n , car on ne sait potentiellement jamais si l'on est arrivé ou non à la dernière version. C'est là que $\emptyset'$ entre en jeu.

On va à présent définir un test de Solovay via les ensembles $\mathcal{V}_n$ suivants : au départ, chaque $\mathcal{V}_n$ est l'ensemble vide. Puis, si n est énuméré dans $\emptyset'$ au temps t , on déclare alors que $\mathcal{V}_n$ est la réunion $\mathcal{U}_{n,0}^k \cup \mathcal{U}_{n,1}^k$ des versions disponibles au temps t , éventuellement tronquées afin que la mesure ne dépasse pas 2^{-n+1} . Il est alors clair que $(\mathcal{V}_n)_{n \in \mathbb{N}}$ forment un test de Solovay.

Supposons à présent que le temps d'entrée de Z dans

$$\mathcal{U}_n = \{X : \bigwedge_{a \in I_n} \Phi(X, a) \downarrow\}$$

(il s'agit du plus petit t tel que $Z \in \mathcal{U}_n[t]$) soit infiniment souvent inférieur au temps d'énumération de n dans $\emptyset'$. Souvenons-nous que Z est forcément dans chaque dernière version, et que ces dernières sont non tronquées. On en déduit que, pour chacun de ces entiers n , l'ensemble Z appartient à $\mathcal{V}_n$, ce qui implique que Z est capturé par le test de Solovay, contredisant le fait que Z est aléatoire au sens de Martin-Löf. On en déduit que le temps d'entrée de Z dans $\mathcal{U}_n$ est pour presque tout entier n supérieur au temps d'énumération de n dans $\emptyset'$. Donc, Z peut savoir si un entier n suffisamment grand appartient à $\emptyset'$ en regardant simplement si $n \in \emptyset'[t]$, où t est le plus petit tel que $Z \in \mathcal{U}_n[t]$. ■

Corollaire 1.8

Si Z est fortement Martin-Löf aléatoire, alors Z n'est pas de degré PA. En particulier, la classe des ensembles de degré PA est de mesure nulle.

Notons qu'un simple argument de « vote à la majorité », à la manière du théorème 18-3.2, permet de prouver que la classe des ensembles de degré PA est de mesure nulle.

2. Relativisation de l'aléatoire

Comme pour la plupart des notions en calculabilité, il est possible de relativiser l'aléatoire de Martin-Löf à un oracle.

Définition 2.1

Soit $X \in 2^{\mathbb{N}}$. Un X -test de Martin-Löf est donné par $\bigcap_n \mathcal{U}_n$, un ensemble $\Pi_2^0(X)$, tel que $\lambda(\mathcal{U}_n) \leq 2^{-n}$. Un ensemble Z est X -aléatoire au sens de Martin-Löf ou $\text{MLR}(X)$ s'il passe tous les X -tests de Martin-Löf. ◆

D'après la définition 17-3.3 et ce qui précède l'exemple 17-3.8, le code d'une classe $\Pi_2^0(X)$ est un entier $\langle 2, 1, e \rangle$ pour lequel W_e^X énumère des entiers $\langle 1, 0, a_n \rangle$ tels que $[W_{a_n}^X]$ est la n -ième composante de notre $\Pi_2^0(X)$. Notre code e peut toutefois être utilisé avec n'importe quel autre oracle Y : l'ensemble W_e^Y énumère lui aussi des entiers. On voit sans peine comment uniformément transformer un code e de manière à ne conserver dans les

énumérations W_e^Y (pour tout oracle Y) que les entiers de la forme $\langle 1, 0, a \rangle$. L'objectif est que, pour tout oracle Y , le même e code pour une classe $\Pi_2^0(Y)$.

Classe et machine à oracle

On parlera de *classe à oracle* pour insister sur le fait que le même code e définisse uniformément une classe $\Sigma_n^0(X)$ pour tout oracle $X \in 2^{\mathbb{N}}$. On parlera de la même manière de *machine à oracle* pour illustrer le fait qu'une machine $M : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ utilisant un oracle X est aussi une machine — c'est-à-dire une fonction partielle de $2^{<\mathbb{N}}$ dans $2^{<\mathbb{N}}$ — avec n'importe quel oracle Y .

Les différents théorèmes que nous avons vus jusqu'ici se relativisent sans problème, en particulier le théorème 18-2.1 de Levin/Schnorr, pour lequel nous relativisons aussi la notion de complexité sans préfixe.

Définition 2.2. Étant donné une machine à oracle M telle que M est sans préfixe sur son oracle X , on note $K_M^X(\sigma)$ la taille de la plus petite chaîne τ telle que $M(X, \tau) \downarrow = \sigma$. $\diamond$

Notons qu'étant donné une machine à oracle M , il est possible que M soit sans préfixe sur certains de ses oracles, mais pas sans préfixe sur d'autres. De la même manière, étant donné $\bigcap_n \mathcal{U}_n$ une classe à oracle Π_2^0 , il est possible que cette classe soit un test de Martin-Löf sur certains de ses oracles, mais pas sur tous. Pour étudier ce phénomène, nous introduisons la notation suivante.

Notation

Pour une machine à oracle M , on écrira $M^X : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ pour la machine utilisée avec l'oracle X . Étant donné $\bigcap_n \mathcal{U}_n$ une classe à oracle Π_2^0 , on écrira $\bigcap_n \mathcal{U}_n^X$ pour la classe $\Pi_2^0(X)$ correspondante.

Étant donné un oracle X , l'existence d'une machine à oracle sans préfixe et universelle sur l'oracle X ne présente pas de difficulté. Il est toutefois possible d'aller plus loin, et de montrer l'existence d'une machine à oracle qui soit à la fois sans préfixe et universelle pour tous les oracles.

Théorème 2.3

Il existe une machine à oracle U telle que pour tout oracle $X \in 2^{\mathbb{N}}$:

- (1) la machine U^X est sans préfixe ;
- (2) pour toute machine à oracle M , telle que M^X est sans préfixe, on a $K_U^X(\sigma) \leq^+ K_M^X(\sigma)$ pour toute chaîne σ .

PREUVE. Il suffit de noter qu'une relativisation de la preuve du théorème 16-2.2 donne une procédure uniforme, et définit donc une fonctionnelle Turing. ■

Le corollaire suivant découle de la relativisation du théorème 18-2.1 de Levin/Schnorr, qui stipule qu'une chaîne est MLR si, et seulement si, chacun de ses préfixes est incompressible.

Corollaire 2.4

Il existe une classe Π_2^0 à oracle qui est un X -test de Martin-Löf universel uniformément en chaque oracle X .

PREUVE. Il suffit de considérer la classe $\Pi_2^0(X)$ suivante :

$$\{Y : \forall c \exists n K^X(Y \upharpoonright_n) \leq n - c\}.$$

D'après le théorème 18-2.1, cette classe est un X -test de Martin-Löf pour tout oracle X . D'après le théorème précédent, elle est uniformément $\Pi_2^0(X)$ pour tout oracle X . ■

Cette universalité uniforme en chaque oracle nous sera utile de temps à autre, nous en voyons une première utilisation avec l'élégant théorème de van Lambalgen.

Théorème 2.5 (van Lambalgen [227])

Soit X_0, X_1 deux ensembles. Alors, $X_0 \oplus X_1$ est MLR si, et seulement si, X_0 est MLR et X_1 est MLR(X_0).

PREUVE. Notons que $\lambda([\sigma_1 \oplus \sigma_2]) = \lambda([\sigma_1]) \times \lambda([\sigma_2])$, pour $\sigma_1, \sigma_2 \in 2^{<\mathbb{N}}$ de même taille. Supposons X non MLR ou Y non MLR(X) afin de montrer $X \oplus Y$ non MLR. Par symétrie, on peut supposer Y non MLR(X). Vu le corollaire 2.4, soit $\bigcap_n \mathcal{U}_n$ un test de Martin-Löf universel pour tous les oracles. Étant donné une chaîne σ , on notera $\mathcal{U}_n^\sigma$ le morceau d'ouvert énuméré avec l'oracle σ .

Comme Y n'est pas MLR(X), alors $Y \in \bigcap_n \mathcal{U}_n^X$. Pour tous n, m , on définit alors la classe Σ_1^0 suivante :

$$\mathcal{U}_{n,m} = \bigcup_{|\tau|=m} \{[\tau] \oplus [\sigma] : [\sigma] \subseteq \mathcal{U}_n^\tau \text{ et } |\sigma| = |\tau|\}.$$

Pour tous entiers m, n on a

$$\lambda(\mathcal{U}_{n,m}) \leq \sum_{|\tau|=m} \lambda([\tau]) \times \lambda(\mathcal{U}_n^\tau) \leq \lambda(\mathcal{U}_n) \sum_{|\tau|=m} \lambda([\tau]) \leq \lambda(\mathcal{U}_n) \leq 2^{-n}.$$

De plus, on a l'inclusion $\mathcal{U}_{n,m} \subseteq \mathcal{U}_{n,m+1}$ pour tous entiers m, n , ce qui implique $\lambda(\bigcup_m \mathcal{U}_{n,m}) \leq 2^{-n}$ pour tout n . Par ailleurs, $X \oplus Y \in \bigcup_m \mathcal{U}_{n,m}$. Dès lors, $\bigcap_n \bigcup_m \mathcal{U}_{n,m}$ est un test de Martin-Löf qui capture $X \oplus Y$.

Supposons à présent $X \oplus Y$ non MLR et capturé par un test de Martin-Löf $\bigcap_n \mathcal{U}_n$. Soit $\mathcal{U}_n^\sigma$ la classe Σ_1^0 donnée par $\{Y : [\sigma \oplus Y \upharpoonright_{|\sigma|}] \subseteq \mathcal{U}_n\}$. Soit $\mathcal{V}_n^X$ la classe $\Sigma_1^0(X)$ donnée par $\bigcup_{\sigma \prec_X} \mathcal{U}_{2n}^\sigma$. Supposons d'abord $\lambda(\mathcal{V}_n^X) \leq 2^{-n}$ pour tout n suffisamment grand. Alors, $\bigcap_n \mathcal{V}_n^X$ est un X -test de Martin-Löf qui capture Y . Sinon, il y a une infinité de n tels que $\lambda(\mathcal{V}_n^X) > 2^{-n}$. Pour tout n , on définit $\mathcal{S}_n$ comme étant la classe Σ_1^0 égale à $\{Z : \lambda(\mathcal{V}_n^Z) > 2^{-n}\}$. Montrons alors que, pour tout n , on a $\lambda(\mathcal{S}_n) \leq 2^{-n}$. Soit W_n un ensemble sans préfixe minimal qui décrit $\mathcal{S}_n$. On a par définition :

$$\sum_{\tau \in W_n} \lambda(\tau) \times \lambda(\mathcal{U}_{2n}^\tau) \leq \lambda(\mathcal{U}_{2n}) \sum_{\tau \in W_n} \lambda(\tau) \leq \lambda(\mathcal{U}_{2n}) \leq 2^{-2n};$$

or, pour tout $\tau \in W_n$, on a $\lambda(\mathcal{U}_{2n}^\tau) > 2^{-n}$, ce qui nous donne :

$$\sum_{\tau \in W_n} \lambda(\tau) \times 2^{-n} \leq \lambda(\mathcal{U}_{2n}) \leq 2^{-2n}.$$

On en déduit $\sum_{\tau \in W_n} \lambda(\tau) \leq 2^{-n}$, et donc $\lambda(\mathcal{S}_n) \leq 2^{-n}$ pour tout n . Il s'ensuit que $(\mathcal{S}_n)_{n \in \mathbb{N}}$ est un test de Solovay, capturant tous les ensembles qui sont dans une infinité de $\mathcal{S}_n$. Ainsi, X n'est pas MLR. ■

Corollaire 2.6

Soient X, Y deux ensembles MLR. Alors, X est MLR(Y) si, et seulement si, Y est MLR(X).

Le corollaire précédent implique en particulier que si X est aléatoire, alors il est atypique de rendre X non aléatoire.

Corollaire 2.7

Soit X un ensemble MLR. Alors, $\{Y \in 2^{\mathbb{N}} : X \text{ est non MLR}(Y)\}$ est une classe de mesure 0.

PREUVE. Si l'ensemble X est non MLR(Y), alors soit Y est non MLR, soit Y est MLR, et dans tous les cas Y est non MLR(X). En particulier, la classe $\{Y \in 2^{\mathbb{N}} : X \text{ est non MLR}(Y)\}$ est incluse dans la réunion de deux classes de mesure 0. ■

La relativisation de l'aléatoire de Martin-Löf nous permet évidemment d'obtenir des notions plus fortes d'aléatoire. N'importe quel oracle X avec suffisamment de puissance de calcul pourra « dé-aléatoriser » quelque chose. Mais de quelle puissance a-t-on besoin exactement pour rendre non aléatoire quelque chose qui l'était ? Existe-t-il au fond un ensemble X non calculable et malgré tout trop faible pour capturer des ensembles MLR dans un X -test ? Ces questions nous amènent à la notion suivante.

Définition 2.8. Un ensemble X est *low* pour l'aléatoire de Martin-Löf si tout ensemble MLR est $\text{MLR}(X)$. $\diamond$

Nous verrons que la classe des ensembles low pour l'aléatoire de Martin-Löf est un peu plus grande que celle des calculables. Il s'agit d'un des résultats les plus beaux et les plus difficiles de l'aléatoire algorithmique : cette classe coïncide avec celle des ensembles K-triviaux.

3. Les 2-aléatoires

Martin-Löf s'autorise des tests Π_2^0 . Pourquoi au fond cette restriction ? On pourrait tout à fait considérer des classes de mesure 0 de complexité arbitraire afin de capturer plus d'ensembles. Nous voyons ici le premier niveau de cette hiérarchie.

Définition 3.1. Un ensemble Z est *2-aléatoire* s'il n'appartient à aucun ensemble Π_3^0 de la forme $\bigcap_n \mathcal{B}_n$ tel que $\lambda(\mathcal{B}_n) \leq 2^{-n}$ pour tout n . $\diamond$

Il existe une caractérisation équivalente, qui est souvent celle utilisée.

Théorème 3.2

Les ensembles 2-aléatoires sont exactement les ensembles $\text{MLR}(\emptyset')$.

PREUVE. Considérons d'abord $\bigcap_n \mathcal{U}_n$, une classe $\Pi_2^0(\emptyset')$. Chaque $\mathcal{U}_n$ est donc une classe $\Sigma_1^0(\emptyset')$ décrite par un ensemble $\Sigma_1^0(\emptyset') W_n \subseteq 2^{<\mathbb{N}}$. D'après le corollaire 5-5.4 et la proposition 5-3.3, l'ensemble W_n est Σ_2^0 , c'est-à-dire $W_n = \{\sigma \in 2^{<\mathbb{N}} : \exists x_1 \forall x_2 R(\sigma, x_1, x_2)\}$ pour un prédicat calculable R . On définit $\mathcal{F}_{x_1, \sigma}$ comme étant la classe Π_1^0 égale à $[\sigma]$ si $\forall x_2 R(\sigma, x_1, x_2)$, et égale à l'ensemble vide sinon. On a $\mathcal{U}_n = \bigcup_{x_1, \sigma} \mathcal{F}_{x_1, \sigma}$, qui est donc un ensemble Σ_2^0 . Donc, $\bigcap_n \mathcal{U}_n$ est une classe Π_3^0 .

Considérons à présent une classe Π_3^0 de la forme $\bigcap_n \mathcal{B}_n$, où chaque $\mathcal{B}_n$ est une classe Σ_2^0 pour laquelle $\lambda(\mathcal{B}_n) < 2^{-n}$. D'après le théorème 17-4.4, pour chaque classe $\mathcal{B}_n$, on peut trouver uniformément une classe $\Pi_2^0(\emptyset')$ de la forme $\bigcap_m \mathcal{U}_m^n$ telle que $\mathcal{B}_n \subseteq \bigcap_m \mathcal{U}_m^n$ et pour laquelle $\lambda(\mathcal{U}_m^n \setminus \mathcal{B}_n) \leq 2^{-m}$ pour tout m . On a donc $\lambda(\mathcal{U}_{n+1}^{n+1}) \leq 2^{-(n+1)} + 2^{-(n+1)} = 2^{-n}$. Il s'ensuit que $\bigcap_n \mathcal{U}_{n+1}^{n+1}$ est un $\emptyset'$ -test de Martin-Löf contenant $\bigcap_n \mathcal{B}_n$. ■

Exercice 3.3. (*) Un ensemble est *n-aléatoire* s'il n'appartient à aucune classe Π_{n+1}^0 de la forme $\bigcap_n \mathcal{B}_n$, avec $\lambda(\mathcal{B}_n) \leq 2^{-n}$.

Montrer que l'on peut itérer le théorème 3.2 : un ensemble est $\text{MLR}(\emptyset^n)$ si, et seulement si, il est *n-aléatoire*. $\diamond$

Cette notion d'aleatoire plus forte nous permet de montrer que la classe des ensembles calculatoirement dominés est atypique.

Théorème 3.4

Aucun 2-aleatoire n'est calculatoirement dominé. En particulier les réels calculatoirement dominés forment une classe de mesure 0.

On utilisera pour montrer le théorème 3.4 le lemme suivant.

Lemme 3.5 (Monin [158])

Toute classe Σ_2^0 qui intersecte n'importe quelle classe Π_1^0 non vide contient tous les ensembles calculatoirement dominés. ★

PREUVE. Soit $\bigcup_n \mathcal{F}_n$ une classe Σ_2^0 qui intersecte toutes les classes Π_1^0 non vides. Supposons par l'absurde qu'il existe X calculatoirement dominé avec $X \notin \bigcup_n \mathcal{F}_n$. Soit $\bigcap_n \mathcal{U}_n$ le complémentaire de $\bigcup_n \mathcal{F}_n$. On définit la fonction X -calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ qui sur n renvoie le plus petit t tel que $X \in \mathcal{U}_n[t]$. Soit $g > f$ une fonction calculable. Alors, X appartient à la classe Π_1^0 donnée par $\bigcap_n \mathcal{U}_n[g(t)]$ et qui est disjointe de $\bigcup_n \mathcal{F}_n$, ce qui est une contradiction. ■

Passons à la preuve du théorème 3.4.

PREUVE DU THÉORÈME 3.4. Nous allons construire pour tout n une classe Σ_2^0 qui intersecte n'importe quelle classe Π_1^0 non vide, et de mesure inférieure à 2^{-n} . Nous allons décrire un ensemble calculatoirement énumérable W de codes pour les classes Π_1^0 qui constituent notre Σ_2^0 . Soit $\mathcal{F}_0, \mathcal{F}_1, \dots$ une énumération de toutes les classes Π_1^0 .

Pour tout e , soit σ_e la plus petite chaîne dans l'ordre lexicographique de taille $n + e + 1$. À l'étape t , pour tout $e \leq t$, si $\mathcal{F}_e[t] \cap [\sigma_e] = \emptyset$, on change σ_e qui devient la prochaine chaîne de taille $n + e + 1$ dans l'ordre lexicographique (sauf si c'est la dernière possible, auquel cas on ne fait rien). Puis, on énumère un code de $\mathcal{F}_e \cap [\sigma_e]$ dans W .

Il est clair que si $\mathcal{F}_e$ est non vide, alors on aura un code de $\mathcal{F}_e \cap [\sigma_e]$ énuméré dans W pour la première chaîne σ_e de taille $n + e + 1$ telle que $\mathcal{F}_e \cap [\sigma_e] \neq \emptyset$. Donc, W intersecte toutes les classes Π_1^0 non vides. Par ailleurs, pour chaque e , on ajoute à la classe Σ_2^0 quelque chose de mesure bornée par 2^{-n-e-1} . La mesure totale de la classe Σ_2^0 est dès lors bornée par $\sum_e 2^{-n-e-1} = 2^{-n}$.

On construit donc de la sorte un test Π_3^0 de la forme $\bigcap_n \bigcup_e \mathcal{F}_{e,n}$ tel que

$$\lambda\left(\bigcup_e \mathcal{F}_{e,n}\right) < 2^{-n}.$$

Le test ne contient donc aucun 2-aleatoire. En revanche, il contient par le lemme 3.5 tous les ensembles calculatoirement dominés. ■

D'après une relativisation du théorème 18-4.1, tout ensemble 2-aléatoire calcule une fonction DNC($\emptyset'$). Il est alors possible d'améliorer le théorème précédent.

Exercice 3.6. (★★) Montrer qu'aucun ensemble de degré DNC($\emptyset'$) n'est calculatoirement dominé (on pourra utiliser une technique similaire à celle présentée dans la preuve ci-dessus). $\diamond$

Jockusch et Stephan [103] ont montré quelque chose de plus fort que dans l'exercice précédent : aucun ensemble X tel que X' est de degré DNC($\emptyset'$) n'est calculatoirement dominé. Voyons à présent une preuve alternative de l'existence d'un ensemble high ne calculant pas $\emptyset'$. Il suffit de considérer $\Omega^{\emptyset'}$, le nombre Ω de Chaitin relativisé à $\emptyset'$.

Proposition 3.7. L'ensemble $\Omega^{\emptyset'}$ est high, mais ne calcule pas $\emptyset'$. $\star$

PREUVE. En utilisant le même algorithme que celui de la preuve du théorème 16-2.13, avec les n premiers bits de $\Omega^{\emptyset'}$, et l'aide de $\emptyset'$, on peut savoir quelles sont les chaînes σ de taille inférieure à n telles que $U(\emptyset', \sigma) \downarrow$. On a dès lors $\Omega^{\emptyset'} \oplus \emptyset' \geq_T \emptyset''$. En revanche, d'après le théorème 1.5, le réel $\Omega^{\emptyset'}$ étant 2-aléatoire, il ne calcule pas $\emptyset'$. $\blacksquare$

La classe des 2-aléatoires permet également de mettre en évidence une propriété d'analyse bien connue : soit f une fonction limite de fonctions continues, alors il existe des fermés de mesure arbitrairement grande sur lesquels la restriction de f est continue. Il s'agit d'un résultat qui fait écho à son analogue catégorique : toute limite de fonctions continues est continue sur un ensemble co-maigre. Nous l'avons montré avec le théorème 10-3.20 qui dit que tout ensemble 1-générique est low généralisé. Nous montrons à présent un théorème équivalent, mais pour l'aléatoire.

Théorème 3.8 (Kautz [108])

Les ensembles 2-aléatoires sont low généralisés.

PREUVE. Étant donné un code e , on veut savoir si Z appartient à la classe

$$\mathcal{U}_e = \{X : \Phi_e(X, e) \downarrow\}.$$

Notons que $\mathcal{U}_e$ est une classe Σ_1^0 . En utilisant $\emptyset'$, on calcule le $\emptyset'$ -test de Martin-Löf suivant : pour tout e , on cherche le plus petit t_e tel que

$$\lambda(\mathcal{U}_e \setminus \mathcal{U}_e[t_e]) < 2^{-e}.$$

On pose ensuite $\mathcal{V}_e = \mathcal{U}_e \setminus \mathcal{U}_e[t_e]$. La classe $\bigcap_d \bigcup_{e>d} \mathcal{V}_e$ est un $\emptyset'$ -test de Martin-Löf. Il existe donc d tel que, pour tout $e > d$, l'ensemble Z n'appartient pas à $\mathcal{V}_e$.

Pour tout entier $e > d$, pour savoir si $Z \in \mathcal{U}_e$, il suffit alors de regarder si $Z \in \mathcal{U}_e[t_e]$. Si c'est le cas, alors $\Phi_e(X, e) \downarrow$; et, si ce n'est pas le cas, alors comme également $Z \notin \mathcal{U}_e \setminus \mathcal{U}_e[t_e]$, on a $Z \notin \mathcal{U}_e$, et donc $\Phi_e(Z, e) \uparrow$. ■

Notons que dans le cas de la théorie des catégories, la même fonctionnelle est utilisée pour calculer G' à partir de $\emptyset' \oplus G$ pour tout ensemble 1-générique G . Pour le théorème ci-dessus, il n'y a pas une unique fonctionnelle qui donne le bon résultat pour n'importe quel ensemble 2-aléatoire Z : cela dépend de la plus petite composante $\mathcal{W}_e$ d'un $\emptyset'$ -test de Martin-Löf auquel Z n'appartient pas. Il faut de plus écrire « en dur » les e premiers bits de Z' . Cette étape peut toutefois être uniformisée grâce à la redondance d'information de l'arrêt : pour connaître $Z'(n)$ pour $n < e$, il suffit d'utiliser le lemme 3-5.1 de remplissage pour trouver un code $m > e$ équivalent à n . Reste que la procédure dépend toujours de e : plus celui-ci est grand, plus proche de 1 sera la classe des 2-aléatoires sur lesquels la fonctionnelle donnera le bon résultat. Ces idées ont été précisément formalisées par Hoyrup et Rojas [94] via les notions de *déficit d'aléatoire* et de *fonctions calculables par couches*.

Corollaire 3.9

La classe des ensembles high est de mesure 0.

PREUVE. D'après le théorème 3.8, si Z est 2-aléatoire, alors $Z' \geq_T \emptyset''$ si, et seulement si, $Z \oplus \emptyset' \geq_T \emptyset''$. Aussi, par une relativisation de la preuve du théorème 18-3.2 de Sacks, on a $\lambda(\{X : X \oplus \emptyset' \geq_T \emptyset''\}) = 0$. La classe des ensembles high est donc de mesure 0. ■

4. Aléatoires incomplets

Les aléatoires incomplets — qui ne calculent pas $\emptyset'$ — forment une notion d'aléatoire intéressante, juste un peu plus forte que MLR et bien plus faible que fortement MLR. Leur étude a réellement été initiée par Franklin et Ng qui ont découvert la notion de *test de différence*, permettant de les capturer.

Définition 4.1 (Franklin et Ng [61]). Un *test de différence* est donné par $\bigcap_n \mathcal{U}_n$ une classe Π_2^0 et $\mathcal{F}$ une classe Π_1^0 telles que $\lambda(\mathcal{U}_n \cap \mathcal{F}) \leq 2^{-n}$ pour tout n . Un ensemble Z est *capturé* par le test si $Z \in \bigcap_n \mathcal{U}_n \cap \mathcal{F}$. Sinon, Z *passé* le test. Un ensemble Z est un *aléatoire de différence* si Z passe tous les tests de différence. ◇

La notion de test différence correspond en quelque sorte après avoir énuméré une chaîne σ dans un ouvert le composant, à se réserver le droit de changer d'avis, et finalement d'enlever cette chaîne de l'énumération. On ne peut en revanche pas changer d'avis une seconde fois et décider finalement de remettre la chaîne σ .

Théorème 4.2 (Franklin et Ng [61])

Soit Z un ensemble MLR. Les énoncés suivants sont équivalents.

- (1) Z calcule $\emptyset'$.
- (2) Z n'est pas un aléatoire de différence.

Afin de montrer le théorème 4.2 nous utiliserons un lemme qui a son intérêt propre : si X est un ensemble MLR, alors « peu » d'ensembles permettent de calculer des préfixes de X .

Lemme 4.3. Soit Z un ensemble MLR et soit Φ une fonctionnelle. Alors, il existe une constante $c \in \mathbb{N}$ telle que $\lambda(\{X : \Phi(X) \succeq Z \upharpoonright_n\}) \leq 2^{-n} \times 2^c$ pour tout $n \in \mathbb{N}$. ★

PREUVE. Supposons que, pour toute constante $c \in \mathbb{N}$, il existe $n \in \mathbb{N}$ tel que

$$\lambda(\{X : \Phi(X) \succeq Z \upharpoonright_n\}) > 2^{-n} \times 2^c.$$

Montrons que Z n'est pas aléatoire au sens de Martin-Löf. Soit $\mathcal{U}_c$ la classe Σ_1^0 générée par les chaînes σ telles que $\lambda(\{X : \Phi(X) \succeq \sigma\}) > 2^{-|\sigma|} \times 2^c$. Soit W un ensemble minimal de chaînes sans préfixe tel que $\mathcal{U}_c = [W]$. En particulier, tout $\sigma \in W$ satisfait l'inégalité ci-dessus. Alors,

$$\lambda(\mathcal{U}_c) = \sum_{\sigma \in W} 2^{-|\sigma|} \leq 2^{-c} \sum_{\sigma \in W} \lambda(\{X : \Phi(X) \succeq \sigma\}).$$

Comme la classe W est sans préfixe, alors pour deux chaînes distinctes σ_1, σ_2 de W on a $\{X : \Phi(X) \succeq \sigma_1\} \cap \{X : \Phi(X) \succeq \sigma_2\} = \emptyset$. Par additivité de la mesure, on a donc $\sum_{\sigma \in W} \lambda(\{X : \Phi(X) \succeq \sigma\}) \leq 1$, et donc $\lambda(\mathcal{U}_c) \leq 2^{-c}$. On peut à présent capturer Z par le test de Martin-Löf donné par $\bigcap_c \mathcal{U}_c$. ■

On peut à présent montrer la caractérisation de Franklin et Ng.

PREUVE DU THÉORÈME 4.2. Supposons que Z calcule $\emptyset'$. Alors, Z calcule également l'ensemble Ω de Chaitin via une fonctionnelle Φ . Soit c la constante du lemme 4.3 telle que $\lambda(\{X : \Phi(X) \succeq \Omega \upharpoonright_n\}) < 2^{-n} \times 2^c$ pour tout n . Soit $\mathcal{C}_{n,s}$ la classe Σ_1^0 donnée par $\{X : \Phi(X) \succeq \Omega_s \upharpoonright_n\}$, où Ω_s est l'approximation de Ω à l'étape s . Soit à présent $\mathcal{C}'_{n,s}$ la classe $\mathcal{C}_{n,s}$ pour laquelle on bloque l'énumération si nécessaire afin que la mesure ne dépasse pas $2^{-n} \times 2^c$.

Soit $\mathcal{U}$ la réunion des classes $\mathcal{C}'_{n,s}$ pour tous n et s tels que $\Omega_s \upharpoonright_n \neq \Omega_{s+1} \upharpoonright_n$. Notons qu'aucun élément de $\mathcal{U}$ ne calcule Ω via Φ , car si $\Omega_s \upharpoonright_n \neq \Omega_{s+1} \upharpoonright_n$, alors $\Omega_s \upharpoonright_n \neq \Omega \upharpoonright_n$. En particulier, Z ne peut pas être dans $\mathcal{U}$. Enfin, pour tout n , soit $\mathcal{U}_n$ la réunion des classes $\mathcal{C}'_{n,s}$ pour tout s , et soit $\mathcal{F}$ le complémentaire de $\mathcal{U}$.

Soit s_n tel que $\Omega_{s_n} \upharpoonright_n = \Omega \upharpoonright_n$. Alors, $Z \in \mathcal{C}_{n,s_n}$, et d'après le choix de la constante c , l'ensemble Z appartient aussi à $\mathcal{C}'_{n,s_n}$. Il est donc clair que Z est dans $\bigcap_n \mathcal{U}_n$, et comme Z n'est pas dans $\mathcal{U}$, il est clair que $Z \in \mathcal{F} \cap \bigcap_n \mathcal{U}_n$. Par ailleurs, les éléments de $\mathcal{F} \cap \mathcal{U}_n$ sont uniquement ceux qui appartiennent à $\mathcal{C}'_{n,s_n}$, et l'on a par définition $\lambda(\mathcal{C}'_{n,s_n}) \leq 2^{-n} \times 2^c$. Donc, $\mathcal{F} \cap \bigcap_n \mathcal{U}_n$ est un test de différence qui contient Z .

Supposons à présent que Z est capturé par un test de différence $\mathcal{F} \cap \bigcap_n \mathcal{U}_n$. Pour tout n , soit $\mathcal{V}_n$ la classe Σ_1^0 suivante : si n est énuméré dans $\emptyset'$ à l'étape t , alors $\mathcal{V}_n$ est défini comme étant $\mathcal{U}_n[t] \cap \mathcal{F}[s_t]$ où s_t est le plus petit entier tel que $\lambda(\mathcal{U}_n[t] \cap \mathcal{F}[s_t]) \leq 2^{-n}$. Si n n'est jamais énuméré, alors $\mathcal{V}_n$ reste vide. La suite $(\mathcal{V}_n)_{n \in \mathbb{N}}$ forme un test de Solovay. En particulier, il existe n tel que pour tout $m > n$, $Z \notin \mathcal{V}_m$. On peut alors calculer $\emptyset'$ à l'aide de Z de la manière suivante : pour tout $m > n$, il suffit de chercher le plus petit t tel que $Z \in \mathcal{U}_m[t]$ et de voir si $m \in \emptyset'[t]$. Si ce n'est pas le cas, alors $m \notin \emptyset'$, car on aurait alors dans le cas inverse $Z \in \mathcal{V}_m$. ■

MLR approchable par la gauche

La preuve du théorème 4.2 fonctionne en remplaçant Ω par n'importe quel MRL approchable par la gauche. En particulier, tout MLR approchable par la gauche est capturé par un test de différence, donc est complet, comme nous en avons discuté dans la section 16-2.4.

Nous voyons à présent une application de la notion de test de différence et d'aléatoire incomplet. Avec l'apparition de la notion de mesure, est apparue en analyse la notion de propriété satisfaite « presque partout ». Ainsi, une propriété sur les réels est vraie presque partout si l'ensemble des réels pour laquelle elle est fausse a pour mesure 0. Un théorème classique nous dit par exemple qu'une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ de graphe borélien est continue presque partout.

Un des champs d'études de l'aléatoire algorithmique consiste à déterminer, pour les théorèmes qui sont vrais presque partout, le niveau exact d'aléatoire pour lequel un tel théorème est vrai. Nous en voyons ici un exemple, avec le théorème de densité de Lebesgue et les aléatoires de différence. Cette étude nous sera par ailleurs utile dans le chapitre à venir sur les ensembles K-triviaux.

Voyons pour commencer le théorème de densité de Lebesgue. Nous n'avons pour le moment exposé que le lemme de densité de Lebesgue, que nous rappelons ci-après.

Lemme (18-3.1 de densité de Lebesgue). Soit $\mathcal{B}$ un ensemble Borélien de mesure positive. Alors, pour tout $\varepsilon > 0$, il existe un cylindre $[\sigma]$ tel

que $\lambda(\mathcal{B} \mid [\sigma]) > 1 - \varepsilon$. ★

Le théorème de densité de Lebesgue est un renforcement de ce lemme-là. Nous introduisons pour cela le concept de valeur asymptotique de la densité d'un borélien le long des préfixes d'un ensemble.

Définition 4.5. Soit $\mathcal{B} \subseteq 2^{\mathbb{N}}$ un borélien et soit $Z \in 2^{\mathbb{N}}$.

- (1) On note $\underline{\rho}(\mathcal{B} \mid Z)$ la limite inférieure de la densité qu'occupe $\mathcal{B}$ à l'intérieur des préfixes de Z , c'est-à-dire

$$\underline{\rho}(\mathcal{B} \mid Z) = \liminf_{\sigma \prec Z} \lambda(\mathcal{B} \mid [\sigma]).$$

- (2) On note de manière similaire $\bar{\rho}(\mathcal{B} \mid Z)$ la limite supérieure de la densité qu'occupe $\mathcal{B}$ à l'intérieur des préfixes de Z , c'est-à-dire

$$\bar{\rho}(\mathcal{B} \mid Z) = \limsup_{\sigma \prec Z} \lambda(\mathcal{B} \mid [\sigma]).$$

On dira que $\underline{\rho}(\mathcal{B} \mid Z)$ est la *densité inférieure* de Z dans $\mathcal{B}$, et $\bar{\rho}(\mathcal{B} \mid Z)$ sa *densité supérieure*. ◇

On peut à présent énoncer le théorème annoncé de densité de Lebesgue.

Théorème 4.6 (Théorème de densité de Lebesgue)

Soit $\mathcal{B} \subseteq 2^{\mathbb{N}}$ une classe borélienne de mesure positive.

Alors, la classe $\{Z \in \mathcal{B} : \underline{\rho}(\mathcal{B} \mid Z) = 1\}$ a la même mesure que $\mathcal{B}$.

Nous utiliserons pour le montrer le lemme suivant.

Lemme 4.7. Soit $\mathcal{F}$ une classe fermée, et soit $\sigma \in 2^{<\mathbb{N}}$ une chaîne telle que

$$\lambda(\mathcal{F} \mid [\sigma]) \geq \varepsilon.$$

Alors, il existe $X \in \mathcal{F} \cap [\sigma]$ tel que $\lambda(\mathcal{F} \mid [X \upharpoonright_n]) \geq \varepsilon$ pour tout $n > |\sigma|$. ★

PREUVE. Il suffit de voir que pour toute chaîne τ et tout borélien $\mathcal{A}$, si $\lambda(\mathcal{A} \mid [\tau]) \geq \varepsilon$, alors pour $i = 0$ ou $i = 1$ on aura $\lambda(\mathcal{A} \mid [\tau i]) \geq \varepsilon$. En effet, si $\mathcal{A}$ occupe pour moins de ε de mesure dans $[\tau 0]$ et dans $[\tau 1]$, il occupera aussi moins de ε de mesure dans $[\tau]$. On construit alors de cette manière un ensemble $X \succ \sigma$ tel que $\lambda(\mathcal{F} \mid [X \upharpoonright_n]) \geq \varepsilon$ pour tout $n > |\sigma|$. Comme $\mathcal{F}$ est une classe fermée et que $\mathcal{F} \cap [X \upharpoonright_n]$ est non vide pour tout n , alors $X \in \mathcal{F}$. ■

PREUVE DU THÉORÈME 4.6. Supposons par l'absurde que la classe

$$\mathcal{B}' = \{Z \in \mathcal{B} : \underline{\rho}(\mathcal{B} \mid Z) < 1\}.$$

est de mesure positive.

Soit $\mathcal{B}'_\varepsilon = \{Z \in \mathcal{B} : \underline{\rho}(\mathcal{B} \mid Z) < 1 - \varepsilon\}$. Comme $\mathcal{B}' = \bigcup_{\varepsilon \in \mathbb{Q} \cap [0,1]} \mathcal{B}'_\varepsilon$, il doit exister par additivité dénombrable de la mesure ε tel que la mesure de $\mathcal{B}'_\varepsilon$ est positive. Soit enfin, en utilisant le théorème 17-4.4, une classe fermée $\mathcal{F} \subseteq \mathcal{B}'_\varepsilon$ de mesure positive. Notons que, pour tout $Z \in \mathcal{F}$, on a également $\underline{\rho}(\mathcal{F} \mid Z) < 1 - \varepsilon$, car $\underline{\rho}(\mathcal{F} \mid [\sigma]) \leq \underline{\rho}(\mathcal{B} \mid [\sigma])$ pour toute chaîne σ .

À présent, d'après le lemme 18-3.1 de densité de Lebesgue, il doit exister une chaîne σ telle que $\lambda(\mathcal{F} \mid [\sigma]) > 1 - \varepsilon$. D'après le lemme 4.7, il existe alors $Z \in \mathcal{F}$ tel que $\lambda(\mathcal{F} \mid [Z \upharpoonright_n]) \geq 1 - \varepsilon$ pour tout $n > |\sigma|$, et donc tel que $\underline{\rho}(\mathcal{F} \mid Z) \geq 1 - \varepsilon$, ce qui contredit $\underline{\rho}(\mathcal{F} \mid Z) < 1 - \varepsilon$. Ainsi, la classe $\mathcal{B}' = \{Z \in \mathcal{B} : \underline{\rho}(\mathcal{B} \mid Z) < 1\}$ est de mesure 0. ■

On ne peut bien sûr pas attendre d'une notion d'aléatoire fixe de satisfaire le théorème de densité de Lebesgue pour n'importe quelle classe borélienne, mais on peut le faire pour un certain niveau dans la hiérarchie borélienne effective. Nous nous limiterons au cas le plus simple : les classes Π_1^0 .

Définition 4.8. Un ensemble $Z \in 2^{\mathbb{N}}$ est dit de *densité inférieure* (resp. *supérieure*) *positive* si $\underline{\rho}(\mathcal{F} \mid Z) > 0$ (resp. $\bar{\rho}(\mathcal{F} \mid Z) > 0$), pour toute classe Π_1^0 $\mathcal{F}$ contenant Z . Un ensemble Z est dit de *densité inférieure* (resp. *supérieure*) *1* si $\underline{\rho}(\mathcal{F} \mid Z) = 1$ (resp. $\bar{\rho}(\mathcal{F} \mid Z) = 1$), pour toute classe Π_1^0 $\mathcal{F}$ contenant Z . ◇

Exercice 4.9. (★★) Montrer que les ensembles MLR sont de densité supérieure 1. ◇

Nous donnons à présent une caractérisation des ensembles aléatoires incomplets, en rapport avec le théorème de densité de Lebesgue.

Théorème 4.10 (Bienvenu, Hölz, Miller, Nies [18])

Soit Z un ensemble MLR. Alors, les énoncés suivants sont équivalents.

1. Z est un ensemble incomplet (c'est-à-dire avec $Z \not\geq_T \emptyset'$).
2. Z est un ensemble de densité inférieure positive.

PREUVE. Supposons Z de densité inférieure nulle, et soit $\mathcal{F}$ une classe Π_1^0 telle que $\underline{\rho}(\mathcal{F} \mid Z) = 0$. Soit $\mathcal{U}_n$ la classe Σ_1^0 générée par les chaînes σ telles que $\lambda(\mathcal{F} \mid [\sigma]) < 2^{-n}$. Il est clair que $\lambda(\mathcal{U}_n \cap \mathcal{F}) \leq 2^{-n} \lambda(\mathcal{F}) \leq 2^{-n}$. Comme $\underline{\rho}(\mathcal{F} \mid Z) = 0$, alors Z est capturé par le test $\mathcal{F} \cap \bigcap_n \mathcal{U}_n$. Donc, Z est capturé par un test de différence. D'après le théorème 4.2, $Z \geq_T \emptyset'$.

Supposons à présent que $Z \geq_T \emptyset'$. Alors, d'après le théorème 4.2, Z est capturé par un test de différence $\mathcal{F} \cap \bigcap_n \mathcal{U}_n$. Montrons que l'on a $\underline{\rho}(\mathcal{F} \mid Z) = 0$. Pour chaque $r \in \mathbb{N}$, nous allons construire un test de Martin-Löf $\bigcap_n \mathcal{V}_n$ tel

que pour tout $X \in 2^{\mathbb{N}}$, si $X \in \mathcal{F} \cap \bigcap_n \mathcal{U}_n$ et $X \notin \bigcap_n \mathcal{V}_n$, alors il existe un préfixe $\sigma \prec X$ tel que $\lambda(\mathcal{F} \mid [\sigma]) < 2^{-r}$. On définit $\mathcal{V}_0 = \mathcal{U}_0$, puis une fois $\mathcal{V}_n$ défini, pour toute chaîne σ énumérée dans $\mathcal{V}_n$ on énumère dans $\mathcal{V}_{n+1}$ les chaînes de $\mathcal{U}_{|\sigma|+r+1} \cap [\sigma]$ en bloquant si besoin l'énumération pour ne pas dépasser une mesure de $2^{-|\sigma|}(1-2^{-r-1})$. On a $\lambda(\mathcal{V}_{n+1}) \leq (1-2^{-r-1})\lambda(\mathcal{V}_n)$, et l'on peut donc borner la mesure de $\mathcal{V}_n$ par une fonction calculable et décroissante qui tend vers 0. On peut alors aisément transformer $\bigcap_n \mathcal{V}_n$ en un test de Martin-Löf. Comme Z est MLR, alors $Z \notin \bigcap_n \mathcal{V}_n$. Soit n le plus petit entier tel que $Z \notin \mathcal{V}_n$, et soit $\sigma \prec Z$ tel que σ est énuméré dans $\mathcal{V}_{n-1}$. Comme $Z \in \bigcap_n \mathcal{U}_n$, cela signifie que l'énumération de $\mathcal{V}_n$ a été bloquée et que l'on a $\lambda(\mathcal{U}_{|\sigma|+r+1} \mid [\sigma]) > 1 - 2^{-r-1}$. Supposons à présent par l'absurde $\lambda(\mathcal{F} \mid [\sigma]) > 2^{-r}$. Alors, $\lambda(\mathcal{U}_{|\sigma|+r+1} \cap \mathcal{F} \mid [\sigma]) > 2^{-r-1}$, et donc $\lambda(\mathcal{F} \cap \mathcal{U}_{|\sigma|+r+1}) > 2^{-|\sigma|-r-1}$, ce qui contredit

$$\lambda(\mathcal{F} \cap \mathcal{U}_{|\sigma|+r+1}) < 2^{-|\sigma|-r-1}.$$

Donc, $\lambda(\mathcal{F} \mid [\sigma]) < 2^{-r}$.

Comme on peut faire la même chose pour tout r , il existe pour tout r un préfixe $\sigma \prec Z$ tel que $\lambda(\mathcal{F} \mid [\sigma]) < 2^{-r}$, et Z est donc de densité inférieure nulle dans $\mathcal{F}$. ■

Chapitre 20

Les K-triviaux

Les K-triviaux constituent une des classes les plus fascinantes de l'aléatoire algorithmique, et qui n'a pas cessé de surprendre au fur et à mesure des découvertes la concernant. Le théorème central est sans aucun doute le suivant : un ensemble A est K-trivial si, et seulement si, il est low pour l'aléatoire. Nous commençons notre aventure par une étude de la notion d'être low pour l'aléatoire, qui connaîtra son point culminant avec la remarquable preuve dite « des ensembles affamés ».

1. Lowness et bases pour l'aléatoire

Nous avons vu avec la définition 19-2.8 le concept d'oracle ne modifiant pas la classe des aléatoires de Martin-Löf. Nous voyons ici un concept *a priori* plus fort encore : les oracles ne modifiant pas la complexité sans préfixe.

1.1. Lowness

De manière générale, toute propriété relativisable P induit une notion de lowness, où X est low pour P ssi P^X et $P^\emptyset$ coïncident. Par exemple, un ensemble X est low (pour le saut Turing) si $X' = \emptyset'$. De même, un ensemble X est low pour l'aléatoire de Martin-Löf si tout ensemble MLR est $\text{MLR}(X)$. Nous définissons la notion correspondante pour la complexité sans préfixe.

Définition 1.1. Un ensemble $A \in 2^{\mathbb{N}}$ est *low-pour-K* si $K^A(\sigma) \leq^+ K(\sigma)$ pour toute chaîne σ . ◇

Il est clair que tout ensemble calculable est low-pour-K, et il est possible de montrer que ce ne sont pas les seuls.

Exercice 1.2. (★★) En utilisant la même technique que celle de la preuve du théorème 16-4.5, montrer qu'il existe des ensembles c. e. non calculables et low-pour-K. $\diamond$

Voyons tout de suite deux implications faciles à montrer : si A est low-pour-K, alors il est K-trivial, et aussi low pour l'aléatoire de Martin-Löf.

Proposition 1.3. Si A est low-pour-K, alors A est K-trivial. $\star$

PREUVE. Pour tout oracle X , on a $K^X(X \upharpoonright_n) \leq^+ K^X(n) \leq^+ K(n)$, pour tout n . Pour la première inégalité, étant donné n , il suffit d'utiliser l'oracle X pour produire $X \upharpoonright_n$. Pour la deuxième inégalité, si une machine peut produire n sans oracle, elle le peut aussi avec oracle. Si A est un ensemble A low-pour-K, on a de plus $K(A \upharpoonright_n) \leq^+ K^A(A \upharpoonright_n)$, donc $K(A \upharpoonright_n) \leq^+ K(n)$ pour tout n . ■

Proposition 1.4. Si A est low-pour-K, alors A est low-pour-MLR. $\star$

PREUVE. Supposons que A soit low-pour-K, et considérons un A -test de Martin-Löf $\bigcap_n \mathcal{U}_n$. D'après le théorème 18-2.1 relativisé, pour tout X dans $\bigcap_n \mathcal{U}_n$, pour tout c il existe m tel que $K^A(X \upharpoonright_m) \leq m - c$, et donc tel que $K(X \upharpoonright_m) \leq m - c + d$, pour une certaine constante d indépendante de m . Donc, pour tout X dans $\bigcap_n \mathcal{U}_n$, pour tout c il existe m tel que $K(X \upharpoonright_m)$ est $\leq m - c$. D'après le corollaire 18-2.2, le A -test de Martin-Löf $\bigcap_n \mathcal{U}_n$ est alors inclus dans le test de Martin-Löf universel (sans oracle). ■

Exercice 1.5. (★) Montrer que tout ensemble low-pour-K est de degré low. $\diamond$

1.2. Base pour l'aléatoire et ensembles affamés

Nous voyons à présent que la classe des low-pour-K coïncide avec celle des low pour l'aléatoire de Martin-Löf. Nous utilisons pour cela une troisième notion qui présente son intérêt propre.

Définition 1.6. Un ensemble $A \in 2^{\mathbb{N}}$ est une *base pour l'aléatoire de Martin-Löf* s'il existe un ensemble $Z \in \text{MLR}(A)$ qui calcule A . $\diamond$

Nous avons vu avec le théorème 18-3.2 que pour A non calculable, la classe des ensembles qui calculent A est de mesure 0. De plus, pour une fonctionnelle Φ fixée, la classe $\{X \in 2^{\mathbb{N}} : \forall n \Phi(X, n) \downarrow = A(n)\}$ est une classe $\Pi_2^0(A)$ de mesure 0.

On en déduit que pour tout A non calculable aucun ensemble calculant A n'est fortement $\text{MLR}(A)$. On ne peut pas en revanche toujours montrer qu'un tel ensemble n'est jamais $\text{MLR}(A)$. Il suffit en effet de considérer un ensemble A non calculable et low pour l'aléatoire de Martin-Löf, et d'après le théorème 18-3.4, il existe un ensemble MLR , et donc $\text{MLR}(A)$ qui calcule A . En particulier, si A est low-pour- K alors A est une base pour l'aléatoire. Le théorème suivant montre que la réciproque est vraie, en utilisant une construction sophistiquée et astucieuse, baptisée « preuve des ensembles affamés ».

Théorème 1.7 (Hirschfeldt, Nies et Stephan [90])

Si A est une base pour l'aléatoire, alors A est low-pour- K .

PREUVE. Le lecteur pourra s'aider de la figure 1.8 pour suivre la preuve. Soit Z un ensemble $\text{MLR}(A)$ tel que $\Phi(Z) = A$ pour une fonctionnelle Φ . Soit U la machine à oracle sans préfixe universelle du théorème 19-2.3. On peut voir U comme une énumération de triplets (ρ, τ, x) signifiant alors $U^\rho(\tau) \downarrow = x$.

On peut considérer sans perte de généralité que si $U^X(\tau) \downarrow = x$ pour un ensemble X , alors il y a un unique préfixe $\rho \prec X$ tel que $U^\rho(\tau) \downarrow = x$.

Nous allons décrire un algorithme paramétré par un entier d . Aussi pour tout d l'algorithme énumérera-t-il un ensemble borné de requêtes L_d , et nous montrerons que pour d suffisamment grand, la machine sans préfixe M_d issue de cet ensemble de requêtes sera telle que $K_{M_d}(\sigma) \leq^+ K^A(\sigma)$, pour toute chaîne σ . Durant l'algorithme, pour chaque triplet (ρ, τ, x) , nous allons énumérer des « ensembles affamés » $C_{\tau,x}^\rho \subseteq 2^{<\mathbb{N}}$ qui demandent à être nourris par des chaînes finies jusqu'à être repus. Un ensemble affamé continuera à « manger » des chaînes finies tant qu'il n'est pas rassasié, mais il refusera en revanche de manger quoi que ce soit qui le nourrirait plus que de raison : chaque ensemble $C_{\tau,x}^\rho$ veut atteindre un poids de $2^{-d}2^{-|\tau|}$, mais refusera toujours de dépasser ce poids.

L'algorithme avec paramètre d est le suivant : à une étape de calcul t , pour tout $\langle \rho, \tau, x \rangle \leq t$, si $U^\rho(\tau)[t] \downarrow = x$, on considère alors l'ensemble ouvert/fermé $\mathcal{U} = \{X : \Phi(X)[t] \succeq \rho\}$. Soit W un ensemble de chaînes sans préfixe tel que $[W] = \mathcal{U}$. L'objectif est de rajouter chaque chaîne $\sigma \in W$ dans $C_{\tau,x}^\rho$, mais tout en gardant les ensembles affamés — vus comme des ouverts — deux à deux disjoints. Donc, si un préfixe de σ est dans un autre ensemble affamé, on n'énumérera rien du tout ; et si une extension de σ est dans un autre ensemble affamé, on énumérera seulement les extensions de σ qui ne sont encore dans aucun d'entre eux. Formellement, pour chaque chaîne $\sigma \in W$, on cherche deux ensembles de chaînes finies $B_{0,\sigma}$ et $B_{1,\sigma}$ tels que $[B_{0,\sigma}] \cup [B_{1,\sigma}] = [\sigma]$, tels que $[B_{0,\sigma}] \cap [B_{1,\sigma}] = \emptyset$, et tels que $[\iota]$ est inclus

dans un ensemble affamé pour chaque $\iota \in B_{0,\sigma}$ et $[\iota]$ a une intersection vide avec tous les ensembles affamés pour chaque $\iota \in B_{1,\sigma}$. Finalement, pour tout $\sigma \in W$ et toute chaîne $\iota \in B_{1,\sigma}$, on énumère dans l'ordre ι dans $C_{\tau,x}^\rho$, à condition que l'on conserve toujours $\lambda([C_{\tau,x}^\rho]) \leq 2^{-d} \times 2^{-|\tau|}$. Si l'énumération d'une chaîne fait passer la mesure de $[C_{\tau,x}^\rho]$ au-dessus de $2^{-d} \times 2^{-|\tau|}$, alors on ne l'énumère pas. Enfin, si après l'énumération de toutes ces chaînes on a $\lambda([C_{\tau,x}^\rho]) \geq 2^{-d-1} \times 2^{-|\tau|}$, on énumère $\langle x, |\tau| + d + 1 \rangle$ dans l'ensemble borné de requêtes L_d . Cela conclut la construction.

Montrons que L_d est bien un ensemble borné de requêtes pour tout d . On énumère au plus un élément $\langle x, |\tau| + d + 1 \rangle$ dans L_d pour chaque ensemble affamé, auquel cas cette énumération arrive au moment où l'ensemble $C_{\tau,x}^\rho$ correspondant est « à moitié repu », c'est-à-dire vérifie $\lambda([C_{\tau,x}^\rho]) \geq 2^{-d-1} \times 2^{-|\tau|}$. On a en particulier

$$\text{poids}(L_d) = \sum_{\langle x, |\tau| + d + 1 \rangle \in L_d} 2^{-d-1} \times 2^{-|\tau|} \leq \sum_{\langle \rho, \tau, x \rangle} \lambda([C_{\tau,x}^\rho]).$$

Il suffit alors de remarquer que par construction les ouverts $[C_{\tau,x}^\rho]$ sont deux à deux disjoints. On a alors en particulier par additivité dénombrable de la mesure :

$$\begin{aligned} \text{poids}(L_d) &\leq \sum_{\langle \rho, \tau, x \rangle} \lambda([C_{\tau,x}^\rho]) = \lambda(\bigcup_{\langle \rho, \tau, x \rangle} [C_{\tau,x}^\rho]) \\ &\leq 1. \end{aligned}$$

L'ensemble L_d est donc bien un ensemble borné de requêtes.

Pour d fixé, on définit $\mathcal{U}_d^A = \bigcup_{\rho \prec A, \tau, x} [C_{\tau,x}^\rho]$, où les ensembles $C_{\tau,x}^\rho$ sont ceux produits par l'algorithme avec d comme paramètre. Montrons que $\bigcap_d \mathcal{U}_d^A$ est un A -test de Martin-Löf. On a

$$\lambda(\mathcal{U}_d^A) \leq \sum_{\rho \prec A, U^\rho(\tau) \downarrow = x} \lambda([C_{\tau,x}^\rho]) \leq \sum_{\rho \prec A, U^\rho(\tau) \downarrow} 2^{-d} \times 2^{-|\tau|} \leq 2^{-d} \times \Omega^A \leq 2^{-d}.$$

Donc, $\bigcap_d \mathcal{U}_d^A$ est bien un A -test de Martin-Löf.

Finissons à présent la preuve : comme Z est $\text{MLR}(A)$, alors il existe d tel que $Z \notin \mathcal{U}_d^A$. Montrons que pour un tel d , pour tous τ, x tels que $U^\rho(\tau) \downarrow = x$ pour $\rho \prec A$, l'ensemble $C_{\tau,x}^\rho$ est toujours « à moitié repu », i.e. que la mesure de $[C_{\tau,x}^\rho]$ atteint forcément $2^{-d-1} \times 2^{-|\tau|}$. Notons que si c'est bien le cas, alors aussi pour tous τ, x tels que $U^A(\tau) \downarrow = x$, on énumère $\langle x, |\tau| + d + 1 \rangle$ dans L_d , ce qui implique bien $K_{M_d}(\sigma) \leq^+ K^A(\sigma)$ pour toute chaîne σ . Soient τ, x et $\rho \prec A$ tels que $U^\rho(\tau) \downarrow = x$. Supposons par l'absurde que

$$\lambda([C_{\tau,x}^\rho]) < 2^{-d-1} \times 2^{-|\tau|}.$$

Soit $\sigma \prec Z$ suffisamment grand tel que $2^{-|\sigma|} < 2^{-d-1} \times 2^{-|\tau|}$ et tel que $\Phi(\sigma) \succeq \rho$. Si à ce moment une chaîne σ' telle que $\sigma' \prec \sigma$ ou tel que $\sigma \prec \sigma' \prec Z$ est déjà énumérée dans un ensemble affamé $C_{\tau',x'}^{\rho'}$, ce sera forcément pour $\rho' \prec A$ puisque $\Phi(Z) = A$. Dans ce cas, $Z \in \mathcal{U}_d^A$, ce qui est

une contradiction. Sinon, il existera une chaîne ι avec $\sigma \preceq \iota \prec Z$ tel que $[\iota]$ admet à ce moment une intersection vide avec tous les ensembles affamés. Comme l'énumération de ι dans $C_{\tau,x}^\rho$ laisse la mesure de $[C_{\tau,x}^\rho]$ en dessous de $2^{-d} \times 2^{-|\tau|}$, alors ι est énuméré dans $C_{\tau,x}^\rho$ et $Z \in \mathcal{U}_d^A$, ce qui est encore une contradiction. Donc, pour tous τ, x et pour $\rho \prec A$ tels que $U^\rho(\tau) \downarrow = x$, la mesure de l'ensemble $[C_{\tau,x}^\rho]$ atteint forcément $2^{-d-1} \times 2^{-|\tau|}$, et par conséquent $\langle x, |\tau| + d + 1 \rangle$ est énuméré dans L_d . Ainsi, la machine M_d construite à partir de L_d est telle que $K_{M_d}(\sigma) \leq^+ K^A(\sigma)$ pour toute chaîne σ . Par suite, A est low-pour-K. ■

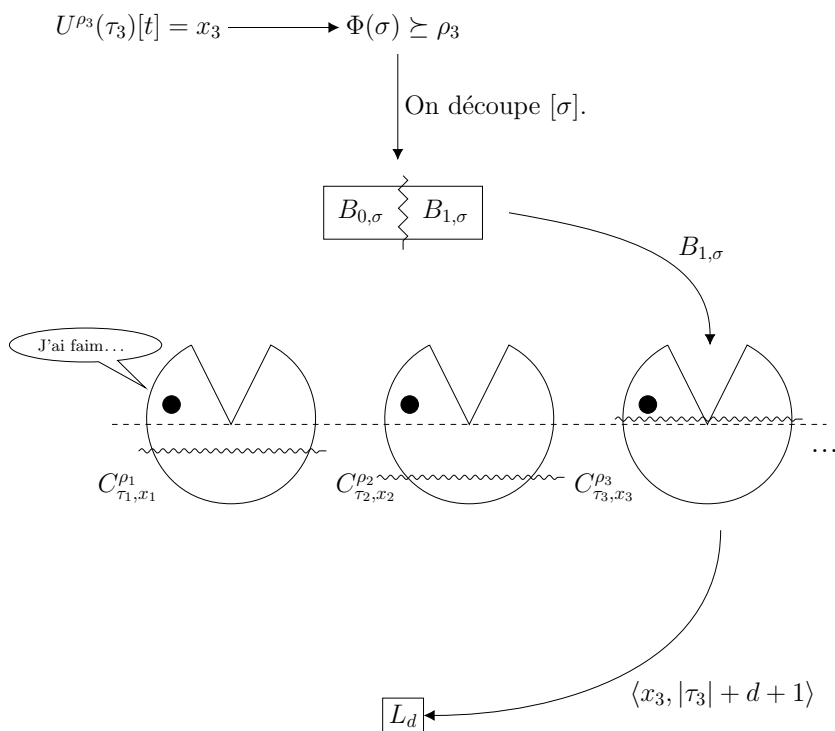


FIGURE 1.8 – Illustration du mécanisme des ensembles affamés. Quand $U^{\rho_3}(\tau_3)[t] = x_3$, on cherche à remplir l'ensemble affamé $C_{\tau_3,x_3}^{\rho_3}$ avec des chaînes σ telles que $\Phi(\sigma) \succeq \rho_3$, tout en gardant les ensembles affamés deux à deux disjoints. C'est pour cela que l'on découpe σ en deux parties $B_{0,\sigma}$ et $B_{1,\sigma}$, où $B_{1,\sigma}$ contiendra le morceau de σ qui n'est pas déjà dans un ensemble affamé. Si un jour $C_{\tau_3,x_3}^{\rho_3}$ est à moitié rempli (c'est-à-dire atteint un poids de $2^{-|\tau_3|-d-1}$), on énumère $\langle x_3, |\tau_3| + d + 1 \rangle$ dans notre ensemble borné de requêtes L_d .

Corollaire 1.9 (Hirschfeldt, Nies and Stephan [90])

Les trois énoncés suivants sont équivalents.

- (1) *A est low-pour-K.*
- (2) *A est low-pour-MLR.*
- (3) *A est une base pour l'aléatoire.*

PREUVE. (1) $\rightarrow$ (2) vient de la proposition 1.4. Montrons (2) $\rightarrow$ (3). Supposons A low-pour-MLR. D'après le théorème 18-3.4, il existe un MLR Z tel que $Z \geq_T A$. Comme A est low-pour-MLR, alors Z est aussi MLR(A), et donc A est une base pour l'aléatoire. Pour finir, (3) $\rightarrow$ (1) est donné par le théorème précédent. ■

Il est temps à présent de rentrer dans le vif du sujet, et de montrer que les ensembles K-triviaux sont tous low-pour-K.

2. Le processus d'or

La preuve des ensembles affamés aura peut-être ouvert l'appétit du lecteur pour les constructions complexes. Que ce dernier se rassure, nous abordons à présent la construction la plus difficile de l'aléatoire algorithmique, dite « du processus d'or », et dont l'objectif sera de montrer le théorème suivant.

Théorème 2.1 (Hirschfeldt, Nies [166])

Les ensembles K-triviaux coïncident avec les ensembles low-pour-K.

Le lecteur préférant la simplicité peut toutefois se rassurer : il existe une autre preuve du fait que tous les ensembles K-triviaux sont low-pour-K, qui elle ne repose pas sur une construction si compliquée et que nous présentons à la fin de ce chapitre. Insistons toutefois sur le fait que cette deuxième preuve n'est pas réellement « plus simple » : elle utilise une série de résultats intermédiaires — dont la preuve des ensembles affamés — tous plus simples à montrer, et qui mis bout à bout arrivent au résultat final. Il est en revanche parfaitement impossible avec cette deuxième preuve de comprendre *pourquoi* les ensembles K-triviaux sont tous low-pour-K. La première preuve que nous présentons ci-après est certes difficile, mais honnête et sans détours. La construction du processus d'or, pour aussi complexe qu'elle soit, montre parfaitement les mécanismes à l'œuvre, et toute personne suffisamment vaillante pour en comprendre les rouages saura réellement pourquoi et comment les ensembles K-triviaux sont tous low-pour-K. La deuxième preuve a bien entendu elle aussi ses avantages, ne serait-ce que parce que chaque résultat intermédiaire de cette preuve détournée a son intérêt propre.

La présente section est entièrement consacrée à la preuve du théorème 2.1. Soit U la machine à oracle sans préfixe universelle du théorème 19-2.3. Soit A un ensemble K-trivial via une constante c , c'est-à-dire tel que

$$K(A \upharpoonright_n) \leq K(n) + c, \text{ pour tout } n.$$

Nous allons prouver qu'un tel ensemble A est forcément low-pour-K, c'est-à-dire que l'on a aussi $K(\sigma) \leq^+ K^A(\sigma)$ pour toute chaîne σ . Nous construirons à cet effet une machine sans préfixe M telle que $K_M(\sigma) \leq^+ K^A(\sigma)$ pour toute chaîne σ .

Idée générale. Comme A est K-trivial, nous savons d'après le corollaire 16-4.9 qu'il est Δ_2^0 . Nous allons donc utiliser l'approximation de A pour construire notre machine M , dont le but sera de faire aussi bien — à constante près — que la machine universelle U sur l'oracle A . Pour ce faire, dès que l'on a $U^\rho(\tau)[s] = \sigma$ à une étape de calcul s pour $\rho \prec A_s$ et pour des chaînes τ et σ , on aimerait énumérer le couple $(\sigma, |\tau|)$ dans un ensemble borné de requêtes. Le problème est évidemment que $\rho \prec A_s$ — l'approximation courante de A utilisée pour envoyer τ sur σ — n'est peut-être pas la bonne. Si cette dernière change, on aura pour ainsi dire « perdu » l'envoi d'une chaîne de taille $|\tau|$ vers σ , sans que cela ne corresponde à ce qui se passe avec le vrai oracle A . Nous dirons alors que l'on perd la quantité $2^{-|\tau|}$, qui sera décomptée de notre ensemble borné de requêtes, dont le poids, rappelons-le, ne doit pas dépasser 1.

Afin d'éviter trop de perte, nous n'allons pas tout de suite énumérer le couple $(\sigma, |\tau|)$ dans notre ensemble borné de requêtes. Nous demanderons d'abord à l'approximation courante de A , des « preuves » de sa solidité. Cela se fera via la construction d'une machine tierce M_d , construite elle aussi via un ensemble borné de requêtes L_d . Soit c_d un entier tel que

$$K(\sigma) \leq K_{M_d}(\sigma) + c_d, \text{ pour toute chaîne } \sigma.$$

On va énumérer dans L_d le couple (n, l) pour une certaine taille l et un certain $n \geq |\rho|$, puis attendre d'avoir $K(A_s \upharpoonright_n)[s] \leq l + c + c_d$ (rappelons que c est la constante de K-trivialité de notre ensemble A). Notons que si L_d est réellement un ensemble borné de requêtes, on doit avoir $K_{M_d}(n) \leq l$. Donc, $K(n) \leq K_{M_d}(n) + c_d \leq l + c_d$. Comme $K(A \upharpoonright_n) \leq K(n) + c$, on doit avoir aussi $K(A \upharpoonright_n) \leq l + c + c_d$.

Si durant cette attente de validation le préfixe $\rho \prec A_s$ reste inchangé, alors il aura prouvé sa solidité, et seulement à ce moment-là nous énumérerons le couple $(\sigma, |\tau|)$ dans notre ensemble borné de requêtes. Bien sûr, une fois que ρ a été validé, cela ne signifie pas qu'il ne changera plus, mais à chaque fois qu'il change on a forcé $K(A_s \upharpoonright_n)$ à être en dessous d'une certaine valeur, pour un mauvais préfixe de A_s , l'objectif est de faire en sorte que cela ne puisse pas arriver trop souvent sans que l'on ait $\sum_\sigma 2^{-K(\sigma)} > 1$.

La construction utilisera un arbre de processus, à branchement infini, mais de hauteur finie. Chacun de ces processus s'efforcera de construire une machine M telle que $K_M(\sigma) \leq^+ K^A(\sigma)$. Si chaque processus travaillera à sa propre tentative de montrer que A est low-pour K , via sa propre machine, tous les processus partageront en revanche un même ensemble borné de requêtes L_d dont le rôle est discuté ci-dessus. L'algorithme dans sa globalité utilise le théorème du point fixe pour avoir accès à la constante c_d associée à la machine M_d que nous construisons.

Notion de i -ensemble. On dit qu'un ensemble $E \subseteq \mathbb{N} \times \mathbb{N}$ est un i -ensemble à l'étape s si tout $(n, l) \in E$ a été énuméré dans L_d à une étape $t \leq s$ et si l'on a i approximations distinctes $A_r \upharpoonright_n$, avec $t \leq r \leq s$, qui sont chacune telle que $K(A_r \upharpoonright_n) \leq l + c + c_d$. Le poids d'un i -ensemble est donné par $\text{poids}(E) = \sum_{(n,l) \in E} 2^{-l}$. On suppose de plus que, pour deux éléments distincts $(n_1, l_1), (n_1, l_2) \in E$, on a $n_1 \neq n_2$.

On fixe $k = 2^{c+c_d+1}$. Chacun de nos processus va créer un i -ensemble pour $i \leq k$. Le point de départ du futur argument du processus d'or est le lemme suivant.

Lemme 2.2. Si E est un k -ensemble, alors $\text{poids}(E) < \frac{1}{2}$. ★

PREUVE. Pour tout (n, l) dans E , on a k chaînes distinctes $\rho_1, \dots, \rho_k$ de taille n telles que $K(\rho_i) \leq l + c + c_d$ pour $i \leq k$, et donc telles que

$$2^{-l-c-c_d} \leq 2^{-K(\rho_i)}, \text{ pour } i \leq k.$$

En particulier, $k \times 2^{-l-c-c_d} \leq \sum_{i \leq k} 2^{-K(\rho_i)}$. De plus, les éléments de E sont tous distincts deux à deux sur leur première coordonnée. On a donc

$$k \times \sum_{(n,l) \in E} 2^{-l-c-c_d} \leq \sum_{\rho} 2^{-K(\rho)} < 1.$$

Par suite, $k \times 2^{-c-c_d} \times \text{poids}(E) < 1$. Comme $k = 2^{c+c_d+1}$, on a par conséquent $\text{poids}(E) < \frac{1}{2}$. ■

Nous allons donner d'ici peu la description de k programmes $P_1, \dots, P_k$ distincts, où chaque instance de P_i sera un processus chargé de produire un i -ensemble.

Arbre des processus. L'algorithme consistera en un arbre dynamique de processus, où chaque nœud de hauteur i sera une instance du programme P_{k-i} . Dans cet arbre, les processus fils d'un nœud correspondent aux processus appelés par ce nœud. Chaque nœud reste dans l'arbre tant que son processus correspondant tourne, et disparaît avec tous ses fils quand le processus s'arrête, ou bien est stoppé par un de ses processus ascendants.

L'algorithme commence par exécuter une unique instance de P_k , qui sera la racine de notre arbre. Cette instance de P_k appellera une infinité de processus fils, chacun étant une instance de P_{k-1} , chacun de ses processus fils appellera à son tour des instances de P_{k-2} et ainsi de suite. Les feuilles de cet arbre seront donc des instances de P_1 .

Chaque instance du programme P_i aura la charge d'énumérer son propre i -ensemble, avec l'aide des processus fils qu'il appelle. Ainsi, chaque instance de P_1 — les feuilles de l'arbre — entamera l'énumération d'un 1-ensemble. Chaque instance de P_2 entamera l'énumération d'un 2-ensemble, avec l'aide des 1-ensembles énumérés par ses processus fils, puis chaque instance de P_3 entamera l'énumération d'un 3-ensemble, avec l'aide des 2-ensembles énumérés par ses processus fils, et ainsi de suite.

Paramètres des processus. Chaque programme P_i prend trois paramètres en entrée.

1. Le paramètre de *seuil* p : il s'agit d'un nombre rationnel plus petit que 1, qui correspond à un *seuil* à atteindre pour P_i , c'est-à-dire que P_i essaiera d'énumérer un i -ensemble F_i de poids p .
2. Le paramètre de *pas* δ : il s'agit d'un rationnel $\leq p$ et qui correspond à « la vitesse » à laquelle P_i essaiera d'atteindre son seuil p , c'est-à-dire qu'il énumérera dans son i -ensemble des couples $(n, -\log_2(\delta))$. Pour cette raison, on demande donc que p soit un multiple de δ .
3. Le paramètre de *taille* w : le dernier paramètre de P_i correspond à la taille d'un préfixe d'une certaine approximation de A , cette taille devant augmenter au fur et à mesure des appels récursifs. Cela sera clarifié par la suite.

Expliquons brièvement l'idée du paramètre de seuil p . Quand le poids du i -ensemble d'un processus atteint son seuil p , ce processus s'arrête, et arrête récursivement tous ses processus fils : les éléments de son i -ensemble sont à présent prêts à être promus en éléments du $(i+1)$ -ensemble de son processus parent. Cela arrivera si jamais l'approximation d'un préfixe de A — d'une taille bien spécifique, que l'on détaillera par la suite — change à nouveau.

Expliquons brièvement l'idée du paramètre de pas δ . Par définition, avant d'énumérer (n, l) dans un i -ensemble, on doit d'abord l'énumérer dans L_d . Le souci est que le poids 2^{-l} correspondant à (n, l) est susceptible d'être « dépensé pour rien ». Cela arrive si après l'énumération de (n, l) dans L_d , l'approximation correspondante de $A_s \upharpoonright_n$ de $A \upharpoonright_n$ change avant que l'on ait $K(A_s \upharpoonright_n) \leq l + c + c_d$. Une instance de P_i va donc remplir son i -ensemble petit à petit avec un pas de δ , où δ correspond à une perte potentielle. L'ensemble de tous les paramètres δ à choisir doit donc être tel que le poids total qui peut être perdu — c'est-à-dire la somme des paramètres δ — n'est pas trop importante.

Expliquons enfin l'idée du paramètre taille w . Il s'agit de la taille du préfixe courant de A qui est en attente de validation par le processus parent. Il sera nécessaire pour le processus courant de ne faire valider que des préfixes de taille supérieure à w , et ainsi de suite, jusqu'aux processus feuilles.

Le processus d'or. Le cœur de la preuve réside dans l'idée du **processus d'or**. Chaque instance de P_i essaiera de créer sa propre machine M , via un ensemble borné de requêtes L , telle que $K_M(\sigma) \leq^+ K^A(\sigma)$ pour toute chaîne σ . Le i -ensemble d'une instance de P_i ne sert pas uniquement comme étape à la création du $(i + 1)$ -ensemble de son processus parent. Il correspond également au poids qui est « perdu » pour cette instance lors de sa tentative de création d'une machine M telle que $K_M(\sigma) \leq^+ K^A(\sigma)$ pour toute chaîne σ .

Nous avons vu que le poids du k -ensemble du processus racine de l'arbre reste toujours en dessous de $1/2$. De ce fait, ce qui est perdu par le processus racine est borné, et c'est là tout l'objectif : le poids qui est perdu est celui qui a été dépensé pour essayer de faire aussi bien que la machine universelle U^A , mais sur de mauvais préfixes de A . Si cette quantité perdue est toujours plus petite que $1/2$, alors il reste « la moitié » de la place totale pour réellement faire aussi bien que U^A — à constante près, bien sûr. On ne peut toutefois pas en conclure que le processus racine réussira à énumérer la machine M désirée : il se pourrait que la validation d'un certain préfixe de A ne vienne jamais, et il suffit pour cela qu'une instance d'un processus fils n'atteigne lui-même jamais son seuil. Néanmoins, si cela arrive, ce processus fils sera lui-même un candidat pour la création d'une machine M telle que $K_M(\sigma) \leq^+ K^A(\sigma)$, pour toute chaîne σ . Cela fonctionnera si toutes ses demandes de validations aboutissent — tant que A ne change pas en cours de validation, bien sûr —, c'est-à-dire si tous ses processus fils sont soit stoppés avant validation, ou alors atteignent eux-mêmes leur seuil. Si l'un d'eux n'atteint jamais son seuil, on descend alors encore dans l'arbre jusqu'à arriver à des instances de P_2 , pour lesquelles nous montrerons que les validations aboutissent toujours.

Ainsi, un processus qui n'est jamais stoppé par un de ses nœuds ascendants, qui n'atteint jamais son seuil et pour lequel toutes ses demandes de validation aboutissent sera un **processus d'or**, qui aboutira à la création d'une machine M telle que $K_M(\sigma) \leq^+ K^A(\sigma)$ pour toute chaîne σ .

Les étapes spéciales de calcul. Afin de simplifier la description de l'algorithme, nous travaillerons avec des étapes de calcul spéciales, c'est-à-dire des étapes pour lesquelles l'ensemble A *semble* K-trivial. Formellement, on a $s_0 = 1$ et s_{n+1} est le plus petit temps de calcul strictement plus grand que s_n et tel que $K(A_{s_{n+1}} \upharpoonright_m)[s_{n+1}] \leq K(m)[s_{n+1}] + c$, pour tout $m \leq s_n$. Nous noterons *étapes de calcul* (en italique) ces étapes de

calcul spéciales $s_0, s_1, s_2, \dots$, et l'algorithme se restreindra à ces étapes spéciales.

La raison de ces *étapes de calcul* est qu'une fois $K(A_s \upharpoonright_n)[s] \leq l + c + c_d$ pour un certain l , si $A_{s+1} \upharpoonright_n \neq A_s \upharpoonright_n$ on veut mettre (n, l) dans un 2-ensemble, mais cela n'est possible que si l'on a aussi $K(A_{s+1} \upharpoonright_n)[s+1] \leq l + c + c_d$. Pour cette raison-là, à la place de chercher une étape de calcul s telle que $K(A_s \upharpoonright_n)[s] \leq l + c + c_d$, on cherchera à la place une *étape* $s > n$ telle que $K(n)[s] \leq l + c_d$. De cette manière, tant que l'on ne travaille que sur nos *étapes de calcul*, on aura toujours $K(A_t \upharpoonright_n)[t] \leq K(n)[t] + c \leq l + c + c_d$, pour n'importe quelle *étape* $t \geq s$.

Description de l'algorithme. Nous commençons par rappeler ici les différentes notations utilisées pour l'algorithme :

- ▷ A est notre ensemble Δ_2^0 et K -trivial via la constante c ,
- ▷ U^A est la machine universelle qui définit notre complexité K^A et pour laquelle on essaye de construire M telle que $K_M(\sigma) \leq^+ K^A(\sigma)$ pour toute chaîne σ .

L'algorithme global crée un arbre de processus en commençant par appeler une instance du programme P_k décrit ci-dessous, avec comme paramètre de seuil $p = \frac{3}{4}$, comme paramètre de pas $\delta = \frac{1}{4} \times 2^{-1}$ et comme paramètre taille $w = 0$. L'instance de P_k appellera à son tour des instances du programme P_{k-1} , qui elles-mêmes appelleront des instances du programme P_{k-2} , et ainsi de suite.

Chaque processus appelé participe à la création d'un ensemble borné de requêtes commun L_d , qui sert lui-même à construire une machine M_d . À l'aide du théorème du point fixe, l'algorithme global utilise une constante c_d telle que $K(\sigma) \leq K_{M_d}(\sigma) + c_d$ et pour laquelle on a $k = 2^{c+c_d+1}$.

On termine avec une dernière considération sur les étapes de calcul et la manière dont s'effectue le parallélisme entre les différents processus. En pratique, un « agenceur global » simule le parallélisme en répartissant les différentes *étapes* de calcul entre les différents processus à exécuter. Pour simplifier la présentation, cet agencement est implicite, et le déroulement de chaque processus est décrit *étape* par *étape*. Nous donnons juste la contrainte suivante pour l'agenceur global : une *étape* s d'un processus parent est exécuté avant la même *étape* s de ses processus fils.

Il sera enfin utile, pour la création de l'ensemble borné de requêtes L_d qui sera partagé entre tous les processus, d'avoir accès à un entier unique par processus et temps de calcul. Nous utiliserons pour cela le temps de calcul de l'agenceur global et pour une *étape* de calcul s au sein d'un processus, on notera s^* l'étape de calcul globale correspondante. Sans plus tarder voici, sur la page ci-après, l'algorithme.

Programme P_i pour $1 < i \leq k$

Entrée: Le seuil p , le pas δ et la taille w

Sortie : Un i -ensemble F_i et un ensemble borné de requêtes L

Pour toute étape s **faire**

Pour toute sous-étape $\langle \tau, \sigma \rangle \leq s$ **faire**

Si $\langle \tau, \sigma \rangle$ est marqué comme disponible **alors**

Si $\exists \rho \preceq A_s$ tel que $U^\rho(\tau)[s] = \sigma$ **alors**

 Soit $\rho_{\tau, \sigma}$ le plus petit préfixe de A_s de taille plus grande que $\min(w, |\rho|)$.

 Soit $w_{\tau, \sigma}$ la taille de $\rho_{\tau, \sigma}$.

 Soit δ' le plus grand rationnel plus petit que $\frac{1}{4} \times 2^{-s^*}$
 tel que $2^{-|\tau|} \times \delta$ est un multiple de δ' .

 On appelle alors le programme $P_{i-1}(2^{-|\tau|} \times \delta, \delta', w_{\tau, \sigma})$.

 On note $P_{i-1, \tau, \sigma}$ le processus correspondant et $F_{i-1, \tau, \sigma}$ son $(i-1)$ -ensemble.

 On marque $\langle \tau, \sigma \rangle$ comme indisponible.

Fin

Fin

Si

$\langle \tau, \sigma \rangle$ est marqué comme indisponible après un appel à $P_{i-1, \tau, \sigma}$
 alors

Si le processus $P_{i-1, \tau, \sigma}$ est terminé **alors**

 On énumère $\left(\sigma, |\tau| - \log_2 \left(\frac{\delta}{p + \delta} \right) \right)$ dans L .

Fin

Si $A_s \upharpoonright_{w_{\tau, \sigma}}$ est différent de $\rho_{\tau, \sigma}$ **alors**

 On marque $\langle \tau, \sigma \rangle$ comme disponible.

 Si le processus $P_{i-1, \tau, \sigma}$ n'est pas terminé, on l'annule
 ainsi que tous ses sous-processus (en
 conservant $F_{i-1, \tau, \sigma}$)

 On énumère les éléments de $F_{i-1, \tau, \sigma}$ dans F_i .

Si $\text{poids}(F_i) \geq p$ **alors**

 On arrête le programme ainsi que tous ses
 sous-processus.

Fin

Fin

Fin

Fin

Fin

Programme P_1

Entrée: Le seuil p , le pas δ et la taille w
Sortie : Un 1-ensemble F_1

Pour toute *étape* $s > w$ **faire**

Soit $n = s^*$ (que l'on suppose aussi plus grand que w).

On énumère $(n, -\log_2(\delta))$ dans L_d .

On attend une *étape* $t > n$ telle que $K(n)[t] \leq -\log_2(\delta) + c_d$

On énumère $(n, -\log_2(\delta))$ dans F_1

Si $\text{poids}(F_1) \geq p$ **alors**
 | On arrête le programme.

Fin

Fin

Passons à présent à la vérification formelle.

Montrons que la sortie F_i de chaque instance de P_i est un i -ensemble. Commençons par le fait suivant.

- (1) Soient $(n_1, l_1), (n_2, l_2)$ deux couples énumérés dans n'importe quel 1-ensemble durant l'algorithme à des étapes différentes ou par des processus différents. Alors, $n_1 \neq n_2$.

Cela vient du fait que de tels couples $(n_1, l_1), (n_2, l_2)$ sont énumérés à deux étapes globales distinctes $s_1^* \neq s_2^*$ qui seront par construction telles que $n_1 = s_1^*$ et $n_2 = s_2^*$. Montrons que chaque ensemble F_1 énuméré par une instance de P_1 est un 1-ensemble à chaque *étape* de calcul. Tout d'abord (1) permet de satisfaire l'une des conditions pour être un 1-ensemble, c'est-à-dire avoir des éléments deux à deux distincts sur leur première coordonnée.

Pour l'autre condition, si à une *étape* s on énumère (n, l) dans un 1-ensemble, cela signifie que $K(n)[t] \leq l + c_d$. Comme t est une *étape de calcul*, alors $K(A_t \upharpoonright_n)[t] \leq K(n)[t] + c \leq l + c + c_d$, et l'on satisfait donc bien l'autre condition pour être un 1-ensemble.

Supposons à présent que chaque ensemble $F_{i,\tau,\sigma}$ associé à une instance $P_{i,\tau,\sigma}$ de P_i soit un i -ensemble à chaque *étape de calcul*. Supposons qu'à une étape s , on promet un élément (n, l) d'un i -ensemble $F_{i,\tau,\sigma}$ vers le $(i+1)$ -ensemble F_{i+1} d'une instance de P_{i+1} . D'après (1), tous les éléments de F_{i+1} après cette énumération restent deux à deux distincts sur leur première coordonnée.

Par construction, on a également :

- ▷ une *étape* $t < s$ et un entier $w_{\tau,\sigma} < n$ tels que $A_t \upharpoonright_{w_{\tau,\sigma}} \neq A_s \upharpoonright_{w_{\tau,\sigma}}$;
- ▷ une *étape* r avec $t < r < s$ telle que $K(n)[r] \leq l + c_d$ (et donc telle que (n, l) entre dans un 1-ensemble) ;
- ▷ une *étape* entre r et s telle que le couple (n, l) arrive dans le i -ensemble $F_{i,\tau,\sigma}$.

On a $K(A_s \upharpoonright_n)[s] \leq K(n)[s] + c \leq K(n)[r] + c \leq l + c + c_d$, car s est une *étape de calcul* plus grande que r . Donc, $A_s \upharpoonright_n$ est un bon candidat pour faire de (n, l) l'élément d'un $(i + 1)$ -ensemble. On doit toutefois vérifier que $A_s \upharpoonright_n$ est différent de toutes les autres chaînes $\rho_1, \dots, \rho_i$ de taille n qui font de (n, l) l'élément du i -ensemble $F_{i,\tau,\sigma}$. N'oublions pas en effet qu'une approximation Δ_2^0 peut revenir à une valeur déjà rencontrée par le passé. Il suffit de montrer que A_t partage ses $w_{\tau,\sigma}$ premiers bits avec chaque ρ_j pour $j \leq i$. Si c'est bien le cas, comme on a $A_t \upharpoonright_{w_{\tau,\sigma}} \neq A_s \upharpoonright_{w_{\tau,\sigma}}$, alors $A_s \upharpoonright_n$ sera bien une chaîne distincte de chaque ρ_j . Il suffit pour cela de voir que s est, au sein de notre instance de P_{i+1} , la plus petite *étape* telle que $A_t \upharpoonright_{w_{\tau,\sigma}} \neq A_s \upharpoonright_{w_{\tau,\sigma}}$ et de se souvenir que les *étapes* des processus parents sont traités avant les étapes des processus *fils*.

Montrons que L_d est bien un ensemble borné de requêtes. Le poids total qui est énuméré dans L_d par un processus $P_{1,\tau,\sigma}$ exécuté avec paramètre (p, δ, w) est borné par le poids de son 1-ensemble plus δ . En effet, par construction, pour chaque $(n, -\log_2(\delta))$ énuméré dans L_d , on énumère aussi $(n, -\log_2(\delta))$ dans le 1-ensemble à moins que le processus ne soit stoppé par un de ses nœuds ascendants, ce qui n'arrive qu'une fois. Le poids que l'on perd est alors de δ .

Étant donné une instance de P_{i+1} exécutée avec paramètre (p, δ, w) , soit C l'ensemble des éléments qui sont énumérés dans le i -ensemble d'un processus fils de l'instance de P_{i+1} , mais qui ne sont pas énumérés dans son $(i + 1)$ -ensemble. Notons que des éléments sont dans C si l'instance de P_{i+1} est annulée par un de ses nœuds ascendants. Montrons que le poids de C est borné par δ . Au moment où l'instance de P_{i+1} est annulée, soit $(P_{i,\tau_j,\sigma_j})_{j \in \mathbb{N}}$ l'ensemble des processus fils qui ne sont pas terminés. Notons que les τ_j sont forcément dans le domaine de U^ρ pour des chaînes ρ deux à deux compatibles : ces chaînes ρ sont des préfixes d'une certaine approximation de A_s , et quand cette approximation change le processus fils correspondant est annulé par P_{i+1} lui-même. Soit ρ la plus grande de toutes ces chaînes. Comme le poids du i -ensemble correspondant à P_{i,τ_j,σ_j} est borné par $2^{-\tau_j} \delta$, alors le poids total est borné par $\delta \sum_{U^\rho(\tau) \downarrow} 2^{-|\tau|} \leq \delta$.

En itérant cette idée, on obtient que la somme des poids totaux de tous les 1-ensembles est bornée par le poids du k -ensemble du processus racine de

l'arbre, plus la somme de tous les paramètres de pas δ . Aussi ces paramètres sont-ils choisis de manière à ce que leur somme soit inférieure à $1/4$. Comme le poids du k -ensemble du processus racine est toujours inférieur à $1/2$, le poids total de ce qui est énuméré dans L_d est bien inférieur à 1.

Fin de la preuve : le processus d'or. Montrons à présent qu'il y a un processus $P_{i,\tau,\sigma}$ appelé avec paramètre (p, δ, w) , qui n'est jamais stoppé par un de ses noeuds ascendants, qui n'atteint jamais son seuil p , et tel que chaque processus qu'il appelle termine à moins qu'il ne soit stoppé par $P_{i,\tau,\sigma}$.

On sait que le processus racine ne peut pas être stoppé par un autre processus et n'atteint jamais son seuil, par le lemme 2.2. Si chacun de ses processus fils se termine ou est annulé, alors le processus racine est le processus d'or. Sinon, au moins un des processus fils de la racine n'est jamais annulé et n'atteint jamais son seuil. Soit il s'agit du processus d'or, soit également un de ses fils n'est jamais stoppé et n'atteint jamais son seuil. Si l'induction se poursuit jusqu'à un processus de la forme $P_{2,\tau,\sigma}$, ce processus est alors nécessairement le processus d'or, car il n'est jamais annulé, n'atteint jamais son seuil, et tous les processus feuilles de l'arbre atteignent nécessairement leur objectif à moins qu'ils ne soient stoppés par un processus ascendant.

Soit à présent un processus d'or donné par une instance de P_i , appelée avec paramètre (p, δ, w) . Soit F_i son i -ensemble et soit L l'ensemble énuméré par ce processus.

Montrons que L est bien un ensemble borné de requêtes. Pour chaque couple

$$\left(\sigma, |\tau| - \log_2 \left(\frac{\delta}{p + \delta} \right) \right)$$

énuméré dans L , on a $U^\rho(\tau) = \sigma$ pour $\rho \prec A_s$ pour un certain s . Ce couple appartient à un $(i-1)$ -ensemble créé par le processus fils $P_{i-1,\tau,\sigma}$. Deux cas se présentent.

Cas 1. La chaîne ρ est un préfixe de A . Si l'on considère la somme des poids des couples associés aux préfixes de A , on obtient alors

$$\sum_{U^A(\tau) \downarrow} 2^{-|\tau| + \log_2 \left(\frac{\delta}{p + \delta} \right)} = \frac{\delta}{p + \delta} \sum_{U^A(\tau) \downarrow} 2^{-|\tau|} \leq \frac{\delta}{p + \delta}.$$

Cas 2. La chaîne ρ n'est pas un préfixe de A . Ainsi, lors d'une étape s future, $A_s \upharpoonright_{|\rho|} \neq \rho$, donc le $(i-1)$ -ensemble créé par le processus fils $P_{i-1,\tau,\sigma}$ va être intégré au i -ensemble F_i créé par le processus d'or P_i . La somme des poids des couples associées aux ρ qui ne sont pas préfixes de A est bornée

par la somme suivante, que nous allons simplifier :

$$\begin{aligned}
 \sum_{P_{i-1,\tau,\sigma}} 2^{-|\tau|+\log_2\left(\frac{\delta}{p+\delta}\right)} &= \frac{\delta}{p+\delta} \times \frac{\sum_{P_{i-1,\tau,\sigma}} 2^{-|\tau|+\log_2(\delta)}}{\delta} \\
 &= \frac{\delta}{p+\delta} \times \frac{\sum_{P_{i-1,\tau,\sigma}} \text{poids}(F_{i-1,\tau,\sigma})}{\delta} \\
 &= \frac{\delta}{p+\delta} \times \frac{\text{poids}(F_i)}{\delta}.
 \end{aligned}$$

Le poids total de L sera donc borné par

$$\frac{\delta}{p+\delta} + \frac{\delta}{p+\delta} \times \frac{\text{poids}(F_i)}{\delta}.$$

Comme F_i n'atteint jamais son seuil, qui est p , on a donc :

$$\frac{\delta}{p+\delta} \left(1 + \frac{\text{poids}(F_i)}{\delta}\right) \leq \frac{\delta}{p+\delta} (1 + p/\delta) \leq 1.$$

Donc, L est bien un ensemble borné de requêtes. Soit M la machine correspondante.

Montrons que $K_M(\sigma) \leq^+ K^A(\sigma)$, pour toute chaîne σ . À présent, si $U^\rho(\tau)[s] = \sigma$ tel que $A_s \upharpoonright_{|\rho|}$ ne changera plus jamais, alors le processus $P_{i-1,\tau,\sigma}$ ne sera jamais annulé. Il atteindra donc son seuil, et l'on énumérera alors $(\sigma, |\tau| - \log_2(\delta/(p+\delta)))$ dans L . Cela conclut la preuve, car alors $K_M(\sigma) \leq K^A(\sigma) - \log_2(\delta/(p+\delta))$ pour toute chaîne σ .

2.1. Conséquences du théorème et de sa preuve

Nous avons à présent terminé la preuve du **processus d'or**. Voyons-en tout de suite une conséquence intéressante.

Théorème 2.3 (Nies [166])

La classe des K-triviaux forme un idéal Turing :

- (1) si A est K-trivial alors tout $B \leq_T A$ est K-trivial ;
- (2) si A_0, A_1 sont K-triviaux, alors $A_0 \oplus A_1$ est K-trivial.

PREUVE. Nous avons vu avec l'exercice 16-4.7 que les ensembles K-triviaux sont clos par jointure Turing. Il est clair que les ensembles low-pour-K sont clos par le bas dans les degrés Turing. ■

Remarquons que dans le théorème précédent, ni (1) n'est évident pour les ensembles K-triviaux, ni (2) pour les ensembles low-pour-K. Le fait que les deux classes coïncident est remarquable, et nous donne ce joli résultat.

Souvenons-nous un instant de la preuve du théorème 16-4.5, où l'on construit un ensemble c. e. A non calculable tel que $K(A \upharpoonright_n) \leq^+ K(n)$ pour tout n . Durant l'approximation, on assignait le poids $2^{-K(n)[s]}$ à chaque préfixe $A_s \upharpoonright_n$ de l'approximation courante de A pour $n \leq s$, c'est-à-dire la taille à l'étape s de la plus petite chaîne produisant n . Cette assignation correspond à ce que l'on place dans un ensemble borné de requêtes pour produire $A_s \upharpoonright_n$. Si à l'étape $s + 1$ un entier $x_s < s$ est énuméré dans A , alors on a perdu tout ce que l'on avait assigné à $A_s \upharpoonright_n$ pour $x_s < n \leq s$. La construction garantit alors que la somme totale de tout ce qui est perdu est finie.

Pour construire un ensemble A c. e. non calculable et low-pour- K , on procède de même, mais où l'on assigne cette fois un poids de $\Omega_s^{A_s \upharpoonright_n} - \Omega_s^{A_s \upharpoonright_{n-1}}$ à chaque préfixe $A_s \upharpoonright_n$ de l'approximation courante de A . Ici, $\Omega_s^{A_s \upharpoonright_n} - \Omega_s^{A_s \upharpoonright_{n-1}}$ est la somme des $2^{-|\tau|}$ tels que $U^{A_s}(\tau)[s] \downarrow$ avec un usage de A_s exactement de n . Comme pour la construction d'un ensemble c. e. K -trivial, si $x_s < s$ est énuméré dans A à l'étape $s + 1$, on voit alors que l'on perd pour tout $x_s < n \leq s$ la quantité $\Omega_s^{A_s \upharpoonright_n} - \Omega_s^{A_s \upharpoonright_{n-1}}$, c'est-à-dire tout ce que l'on a assigné à de mauvais préfixes de A . On peut abstraire ce raisonnement et considérer ce que l'on appelle une *fonction de coût*.

Définition 2.4. Une *fonction de coût* est donnée par $c : \mathbb{N} \rightarrow \mathbb{Q}$, une fonction Δ_2^0 telle que $\sum_n c(n) < 1$ et telle que $\sum_n c_s(n) < 1$ pour toute approximation c_s . L'approximation $(c_s)_{s \in \mathbb{N}}$ d'une fonction de coût peut éventuellement dépendre d'une approximation $\Delta_2^0(A_s)_{s \in \mathbb{N}}$ d'un ensemble A auquel cas la fonction sera dite *adaptative*. $\diamond$

La fonction de coût pour construire un ensemble K -trivial est $c(n) = 2^{-K(n)}$. Celle utilisée pour construire un ensemble low-pour- K est adaptative, et donnée par l'approximation $c_s(n) = \Omega_s^{A_s \upharpoonright_n} - \Omega_s^{A_s \upharpoonright_{n-1}}$. Dans les deux cas, on peut construire un ensemble c. e. non calculable tel que ce que l'on perd est fourni par la fonction de coût, donc est finie. Cela nous donne la définition suivante.

Définition 2.5. Soit $(A_s)_{s \in \mathbb{N}}$ une approximation Δ_2^0 d'un ensemble A . Soit $S = \{s \in \mathbb{N} : A_s \neq A_{s+1}\}$ et, pour $s \in S$, soit $x_s < s$ le plus petit entier tel que $A_{s+1} \upharpoonright_{x_s} \neq A_s \upharpoonright_{x_s}$. L'approximation de A satisfait une fonction de coût c —éventuellement adaptative— si $\sum_{s \in S} \sum_{x_s < n \leq s} c_s(n)$ est fini. $\diamond$

La preuve du processus d'or peut être adaptée pour montrer que l'on peut accélérer l'approximation Δ_2^0 de tout ensemble K -trivial, de manière à montrer qu'elle satisfait n'importe quelle fonction de coût fixée à l'avance. Cela nous donne le théorème suivant.

Théorème 2.6 (Nies [166])

Soit c une fonction de coût éventuellement adaptative. Tout ensemble K-trivial A admet une approximation $\Delta_2^0(A_s)_{s \in \mathbb{N}}$ qui satisfait c .

On peut alors utiliser cela pour montrer une plusieurs propriétés intéressantes, par exemple le fait que tout K-trivial admet une approximation avec peu de changements.

Théorème 2.7 (Nies [166])

Tout ensemble K-trivial admet une approximation Δ_2^0 telle que chaque préfixe de taille n change au plus $n^2 \times d$ fois pour une certaine constante d .

PREUVE. Par la proposition 16-2.5, il existe un d tel que $K(n) \leq 2\log_2(n) + d$ pour tout n . On peut considérer sans perte de généralité que, pour tout s , on a $K(n)[s] \leq 2\log_2(n) + d$. Il suffit alors d'utiliser la fonction de coût donnée par $c_s(n) = 2^{-K(n)[s]}$. Soit $(A_s)_{s \in \mathbb{N}}$ une approximation de A qui satisfait cette fonction de coût et soit e la somme totale du coût qui est perdu. À chaque fois qu'un préfixe de A de taille n change à une étape s , cela coûte au moins $2^{-K(n)[s]} \geq 2^{-2\log_2(n) - d} = n^{-2} \times 2^{-d}$. Cela ne peut donc pas arriver plus de $n^2 \times 2^d \times e$ fois sans faire passer le coût au-dessus de e . ■

Il est également possible d'utiliser la technique des fonctions de coût pour montrer que tout ensemble K-trivial est borné dans les degrés Turing par un K-trivial c. e. On définit pour cela l'ensemble c. e. C tel que C énumère $\langle n, i \rangle$ à l'étape s si le préfixe de taille n de A change pour la i -ième fois à l'étape s . On vérifie sans peine que $C \geq_T A$. En effet, pour déterminer le segment initial de A de longueur n , il suffit de chercher le plus grand i tel que $\langle n, i \rangle \in C$, et d'exécuter l'approximation de A jusqu'à ce que $A_s \upharpoonright_n$ ait changé i fois. On montre alors que si une approximation de A satisfait la fonction de coût donnée par $c(n) = 2^{-K(n)}$, alors l'approximation de C doit aussi satisfaire cette fonction de coût, ce que l'on peut utiliser pour montrer que C est lui aussi K-trivial. Nous verrons une preuve alternative de ce fait avec le théorème 4.4.

3. Caractérisation des K-triviaux c. e.

Nous continuons ici notre étude des K-triviaux avec la présentation d'une question qui resta ouverte un certain temps, et dont l'étude amena de nombreux développements, qui entre autres choses permirent de donner une autre preuve du fait que tout K-trivial est low-pour-K.

3.1. La question

La question qui vient de l'étude des bases pour l'aléatoire s'est posée après la découverte de la proposition suivante et de son corollaire.

Proposition 3.1. Soit A un ensemble c.e., et soit Z un ensemble MLR mais non MLR(A). Alors, $Z \oplus A \geq_T \emptyset'$. ★

PREUVE. Soit $\bigcap_n \mathcal{U}_n^A$ un A -test de Martin-Löf tel que $Z \in \bigcap_n \mathcal{U}_n^A$. Notons que l'on peut supposer d'après le théorème 19-2.3 que $\bigcap_n \mathcal{U}_n^X$ est un test de Martin-Löf pour tout ensemble X . On décrit à présent $(\mathcal{V}_n)_{n \in \mathbb{N}}$ des classes Σ_1^0 utilisées pour créer un tel test. Si n est énuméré dans $\emptyset'$ à l'étape t , on énumère alors $\mathcal{U}_n^{A_t}[t]$ dans $\mathcal{V}_n$. Si n n'est jamais énuméré dans $\emptyset'$, alors $\mathcal{V}_n$ reste vide. Comme Z est MLR, il existe donc n tel que $Z \notin \bigcup_{m \geq n} \mathcal{V}_m$. On peut à présent décider si un entier $m \geq n$ appartient à $\emptyset'$ à l'aide de $Z \oplus A$ de la manière suivante. Soit t le plus petit temps de calcul tel que $Z \in \mathcal{U}_m^A[t]$, et soit ρ le préfixe de A tel que $\mathcal{U}_m^\rho[t] = \mathcal{U}_m^A[t]$. On cherche alors t' le plus petit temps de calcul tel que $A_{t'} \upharpoonright_{|\rho|} = A \upharpoonright_{|\rho|}$. On a alors $m \in \emptyset'$ ssi $m \in \emptyset'[\max(t, t')]$. En effet, si m rentre dans $\emptyset'$ à un temps de calcul $s > \max(t, t')$, alors on aura toujours $A_s \upharpoonright_{|\rho|} = A \upharpoonright_{|\rho|}$, car A est c.e., et l'on a donc $Z \in \mathcal{U}_m^\rho[s] \subseteq \mathcal{U}_m^{A_s}[s] = \mathcal{V}_m$, ce qui contredit $Z \notin \mathcal{V}_m$. ■

Corollaire 3.2 (Hirschfeldt, Nies and Stephan [90])

Soit Z un ensemble MLR incomplet, et soit A un ensemble c.e. tel que $Z \geq_T A$. Alors, A est K-trivial.

PREUVE. Puisque $Z \geq_T A$ et que Z est incomplet, alors $Z \oplus A \not\geq_T \emptyset'$. Donc, d'après la proposition 3.1, Z est MLR(A), et A est alors une base pour l'aléatoire. D'après le théorème 1.7, A est donc low-pour-K, et dès lors K-trivial. ■

Qu'en est-il de la réciproque du corollaire 3.2? Les ensembles K-triviaux c.e. sont-ils nécessairement calculables par un ensemble MLR incomplet? Cette question a fait l'objet de nombreux travaux et développements, pour ne citer que [13] [17] [18] [19] [43], aboutissant à une réponse positive, permettant au passage, conjointement avec d'autres résultats, de donner une preuve alternative du fait que les ensembles K-triviaux sont, comme il se doit, tous low-pour-K.

Exercice 3.3. (★) Montrer que le corollaire 3.2 ne peut pas être étendu aux ensembles Δ_2^0 . ◇

3.2. L'aléatoire d'Oberwolfach

Oberwolfach est une petite commune de quelques milliers d'habitants, perdue au cœur de la Forêt noire du Pays de Bade, en Allemagne. C'est ici qu'émergera durant la Seconde Guerre mondiale un centre de recherche en mathématiques, qui deviendra l'un des plus célèbres et des plus importants au monde. Toute l'année, des mathématiciens de tous les pays s'y retrouvent pour des conférences. Le calme de l'endroit et de sa nature environnante en font un lieu propice à la concentration et au travail mathématique. De ce fait, en plus de ses conférences, le centre de recherche d'Oberwolfach organise en parallèle des programmes de « research in pairs » : un petit groupe de deux à cinq mathématiciens s'y retrouve pour quelques semaines afin de travailler sur un sujet précis. C'est ainsi qu'en 2012, Laurent Bienvenu (chargé de recherche au CNRS), Noam Greenberg (professeur à l'université de Wellington), Antonín Kučera (professeur à l'université de Prague), André Nies (professeur à l'université d'Auckland) et Dan Turetsky (post-doctorant au centre de recherche Kurt Gödel à Vienne) se retrouvent à Oberwolfach pour s'attaquer à la question de la réciproque du corollaire 3.2.

Si leurs travaux n'aboutiront pas tout de suite à la résolution de cette question-là, ils constitueront néanmoins une avancée fondamentale, via le concept qu'ils baptisèrent en l'honneur du lieu qui les accueillait : *l'aléatoire d'Oberwolfach*. Nous avons vu avec le théorème 19-1.5 que si X est MLR mais non fortement MLR, alors X calcule un ensemble c.e. non calculable. Nous avons vu avec le théorème 19-4.2 que si X est MLR mais non aléatoire de différence, alors X calcule le problème de l'arrêt. Étant donné un K-trivial c.e. A , l'idée est alors de rechercher une notion d'aléatoire qui se situerait entre aléatoire de différence et aléatoire de Martin-Löf fort, telle que tous les MLR qui ne sont pas aléatoires pour cette notion calculeraient forcément A .

Définition 3.4 (BGKNT [15])

Un *test d'Oberwolfach* est donné par $\bigcap_n \mathcal{U}_n$, une classe Π_2^0 , et un réel positif approchable par la gauche $r < 1$ tels que $\lambda(\mathcal{U}_n) \leq r - r_n$, où $(r_n)_{n \in \mathbb{N}}$ est l'approximation de r . Un ensemble Z est *aléatoire au sens d'Oberwolfach* si Z n'appartient à aucun test d'Oberwolfach. $\diamond$

La condition de convergence de la mesure des ouverts vers 0 est donc relaxée dans un test d'Oberwolfach par rapport à un test de Martin-Löf. En effet, si r n'est pas calculable, alors la borne $r - r_n$ ne le sera pas non plus. Il y a malgré tout une restriction sur la convergence des ouverts vers 0, et il est de fait possible de montrer que cette notion d'aléatoire est strictement plus faible que l'aléatoire de Martin-Löf fort, pour lequel il n'y a aucune restriction. Nous verrons avec le théorème 3.7 que tout ensemble aléatoire au

sens d'Oberwolfach est aléatoire de différence. Le théorème suivant montre que l'objectif de l'aléatoire d'Oberwolfach est rempli d'un seul coup pour tous les K-triviaux.

Théorème 3.5 (BGKNT [15])

Soit Z un ensemble MLR et non aléatoire au sens d'Oberwolfach. Alors, Z calcule tous les K-triviaux c.e.

Nous avons besoin pour montrer le théorème 3.5 d'un lemme qui au premier abord peut sembler sibyllin, mais qui prend tout son sens quand on le place dans le contexte des fonctions de coût, que l'on a vues avec la définition 2.4. Le lemme suivant nous dit que pour tout réel r approchable par la gauche, tout K-trivial c.e. admet une approximation c.e. qui satisfait la fonction de coût calculable donnée par $c(s) = r_s - r_{s-1}$. Il s'agit bien entendu d'une conséquence du théorème 2.6, mais que l'on peut montrer ici directement.

Lemme 3.6 (BGKNT [15])

Soit A un K-trivial c.e. avec une constante d . Soit $r < 1$ un réel positif approchable par la gauche avec son approximation $(r_s)_{s \in \mathbb{N}}$. Alors, il existe une approximation $(A_s)_{s \in \mathbb{N}}$ de A telle que la somme

$$\sum_{s \in S} r_s - r_{x_s} \text{ est finie,}$$

où $S = \{s \in \mathbb{N} : A_s \neq A_{s+1}\}$ et où $x_s < s$ est le plus petit entier tel que $A_{s+1} \upharpoonright_{x_s} \neq A_s \upharpoonright_{x_s}$ ★

PREUVE. Soit la fonction

$$f : \mathbb{N} \rightarrow \mathbb{N}, \quad f(s) = -\log_2(r_{s+1} - r_s).$$

On a en particulier $\sum_s 2^{-f(s)} = r < 1$. D'après le théorème 16-3.1, il existe une constante c telle que $2^{-f(s)} \leq 2^{-K(s)} \times 2^c$, pour tout s . Comme A est K-trivial avec constante d , on peut donc accélérer son approximation $(A_s)_{s \in \mathbb{N}}$ afin d'avoir $K(A_s \upharpoonright_t)[s] \leq f(t) + c + d$, pour tout s et tout $t \leq s$. On a alors, pour $s \in S$,

$$r_s - r_{x_s} = \sum_{x_s \leq t < s} 2^{-f(s)} \leq \sum_{x_s \leq t < s} 2^{-K(A_s \upharpoonright_t)} \times 2^{c+d}$$

Comme A est c.e., alors si $A_s \upharpoonright_{x_s} \neq A_{s+1} \upharpoonright_{x_s}$ on a aussi $A_s \upharpoonright_{x_s} \neq A_{s'} \upharpoonright_{x_s}$, pour tout $s' > s+1$. En particulier, on a $A_s \upharpoonright_{x_s} \neq A_{s'} \upharpoonright_{x_{s'}}$ pour s, s' distincts appartenant à S , et donc

$$\sum_{s \in S} r_s - r_{x_s} \leq \sum_{s \in S} \sum_{x_s \leq t < s} 2^{-K(A_s \upharpoonright_t)} \times 2^{c+d} \leq \sum_{U(\tau) \downarrow} 2^{-|\tau|} \times 2^{c+d} \leq \Omega \times 2^{c+d} \leq 2^{c+d}.$$

Cela conclut la preuve. ■

PREUVE DU THÉORÈME 3.5. Soit Z un ensemble MLR et capturé par un test d'Oberwolfach $\bigcap_n \mathcal{U}_n$, avec $\lambda(\mathcal{U}_n) \leq r - r_n$ pour un réel positif $r < 1$ approchable par la gauche. Soit A un K-trivial c.e. En utilisant le lemme 3.6, on peut supposer que l'énumération $(A_s)_{s \in \mathbb{N}}$ de A est telle que $\sum_{s \in S} r_s - r_{x_s}$ est finie, où $S = \{s \in \mathbb{N} : A_s \neq A_{s+1}\}$ et où pour $s \in S$ l'entier x_s est le plus petit tel que $A_{s+1} \upharpoonright_{x_s} \neq A_s \upharpoonright_{x_s}$. On accélère l'énumération de chaque ouvert $\mathcal{U}_n$ de manière à avoir $\lambda(\mathcal{U}_n[s]) \leq r_s - r_n$ pour tout s . On définit la fonctionnelle Φ par $\Phi(X, n) = A_s(n)$, pour tout $X \in \mathcal{U}_{n+1}[s+1] \setminus \mathcal{U}_{n+1}[s]$. Pour voir que l'on a bien $\Phi(Z, n) = A(n)$, on définit le test de Solovay $\mathcal{V}_s = \mathcal{U}_{x_s}[s]$, pour tout $s \in S$. Si $s \notin S$, alors $\mathcal{V}_s$ reste vide. Notons que l'on a $\sum_{s \in S} \lambda(\mathcal{V}_s) = \sum_{s \in S} \lambda(\mathcal{U}_{x_s}[s]) \leq \sum_{s \in S} r_s - r_{x_s}$. Donc, $\sum_{s \in S} \lambda(\mathcal{V}_s)$ est une quantité finie, et $(\mathcal{V}_s)_{s \in \mathbb{N}}$ est bien un test de Solovay.

Comme $Z \in \bigcap_n \mathcal{U}_n$, alors la fonctionnelle est bien totale sur l'oracle Z . Supposons que $\Phi(Z, n) \neq A(n)$ pour un entier n fixé. Soit s tel que

$$Z \in \mathcal{U}_{n+1}[s+1] - \mathcal{U}_{n+1}[s].$$

Alors, il doit exister $t > s$ tel que $A_t(n) \neq A_{t+1}(n)$. Soit $x_t \leq n+1$ le plus petit tel que $A_t \upharpoonright_{x_t} \neq A_{t+1} \upharpoonright_{x_t}$. On a $\mathcal{V}_t = \mathcal{U}_{x_t}[t] \supseteq \mathcal{U}_{n+1}[s+1]$, et donc $Z \in \mathcal{V}_t$. Comme Z est MLR, cela ne peut pas arriver une infinité de fois. Donc, $\Phi(Z, n) = A(n)$ pour n suffisamment grand. ■

Les protagonistes de l'aléatoire d'Oberwolfach ont également montré que leur notion était la meilleure possible pour l'objectif donné : il est possible de construire un K-trivial tel que tout ensemble qui le calcule est capturé par un test d'Oberwolfach. Aucun ensemble aléatoire au sens d'Oberwolfach ne peut donc calculer tous les K-triviaux, et il existe en particulier des K-triviaux plus dégourdis que les autres, dits « K-triviaux intelligents », tels que tout aléatoire qui les calcule peut alors calculer tous les K-triviaux.

3.3. Résolution de la question

Nous sommes à présent proches de la résolution de la question de savoir si tout K-trivial c.e. est calculé par un aléatoire incomplet. Il s'agit de construire un ensemble qui passe tous les tests de différence — afin d'être incomplet — mais qui ne soit pas aléatoire au sens d'Oberwolfach. Une telle construction s'avère toutefois fort complexe, et n'a pu être achevée directement. La solution viendra à la place d'un détour inattendu par le théorème de densité de Lebesgue. Souvenons-nous qu'un ensemble MLR est incomplet si, et seulement si, c'est un point de densité positive dans toute classe Π_1^0 qui le contient. Bienvenu, Greenberg, Kučera, Nies et Turetsky ont également montré durant leur séjour à Oberwolfach que la notion d'aléatoire qu'ils y ont définie est suffisante pour satisfaire le théorème de densité de Lebesgue pour toute classe Π_1^0 .

Théorème 3.7 (BGKNT [15])

Soit Z un ensemble aléatoire au sens d'Oberwolfach. Alors, $\underline{\rho}(\mathcal{P} \mid Z) = 1$ pour tout $\mathcal{P}$ classe Π_1^0 contenant Z .

PREUVE. Soit $\mathcal{P}$ une classe Π_1^0 contenant Z . Supposons $\underline{\rho}(\mathcal{P} \mid Z) < 1$, par l'absurde. Soit $\alpha < 1$ tel que $\underline{\rho}(\mathcal{P} \mid Z) < \alpha$. Soit T l'arbre calculable dont les chemins infinis sont les éléments de $\mathcal{P}$. On définit

$$U_n = \{\sigma \in T : |\sigma| \geq n \text{ et } \lambda(\mathcal{P} \mid [\sigma]) < \alpha\}.$$

Chaque ouvert $\mathcal{U}_n = [U_n]$ est bien Σ_1^0 , et il est clair que $Z \in \bigcap_n \mathcal{U}_n$. Nous prétendons que $\bigcap_n \mathcal{U}_n$ est un test d'Oberwolfach. On se doit de trouver pour cela un réel approchable par la gauche r tel que $\lambda(\mathcal{U}_n) \leq r - r_n$.

Soit $\mathcal{P}[n]$ l'approximation de $\mathcal{P}$ donnée par l'ouvert-fermé constitué des chaînes $\sigma \in T$ de taille n . Il est clair que l'on a $\mathcal{U}_n \subseteq \mathcal{P}[n]$. Cherchons à présent une borne pour la mesure de $\mathcal{U}_n \subseteq \mathcal{P}[n]$. On a

$$\lambda(\mathcal{P} \setminus \mathcal{U}_n) \leq \lambda(\mathcal{P}[n] \setminus \mathcal{U}_n) = \lambda(\mathcal{P}[n]) - \lambda(\mathcal{U}_n).$$

Par ailleurs, les éléments de $\mathcal{P}$ sont soit dans $\mathcal{U}_n$, soit dans $\mathcal{P} \setminus \mathcal{U}_n$. Par suite, $\lambda(\mathcal{P}) = \lambda(\mathcal{P} \cap \mathcal{U}_n) + \lambda(\mathcal{P} \setminus \mathcal{U}_n)$. De plus, si $\sigma \in U_n$, alors $\lambda(\mathcal{P} \mid [\sigma]) < \alpha$. On a donc $\lambda(\mathcal{P} \cap \mathcal{U}_n) \leq \alpha \lambda(\mathcal{U}_n)$. Cela donne

$$\lambda(\mathcal{P}) = \lambda(\mathcal{P} \cap \mathcal{U}_n) + \lambda(\mathcal{P} \setminus \mathcal{U}_n) \leq \alpha \lambda(\mathcal{U}_n) + \lambda(\mathcal{P}[n]) - \lambda(\mathcal{U}_n).$$

On a alors $\lambda(\mathcal{P}) - \lambda(\mathcal{P}[n]) \leq (\alpha - 1)\lambda(\mathcal{U}_n)$, et donc

$$(1 - \alpha)\lambda(\mathcal{U}_n) \leq \lambda(\mathcal{P}[n]) - \lambda(\mathcal{P}) = \lambda(\mathcal{P}^c) - \lambda(\mathcal{P}[n]^c),$$

où $\mathcal{P}^c$ et $\lambda(\mathcal{P}[n]^c)$ désignent respectivement les complémentaires des ensembles $\mathcal{P}$ et $\mathcal{P}[n]$. Ainsi,

$$\lambda(\mathcal{U}_n) \leq \frac{1}{1 - \alpha} (\lambda(\mathcal{P}^c) - \lambda(\mathcal{P}[n]^c)).$$

Notre réel approchable par la gauche sera donc $r = \frac{1}{1 - \alpha} \lambda(\mathcal{P}^c)$, via l'approximation $r_n = \left(\frac{1}{1 - \alpha} \lambda(\mathcal{P}[n]^c) \right)_{n \in \mathbb{N}}$. Il est bien entendu possible que ce réel soit plus grand que 1, auquel cas il suffit de commencer l'approximation à partir de n suffisamment grand tel que $r - r_n < 1$. ■

Corollaire 3.8

Soit $X \in 2^{\mathbb{N}}$. Alors, X fortement MLR implique X aléatoire au sens d'Oberwolfach implique X différence aléatoire.

PREUVE. Il est clair que les tests d'Oberwolfach sont tous des classes Π_2^0 de mesure 0. Donc, si X est fortement MLR, il est aléatoire au sens d'Oberwolfach. À présent, si X est aléatoire au sens d'Oberwolfach, il est de densité 1, et donc de densité positive dans toute classe Π_1^0 de mesure positive. D'après le théorème 19-4.10, il est donc différence aléatoire. ■

Il suffirait donc pour résoudre la question de construire un aléatoire de Martin-Löf qui soit un point de densité positive dans toute classe Π_1^0 , et qui ne soit pas un point de densité 1 pour au moins une classe Π_1^0 , afin d'exhiber un aléatoire de Martin-Löf incomplet calculant tous les K-triviaux. C'est ce qui fut fait par Day et Miller, qui apporteront à l'aide d'un forcing approprié, un point final à la question.

Théorème 3.9 (Day et Miller [43])

Il existe un ensemble MLR $Z \in 2^{\mathbb{N}}$ tel que :

- (1) $\rho(\mathcal{P} \mid Z) > 0$, pour tout $\mathcal{P}$ classe Π_1^0 ;
- (2) $\rho(\mathcal{Q} \mid Z) < 1$, pour $\mathcal{Q}$ une classe Π_1^0 spécifique.

PREUVE. Soit $\mathcal{P}$ une classe Π_1^0 ne contenant que des ensembles MLR. On définit les conditions de forcing $\mathbb{P}$ par $\langle \sigma, \mathcal{Q} \rangle \in \mathbb{P}$ si :

- (1) $\sigma \in 2^{<\mathbb{N}}$;
- (2) $\mathcal{Q} \subseteq \mathcal{P}$ est une classe Π_1^0 ;
- (3) $[\sigma] \cap \mathcal{Q} \neq \emptyset$ (et donc $\lambda(\mathcal{Q} \mid [\sigma]) > 0$, car $\mathcal{Q}$ ne contient que des MLR) ;
- (4) Il existe $\delta < 1/2$ tel que pour tous $\tau \succeq \sigma$, si $[\tau] \cap \mathcal{Q}$ est non vide, alors $\lambda(\mathcal{P} \mid [\tau]) \leq \lambda(\mathcal{Q} \mid [\tau]) + \delta$.

On dit que $\langle \tau, \mathcal{R} \rangle$ étend $\langle \sigma, \mathcal{Q} \rangle$ si $\tau \succeq \sigma$ et $\mathcal{R} \subseteq \mathcal{Q}$. Notons que $(\epsilon, \mathcal{P})$, où ϵ est la chaîne vide, est une condition avec $\delta = 0$. De même, si $(\sigma, \mathcal{Q})$ est une condition satisfaisant (4) pour un δ , alors pour tout $\tau \succeq \sigma$ tel que $[\tau] \cap \mathcal{Q} \neq \emptyset$, l'extension $(\tau, \mathcal{Q})$ de $(\sigma, \mathcal{Q})$ satisfait (4) pour le même δ . Ainsi, tout filtre suffisamment générique contiendra des conditions $(\sigma, \mathcal{Q})$, avec $|\sigma|$ de longueur arbitraire.

Soit $G \subseteq \mathbb{P}$ un filtre suffisamment générique. Soit X_G l'unique point limite des σ_i pour $\langle \sigma_i, \mathcal{Q}_i \rangle \in G$. Comme $\mathcal{P}$ est une classe fermée, il est clair que $X_G \in \mathcal{P}$, et donc X_G est un ensemble MLR.

Montrons à présent que si l'ensemble G est suffisamment générique, on a alors $\rho(\mathcal{P} \mid X_G) < 1/2$. Soit $\langle \sigma, \mathcal{Q} \rangle$ une condition de forcing avec δ satisfaisant la condition (4). Soit X le chemin le plus à gauche de $\mathcal{Q} \cap [\sigma]$. La classe $\mathcal{Q} \cap [\sigma] \subseteq \mathcal{P}$ est une classe ne contenant que des ensembles MLR, si bien que X est un ensemble MLR. D'après l'exercice 18-1.3, les ensembles MLR contiennent des suites de 1 arbitrairement grandes. Soit m tel que $2^{-m} + \delta < 1/2$, et soit τ tel que $\sigma \prec \tau 1^m \prec X$. Comme X est le chemin le plus à gauche de $\mathcal{Q} \cap [\sigma]$, on a $\lambda(\mathcal{Q} \mid [\tau]) \leq 2^{-m}$, et par conséquent $\lambda(\mathcal{P} \mid [\tau]) \leq 2^{-m} + \delta \leq 1/2$. La condition de forcing $\langle \tau, \mathcal{Q} \rangle$ étend bien $\langle \sigma, \mathcal{Q} \rangle$. Dès lors, si le filtre G est suffisamment générique, on aura $\lambda(\mathcal{P} \mid X_G \restriction_m) < 1/2$ pour une infinité d'entiers m , et donc $\rho(\mathcal{P} \mid X_G) < 1/2$.

Montrons finalement que si G est suffisamment générique, alors pour toute $\mathcal{S}$ classe Π_1^0 on a $X_G \in \mathcal{S}$ implique $\rho(\mathcal{S} \mid X_G) > 0$. Soit $\langle \sigma, \mathcal{Q} \rangle$ une condition de forcing, avec δ satisfaisant la condition (4). S'il existe $\tau \succeq \sigma$ telle que $\mathcal{Q} \cap [\tau] \neq \emptyset$ et telle que $\mathcal{S} \cap [\tau] = \emptyset$, on prend $\langle \tau, \mathcal{Q} \rangle$ comme extension de forcing, et l'on aura alors $X_G \notin \mathcal{S}$. Sinon, cela signifie $[\sigma] \cap \mathcal{Q} \subseteq \mathcal{S}$. Soit $\varepsilon < \min(1/2 - \delta, \lambda(\mathcal{Q} \mid [\sigma]))$. Considérons la classe

$$\mathcal{Q}' = \{X \in \mathcal{Q} \cap [\sigma] : \forall n \geq |\sigma| \lambda(\mathcal{Q} \mid [X \upharpoonright_n]) \geq \varepsilon\}.$$

Il est clair que $\rho(\mathcal{Q} \mid X) \geq \varepsilon$, pour tout $X \in \mathcal{Q}'$. Comme $[\sigma] \cap \mathcal{Q} \subseteq \mathcal{S}$, alors également $\rho(\mathcal{S} \mid X) \geq \varepsilon$, pour tout $X \in \mathcal{Q}'$. Il nous reste dès lors à montrer que $\langle \mathcal{Q}', \sigma \rangle$ est une extension valide, pour laquelle on aura par conséquent $\rho(\mathcal{S} \mid X_G) \geq \varepsilon$.

Montrons d'abord que la classe $\mathcal{Q}' \cap [\sigma]$ est non vide. Par le choix de ε , on a $\lambda(\mathcal{Q} \mid [\sigma]) \geq \varepsilon$. Il doit donc exister d'après le lemme 19-4.7 une suite infinie $Y \succ \sigma$, avec $Y \in \mathcal{Q}$ telle que $\lambda(\mathcal{Q} \mid [Y \upharpoonright_n]) \geq \varepsilon$ pour tout $n \geq |\sigma|$. Donc, $\mathcal{Q}' \cap [\sigma]$ est non vide. Soit à présent une chaîne $\tau \succeq \sigma$ telle que $\mathcal{Q}' \cap [\tau]$ est non vide. Montrons que l'on a alors $\lambda(\mathcal{P} \mid [\tau]) \leq \lambda(\mathcal{Q}' \mid [\tau]) + \delta + \varepsilon$ faisant de $\delta + \varepsilon < 1/2$ le rationnel satisfaisant la condition (4). Soit

$$\mathcal{U} = \{X \succeq \sigma : \exists n \lambda(\mathcal{Q} \mid [X \upharpoonright_n]) < \varepsilon\}.$$

Comme $\mathcal{Q}' \cap [\tau]$ est non vide, on a $\lambda(\mathcal{Q} \mid [\tau \upharpoonright_n]) \geq \varepsilon$, pour tout entier n avec $|\sigma| < n \leq |\tau|$, et donc $[\tau] \not\subseteq \mathcal{U}$. Soit W l'ensemble minimal de chaînes sans préfixes décrivant $\mathcal{U}$. Comme $[\tau] \not\subseteq \mathcal{U}$, on a alors

$$\lambda(\mathcal{Q} \cap \mathcal{U} \cap [\tau]) = \sum_{\rho \in W, \rho \succ \tau} \lambda(\mathcal{Q} \cap [\rho]).$$

Par ailleurs, pour toute chaîne $\rho \in W$ on a $\lambda(\mathcal{Q} \mid [\rho]) < \varepsilon$ par définition de $\mathcal{U}$. Donc,

$$\lambda(\mathcal{Q} \cap \mathcal{U} \cap [\tau]) \leq \varepsilon \sum_{\rho \in W, \rho \succ \tau} 2^{-|\rho|} \leq \varepsilon 2^{-|\tau|}.$$

Comme on a $\mathcal{Q}' \cap [\tau] = \mathcal{Q} \cap [\tau] \setminus (\mathcal{Q} \cap \mathcal{U} \cap [\tau])$, alors

$$\lambda(\mathcal{Q}' \cap [\tau]) = \lambda(\mathcal{Q} \cap [\tau]) - \lambda(\mathcal{Q} \cap \mathcal{U} \cap [\tau]),$$

et donc $\lambda(\mathcal{Q}' \cap [\tau]) \geq \lambda(\mathcal{Q} \cap [\tau]) - \varepsilon 2^{-|\tau|}$. Ainsi, $\lambda(\mathcal{Q}' \mid [\tau]) \geq \lambda(\mathcal{Q} \mid [\tau]) - \varepsilon$. Comme $\mathcal{Q}' \cap [\tau]$ est non vide, alors aussi $\mathcal{Q} \cap [\tau]$ est non vide, et l'on a donc $\lambda(\mathcal{P} \mid [\tau]) \leq \lambda(\mathcal{Q} \mid [\tau]) + \delta \leq \lambda(\mathcal{Q}' \mid [\tau]) + \delta + \varepsilon$. Donc, $\langle \sigma, \mathcal{Q}' \rangle$ est une condition valide pour $\delta + \varepsilon < 1/2$. ■

Corollaire 3.10 (BDGKMNT [13])

Tout ensemble K-trivial c.e. est calculable par un MLR incomplet.

PREUVE. Il suffit, d'après le théorème 3.9, de considérer un MLR Z qui est un point de densité inférieure positive sans être un point de densité

inférieur 1. D'après le théorème 19-4.10, un tel ensemble ne peut pas calculer $\emptyset'$. D'après le théorème 3.7, un tel ensemble ne peut pas être aléatoire au sens d'Oberwolfach, car il serait sinon un point de densité inférieure 1. D'après le théorème 3.5, cet ensemble Z calcule donc tous les K-triviaux c. e. ■

4. Une nouvelle preuve de K-trivial implique low-pour-K

Nous montrons à présent comment combiner les résultats vus jusqu'ici pour montrer que tout K-trivial c. e. est low-pour-K, sans utiliser la preuve du processus d'or. Soit A un ensemble c. e. K-trivial. Alors, d'après le corollaire 3.10, A est calculé par un MLR incomplet Z . Supposons par l'absurde que Z n'est pas $\text{MLR}(A)$. Alors, d'après la proposition 3.1, on doit avoir $A \oplus Z \geq \emptyset'$, et donc $Z \geq \emptyset'$, ce qui contredit le fait que Z soit incomplet. Ainsi, Z est $\text{MLR}(A)$, et A est une base pour l'aléatoire. D'après le théorème 1.7, A est donc low-pour-K.

Il ne nous reste plus qu'à montrer que tout K-trivial est calculé par un K-trivial c. e., et ce sans utiliser non plus la preuve du processus d'or. Comme tout K-trivial c. e. est low-pour-K et que les low-pour-K sont clos par le bas dans les degrés Turing, on en déduit facilement que tout K-trivial est aussi low-pour-K. Nous faisons appel pour cela à une utilisation astucieuse d'un concept introduit par Solovay : des fonctions de compressions calculables qui font aussi bien que K infiniment souvent.

Définition 4.1. Une *fonction de Solovay* est une fonction $g : \mathbb{N} \rightarrow \mathbb{N}$ calculable telle que $K(n) \leq^+ g(n)$ pour tout n , et telle que $g(n) \leq^+ K(n)$ pour une infinité d'entiers n . ◇

Proposition 4.2. Il existe une fonction de Solovay. ★

PREUVE. Soit $g_S : \mathbb{N} \rightarrow \mathbb{N}$ telle que $g_S(\langle \sigma, n, t \rangle) = |\sigma|$ si t est le plus petit temps de calcul tel que $U(\sigma)[t] \downarrow = n$. Sinon, $g_S(\langle \sigma, n, t \rangle) = 2\langle \sigma, n, t \rangle$. Comme $K(\langle \sigma, n, t \rangle) \leq^+ 2\langle \sigma, n, t \rangle$, on a bien $K(n) \leq^+ g_S(n)$ pour tout n . Montrons que l'on a $K(n) =^+ g_S(n)$ pour une infinité d'entiers n .

Soit M la machine sans préfixe qui sur σ tel que t est le plus petit pour lequel $U(\sigma)[t] \downarrow = n$, renvoie $\langle \sigma, n, t \rangle$. Via M , on a $K(\langle \sigma, n, t \rangle) \leq^+ |\sigma|$. Comme on peut calculer n à partir de $\langle \sigma, n, t \rangle$, on a également $K(n) \leq^+ K(\langle \sigma, n, t \rangle)$. Enfin, si σ est la plus petite chaîne telle que $U(\sigma) \downarrow = n$, on a par définition $|\sigma| = K(n)$.

On a donc dans ce cas $|\sigma| = K(n) \leq^+ K(\langle \sigma, n, t \rangle) \leq^+ |\sigma|$, d'où il résulte que $g_S(\langle \sigma, n, t \rangle) =^+ K(\langle \sigma, n, t \rangle)$. ■

Bienvenu et Downey ont montré que la fonction définie par Solovay pouvait être utilisée pour caractériser la K-trivialité.

Proposition 4.3 (Bienvenu et Downey, [14]). Soit g_S la fonction de Solovay de la proposition 4.2. Si $K(A \upharpoonright_n) \leq g_S(n) + c$ pour tout n , alors A est K-trivial via une certaine constante $c + d$. ★

PREUVE. Pour tout n , soit σ_n la plus petite chaîne telle que $U(\sigma_n) = n$, et soit t_n le plus petit temps de calcul tel que $U(\sigma_n)[t_n] = n$. On a

$$K(A \upharpoonright_{\langle \sigma_n, n, t_n \rangle}) \leq g_S(\langle \sigma_n, n, t_n \rangle) + c = |\sigma_n| + c = K(n) + c.$$

Comme on peut calculer $A \upharpoonright_n$ à partir de $A \upharpoonright_{\langle \sigma_n, n, t_n \rangle}$, on a une constante d telle que $K(A \upharpoonright_n) \leq K(A \upharpoonright_{\langle \sigma_n, n, t_n \rangle}) + d$, et donc telle que $K(A \upharpoonright_n) \leq K(n) + c + d$. ■

Plus tard, Bienvenu, Merkle et Nies [20] montreront que la proposition précédente fonctionne pour toutes les fonctions de Solovay, mais il s'agit d'une preuve bien plus difficile. Nous avons à présent tous les ingrédients nécessaires pour montrer que tout K-trivial est borné dans les degrés Turing par un K-trivial c. e., résultat qui fut démontré d'abord par Nies à partir de la technique des fonctions de coût et de la preuve du processus d'or, et qui peut se montrer à présent de manière complètement différente.

Théorème 4.4 (Nies [166])

Tout K-trivial est calculé par un K-trivial c. e.

PREUVE. Soit $g_S : \mathbb{N} \rightarrow \mathbb{N}$ la fonction de Solovay de la proposition 4.2. Soit $T = \{\sigma \in 2^{<\mathbb{N}} : \forall n < |\sigma| \ K(X \upharpoonright_n) \leq g_S(n) + c\}$ l'arbre c. e. qui représente la classe $\Pi_1^0(\emptyset')$ de l'ensemble $\{X : K(X \upharpoonright_n) \leq g_S(n) + c\}$. Il est clair que $[T]$ contient tous les K-triviaux avec constante c , et d'après la proposition 4.3 $[T]$ ne contient que des K-triviaux avec constante $c + d$, et donc en particulier ne contient qu'un nombre fini d'éléments. Soit A l'un de ces K-triviaux, et soit $\sigma \prec A$ tel que A et le seul chemin infini de T qui étend σ . Soit $T \upharpoonright_\sigma$ l'arbre T restreint aux chaînes comparables avec σ . On modifie l'énumération de $T \upharpoonright_\sigma$ de la manière suivante : étant donné $T_s \upharpoonright_\sigma$ à l'étape s , on énumère dans $T_s \upharpoonright_\sigma$ une nouvelle chaîne $\tau \succeq \sigma$ uniquement si celle-ci est de taille supérieure ou égale à toutes les chaînes actuellement dans $T_s \upharpoonright_\sigma$ — auquel cas on énumère également tous les préfixes de τ dans $T_s \upharpoonright_\sigma$. Soit T_σ l'arbre c. e. résultant de cette énumération.

Notons que comme T contient des extensions de σ de taille arbitrairement grande, alors notre nouvelle énumération T_σ contient de même des extensions de σ de taille arbitrairement grande. En particulier, par le lemme de König, T_σ contient un chemin infini. Comme A est l'unique chemin infini de T qui étend σ , ce chemin infini est dès lors forcément A . Comme A est l'unique chemin infini de T_σ , alors T_σ permet de calculer A .

Montrons que T_σ est K-trivial via sa représentation X_σ où $X_\sigma(n) = 1$ si, et seulement si, n correspond à une chaîne énumérée dans T_σ . Notons que $X_\sigma \upharpoonright_{2^n}$ correspond à toutes les chaînes de T_σ de taille inférieure ou égale à n . Montrons $K(X_\sigma \upharpoonright_{2^n}) \leq^+ g_S(n)$ pour tout n . Pour tout n , soit σ_n la dernière chaîne de taille égale à n qui est énumérée dans T .

Notons que σ_n permet de calculer uniformément $X_\sigma \upharpoonright_{2^n}$: il suffit d'attendre que σ_n soit énumérée dans T_σ , et de renvoyer alors l'ensemble des chaînes de taille inférieure ou égale à $|\sigma_n|$ énumérées dans T_σ jusqu'ici. Par la propriété d'énumération de T_σ , on sait qu'aucune autre chaîne de taille inférieure ou égale à n ne sera énumérée par la suite. Comme $\sigma_n \in T$, par définition de T , $K(\sigma_n) \leq g_S(n) + c$. Enfin, comme $X_\sigma \upharpoonright_n$ est calculable à partir de $X_\sigma \upharpoonright_{2^n}$, on a $K(X_\sigma \upharpoonright_n) \leq^+ K(X_\sigma \upharpoonright_{2^n})$. Cela nous donne

$$K(X_\sigma \upharpoonright_n) \leq^+ K(X_\sigma \upharpoonright_{2^n}) \leq^+ K(\sigma_n) \leq^+ g_S(n),$$

pour tout n . Donc, X_σ est K-trivial. ■

Schéma Récapitulatif

Voici un tableau qui récapitule les différentes notions d'aléatoire abordées jusqu'ici.

Aléatoire	Tests	Caractérisation	Densité dans les Π_1^0 le contenant
Fortement MLR	Π_2^0 de mesure 0	MLR et ne calcule aucun Δ_2^0 non calculable.	
Aléatoire d'Oberwolfach	$\bigcap_n \mathcal{U}_n$ tel que $\lambda(\mathcal{U}_n) < r - r_n$, avec $\lim_n r_n = r$	MLR et ne calcule pas tous les K-triviaux c. e.	densité inférieure 1
Aléatoire de différence	$\mathcal{F} \cap \bigcap_n \mathcal{U}_n$ tel que $\lambda(\mathcal{U}_n \cap \mathcal{F}) \leq 2^{-n}$	MLR incomplet	densité inférieure positive
MLR	$\bigcap_n \mathcal{U}_n$ tel que $\lambda(\mathcal{U}_n) < 2^{-n}$		densité supérieure 1

Troisième partie

Mathématiques à rebours

Chapitre 21

Introduction

Les mathématiques à rebours¹ sont un ensemble d'outils provenant de la théorie de la preuve et de la calculabilité pour analyser le contenu calculatoire des théorèmes mathématiques. Elles permettent de répondre à des questions méta-mathématiques fondamentales telles que : « Quels sont les axiomes optimaux pour prouver les théorèmes ordinaires ? », ou bien : « Quel sens donner à l'implication d'un théorème par un autre ? »

1. Quête des axiomes optimaux

La quête des axiomes optimaux pour prouver les théorèmes usuels est la motivation historique des mathématiques à rebours, telles que conçues par Harvey Friedman en 1975-1976. Comme nous l'avons vu dans le chapitre 9 sur la crise des fondements, les développements de la théorie des ensembles repoussant les limites de l'intuition, on a vu naître des paradoxes semant le doute dans la solidité de l'édifice des mathématiques. La crise des fondements a connu son paroxysme avec le second théorème d'incomplétude de Gödel, énonçant qu'une théorie calculatoirement énumérable cohérente capable de parler des entiers naturels,



Harvey Friedman, 1948–

1. « Reverse mathematics », en anglais.

ne pouvait prouver sa propre cohérence. Les mathématiciens sont donc condamnés à reléguer la cohérence des mathématiques au niveau de croyance, justifiée uniquement par l'absence de contradiction repérée à ce jour malgré un usage intensif des mathématiques au quotidien.

Les théorèmes peuvent donc être vus comme des édifices de connaissances construits sur la croyance en la cohérence d'un système d'axiomes. Plus la preuve d'un théorème fera appel à des axiomes forts, plus fragile sera cette connaissance. La démarche de Friedman consista donc à essayer de maîtriser ces risques, en déterminant quels théorèmes seraient invalidés si l'on devait renoncer à certains axiomes. En d'autres termes, il s'agissait de déterminer le niveau de fiabilité des théorèmes, en réponse directe à la crise des fondements. La question historique des mathématiques à rebours fut donc la suivante.

« Quels sont les *axiomes optimaux* permettant de prouver les théorèmes des *mathématiques ordinaires* ? »

Cette question comporte plusieurs composants importants sur lesquels il convient de s'arrêter.

Tout d'abord, qu'entend-on par « mathématiques ordinaires » ? Il est important de comprendre que la démarche de Friedman était avant tout empirique et ne recherchait pas l'exhaustivité : il ne s'agit pas d'être capable d'analyser la puissance axiomatique des tout derniers résultats de théorie des ensembles, qui est une branche méta-mathématique à puissance logique très éloignée des mathématiques traditionnelles. Il s'agit plutôt d'étudier les mathématiques « de la vie de tous les jours », comme les théorèmes usuels d'algèbre ou d'analyse.

Comment prouver qu'un ensemble d'axiomes est optimal pour prouver un théorème ? Analyser avec soin la preuve du théorème pour en identifier les hypothèses ou les axiomes ne suffit pas. Il peut très bien exister une nouvelle preuve faisant appel à des axiomes plus élémentaires. La notion d'axiome optimal n'est donc pas une propriété relative à une preuve, mais à un théorème. Soient $A_1, \dots, A_n$ des axiomes *suffisants* pour prouver un théorème T . Autrement dit, $A_1 \wedge \dots \wedge A_n \rightarrow T$. Pour s'assurer que les axiomes sont *nécessaires*, il suffit de prouver l'implication inverse (d'où le nom de « mathématiques à rebours »). On se retrouve alors avec l'équivalence

$$A_1 \wedge \dots \wedge A_n \leftrightarrow T$$

1.1. Choix d'une théorie de base

La démarche serait probablement vouée à l'échec si l'on se restreignait à l'implication logique sans fixer une théorie de base. L'implication logique est en effet une relation très fine qui placerait chaque théorème — et même différentes formulations d'un même théorème — dans des « degrés

logiques » différents. Il est naturel de chercher à s'abstraire des détails de formulation et de travailler modulo des opérations « élémentaires » : nos implications logiques pourront s'établir relativement à une théorie de base permettant de réaliser ces opérations élémentaires. Il convient d'être prudent dans le choix de cette théorie : si celle-ci est trop forte — par exemple si elle permet déjà de prouver la plupart des théorèmes —, la valeur informative d'une implication $P \rightarrow Q$, relativement à cette théorie, sera affaiblie.

C'est là qu'intervient la calculabilité : nous allons fixer une théorie de base, RCA_0 , capturant les *mathématiques calculables*. Le choix de cette théorie définit l'interprétation sémantique de la relation d'implication. Par exemple, si l'on considère que la théorie de Zermelo-Fraenkel (ZF) représente les mathématiques consensuelles, c'est-à-dire les mathématiques largement acceptées par la communauté, une implication $\text{ZF} \vdash P \rightarrow Q$ signifie que consensuellement, si l'on admet P , alors on admettra Q . De la même manière, une preuve de $\text{RCA}_0 \vdash P \rightarrow Q$ signifie que si l'on fait confiance aux mathématiques calculables, et si l'on admet P , alors il est logique de considérer Q comme vrai. En pratique, la théorie RCA_0 est beaucoup plus faible que ZF, et en particulier beaucoup de théorèmes classiques ne sont pas prouvables dans RCA_0 , comme nous allons le voir.

1.2. Arithmétique du second ordre

Pour mener à bien le programme historique des mathématiques à rebours, à savoir la quête des axiomes optimaux, il faut fixer un cadre formel, à commencer par le choix d'un langage. À première vue, le choix le plus évident est celui d'une théorie fondationnelle comme celle de la théorie des ensembles. En effet, comme nous l'avons vu dans la section 9-4, le langage ensembliste permet une formalisation unifiée de la totalité des mathématiques. Le choix des mathématiques à rebours s'est cependant porté sur le langage de l'arithmétique du second ordre, c'est-à-dire un langage où l'on manipule et quantifie sur les entiers naturels et les ensembles d'entiers.

Ce choix peut paraître surprenant : les ensembles d'entiers naturels étant par nature dénombrables, l'arithmétique du second ordre ne peut manipuler que des objets dénombrables. Comment alors parler des objets usuels qui ne le sont pas, comme les fonctions de $\mathbb{R}$ dans $\mathbb{R}$? Ce n'est de fait pas possible en toute généralité, mais les travaux de Hilbert et Bernays, *Grundlagen der Mathematik* (1934-1936), ont montré qu'une grande partie des mathématiques pouvait être formalisée, modulo un codage approprié, dans l'arithmétique du second ordre. Ainsi, par exemple les fonctions *continues* — et même boréliennes — de $\mathbb{R}$ dans $\mathbb{R}$ admettent toutes une représentation dénombrable. Rappelons que le but des mathématiques à rebours n'est pas d'être exhaustif, mais de rendre compte d'une tendance générale des mathématiques.

En contrepartie de cette légère perte de généralité, l'arithmétique du second ordre présente un avantage considérable : les objets mathématiques manipulés étant tous représentés par des entiers et des ensembles d'entiers, nous pouvons bénéficier des outils de la calculabilité pour parler de la complexité des axiomes. Il existe en effet une notion robuste de calculabilité sur les entiers, et notamment une caractérisation des ensembles calculables comme ceux définissables par des prédicats Δ_1^0 . Les différentes représentations d'un même théorème dans l'arithmétique du second ordre à l'aide de différentes fonctions de codage n'auront en pratique pas d'impact sur la force du théorème, car la plupart des codages sont calculables, et donc équivalents dans la théorie de base RCA_0 .

1.3. Phénomène de structure

Depuis le lancement des mathématiques à rebours, des centaines de théorèmes ont été analysés, provenant de toutes les branches des mathématiques. On retrouvera cette démarche dans l'excellent ouvrage de Steven Simpson, *Subsystems of Second-Order Arithmetics*. L'étude systématique des théorèmes classiques sous l'angle des mathématiques à rebours a laissé entrevoir deux observations empiriques :

La plupart des théorèmes ordinaires requièrent une puissance axiomatique faible. Cette observation peut être vue comme une réponse partielle à la crise des fondements, en délivrant un message d'optimisme : oui, la théorie de l'arithmétique est incomplète et nous devons nous contenter d'une croyance expérimentale en sa cohérence, oui, l'ajout d'axiomes à cette théorie ne fait potentiellement qu'empirer les risques, mais la plupart des théorèmes usuels ne requièrent qu'une partie « faible » de ces axiomes supplémentaires. Les mathématiques sont donc robustes face à d'éventuelles incohérences dues à des axiomes trop forts.

Les mathématiques sont calculatoirement très structurées. Plus précisément, il existe quatre grands systèmes d'axiomes, linéairement ordonnés par l'implication modulo RCA_0 , tels que si l'on prend un théorème classique au hasard², il y a de fortes chances pour qu'il soit équivalent à l'un des quatre systèmes modulo RCA_0 , ou bien même déjà prouvable dans RCA_0 . On appelle cette observation le *phénomène du Club des cinq* (Big Five, en anglais).

Cette seconde observation est cependant à relativiser. Il existe des contre-exemples — provenant notamment de la théorie de Ramsey — se comportant de manière beaucoup plus chaotique. Cela a conduit certains chercheurs à soulever l'objection selon laquelle le phénomène de structure observé en mathématiques à rebours relève d'un biais humain, et que cette structure serait davantage celle du cerveau des mathématiciens que des mathématiques elles-mêmes.

2. « Hasard » est à prendre au sens informel du terme. Il s'agit d'une observation empirique et non d'un résultat de probabilités.

2. Comparaison des théorèmes

Les mathématiques à rebours ont petit à petit évolué, et de nombreuses branches sont nées sur le terreau originel de la recherche de l'optimalité des axiomes. Une grande partie de la discipline consiste en la comparaison et en la classification des théorèmes. Il est courant en mathématiques d'entendre des énoncés comme « les théorèmes A et B sont équivalents », ou encore « le théorème A n'est pas une conséquence du théorème B ». Les mathématiques à rebours permettent de donner un sens précis à ces affirmations informelles.

D'un point de vue purement logique, tous les théorèmes sont équivalents, au sens où ils sont tous interprétés par la valeur de vérité **vrai**. Quel sens donner à l'intuition de l'implication ou de l'équivalence entre deux théorèmes ? On pourrait par exemple considérer qu'un théorème T_0 implique un autre théorème T_1 si la preuve de T_1 fait intervenir l'énoncé T_0 . Cependant, il est possible de remplacer chaque occurrence de T_0 par sa preuve, dans la preuve de T_1 pour obtenir une nouvelle démonstration ne faisant pas intervenir T_0 . Cette tentative de formalisation n'est donc pas la bonne.

Si l'on en revient à l'intuition première, un théorème T_0 implique un théorème T_1 si T_1 peut être prouvé *de manière élémentaire*, en faisant appel à T_0 comme une boîte noire. Les mathématiques à rebours permettent justement de formaliser la notion de raisonnement élémentaire à l'aide du système RCA_0 . Il est alors possible de donner un sens formel à l'implication $T_0 \rightarrow T_1$ en la prouvant dans RCA_0 . Les mathématiques à rebours, initialement conçues pour identifier les axiomes nécessaires aux mathématiques, deviennent un outil de classification de théorèmes.

Exemple 2.1. On trouvera dans la littérature des affirmations comme « Le théorème des valeurs intermédiaires est une conséquence de la propriété de la borne supérieure ». Une version faible du théorème des valeurs intermédiaires — le théorème de Bolzano — affirme que si une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ est continue sur un intervalle $[a, b]$ avec $f(a) < 0$ et $f(b) > 0$, alors il existe $c \in]a, b[$ tel que $f(c) = 0$. La *propriété de la borne supérieure* affirme que toute partie de $\mathbb{R}$ non vide et majorée admet une borne supérieure.

Supposons la propriété de la borne supérieure afin de montrer le théorème de Bolzano. Soit $f : \mathbb{R} \rightarrow \mathbb{R}$ une fonction continue telle que $f(a) < 0$ et $f(b) > 0$. Soit $S = \{x \in [a, b] : f(x) < 0\}$. L'ensemble S contient a et est majoré par b , donc il admet une borne supérieure c .

En supposant par l'absurde $f(c) \neq 0$, alors pour un intervalle ouvert I tel que $f(c) \in I$ avec $\max I < 0$, on a par continuité de f un intervalle ouvert J contenant c tel que $f(J) \subseteq I$.

L'inégalité $c < \sup J$ contredit la qualité de borne supérieure à c . Par conséquent, $f(c) = 0$.

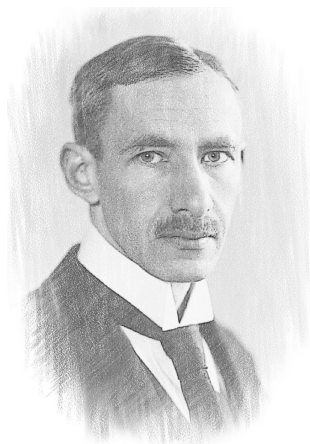
L'exemple précédent est une démonstration « élémentaire » (RCA_0 donnera un sens précis à cela) du théorème de Bolzano à partir de la propriété de la borne supérieure. Du point de vue des mathématiques à rebours, ces théorèmes doivent se formaliser dans le langage de l'arithmétique du second ordre. Une fonction continue pourra être représentée par un encodage de tous les ouverts $f^{-1}(]a, b[)$, pour des rationnels $a < b$ quelconques. La propriété de borne supérieure ne peut pas se formaliser pour des ensembles de réels quelconques, mais on pourra la formaliser pour toutes les classes boréliennes, qui peuvent se coder par des objets dénombrables.

L'intérêt de l'implication dans RCA_0 pour classifier les théorèmes reste cependant relativement limité, dans la mesure où la plupart des théorèmes sont équivalents à cinq grands ensembles d'axiomes. Il existe toutefois des raffinements de l'implication où l'on va contrôler les utilisations de T_0 dans la preuve de T_1 , ou même interdire les analyses de cas. Ces restrictions aboutissent respectivement à la *réduction calculatoire* et la *réduction de Weihrauch*, que nous verrons en détail dans cette partie. De nos jours, les mathématiques à rebours ont pris un sens plus large, et forment une bannière sous laquelle se regroupent la démarche fondationnelle de recherche des axiomes optimaux et la classification des théorèmes à travers différentes réductions calculatoires.

Chapitre 22

Arithmétique du second ordre

L'arithmétique du second ordre, notée Z_2 , est une théorie introduite par Hilbert et Bernays dans *Grundlagen der Mathematik*, permettant de parler des entiers et des ensembles d'entiers. Bien que la théorie ne manipule que des objets dénombrables, Hilbert et Bernays ont montré qu'une grande partie des mathématiques classiques était déjà prouvable dans Z_2 . Les axiomes de l'arithmétique du second ordre qui sont relatifs aux entiers (et ne parlent pas des ensembles d'entiers) coïncident avec ceux de l'arithmétique de Peano. La théorie Z_2 n'en est pas pour autant une *extension conservative* de PA (voir la définition 23-6.1) : il existe des formules ne faisant intervenir que des entiers naturels qui sont prouvables dans Z_2 mais pas dans PA, à commencer par l'énoncé de la cohérence de PA.



Paul Bernays, 1888–1977

Dans ce chapitre, nous allons étudier plusieurs sous-systèmes de l'arithmétique du second ordre qui se sont révélés particulièrement importants en mathématiques à rebours.

1. Langage de Z_2

Le langage de l'arithmétique du second ordre correspond à celui de l'arithmétique de Peano, augmenté de variables sur les ensembles d'entiers, de quantificateurs sur ces ensembles, et de la relation binaire d'appartenance entre un entier et un ensemble.

Définition 1.1. Le langage $\mathcal{L}_{Z_2}$ de l'arithmétique du second ordre comprend :

- (1) des symboles de *variables du premier ordre* $x, y, z, \dots$ pour les entiers naturels ;
- (2) des symboles de *variables du second ordre* $X, Y, Z, \dots$ pour les ensembles d'entiers ;
- (3) les symboles de *connecteurs logiques* : $\wedge, \vee, \rightarrow, \neg$ et les parenthèses $()$;
- (4) les symboles de *quantificateurs* sur les entiers et sur les ensembles d'entiers : $\forall, \exists$;
- (5) les symboles de *fonctions binaires* sur les entiers suivantes : $+, \times$;
- (6) les symboles de *relations binaires* sur les entiers suivants : $=, <$;
- (7) le symbole de relation d'appartenance : $\in$;
- (8) des symboles de *constantes* $0, 1$. ◇

Le langage de l'arithmétique du second ordre est dit *à deux sortes*, au sens où il manipule deux types d'objets distincts : les entiers (premier ordre), et les ensembles d'entiers (second ordre). On distingue donc deux types de termes : les termes du premier ordre et ceux du second ordre.

Définition 1.2. Les *termes du premier ordre* de Z_2 sont définis inductivement de la manière suivante.

- (1) Une variable du premier ordre ou une constante est un terme du premier ordre.
- (2) Si t_1, t_2 sont des termes du premier ordre, alors $(t_1 + t_2)$ et $(t_1 \times t_2)$ en sont aussi.

Les *termes du second ordre* sont simplement les variables du second ordre. ◇

La simplicité des termes du second ordre provient de l'absence de fonctions sur les ensembles d'entiers dans le langage de l'arithmétique du second ordre. Si l'on avait ajouté le symbole de réunion $\cup$, on aurait alors considéré $(X \cup Y) \cup Z$ comme un terme du second ordre. Les opérations ensemblistes standard pourront toutefois être définies à l'aide de certains axiomes.

Exemple 1.3. Les expressions suivantes sont des termes du premier ordre : x , $(((((x + \dot{1}) + \dot{1}) + \dot{1}) \times \dot{1}) + \dot{1}), (\dot{1} + \dot{0}), (x + (y \times z))$. Les seuls termes du second ordre sont les variables du second ordre.

Comme pour l'arithmétique de Peano, si les symboles de fonction sont utilisés pour créer les termes du langage, les symboles de relation sont eux utilisés pour créer les *formules* du langage.

Définition 1.4. Les *formules de l'arithmétique du second ordre* sont définies de la manière suivante.

- (1) Pour tous termes du premier ordre t_1, t_2 et tout terme du second ordre X , alors $t_1 = t_2$, $t_1 < t_2$, et $t_1 \in X$ sont des formules. Ces formules sont appelées *formules atomiques*.
- (2) Pour toutes formules F_1, F_2 , alors $(F_1 \wedge F_2), (F_1 \vee F_2), (F_1 \rightarrow F_2)$ et $\neg F_1$ sont des formules.
- (3) Pour toute formule F , alors $\forall xF$, $\exists xF$, $\forall XF$ et $\exists XF$ sont des formules. ◇

Exemple 1.5. La formule

$$\forall X ([\dot{0} \in X \wedge \forall y (y \in X \rightarrow y + \dot{1} \in X)] \rightarrow \forall z z \in X)$$

exprime l'induction.

Une formule est *close* si elle ne possède aucune variable libre. Sauf spécifié autrement, on ne supposera pas par défaut qu'une formule est close.

Formules et variables libres

Souvenons-nous de la notation $F(x)$ signifiant que x est libre dans F . Afin de ne pas alourdir certains énoncés, on ne rendra pas toujours explicites toutes les variables libres de F . Ce sera par exemple le cas pour des formules $F(x) \equiv G(X_1, \dots, X_s, a_1, \dots, a_t, x)$ où $X_1, \dots, X_s$ et $a_1, \dots, a_t$ sont des variables libres, respectivement du second et du premier ordre, qui auront vocation à être remplacées par des *paramètres* dans une certaine structure.

Rappelons que les formules de l'arithmétique du second ordre peuvent être hiérarchisées en fonction de leurs alternances de quantificateurs. En particulier, une formule *arithmétique* est une formule de $\mathcal{L}_{Z_2}$ où seuls les quantificateurs sur les entiers sont autorisés, bien que la formule puisse contenir des variables libres du premier ou du second ordre (voir la section 10-1).

2. La théorie Z_2

Venons-en aux axiomes de l'arithmétique du second ordre. On y retrouve les axiomes de l'arithmétique de Robinson, pour régir le comportement des entiers naturels et des opérations usuelles (voir la section 9-2.3). Nous les redonnons ici à des fins de complétude :

(1) $\forall x \neg(x + \dot{1} = \dot{0})$	(5) $\forall x \forall y (x + (y + \dot{1}) = (x + y) + \dot{1})$
(2) $\forall x (x = \dot{0} \vee \exists y (x = y + \dot{1}))$	(6) $\forall x (x \times \dot{0} = \dot{0})$
(3) $\forall x \forall y (x + \dot{1} = y + \dot{1} \rightarrow x = y)$	(7) $\forall x \forall y (x \times (y + \dot{1}) = (x \times y) + x)$
(4) $\forall x (x + \dot{0} = x)$	(8) $\forall x \forall y$ $(x < y \leftrightarrow (\exists z (z \neq \dot{0} \wedge x + z = y)))$

Rappelons que l'on note Q la théorie de l'arithmétique de Robinson constituée des axiomes (1)-(8), ci-dessus.

2.1. Schéma de compréhension

Jusqu'ici, nous n'avons spécifié que le comportement des entiers naturels. Nous allons maintenant définir un schéma d'axiomes qui nous permettra de « construire » des ensembles, autrement dit de s'assurer que ces ensembles existeront dans tout modèle. Nous avons vu dans la définition des axiomes de la théorie des ensembles (voir la section 9-4) le schéma de compréhension restreint. Nous allons ajouter un schéma similaire pour toute formule (avec variables libres) de l'arithmétique du second ordre $F(x)$:

$$\exists X \forall y (y \in X \leftrightarrow F(y)). \quad (9)$$

On supposera que la variable X n'apparaît pas librement dans la formule F .

Clôture universelle des variables libres

Le schéma de compréhension s'applique à *toute formule avec variables libres* : toute formule $F(x) \equiv G(X_1, \dots, X_s, a_1, \dots, a_t, x)$, où $X_1, \dots, X_s$ et $a_1, \dots, a_t$ sont des variables libres, respectivement du second et du premier ordre. L'idée est alors que le schéma de compréhension doit être vrai quelles que soient les valeurs possibles des variables libres. Pour être tout à fait formel, on devrait exprimer (9) sous la forme

$$\forall Y_1 \dots \forall Y_s \forall a_1 \dots \forall a_t \exists X \forall y (y \in X \leftrightarrow G(Y_1, \dots, Y_s, a_1, \dots, a_t, y)).$$

Afin de garder un peu de légèreté dans nos notations, il sera entendu dans la suite que l'on ne note pas toujours explicitement les variables libres, qui doivent alors être quantifiées universellement si nécessaire.

Le schéma de compréhension ajoute des ensembles calculatoirement très complexes aux modèles de l'arithmétique du second ordre. On y retrouve

par exemple toute la hiérarchie arithmétique, à commencer par le problème de l'arrêt.

Exemple 2.1. Le prédicat $\Phi_z(X, z) \downarrow$ correspond à une formule $\Sigma_1^0(X)$ de l'arithmétique. Donc, l'axiome de compréhension appliqué à la formule $F(G, z) \equiv \Phi_z(G, z) \downarrow$ avec un ensemble que l'on suppose existant X :

$$\exists Y \forall z (z \in Y \leftrightarrow \Phi_z(X, z) \downarrow)$$

nous garantit l'existence de X' .

Notons que les formules F peuvent aussi contenir des quantifications du second ordre. Cela nous donne un pouvoir définitionnel considérable, que nous aborderons dans la partie **IV**. À l'aide de quantifications du second ordre, il sera par exemple possible de montrer l'existence de l' ω -saut Turing défini par $\emptyset^{(\omega)} = \bigoplus_n \emptyset^{(n)}$. Notons que ces quantifications du second ordre permettent de faire des définitions auto-référentes, au sens où la formule $F(x)$ peut contenir un quantificateur universel sur les ensembles d'entiers, et donc qui prendra notamment comme valeur l'ensemble que l'on est en train de définir. La théorie de l'arithmétique du second ordre est donc un système foncièrement *imprédicatif* au sens de Poincaré (voir la section **9-1** et l'exemple **8.5**).

2.2. Schéma d'induction

Rappelons que pour définir l'arithmétique de Peano, nous avons ajouté le schéma d'induction pour toute formule arithmétique. La tentation serait d'étendre ce schéma d'axiomes à toutes les formules de l'arithmétique du second ordre $F(x)$ avec des variables libres, à savoir :

$$(F(\dot{0}) \wedge (\forall x (F(x) \rightarrow F(x + \dot{1})))) \rightarrow \forall x F(x). \quad (10)$$

Nous allons cependant utiliser un axiome qui, combiné au schéma de compréhension, implique le schéma d'induction (10) pour toute formule du second ordre.

$$\forall X ([\dot{0} \in X \wedge \forall y (y \in X \rightarrow y + \dot{1} \in X)] \rightarrow \forall z z \in X). \quad (11)$$

Notons que (11) n'est pas un schéma, mais un simple axiome, dans la mesure où il n'est pas paramétré par une formule.

Exercice 2.2. Montrer que l'axiome d'induction sur les ensembles (11) et le schéma de compréhension (9) impliquent le schéma d'induction (10) sur les formules de l'arithmétique du second ordre. $\diamond$

Notation

On note Z_2 la théorie de l'arithmétique du second ordre, composée de Q (les axiomes de Robinson (1)-(8), ci-dessus), du schéma de compréhension (9) et de l'axiome d'induction (11) sur les ensembles.

Comme expliqué précédemment, malgré la simplicité de ses axiomes, Hilbert et Bernays ont montré qu'une grande partie des mathématiques traditionnelles était déjà formalisable dans Z_2 .

3. Sémantiques de l'arithmétique du second ordre

Le modèle intentionnel de l'arithmétique du second ordre est bien entendu $(\mathbb{N}, \mathcal{P}(\mathbb{N}), +, \times, <, 0, 1)$, c'est-à-dire les entiers naturels et ensembles d'entiers, munis des opérations standard.

3.1. Le second ordre

Il existe deux notions de structures pour interpréter l'arithmétique du second ordre : les structures de Henkin et les structures pleines. Dans toute sa généralité, une structure dans le langage de l'arithmétique du second ordre spécifie deux ensembles M et S représentant respectivement le premier et le second ordre, ainsi que des opérations $+$, $\times$ sur M , une relation d'ordre $<$ sur M , et une relation $\in$ entre M et S . Les structures de Henkin se restreignent aux cas où M et S sont disjoints et où $S \subseteq \mathcal{P}(M)$, avec $\in$ dénotant la vraie relation d'appartenance.

Définition 3.1. Une *structure (de Henkin)* dans $\mathcal{L}_{Z_2}$ est donnée par un tuple $\mathcal{M} = (M, S, +^{\mathcal{M}}, \times^{\mathcal{M}}, <^{\mathcal{M}}, 0^{\mathcal{M}}, 1^{\mathcal{M}})$, où M et S sont deux ensembles disjoints tels que $S \subseteq \mathcal{P}(M)$, $+^{\mathcal{M}}, \times^{\mathcal{M}} : M \times M \rightarrow M$ sont deux opérations sur M et $<^{\mathcal{M}} \subseteq M \times M$ est une relation sur M . Nous avons également la relation d'égalité sur les éléments de M , ainsi qu'un élément $0^{\mathcal{M}} \in M$ correspondant au symbole de constante $\bar{0}$ et un élément $1^{\mathcal{M}} \in M$ correspondant au symbole de constante $\bar{1}$. ◇

Notons qu'il n'y a pas de relation d'égalité sur les ensembles. Leon Henkin a prouvé que le théorème 9-2.22 de complétude était toujours valable dans les structures qui portent son nom : on peut essentiellement se ramener à des structures du premier ordre comme vues dans la section 9-2.4 et pour lesquelles la complétude est vérifiée. Les structures usuelles du premier ordre ne contiennent qu'un seul ensemble d'éléments E . Pour de telles structures, on peut ajouter des prédicats M et S permettant de

déterminer quels éléments de E sont du premier ou du second ordre, et des axiomes pour simuler le fait que l'on ait une structure de Henkin, par exemple $\forall x (M(x) \wedge \neg S(x)) \vee (\neg M(x) \wedge S(x))$ indique que tout élément est soit du premier ordre, soit du second ordre.

La sémantique standard associée à l'arithmétique du second ordre se restreint au cas particulier des structures de Henkin où le second ordre contient tous les sous-ensembles de M .

Définition 3.2. Une *structure pleine* dans $\mathcal{L}_{Z_2}$ est une structure de Henkin de la forme $\mathcal{M} = (M, \mathcal{P}(M), +^{\mathcal{M}}, \times^{\mathcal{M}}, <^{\mathcal{M}}, 0^{\mathcal{M}}, 1^{\mathcal{M}})$, c'est-à-dire où le second ordre est $\mathcal{P}(M)$ tout entier. $\diamond$

Dans une structure de Henkin avec M et $S \subseteq \mathcal{P}(M)$, une phrase du type $\forall X \Phi(X)$ sera vérifiée si elle l'est pour tous les éléments de S . Au contraire, dans une structure pleine, la quantification universelle se fait réellement sur tous les sous-ensembles de M . Nous verrons le genre de conséquence que cette distinction peut avoir dans le chapitre 31.

On appelle généralement *sémantique standard* ou *sémantique des modèles pleins* l'étude de l'arithmétique du second ordre où l'interprétation est restreinte aux structures pleines. La sémantique standard ne satisfait pas d'aussi bonnes propriétés que la sémantique plus générale de Henkin. En effet, il existe une formule qui fixe de manière unique ce que peut être l'ensemble M des éléments du premier ordre dans une structure pleine. Il s'agit simplement de la conjonction des axiomes de $\mathbf{Q}$ avec l'axiome d'induction (11), ci-dessus.

$$\forall X ((0 \in X \wedge \forall y (y \in X \rightarrow y + 1 \in X)) \rightarrow \forall z z \in X) \wedge \mathbf{Q}. \quad (\mathbf{a})$$

Soit $\mathcal{M} = (M, \mathcal{P}(M), +^{\mathcal{M}}, \times^{\mathcal{M}}, <^{\mathcal{M}}, 0^{\mathcal{M}}, 1^{\mathcal{M}})$ une structure satisfaisant (a). Soit $\omega = \{0^{\mathcal{M}}, 1^{\mathcal{M}}, 1^{\mathcal{M}} + 1^{\mathcal{M}}, 1^{\mathcal{M}} + 1^{\mathcal{M}} + 1^{\mathcal{M}}, \dots\}$. Notons que $\omega \subseteq M$. Il est possible de montrer que les axiomes de $\mathbf{Q}$ impliquent que

$$(\omega, +^{\mathcal{M}} \upharpoonright_{\omega}, \times^{\mathcal{M}} \upharpoonright_{\omega}, <^{\mathcal{M}} \upharpoonright_{\omega})$$

est nécessairement isomorphe aux entiers standard. L'axiome d'induction assure quant à lui que M ne possède aucun autre élément que ceux de ω : en prenant $X = \omega$, on obtient $\forall z z \in \omega$, donc $M = \omega$. Le modèle est alors forcément le modèle standard des entiers avec l'addition et la multiplication. Ainsi, le seul modèle plein de (a) est, à isomorphisme près, celui des entiers standard. Une théorie ne possédant qu'un seul modèle à isomorphisme près est dite *catégorique*. Une conséquence du corollaire 9-2.26 est que toute théorie catégorique est nécessairement complète. Le théorème 9-3.10 d'incomplétude de Gödel-Rosser affirme qu'il n'existe pas de théorie c.e. complète et cohérente qui étend PA. On en déduit que la théorie $\text{PA} + (\mathbf{a})$ n'est pas c.e., ce qui est absurde. Le problème dans ce

raisonnement est le corollaire 9-2.26 qui découle du théorème 9-2.22 de complétude de Gödel : ce dernier échoue pour la sémantique standard de l'arithmétique du second ordre.

Nous considérerons donc par défaut les structures de Henkin, que nous appellerons désormais tout simplement *structures*, et qui ne seront en général pas pleines¹.

3.2. Le premier ordre

Comme nous l'avons vu dans la section 9-3.3 la partie du premier ordre d'un modèle de Z_2 , c'est-à-dire « les entiers » du modèle, ne correspond pas nécessairement aux « vrais entiers ».

Dans n'importe quel modèle $\mathcal{M} = (M, S, +, \times, <, 0, 1)$ de Z_2 — nous avons supprimé l'exposant $\mathcal{M}$ pour plus de clarté — on aura nécessairement $\omega \subseteq M$ pour $\omega = \{0, 1, 1+1, \dots\}$. Comme mentionné dans la section précédente, en utilisant le fait que $(M, +, \times, <, 0, 1)$ satisfait les axiomes Q de l'arithmétique de Robinson, on construit aisément depuis l'extérieur de $\mathcal{M}$ un isomorphisme (pour l'ordre, l'addition et la multiplication) entre nos vrais entiers $\mathbb{N}$ et $\omega \subseteq M$. Il est parfaitement possible dans un tel modèle d'avoir $\omega \subsetneq M$, auquel cas ω est nécessairement un segment initial de M au sens de $<$ (voir le théorème 9-3.13).

Définition 3.3 (Entiers standard et entiers non standard)

Soit $\mathcal{M} = (M, S, +, \times, <, 0, 1)$ un modèle de la théorie Z_2 . Les éléments de $\omega = \{0, 1, 1+1, \dots\} \subseteq S$ sont appelés *entiers standard*, par opposition aux éléments de $M \setminus \omega$ qui sont les *entiers non standard*. Un modèle vérifiant $M \setminus \omega \neq \emptyset$ est lui-même qualifié de *non standard*. $\diamond$

Notation

Dans un modèle de Z_2 , les entiers standard seront simplement notés $0, 1, 2, \dots$. Cette notation se justifie par l'existence d'un isomorphisme entre $\mathbb{N}$ et la partie standard de notre modèle.

Comprendre les modèles non standard. Un modèle non standard de Z_2 contient par définition des éléments plus grands que nos entiers usuels $0, 1, 2, 3, \dots$, et dans le même temps vérifie bien tous les axiomes de Z_2 . Il peut être difficile au départ d'avoir les idées claires sur de tels objets, pour lesquels de nombreux points peuvent sembler paradoxaux. Voyons un exemple. Il devrait être parfaitement clair que toute suite d'entiers (non strictement) décroissante est constante à partir d'un certain rang.

1. Le théorème dit de Löwenheim-Skolem, bien connu en théorie des modèles, implique par exemple qu'il existe des modèles dénombrables de l'arithmétique du second ordre.

L'intuition immédiate que cet énoncé est vrai repose sur notre vision du modèle standard, mais peut bien entendu aussi se démontrer : tout ensemble d'entiers non vide possède un plus petit élément (nous verrons avec la proposition 23-3.4 qu'il s'agit d'une conséquence de l'induction). Soit a_i le plus petit élément d'une suite d'entiers $(a_n)_{n \in \mathbb{N}}$. Par hypothèse sur notre suite, les éléments suivants sont soit égaux soit strictement inférieurs à a_i . Par minimalité de a_i , ils ne peuvent lui être strictement inférieur. Ils lui sont donc tous égaux, et la suite est alors constante à partir du rang i .

Considérons à présent un modèle $\mathcal{M} = (M, S, +, \times, <, 0, 1)$ non standard de Z_2 et prenons un entier non standard $x \in M$. Comme

$$x > 1, x > 2, x > 3, \dots,$$

alors pour tout entier standard n l'élément $x - n$ est différent de 0. On peut donc utiliser l'axiome (2) de l'arithmétique de Robinson pour construire inductivement la suite décroissante $x, x - 1, x - 2, x - 3, \dots$, qui est aussi strictement décroissante. D'après le théorème de complétude et la preuve du paragraphe précédent, toute suite décroissante d'éléments dans tout modèle de Z_2 devrait pourtant être constante à partir d'un certain rang. Il semble donc que nous arrivions à une contradiction. Que s'est-il passé ? On peut effectivement définir la suite $a_0 = x$ pour $x \in M$ et $a_{n+1} = a_n - 1$ pour tout $n \in \mathbb{N}$. Cette suite-là n'est toutefois infinie (et même définie) que de l'extérieur du modèle. En effet, de l'intérieur du modèle il n'y a pas de distinction entre entiers standard et non standard : le modèle « pense » que tous ses éléments sont des entiers standard. Donnons-en une explication un peu plus précise. Une suite est une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$, et donc un objet du second ordre, dont l'existence est assurée par l'axiome de compréhension, qui doit se baser sur une formule du premier ordre $F(n, y)$ telle que pour tout $n \in M$ il existe exactement un élément y (c'est-à-dire l'élément a_n de notre suite) tel que $\mathcal{M} \models F(n, y)$. Dans une preuve destinée à être lue par les mathématiciens, la suite ci-dessus se définit via la fonction primitive récursive $f(0) = x$ et $f(n + 1) = \max(0, f(n) - 1)$. Il convient toutefois de formaliser cette définition dans le langage de l'arithmétique, pour lequel nous devons utiliser le codage de Gödel des fonctions primitives récursives par des formules de l'arithmétique. En reprenant la fonction $\beta(a, b, i)$ du lemme 9-3.6 permettant de coder les suites d'entiers, la formule sera la suivante (pour plus de clarté nous avons gardé les fonctions $\max$, β et — telles quelles plutôt que de leur substituer les formules qui les représentent) :

$$F(n, y) \equiv (n = 0 \wedge y = x) \vee \exists a, b \left(\begin{array}{l} y = \beta(a, b, n) \wedge \beta(a, b, 0) = x \wedge \forall i < n \\ \beta(a, b, i + 1) = \max(0, \beta(a, b, i) - 1) \end{array} \right).$$

Dans notre modèle de Z_2 , la formule F ci-dessus définira bien une fonction dont le domaine est l'ensemble M tout entier et non plus simplement les entiers standard.

La suite correspondante aura donc pour segment initial $a_0, a_1, a_2, a_3, \dots$ pour tous les entiers standard $0, 1, 2, 3, \dots$, et vérifiera effectivement $a_0 = x, a_1 = x - 1, a_2 = x - 2, a_3 = x - 3, \dots$, mais du point de vue du modèle, elle sera également définie pour tous les éléments de M . Ainsi, l'affirmation que $(a_y)_{y \in M}$ est constante à partir d'un certain rang sera bel et bien vérifiée, simplement elle sera constante à partir d'un rang non standard, qui du point de vue du modèle est un entier tout à fait valide. Notons pour finir que la partie standard de la suite $(a_y)_{y \in M}$, telle que définie initialement depuis l'extérieur du modèle n'est pas distinguable au sein du modèle. Nous verrons en effet dans la section 23-3.2 que l'axiome d'induction implique qu'il est impossible au sein d'un modèle de distinguer ses éléments non standard des autres.

Les ω -structures. Dans cette partie, nous nous intéresserons particulièrement à des modèles de fragments de l'arithmétique du second ordre, au sein desquels le premier ordre correspond aux entiers standard ω , munis des opérations usuelles.

Définition 3.4

Une ω -structure est une structure $\mathcal{M} = (\omega, S, +, \times, <, 0, 1)$ où ω est l'ensemble des entiers standard, $S \subseteq \mathcal{P}(\omega)$, $+$ et $\times$ sont les opérations d'addition et de multiplication usuelles, et $<$ est l'ordre naturel. Une ω -structure est donc entièrement spécifiée par l'ensemble S . $\diamond$

En particulier, $(\omega, \mathcal{P}(\omega), +, \times, <, 0, 1)$ est une ω -structure. On aura tendance à identifier un ensemble $S \subseteq \mathcal{P}(\omega)$ avec l' ω -structure dont S est le second ordre.

3.3. Formules à paramètres

Au cours de cette partie, nous aurons régulièrement recours à des formules à paramètres dans une structure. Intuitivement, une formule à paramètres dans une structure est une formule qui fait directement intervenir des éléments de la structure.

Définition 3.5

Soit $\mathcal{M} = (M, S, +^{\mathcal{M}}, \times^{\mathcal{M}}, <^{\mathcal{M}}, 0^{\mathcal{M}}, 1^{\mathcal{M}})$ une structure de l'arithmétique du second ordre. Une formule de $\mathcal{L}_{\mathbb{Z}_2}$ à paramètres dans $\mathcal{M}$ est une formule dans le langage $\mathcal{L}_{\mathbb{Z}_2} \cup \{c_n : n \in M\} \cup \{c_X : X \in S^{\mathcal{M}}\}$, c'est-à-dire une formule dans le langage de l'arithmétique du second ordre augmenté d'un symbole de constante pour chaque élément de la structure. $\diamond$

On aura tendance à identifier le symbole de constante et son interprétation dans la structure, et directement noter $F(n, X)$ pour $F(c_n, c_X)$ avec $n \in M$ et $X \in S^{\mathcal{M}}$.

Les paramètres sont étroitement liés aux variables libres des schémas d'axiomes. En effet, dans ces schémas, les variables libres sont quantifiées universellement, or une structure $\mathcal{M}$ satisfait une formule de la forme $\forall X F(X)$ ssi $\mathcal{M}$ satisfait la formule close $F(Z)$ pour chaque paramètre $Z \in \mathcal{M}$.

Exemple 3.6. Soit $F(X, y)$ une formule de l'arithmétique du second ordre, avec pour seules variables libres la variable d'ensemble X et la variable d'entier y . Soit $\mathcal{M} \models Z_2$. Par le schéma de compréhension, on a

$$\mathcal{M} \models \forall X \exists Y \forall z (z \in Y \leftrightarrow F(X, z)).$$

En particulier, pour tout paramètre $Z \in \mathcal{M}$, on a

$$\mathcal{M} \models \exists Y \forall z (z \in Y \leftrightarrow F(Z, z)).$$

Ici, Z est un paramètre et non une variable libre, et $F(Z, z)$ est une formule à paramètre.

La même nomenclature pour les variables libres et les paramètres se justifie par le fait que les premières auront normalement vocation à être remplacées par les deuxièmes dans les structures considérées.

4. Formaliser l'analyse dans Z_2

Comme mentionné précédemment, Hilbert et Bernays ont montré qu'une grande partie des mathématiques classiques pouvait se faire dans Z_2 . Nous montrons ici comment y formaliser quelques aspects d'analyse classique. Cela nous permettra par la suite d'illustrer la force de certains sous-systèmes axiomatiques de Z_2 par différents théorèmes d'analyse.

4.1. Les rationnels

Il est courant en calculabilité de manipuler des objets finis — comme les chaînes binaires — via un système de codage par des entiers. Nous utiliserons le codage suivant pour parler des rationnels.

Définition 4.1. Dans Z_2 , un rationnel est un entier de la forme $\langle i, n, m \rangle$ où $i \in \{0, 1\}$ et $m \neq 0$. Le rationnel correspondant est n/m si $i = 0$ et $-n/m$ si $i = 1$. ◇

Notons que nous avons plusieurs représentations possibles pour le même rationnel, ce qui n'est pas un problème, les points importants étant les suivants. En premier lieu, il existe une formule $F_{\mathbb{Q}}(x)$ de Z_2 qui est vraie ssi x est un rationnel (c'est-à-dire si x est un entier qui satisfait la définition ci-dessus). Pour x vérifiant $F_{\mathbb{Q}}$, notons $\iota_{\mathbb{Q}}(x)$ le rationnel représenté par x .

Les opérations usuelles de manipulation de rationnels doivent pouvoir s'exprimer dans Z_2 . Il nous faudra par exemple une formule $F_<$ de Z_2 telle que pour tout n_1, n_2 vérifiant $F_{\mathbb{Q}}$ on a $\iota_{\mathbb{Q}}(n_1) < \iota_{\mathbb{Q}}(n_2)$ ssi $F_<(n_1, n_2)$, ou encore une formule F_+ telle que pour tous n_1, n_2 vérifiant $F_{\mathbb{Q}}$, la formule $F_+(n_1, n_2, r)$ est vérifiée pour un unique entier r tel que $F_{\mathbb{Q}}(r)$ et tel que $\iota_{\mathbb{Q}}(r) = \iota_{\mathbb{Q}}(n_1) + \iota_{\mathbb{Q}}(n_2)$. Les conditions ci-dessus sur les formules $F_<$ et F_+ sont bien sûr exprimées de l'extérieur de Z_2 puisque $\iota_{\mathbb{Q}}$ n'est pas un objet de Z_2 . De l'intérieur de Z_2 , il faudra montrer que les formules permettant de définir les fonctions et relations usuelles sur $\mathbb{Q}$ en font un corps totalement ordonné et dense.

Il devrait être clair que les fonctions et relations usuelles sont primitives récursives sur l'ensemble des codes de rationnels. Nous verrons que cela implique automatiquement que les formules du type $F_<$ ou F_+ , ci-dessus, peuvent être définies dans Z_2 , et même dans le sous système RCA_0 , dont nous montrerons avec le théorème 23-4.6 qu'il est suffisant pour définir toute fonction primitive récursive via une formule de l'arithmétique.

Nous manipulerons comme habituellement nos rationnels sans recours explicite au codage, étant entendu que les différentes opérations se formalisent via ce codage dans le langage de l'arithmétique.

4.2. Nombres réels

Les nombres réels sont proches des ensembles d'entiers. On peut par exemple représenter un réel par un couple $\langle n, X \rangle$ avec $n \in \mathbb{N}$ et $X \in 2^{\mathbb{N}}$, où n indique la partie entière du réel et X sa partie fractionnaire. Ce genre de représentation ne va toutefois pas nous convenir, car elle ne permet pas d'étendre correctement les notions de calculabilité aux fonctions des réels sur les réels. L'exemple suivant est donné via la représentation des réels par leur développement décimal plutôt que binaire, afin d'en simplifier la présentation.

Exemple 4.2. À supposer qu'une fonctionnelle dont l'objectif est d'effectuer la multiplication par 3 lise les chiffres 0,333333..., elle doit se décider au bout d'un moment à sortir le premier chiffre du résultat de cette multiplication, mais comment savoir si l'on doit commencer par 0,999999... ou bien par 1,000000... ? Si la fonctionnelle opte pour la première possibilité, il se pourrait alors qu'un 4 survienne dans le développement décimal de notre entrée, rendant le début du calcul faux. Si la machine opte pour la deuxième possibilité, il se pourrait qu'un 2 survienne dans le développement décimal de notre entrée, avec la même conséquence...

L'écueil de l'exemple ci-dessus vient de la différence topologique entre $\mathbb{R}$ et $2^{\mathbb{N}}$, et notamment du fait que certains réels admettent plusieurs représentations possibles via leur développement décimal ou binaire. Afin de régler le problème, la solution communément utilisée est de travailler via la représentation d'un réel r par une suite de Cauchy de nombres rationnels de convergence suffisamment rapide. Formellement, cela s'exprime comme suit.

Définition 4.3. Dans Z_2 , un réel est une suite de rationnels $(q_n)_{n \in \mathbb{N}}$ telle que $\forall n \forall m |q_n - q_{n+m}| \leq 2^{-n}$. Le nombre $r \in \mathbb{R}$ correspondant à une telle suite est $\lim_{n \rightarrow +\infty} q_n$. $\diamond$

Il est clair que tout réel $r \in \mathbb{R}$ a une représentation dans Z_2 , et réciproquement que tout réel $\mathbb{R}$ du point de vue de Z_2 — c'est-à-dire toute suite $(q_n)_{n \in \mathbb{N}}$ respectant la définition ci-dessus — définit bien un unique réel.

Pourquoi se restreindre aux suites de Cauchy à convergence maîtrisée et ne pas considérer à la place des suites de Cauchy arbitraires ? c'est-à-dire telles que pour tout n il existe a pour lequel $|q_a - q_{a+b}| \leq 2^{-n}$ pour tout b . De telles suites sont elles aussi convergentes, et définissent aussi sans ambiguïté un unique réel. La raison est que nous aimerions garder calculables un certain nombre de relations, afin de minimiser les axiomes de Z_2 utilisés dans nos démonstrations. Si $(q_n)_{n \in \mathbb{N}}$ et $(p_n)_{n \in \mathbb{N}}$ sont deux suites de Cauchy arbitraires représentant des réels r_q et r_p que l'on suppose distincts, on ne peut pas décider de manière calculable, à partir des représentations de nos réels, si $r_p < r_q$ ou si $r_q < r_p$. En revanche, l'ordre devient décidable dès lors que l'on considère les suites de Cauchy de convergence maîtrisée. Notons que la relation $r_p = r_q$ reste dans tous les cas Π_1^0 : elle s'exprime par $\forall n |q_n - p_n| \leq 2^{-n+1}$.

4.3. Les ensembles de réels

Les ensembles de réels sont trop gros pour être décrits par des objets dénombrables. On a notamment $|\mathcal{P}(\mathbb{R})| > |2^{\mathbb{N}}|$. On peut en revanche indirectement parler de classes boréliennes dans Z_2 . Un ouvert $\mathcal{U} \subseteq \mathbb{R}$ peut toujours être exprimé comme une réunion d'intervalles rationnels. Pour des raisons techniques, nous considérerons l'ensemble $\mathbb{Q}^\infty = \mathbb{Q} \cup \{-\infty, +\infty\}$ plutôt que $\mathbb{Q}$. Via une bijection de $\mathbb{N}$ vers $\mathbb{Q}^\infty$, nous définissons une bijection calculable de $\mathbb{N}$ vers les intervalles ouverts de la forme $]a, b[$, pour $a, b \in \mathbb{Q}^\infty$ avec $a < b$. La bijection induit une liste $(I_n)_{n \in \mathbb{N}}$ de ces intervalles. Tout comme nous l'avons expliqué dans la section sur les rationnels, il est là encore entendu que la bijection choisie nous garantit que les opérations usuelles (comme la réunion ou l'intersection de deux intervalles) ainsi que les relations usuelles (comme l'inclusion d'un intervalle dans un autre) sont toutes primitives récursives, et donc exprimables dans Z_2 — et même dans RCA_0 — par le théorème 23-4.6 à venir.

Un ouvert $\mathcal{U} \subseteq \mathbb{R}$ sera alors représenté par un ensemble d'entiers $X \in 2^{\mathbb{N}}$ tel que $\mathcal{U} = \bigcup_{n \in X} I_n$. On définit de l'extérieur de $\mathbb{Z}_2$ la fonction ι_{Σ}^1 par $\iota_{\Sigma}^1(X)$ comme étant l'ouvert représenté par X . Un fermé étant simplement le complémentaire d'un ouvert, tout ensemble X représente également un fermé et l'on définit ι_{Π}^1 par $\iota_{\Pi}^1(X) = \mathbb{R} \setminus \iota_{\Sigma}^1(X)$. On peut continuer ainsi pour définir les boréliens. Une fois la représentation des classes Π_n^0 définie, une classe $\mathcal{B}$ qui est Σ_{n+1}^0 est représentée par $\bigoplus_m X_m$ tel que $\mathcal{B} = \bigcup_m \iota_{\Pi}^n(X_m)$. On a alors $\iota_{\Sigma}^{n+1}(\bigoplus_m X_m) = \mathcal{B}$ et $\iota_{\Pi}^{n+1}(\bigoplus_m X_m) = \mathbb{R} \setminus \iota_{\Sigma}^{n+1}(\bigoplus_m X_m)$.

4.4. Fonctions continues

Les fonctions $f : \mathbb{R} \rightarrow \mathbb{R}$ ne sont en général pas des objets dénombrables (en particulier $|\mathbb{R}^{\mathbb{R}}| > |2^{\mathbb{N}}|$). En revanche, les fonctions *continues* le sont, et l'on peut les représenter sans ambiguïté par un élément de $2^{\mathbb{N}}$.

Une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ est continue ssi pour tout ouvert $\mathcal{U}$, la classe $f^{-1}(\mathcal{U})$ est ouverte. Reprenons notre liste $(I_n)_{n \in \mathbb{N}}$ d'intervalles de la forme $]a, b[$ pour $a, b \in \mathbb{Q}^{\infty}$. Une fonction continue $f : \mathbb{R} \rightarrow \mathbb{R}$ sera alors représentée par une suite $((a_{n,m})_{m \in \mathbb{N}}, b_n)_{n \in \mathbb{N}}$ telle que $f^{-1}(I_{b_n}) = \bigcup_{m \in \mathbb{N}} I_{a_{n,m}}$ pour tout n . Notons qu'une suite $((a_{n,m})_{m \in \mathbb{N}}, b_n)_{n \in \mathbb{N}}$ arbitraire ne représente pas toujours une fonction continue. Il y a certaines des règles à vérifier.

Définition 4.4. Dans $\mathbb{Z}_2$, une fonction continue est une suite

$$((a_{n,m})_{m \in \mathbb{N}}, b_n)_{n \in \mathbb{N}}$$

telle que en considérant $f^{-1}(I_{b_n}) = \bigcup_{m \in \mathbb{N}} I_{a_{n,m}}$, pour tout n , on a :

- (1) $I_{n_1} \cap I_{n_2} = \emptyset \rightarrow f^{-1}(I_{n_1}) \cap f^{-1}(I_{n_2}) = \emptyset$.
- (2) Pour tout réel $(q_n)_{n \in \mathbb{N}}$ et tout n , il existe k suffisamment grand et un intervalle I_m avec $|I_m| \leq 2^{-n}$ tels que $]q_k - 2^{-k}, q_k + 2^{-k}[\subseteq f^{-1}(I_m)$.



La première condition de la définition ci-dessus nous garantit que la relation induite par une représentation $((a_{n,m})_{m \in \mathbb{N}}, b_n)_{n \in \mathbb{N}}$ est bien fonctionnelle, et la deuxième que la fonction ainsi définie est bien totale. Étant donné une fonction continue f représentée par $((a_{n,m})_{m \in \mathbb{N}}, b_n)_{n \in \mathbb{N}}$ et un réel r représenté par $(q_n)_{n \in \mathbb{N}}$, le réel $f(r)$ est défini sans ambiguïté par l'algorithme suivant.

▷ On cherche k_0 et $]x_0, y_0[$ tels que

$$|x_0 - y_0| \leq 1 \quad \text{et} \quad]q_{k_0} - 2^{-k_0}, q_{k_0} + 2^{-k_0}[\subseteq f^{-1}(]x_0, y_0[).$$

Par (2) de la définition ci-dessus, la recherche aboutit nécessairement.

▷ Supposons k_n et x_n, y_n déjà définis avec $|x_n - y_n| \leq 2^{-n}$.

Soit $m = (x_n + y_n)/2$; alors, on cherche $k_{n+1} > k_n$ tel que

- ▷ $]q_{k_{n+1}} - 2^{-k_{n+1}}, q_{k_{n+1}} + 2^{-k_{n+1}}[\subseteq f^{-1}(]x_n, m[)$ ou
- ▷ $]q_{k_{n+1}} - 2^{-k_{n+1}}, q_{k_{n+1}} + 2^{-k_{n+1}}[\subseteq f^{-1}(]m, y_n[)$ ou
- ▷ $]q_{k_{n+1}} - 2^{-k_{n+1}}, q_{k_{n+1}} + 2^{-k_{n+1}}[\subseteq f^{-1}(]m - 2^{-n-2}, m + 2^{-n-2}[)$.

Par (2) et (1) de la définition ci-dessus, la recherche aboutit nécessairement. Une fois $q_{k_{n+1}}$ trouvé, on définit $x_{n+1} = x_n$ et $y_{n+1} = m$ dans le premier cas, $x_{n+1} = m$ et $y_{n+1} = y_n$ dans le deuxième, et enfin $x_{n+1} = m - 2^{-n-2}$ et $y_{n+1} = m + 2^{-n-2}$ dans le troisième.

Le réel $f(r)$ calculé est donné par la suite $(x_n)_{n \in \mathbb{N}}$ de rationnels, qui vérifie bien $|x_n - x_{n+m}| \leq 2^{-n}$ pour tous n, m (on pourrait tout aussi bien choisir $(y_n)_{n \in \mathbb{N}}$ qui représente le même réel). Notons qu'étant donné une représentation de f et une représentation de r , l'algorithme décrit calcule toujours une représentation de $f(r)$.

5. RCA_0 ou les mathématiques calculables

L'arithmétique du second ordre souffre des mêmes problèmes que la théorie des ensembles vis-à-vis de la crise des fondements : ce système manipule des objets mathématiques complexes, pour lesquels il est difficile de se fier à l'intuition. De plus, le schéma de compréhension autorise les quantifications sur les ensembles, ce qui rend le système imprédicatif. Nous avons déjà vu que l'imprédicativité était parfois source de paradoxes (voir la section 9-1), ce qui n'est pas très rassurant en ce qui concerne la cohérence des axiomes.

Le but originel des mathématiques à rebours étant l'analyse des axiomes nécessaires pour montrer un résultat, afin de restaurer la confiance dans nos mathématiques, il est nécessaire de travailler modulo une théorie de base robuste, qui ne laisse pas place au doute quant à sa cohérence. L'infini étant potentiellement source de paradoxes, notamment à cause de certains comportements contre-intuitifs, notre théorie de base, RCA_0 , se cantonnera aux axiomes nécessaires pour parler des ensembles calculables, qui en particulier peuvent être décrits par des moyens finitaires.

5.1. Schéma de compréhension Δ_1^0

Nous avons vu que le schéma de compréhension permettait de construire tous les ensembles de la hiérarchie arithmétique, et bien au-delà. La première étape consiste à le restreindre pour ne permettre de construire que des ensembles calculables. Rappelons qu'une formule du second ordre (avec variables libres) est Σ_n^0 si elle peut s'exprimer sous la forme d'une alternance de n quantificateurs sur les entiers en commençant par une quantification existentielle, suivie d'une formule Δ_0^0 , c'est-à-dire une formule n'ayant que des quantificateurs bornés. En particulier, les formules Σ_1^0 sont de la

forme $\exists x F(x)$ où F est Δ_0^0 . Notons qu'en arithmétique du second ordre, la formule F peut posséder des variables libres d'ensembles. De la même manière, une formule est Π_n^0 si elle peut s'exprimer sous la forme d'une alternance de n quantificateurs sur les entiers en commençant cette fois par un quantificateur universel. Le théorème 10-1.2 établit une correspondance entre calculabilité et définissabilité par les formules de l'arithmétique. Nous le rappelons ci-après.

Théorème (10-1.2)

Soient $A \subseteq \mathbb{N}$ et $Z \in 2^{\mathbb{N}}$. Les énoncés suivants sont équivalents.

- (1) L'ensemble $A \subseteq \mathbb{N}$ est Z -c. e.
- (2) Il existe $F(X, n)$, formule Σ_1^0 de l'arithmétique du second ordre, telle que

$$A = \{n \in \mathbb{N} : \mathbb{N} \models F(Z, n)\}.$$

- (3) Il existe une fonctionnelle Turing $\Phi(X, n)$ telle que

$$A = \{n \in \mathbb{N} : \Phi(Z, n) \downarrow\}.$$

Nous allons restreindre le schéma de compréhension aux prédicats Δ_1^0 , autrement dit aux prédicats à la fois Σ_1^0 et Π_1^0 . Comme la notion de prédicat Δ_1^0 n'est pas syntaxique, contrairement à celle de prédicat Σ_1^0 ou Π_1^0 , nous allons utiliser l'astuce suivante. Pour toute formule $\Sigma_1^0 F(x)$ et toute formule $\Pi_1^0 G(x)$, définissons l'axiome :

$$(\forall x (F(x) \leftrightarrow G(x))) \rightarrow \exists X \forall y (y \in X \leftrightarrow F(y)). \quad (12)$$

La prémisse $\forall x (F(x) \leftrightarrow G(x))$ s'assure que le prédicat est Δ_1^0 avant de définir l'ensemble X . Notons que les formules F et G peuvent contenir d'autres variables libres, du premier ou du second ordre, sur lesquelles on doit quantifier universellement comme expliqué dans 2.1. On obtient alors en substance : pour toute substitution des paramètres aux variables libres, si $F(x) \leftrightarrow G(x)$ avec ces paramètres-là, alors l'ensemble des éléments qui vérifient $F(x)$ avec ces paramètres existe bien. On appelle (12) le *schéma de compréhension* Δ_1^0 .

5.2. Schéma d'induction Σ_1^0

Concentrons-nous à présent sur le premier ordre, qui est régi par les axiomes de l'arithmétique de Robinson et par le schéma d'induction. Dans une démarche de doute face à ce qui touche à l'infini, il est légitime de questionner ce schéma, qui permet de déduire des propriétés *pour tous les entiers*. On l'utilise par exemple pour montrer le théorème fondamental de l'arithmétique : tout nombre entier se décompose de manière unique en produit de nombres premiers. Il nous est impossible de vérifier entier par

entier que c'est effectivement le cas. Il faut pour cela faire confiance à l'induction. Le lecteur pourra éventuellement être choqué de voir l'induction ainsi remise en cause, et de fait, nous verrons dans la section 23-2.1 que si l'on ne s'autorise aucune forme d'induction, on ne peut pas montrer grand-chose d'intéressant. On peut en revanche montrer qu'un niveau d'induction relativement faible est déjà suffisant pour beaucoup de théorèmes usuels, et en particulier pour formaliser la notion de calcul et de fonctions calculables.

Définition 5.2. On appelle *schéma d'induction* Σ_1^0 , le schéma d'induction (10) restreint à toute formule Σ_1^0 avec variables libres. $\diamond$

En suivant le principe de ne pas utiliser plus que nécessaire, c'est ce schéma d'induction qui sera utilisé dans notre système de base.

Il est difficile au départ de développer une bonne intuition sur ce qu'implique l'absence, par exemple de l'induction pour les formules Σ_2^0 . Cette difficulté vient du fait que la plupart des mathématiciens sont habitués à travailler avec les entiers standard, et à réfléchir en se basant sur ce modèle plutôt qu'en se basant sur les axiomes utilisés. Ainsi par exemple admet-on sans y prêter la moindre attention qu'un ensemble d'entiers non vide quelconque a un plus petit élément. Nous verrons dans la section 23-3.2 que cette propriété est équivalente à l'induction. Il est tout à fait possible de construire des modèles qui ne satisfont que l'induction Σ_1^0 , et au sein desquels certains ensembles Π_2^0 non vides n'ont pas de plus petit élément.

Il est en fait nécessaire ou à défaut très utile, pour comprendre les conséquences de l'absence de certains niveaux d'induction, d'avoir en tête les modèles non standard de l'arithmétique, afin de se créer une intuition qui risque sinon de faire défaut. Nous verrons dans la section 23-3 plus de détails sur l'induction, l'absence d'induction et les modèles non standard qui vont avec.

Exercice 5.3. Montrer que le schéma d'induction (12) pour les formules (avec variables libres) Δ_0^0 implique l'axiome d'induction (11). $\diamond$

5.3. Théorie RCA_0

Nous avons à présent les éléments nécessaires pour définir notre théorie RCA_0 .

Définition 5.4. On note RCA_0 le système composé des axiomes de l'arithmétique de Robinson augmentés du schéma de compréhension Δ_1^0 (12) et du schéma d'induction (10) pour les formules Σ_1^0 avec variables libres. $\diamond$

L'acronyme RCA_0 signifie « Recursive Comprehension Axiom ». Rappelons que *récuratif* est un ancien terme pour *calculable*. Ce système doit son

nom à son schéma de compréhension Δ_1^0 qui ne permet de construire que des ensembles calculables. L'indice « 0 » de RCA_0 signifie que le schéma d'induction est restreint aux formules Σ_1^0 . En effet, Friedman avait initialement défini le système RCA avec le schéma d'induction (10) pour toutes les formules.

Tout comme la restriction du schéma de compréhension permet de restreindre la complexité des ensembles *infinis* du système, nous verrons que le schéma d'induction permet — dans un sens qui sera clarifié dans la section 23-5 — de restreindre la complexité des ensembles *finis*. Dans le cas de RCA_0 , nous aurons les ensembles infinis calculables, et les ensembles finis Σ_1^0 . Le lecteur aura éventuellement l'impression que les ensembles finis peuvent tous être représentés par un entier, et que parler de leur complexité n'a pas vraiment de sens. Nous insistons une fois de plus sur le fait que les restrictions de l'induction doivent être appréhendées sous l'éclairage de modèles non standard. Dans un tel modèle, on a un élément a plus grand que tous les entiers dans la partie standard ω du modèle. De l'extérieur du modèle, il y a une infinité d'éléments plus petits que a , et donc une infinité indénombrable de sous-ensembles d'éléments plus petits que a . En particulier, si notre modèle non standard est dénombrable, il y aura fatalement certains de ces sous-ensembles qui ne pourront pas être « codés » par un élément de notre modèle.

5.4. Modèles de RCA_0

Rappelons qu'une ω -structure est une structure de l'arithmétique du second ordre où le premier ordre est constitué des entiers standard munis des opérations usuelles. Les ω -structures sont donc caractérisées par leur partie du second ordre. Les ω -modèles de RCA_0 admettent une caractérisation très simple en termes d'idéaux Turing.

Définition 5.5. Un *idéal Turing* est une classe $\mathcal{I} \subseteq 2^{\mathbb{N}}$ possédant les deux propriétés suivantes :

- (1) clôture par réduction Turing : $\forall X \in \mathcal{I} \forall Y \leq_T X \ Y \in \mathcal{I}$;
- (2) clôture par jointure : $\forall X \in \mathcal{I} \forall Y \in \mathcal{I} \ X \oplus Y \in \mathcal{I}$.

◇

Exemple 5.6. La classe des ensembles calculables est un idéal Turing. De même, pour tout ensemble A , la classe $\{X \in 2^{\mathbb{N}} : X \leq_T A\}$ est un idéal Turing. Les K -triviaux, vus dans le chapitre 20, forment un idéal Turing.

Exercice 5.7. (★) Montrer que la classe des ensembles low ne forme par un idéal Turing.

Indication.— On pourra par exemple utiliser le théorème 10-3.31.

◇

Exercice 5.8. (★★) Montrer que la classe des ensembles calculatoirement dominés ne forme par un idéal Turing. $\diamond$

Notons que par le théorème 14-2.7, tout idéal Turing dénombrable $\mathcal{I}$ admet une paire exacte, c'est-à-dire deux ensembles A, B tels que

$$\mathcal{I} = \{X \in 2^{\mathbb{N}} : X \leq_T A \wedge X \leq_T B\}.$$

Il est aisé de montrer l'équivalence suivante.

Proposition 5.9 (H. Friedman). Une ω -structure $\mathcal{M}$ est un modèle de RCA_0 si, et seulement si, son second ordre est un idéal Turing. $\star$

PREUVE. Supposons que $\mathcal{M} \models \text{RCA}_0$. Soit $\mathcal{I}$ son second ordre. Montrons que $\mathcal{I}$ est clos par réduction Turing. Soit $X \in \mathcal{I}$, et soit $Y \leq_T X$. Par le théorème 10-1.2, il existe une formule $\Sigma_1^0 F(X, x)$ et une formule $\Pi_1^0 G(X, x)$ telles que $Y = \{n \in \mathbb{N} : F(X, n)\} = \{n \in \mathbb{N} : G(X, n)\}$. Par le schéma de compréhension Δ_1^0 , l'ensemble Y existe, donc $Y \in \mathcal{I}$. Montrons que $\mathcal{I}$ est clos par jointure. Soient $X, Y \in \mathcal{I}$, et soit $F(X, Y, x)$ la formule Δ_0^0 définie par

$$\exists y \leq x ((x = \langle 0, y \rangle \wedge y \in X) \vee (x = \langle 1, y \rangle \wedge y \in Y)).$$

Comme la formule F est Δ_0^0 , le schéma de compréhension Δ_1^0 garantit que l'ensemble $X \oplus Y = \{n \in \mathbb{N} : F(X, Y, n)\}$ existe, si bien que $X \oplus Y \in \mathcal{I}$. La classe $\mathcal{I}$ est donc un idéal Turing.

Inversement, supposons que $\mathcal{I}$ est un idéal Turing, et soit $\mathcal{M}$ l' ω -structure ayant $\mathcal{I}$ pour second ordre. Montrons que $\mathcal{M} \models \text{RCA}_0$. Les axiomes de l'arithmétique de Robinson et le schéma d'induction sont satisfaits, car $\mathcal{M}$ est une ω -structure. Il suffit de montrer que $\mathcal{M}$ satisfait le schéma de compréhension Δ_1^0 . Soient $F(X_1, \dots, X_n, y)$ et $G(X_1, \dots, X_n, y)$ des formules respectivement Σ_1^0 et Π_1^0 , avec $X_1, \dots, X_n \in \mathcal{I}$ comme paramètres du second ordre, tels que $\mathcal{M} \models \forall x (F(X_1, \dots, X_n, x) \leftrightarrow G(X_1, \dots, X_n, x))$. Comme $\mathcal{M}$ est une ω -structure,

$$\{m \in \mathbb{N} : F(X_1, \dots, X_n, m)\} = \{m \in \mathbb{N} : G(X_1, \dots, X_n, m)\}.$$

Appelons A cet ensemble. Par le théorème 10-1.2, $A \leq_T X_1 \oplus \dots \oplus X_n$. Comme $X_1, \dots, X_n \in \mathcal{I}$, par clôture par jointure et réduction Turing de $\mathcal{I}$, on obtient $A \in \mathcal{I}$. $\blacksquare$

Exercice 5.10. Soit $\mathcal{I}_0 \subseteq \mathcal{I}_1 \subseteq \mathcal{I}_2 \subseteq \dots$ une suite d'idéaux Turing. Montrer que $\mathcal{I} = \bigcup_n \mathcal{I}_n$ est un idéal Turing. $\diamond$

Exercice 5.11. Soit $Z_0 \leq_T Z_1 \leq \dots$ une suite d'ensembles croissante pour la réduction Turing. Montrer que $\mathcal{I} = \{X \in 2^{\mathbb{N}} : \exists n X \leq_T Z_n\}$ est un idéal Turing. $\diamond$

5.5. Mathématiques dans RCA_0

Par la proposition 5.9, RCA_0 possède un ω -modèle minimal pour l'inclusion, à savoir l' ω -structure $\mathcal{M}_{\text{CALC}}$ dont le second ordre est l'idéal Turing des ensembles calculables. Il s'ensuit — d'après le théorème 9-2.22 de complétude — qu'il suffit, pour conclure qu'un théorème T n'est pas prouvable dans RCA_0 de montrer que le théorème T implique l'existence d'un ensemble non calculable. Voici un exemple.

Proposition 5.12. Le lemme de König n'est pas prouvable dans RCA_0 . ★

PREUVE. Montrons que $\mathcal{M}_{\text{CALC}}$ ne satisfait pas le lemme de König. Soit, à cet effet, $T \subseteq 2^{<\mathbb{N}}$ un arbre infini calculable n'ayant pas de chemins calculables. Par exemple, l'arbre dont tous les chemins sont de degré PA. L'arbre T étant calculable, $T \in \mathcal{M}_{\text{CALC}}$. Cependant, T ne possède aucun chemin dans $\mathcal{M}_{\text{CALC}}$ puisque T n'a pas de chemin calculable. Il s'ensuit que $\mathcal{M}_{\text{CALC}}$ n'est pas un modèle du lemme de König, donc que RCA_0 ne prouve pas le lemme de König. ■

Notons que pour une preuve de séparation dans les ω -structures, il n'est pas nécessaire de se préoccuper du niveau d'induction utilisé dans la preuve. En effet, les ω -structures satisfont le schéma d'induction complet. Les résultats de séparation sont donc souvent des arguments de calculabilité pure.

En revanche, montrer que RCA_0 implique un théorème requiert beaucoup plus d'attention sur la complexité des objets manipulés et le niveau d'induction utilisé. Rappelons que le *théorème des valeurs intermédiaires*² affirme que si $f : \mathbb{R} \rightarrow \mathbb{R}$ est continue sur l'intervalle $[0, 1]$ avec $f(0) < 0 < f(1)$, alors il existe $x \in]0, 1[$ tel que $f(x) = 0$. Nous allons montrer que le théorème des valeurs intermédiaires est prouvable dans RCA_0 .

Nous renvoyons le lecteur à la section 4 pour quelques explications sur la manière de faire de l'analyse dans Z_2 .

Proposition 5.13 (Simpson [203]). Le théorème des valeurs intermédiaires est prouvable dans RCA_0 . ★

PREUVE. Nous allons formaliser la preuve standard du théorème, à savoir la preuve par recherche dichotomique. On suppose fixé une suite $(I_n)_{n \in \mathbb{N}}$ de tous les intervalles de la forme $]a, b[$, pour $a, b \in \mathbb{Q} \cup \{-\infty, +\infty\}$. Soit $f : [0, 1] \rightarrow \mathbb{R}$ une fonction continue satisfaisant $f(0) < 0 < f(1)$ (la fonction nous est donnée via sa représentation, comme expliqué dans la section 4). Supposons que $f(q) \neq 0$ pour tout rationnel $q \in \mathbb{Q} \cap]0, 1[$, sinon q est le réel désiré. Par le schéma de compréhension Δ_1^0 , les ensembles X_- , X_+ tels que $\bigcup_{n \in X_-} I_n = f^{-1}(]-\infty, 0])$ et $\bigcup_{n \in X_+} I_n = f^{-1}(]0, \infty[)$ existent (il

2. Ou sous cette version, théorème de Bolzano.

suffit d'extraire les bonnes informations de la représentation de notre fonction f). Soit $]a_0, b_0[\supseteq]a_1, b_1[\supseteq \dots$ la suite d'intervalles aux bornes rationnelles, définie par $]a_0, b_0[=]0, 1[$, et

$$]a_{n+1}, b_{n+1}[= \begin{cases} \left] \frac{a_n + b_n}{2}, b_n \right[& \text{si } \frac{a_n + b_n}{2} \in \bigcup_{n \in X_-} I_n, \\ \left] a_n, \frac{a_n + b_n}{2} \right[& \text{si } \frac{a_n + b_n}{2} \in \bigcup_{n \in X_+} I_n. \end{cases}$$

Par notre hypothèse, $f(q) \neq 0$ pour tout rationnel $q \in \mathbb{Q} \cap]0, 1[$, la suite $\{]a_n, b_n[: n \in \mathbb{N}\}$ est bien définie pour tout n . Elle est donc prouvablement $\Delta_1^0(X_- \oplus X_+)$ et existe par le schéma de compréhension Δ_1^0 . Par induction, Σ_1^0 , $f(a_n) < 0 < f(b_n)$ pour tout $n \in \mathbb{N}$, et $|a_n - b_n| = 2^{-n}$. Il s'ensuit que $r = (a_n)_{n \in \mathbb{N}}$ est un réel.

Montrons à présent que $f(r) = 0$. Supposons $f(r) < 0$. Soit $n \in \mathbb{N}$ tel que $f(r) < -2^{-n}$. On recherche alors un intervalle I_n avec $|I_n| < 2^{-n}$ et un entier m tels que $f(]a_m, b_m[) \subseteq I_n$ et $a_m < b_{m+1} < b_m$. La distance entre n'importe quels points x, y de I_n , et donc de $f(]a_m, b_m[)$, est bornée par 2^{-n} . C'est en particulier le cas pour les points $f(b_{m+1})$ et $f(r)$. On a alors $f(b_{m+1}) < 0$. Cela qui contredit le fait que $f(b_{m+1}) > 0$. Le cas $f(r) > 0$ est similaire. ■

Notons que la preuve de la proposition 5.13 n'est pas uniforme, au sens où elle distingue le cas d'une solution rationnelle ou non. Nous verrons dans le chapitre 24 la réduction de Weihrauch, qui permet d'exprimer ces considérations d'uniformité.

Nous redirigeons le lecteur intéressé par un développement plus détaillé des mathématiques dans RCA_0 , vers le très complet ouvrage de Simpson [203], *Subsystems of Second Order Arithmetic*.

6. ACA_0 et la hiérarchie arithmétique

Comme mentionné précédemment, Z_2 n'est pas une extension conservative de l'arithmétique de Peano (PA), au sens où il existe des énoncés formulés uniquement dans le langage de l'arithmétique du premier ordre, qui ne sont pas prouvables dans PA, mais le sont dans Z_2 . Nous allons maintenant voir une restriction importante de l'arithmétique du second ordre, qui nous donne plus de pouvoir que RCA_0 mais qui, contrairement à Z_2 , est conservative sur PA.

Notation

On note ACA_0 le système composé des axiomes de l'arithmétique de Robinson, augmentés du schéma de compréhension (9) pour les formules

arithmétiques, et de l'axiome d'induction (11).

L'acronyme ACA_0 signifie « Arithmetical Comprehension Axiom ». L'axiome d'induction et le schéma de compréhension pour les formules arithmétiques permettent de prouver le schéma d'induction (10) pour les formules arithmétiques. Le système ACA_0 implique donc RCA_0 . Nous avons montré que $\mathbb{Z}_2$ est un système imprédictif, car le schéma de compréhension avec des formules arbitraires permet de construire des ensembles qui dépendent d'eux-mêmes (voir la section 9-1 et l'exemple 8.5). Restreindre le schéma de compréhension aux formules arithmétiques permet justement d'éviter cela, et rend le système prédictif. Des systèmes équivalents à ACA_0 avaient déjà été étudiés par le courant prédicativiste, notamment par Weyl en 1918 dans son livre *Das Kontinuum*, soit presque soixante ans avant le début des mathématiques à rebours.

Le système ACA_0 fait partie des grands systèmes des mathématiques à rebours qui constituent le phénomène du Club des cinq. Notons qu'il n'est pas nécessaire de rajouter le schéma de compréhension pour toutes les formules arithmétiques pour obtenir le système ACA_0 . Il suffit d'ajouter le schéma pour les formules Σ_1^0 , comme le montre la proposition suivante.

Proposition 6.1 (Simpson [203]). Le système RCA_0 prouve que ACA_0 est équivalent au schéma de compréhension Σ_1^0 . ★

PREUVE. Les formules Σ_1^0 étant arithmétiques, ACA_0 implique le schéma de compréhension Σ_1^0 . Inversement, supposons que ACA_0 prouve le schéma de compréhension Σ_n^0 , pour un $n \in \mathbb{N}$ fixé. Montrons qu'il implique le schéma de compréhension Σ_{n+1}^0 . Soit $F(x)$ une formule Σ_{n+1}^0 (avec des variables libres); elle peut s'exprimer sous la forme $\exists y G(x, y)$, où $G(x, y)$ est une formule Π_n^0 . L'ensemble $Y = \{(x, y) \in \mathbb{N} \times \mathbb{N} : \neg G(x, y)\}$ existe, par le schéma de compréhension Σ_n^0 . Par le schéma de compréhension Σ_1^0 paramétré par Y , l'ensemble $Z = \{x \in \mathbb{N} : \exists y (x, y) \notin Y\}$ existe à son tour. En particulier, $Z = \{x \in \mathbb{N} : \exists y G(x, y)\}$ existe. Ainsi, ACA_0 prouve le schéma de compréhension Σ_{n+1}^0 . ■

Notons l'induction de la preuve précédente : on utilise n applications successives de la compréhension Σ_1^0 pour montrer la compréhension Σ_n^0 . De fait, on ne tient pas compte du nombre de fois qu'un axiome est utilisé dans une preuve. Ainsi, dès que l'on s'autorise la compréhension Σ_1^0 , les différents niveaux de la hiérarchie arithmétique ne sont pas distinguables du point de vue de la prouvabilité. Nous verrons dans le chapitre 24 des outils permettant de mesurer de manière plus précise la puissance calculatoire des théorèmes, en contrôlant notamment le nombre d'applications d'un même axiome.

Le système ACA_0 nous fait sortir des mathématiques calculables. Il est en fait équivalent à l'existence du saut Turing de tout ensemble.

Exercice 6.2. ($\star$) Montrer que ACA_0 est équivalent dans RCA_0 à l'énoncé

$$\forall X \exists Y Y = X',$$

où $Y = X'$ est une notation pour $\forall e (e \in Y \leftrightarrow \Phi_e^X(e) \downarrow)$. $\diamond$

6.1. Modèles de ACA_0

Tout comme avec RCA_0 , les ω -modèles de ACA_0 possèdent une belle caractérisation en termes d'idéaux Turing.

Définition 6.3. Un *idéal de saut* est un idéal Turing $\mathcal{I} \subseteq 2^{\mathbb{N}}$ tel que pour tout $X \in \mathcal{I}$, on a $X' \in \mathcal{I}$. $\diamond$

L'équivalence suivante est un analogue de la proposition 5.9.

Proposition 6.4 (Simpson [203]). Une ω -structure $\mathcal{M}$ est un modèle de ACA_0 si, et seulement si, son second ordre est un idéal de saut. $\star$

PREUVE. Supposons $\mathcal{M} \models \text{ACA}_0$. Soit $\mathcal{I}$ son second ordre. En particulier, $\mathcal{M} \models \text{RCA}_0$, donc par la proposition 5.9, $\mathcal{I}$ est un idéal Turing. Par l'exercice 6.2, pour tout $X \in \mathcal{I}$, on a $X' \in \mathcal{I}$, et $\mathcal{I}$ est donc un idéal de saut.

Inversement, supposons que $\mathcal{I}$ soit un idéal de saut. En particulier, $\mathcal{I}$ est un idéal Turing, donc par la proposition 5.9, $\mathcal{M} \models \text{RCA}_0$. Comme $\mathcal{I}$ est un idéal de saut, par l'exercice 6.2, $\mathcal{M}$ est un modèle de ACA_0 . $\blacksquare$

En particulier, ACA_0 possède aussi un ω -modèle minimal pour l'inclusion, à savoir l' ω -modèle dont le second ordre est exactement l'idéal de saut des ensembles arithmétiques. Il est aisé de voir que RCA_0 n'implique pas ACA_0 , en considérant l' ω -structure $\mathcal{M}_{\text{CALC}}$ dont le second ordre est l'idéal Turing des ensembles calculables. L' ω -structure $\mathcal{M}_{\text{CALC}}$ est un modèle de RCA_0 , mais les ensembles calculables ne formant pas un idéal de saut, $\mathcal{M}_{\text{CALC}}$ n'est pas un modèle de ACA_0 .

6.2. Mathématiques dans ACA_0

Le système ACA_0 est très puissant, dans la mesure où il suffit à prouver une écrasante majorité de théorèmes. Commençons par montrer que ACA_0 est équivalent dans RCA_0 à un théorème d'analyse bien connu. Le *théorème de Bolzano-Weierstrass* affirme que pour toute suite $(x_n)_{n \in \mathbb{N}}$ de points dans le segment $[0, 1]$, il existe une sous-suite convergente, c'est-à-dire une fonction strictement croissante $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que la suite $(x_{f(n)})_{n \in \mathbb{N}}$ converge vers un point.

Le théorème de Bolzano-Weierstrass énonce deux existences : celle d'une sous-suite convergente, et celle du point de convergence. Le système RCA_0 prouve que l'existence d'un point de convergence implique celle d'une sous-suite convergente : il suffit de calculer à partir de la suite et du point une sous-suite de plus en plus proche du point de convergence. L'inverse n'est cependant pas vrai : l'existence seule d'une sous-suite convergente est équivalente à un système nommé COH (voir la section 25-4.1), compris strictement entre RCA_0 et ACA_0 .

Proposition 6.5 (Simpson [203]). Le système ACA_0 est équivalent dans RCA_0 au théorème de Bolzano-Weierstrass. ★

PREUVE DE BOLZANO-WEIERSTRASS DANS ACA_0 . Soit $(x_n)_{n \in \mathbb{N}}$ une suite de réels dans $[0, 1]$. Soit $X \in 2^{\mathbb{N}}$ une représentation de cette suite. En utilisant X'' , on définit une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que

$$\forall n \forall i |x_{f(n)} - x_{f(n+i)}| \leq 2^{-n}.$$

On calcule pour cela à partir de X'' une suite $[a_0, b_0] \supseteq [a_1, b_1] \supseteq \dots$ d'intervalles telle que chaque segment $[a_n, b_n]$ contient une infinité de x_m et telle que $b_n - a_n \leq 2^{-n}$. On commence par $[a_0, b_0] = [0, 1]$. Une fois $[a_n, b_n]$ calculé, s'il y a une infinité de points dans le segment $[a_n, (a_n + b_n)/2]$ on définit alors $[a_{n+1}, b_{n+1}] = [a_n, (a_n + b_n)/2]$. À défaut, on définit le segment $[a_{n+1}, b_{n+1}]$ comme $[(a_n + b_n)/2, b_n]$. Notons que la question : « il existe une infinité de points tels que... » est Π_2^0 , et peut donc être résolue à l'aide de X'' . En utilisant l'induction arithmétique, on montre conjointement que la suite $(a_n, b_n)_{n \in \mathbb{N}}$ est bien définie et que chaque $[a_n, b_n]$ contient bien une infinité de x_m . On peut alors définir $f(n)$ comme étant le premier m trouvé tel que $x_m \in [a_n, b_n]$. Une fois f définie, on construit aisément une représentation rationnelle $(q_n)_{n \in \mathbb{N}}$ de $\lim_{n \rightarrow \infty} f(n)$ telle que, pour tous $n, i \in \mathbb{N}$, on a $|q_n - q_{n+i}| \leq 2^{-n}$. ■

PREUVE DE ACA_0 DANS BOLZANO-WEIERSTRASS. Soit X un ensemble, et soit $(x_n)_{n \in \mathbb{N}}$ la suite de réels définie par $x_n = 0.X'[n]$, où $X'[n]$ est l'approximation de X' à l'étape de calcul n . Notons que la suite est bornée par 1. D'après le théorème de Bolzano-Weierstrass, cette suite possède une sous-suite convergente et une limite, qui ne peut être que $0.X'$. D'après l'exercice 6.2, on a donc ACA_0 . ■

Les deux preuves précédentes illustrent bien l'insensibilité de ACA_0 aux itérations du saut Turing : nous avons utilisé le double saut Turing pour prouver le théorème de Bolzano-Weierstrass. En revanche, dans le sens inverse, une application du théorème de Bolzano-Weierstrass n'a permis de prouver l'existence que du simple saut. On n'obtient alors le double saut qu'en appliquant le théorème deux fois de suite. Il est naturel de se poser la question de la puissance calculatoire exacte d'une application du théorème

de Bolzano-Weierstrass. Il se trouve que sa puissance est exactement celle d'un degré PA relativement à $\emptyset'$, comme le montre l'exercice suivant.

Exercice 6.6. (*) Montrez que pour toute suite $(x_n)_{n \in \mathbb{N}}$ de réels dans $[0, 1]$, il existe un arbre $\emptyset'$ -calculable dont tous les chemins infinis sont exactement des points de convergence de sous-suites de $(x_n)_{n \in \mathbb{N}}$. Montrez enfin que pour tout arbre $\emptyset'$ -calculable infini, il existe une suite $(x_n)_{n \in \mathbb{N}}$ de points dans $[0, 1]$ dont les points de convergence sont exactement les chemins infinis de l'arbre. $\diamond$

7. WKL_0 et l'argument de compacité

Le système WKL_0 est sans doute le système le plus significatif des mathématiques à rebours, qui les premières en ont identifié l'importance au sein des mathématiques ordinaires. Des formalismes équivalents aux systèmes Z_2 et ACA_0 existaient préalablement, tandis que RCA_0 capture la notion déjà bien établie des mathématiques calculables. WKL_0 constitue l'un des systèmes du phénomène du Club des cinq, et correspond intuitivement aux arguments de compacité.

Notation

On note WKL l'énoncé « Tout arbre binaire infini admet un chemin infini. » On note WKL_0 le système composé de RCA_0 augmenté de l'énoncé WKL .

WKL_0 est l'acronyme de « Weak König's Lemma », et correspond à une puissance bien connue en calculabilité, à savoir celle des degrés PA. Nous allons voir que ce système est équivalent à des théorèmes importants d'analyse et de logique, notamment le théorème de compacité de Borel-Lebesgue et le théorème de complétude de Gödel.

7.1. Modèles de WKL_0

Tout comme pour RCA_0 et ACA_0 , le système WKL_0 admet une caractérisation de ses ω -modèles, qui découle directement de celle de RCA_0 , en termes d'idéaux Turing (voir la proposition 5.9).

Définition 7.1. Un *idéal de Scott* est un idéal Turing $\mathcal{I} \subseteq 2^{\mathbb{N}}$ tel que pour tout $X \in \mathcal{I}$ et tout arbre binaire infini X -calculable $T \subseteq 2^{<\mathbb{N}}$, il existe un chemin infini de T appartenant à $\mathcal{I}$. $\diamond$

Autrement dit, un idéal Turing $\mathcal{I}$ est un idéal de Scott si pour tout $X \in \mathcal{I}$, il existe un ensemble $Y \in \mathcal{I}$ de degré PA relativement à X . Dana Scott a été le premier à étudier ces idéaux [194], qui caractérisent sans surprise les ω -modèles de WKL_0 .

Proposition 7.2. Une ω -structure $\mathcal{M}$ est un modèle de WKL_0 si, et seulement si, son second ordre est un idéal de Scott. ★

PREUVE. Immédiat par la proposition 5.9 et la définition de WKL_0 . ■

Sachant qu'il n'existe pas de degré PA calculable, l'idéal Turing des ensembles calculables n'est pas un idéal de Scott. L' ω -structure $\mathcal{M}_{\text{CALC}}$ est donc un modèle de RCA_0 qui n'est pas modèle de WKL_0 , ce qui montre que RCA_0 n'implique pas WKL_0 .

La plupart des théorèmes de base Π_1^0 se relativisent et permettent de construire des idéaux de Scott avec des propriétés calculatoires faibles. Ces idéaux correspondent à des ω -modèles de WKL_0 , de sorte que nos résultats de calculabilité se traduisent par des résultats de séparation en mathématiques à rebours.

Proposition 7.3. Soit $A_0, A_1, \dots$ une suite dénombrable d'ensembles non calculables. Il existe un idéal de Scott $\mathcal{I}$ ne contenant aucun des A_i . ★

PREUVE. Nous allons définir une suite d'ensembles $Z_0 \leq_T Z_1 \leq_T \dots$ de telle sorte que pour tout $n \in \mathbb{N}$,

- (1) Z_{n+1} est de degré PA relativement à Z_n ;
- (2) $A_0, A_1, \dots$ ne sont pas Z_n -calculables.

On a $Z_0 = \emptyset$. Supposons que l'on a défini Z_n . Par le théorème de base d'évitement de cône (voir le théorème 8-4.7) itéré, il existe un ensemble Z_{n+1} de degré PA relativement à Z_n , tel que $A_0, A_1, \dots$ ne sont pas Z_{n+1} -calculables. En particulier, $Z_{n+1} \geq_T Z_n$ (voir si besoin l'exercice 8-7.6). Soit $\mathcal{I} = \{X \in 2^{\mathbb{N}} : \exists n \ X \leq_T Z_n\}$. Par l'exercice 5.11, $\mathcal{I}$ est un idéal Turing. Montrons que $\mathcal{I}$ est un idéal de Scott. Soit $X \in \mathcal{I}$, et soit $T \subseteq 2^{<\mathbb{N}}$ un arbre infini X -calculable. En particulier, il existe un $n \in \mathbb{N}$ tel que $X \leq_T Z_n$, donc Z_{n+1} est de degré PA relativement à X . Il s'ensuit que Z_{n+1} calcule un chemin infini de T , donc $\mathcal{I}$ contient un chemin infini de T . La classe $\mathcal{I}$ est donc un idéal de Scott. Par construction, $\mathcal{I}$ ne contient aucun des A_i . ■

En particulier, il n'existe pas d' ω -modèle minimal de WKL_0 pour l'inclusion, contrairement à RCA_0 et ACA_0 . En effet, par la proposition 7.3, l'intersection de tous les idéaux de Scott est l'idéal Turing des ensembles calculables, qui n'est pas un idéal de Scott.

Nous pouvons déduire aussi de la proposition 7.3 que WKL_0 n'implique pas ACA_0 dans RCA_0 . En effet, tout ω -modèle de ACA_0 contient $\emptyset'$, mais par la proposition 7.3, il existe un ω -modèle de WKL_0 ne contenant pas $\emptyset'$.

Exercice 7.4. Montrer qu'il existe un idéal de Scott $\mathcal{I}$ ne contenant que des ensembles de degré calculatoirement dominé. $\diamond$

Exercice 7.5. Montrer qu'il existe un idéal de Scott $\mathcal{I}$ ne contenant que des ensembles de degré low. $\diamond$

7.2. Mathématiques dans WKL_0

Comme nous l'avons mentionné, le système WKL_0 correspond intuitivement aux arguments de compacité. Le théorème de compacité par excellence est le théorème de Borel-Lebesgue, qui énonce que pour toute collection $(\mathcal{U}_n)_{n \in \mathbb{N}}$ d'ouverts de $\mathbb{R}$ telle que $[0, 1] \subseteq \bigcup_n \mathcal{U}_n$, il existe k tel que $[0, 1] \subseteq \bigcup_{n \leq k} \mathcal{U}_n$. Nous allons montrer que cet énoncé est équivalent à WKL_0 dans RCA_0 .

Étant donné $[0, 1] \subseteq \bigcup_n \mathcal{U}_n$, on peut supposer sans perte de généralité que chaque ouvert $\mathcal{U}_n$ consiste en un intervalle rationnel $]a_n, b_n[$, en isolant les intervalles de chaque ouvert de notre réunion. Passons à la preuve du théorème.

PREUVE DE BOREL-LEBESGUE DANS WKL_0 . Pour toute chaîne $\sigma \in 2^{<\mathbb{N}}$, on considère l'intervalle $I_\sigma = [0.\sigma 0^\infty, 0.\sigma 1^\infty] \subseteq [0, 1]$. Notons que les intervalles I_σ pour $|\sigma| = n$ forment un découpage de $[0, 1]$ en 2^n parties presque disjointes (par exemple, $[0.01010^\infty, 0.01011^\infty]$ et $[0.01100^\infty, 0.01101^\infty]$ ont exactement le point $0.01011^\infty = 0.01100^\infty$ en commun.)

On construit l'arbre $T \subseteq 2^{<\mathbb{N}}$ tel que $\sigma \in T$ ssi I_σ n'est inclus dans aucun ouvert $\mathcal{U}_n$. Notons que $\sigma \in T$ est un prédicat Π_1^0 . En effet, $\sigma \notin T$ si, et seulement si, $[0.\sigma 0^\infty, 0.\sigma 1^\infty] \subseteq]a_n, b_n[$ pour un certain n .

Donc, T est Π_1^0 relativement à la représentation des ouverts $\mathcal{U}_n$, et l'on peut construire un arbre T -calculable T' contenant T tel que $[T] = [T']$ (voir la proposition 8-1.10). Montrons que $[T]$ ne contient aucun chemin infini. Considérons $X \in 2^\mathbb{N}$, et considérons aussi le réel $r_x = 0.X$. Formellement, $r_x = \sum_{i \in \mathbb{N}^*} 2^{-i} \times X(i)$. Par l'hypothèse du théorème, on a $r_x \in]a_n, b_n[$ pour un certain n . Soit $\sigma \prec X$ suffisamment grand tel que $r_x \in I_\sigma \subseteq]a_n, b_n[$. Alors, $\sigma \notin T$, et donc $X \notin [T]$. Ainsi, T n'a pas de chemin infini et T' non plus. Par le lemme faible de König, on en déduit que T' est fini, et donc T aussi. Il existe dès lors n tel que tout intervalle I_σ pour σ de taille n est inclus dans un certain $\mathcal{U}_m$. Comme le nombre de chaînes de taille n est fini, on a donc un certain k tel que $[0, 1] \subseteq \bigcup_{n \leq k} \mathcal{U}_n$.

Avant de conclure, souvenons-nous que WKL_0 n'a toujours que l'induction pour les formules Σ_1^0 . Il n'est pas toujours évident de bien cerner où l'induction est utilisée et à quel niveau. Le point auquel il faut ici faire attention est le suivant : pour la dernière étape, on utilise le fait que « pour toute chaîne σ de taille n , il existe m pour lequel $I_\sigma \subseteq \mathcal{U}_m$ » implique « il existe k tel que pour toute chaîne σ de taille n , il existe $m \leq k$ tel que $I_\sigma \subseteq \mathcal{U}_m$. »

Il s'agit d'une utilisation du *schéma de collection* Σ_1^0 que nous définirons formellement dans la section 23-3.1, et qui se démontre à partir de l'induction Σ_1^0 . Un examen attentif révélera que ce schéma est également nécessaire pour montrer $[T] = [T']$ ci-dessus. ■

Montrons à présent que WKL_0 est nécessaire pour prouver le théorème de Borel-Lebesgue.

PREUVE DE WKL_0 DANS BOREL-LEBESGUE. Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre binaire infini. Reprenons la notation $I_\sigma = [0.\sigma 0^\infty, 0.\sigma 1^\infty]$ de la preuve précédente. Supposons que T n'ait pas de chemin infini. Alors, pour tout $X \in 2^\mathbb{N}$, il existe $\sigma \notin T$ tel que $X \in I_\sigma$. Il s'ensuit que $(I_\sigma)_{\sigma \notin T}$ forme une couverture de $2^\mathbb{N}$ par des ouverts. Par le théorème de Borel-Lebesgue, il existe un ensemble fini $F \subseteq 2^{<\mathbb{N}} \setminus T$ tel que $2^\mathbb{N} \subseteq \bigcup_{\sigma \in F} I_\sigma$. Soit n la longueur maximale des chaînes dans F . L'arbre T étant infini, il existe $\tau \in T$ de longueur n . Comme $\bigcup_{\sigma \in F} I_\sigma$ est une couverture de $2^\mathbb{N}$, il existe $\sigma \in F$ tel que $I_\tau \subseteq I_\sigma$. En particulier, $\sigma \preceq \tau$, mais $\sigma \notin T$ et $\tau \in T$, ce qui contredit le fait que T soit un arbre.

Là encore, il faut faire attention à ne pas utiliser plus que l'induction Σ_1^0 , ce qui est bien le cas. ■

Le système WKL_0 est équivalent à des théorèmes plus généraux de compacité sur les espaces métriques compacts. En particulier, WKL_0 est équivalent à l'énoncé « Toute couverture d'un espace métrique compact par des ouverts admet une sous-couverture finie. » Le système WKL_0 est également équivalent au théorème 9-2.24 : toute théorie cohérente peut être étendue en une théorie complète et cohérente. L'équivalence entre la capacité à calculer une extension complète et cohérente de l'arithmétique de Peano et la capacité à calculer un chemin dans tout arbre calculable infini illustre cette équivalence. Le lecteur trouvera un développement plus détaillé des mathématiques dans WKL_0 dans l'ouvrage de Simpson [203].

7.3. WWKL_0 et l'aléatoire

Il convient de s'arrêter sur un sous-système de WKL_0 particulièrement important, au point qu'il mériterait, par sa robustesse, par son interprétation épistémologique, et par le nombre de théorèmes qui lui sont équivalents, d'être considéré comme le sixième membre d'honneur du Club des cinq. Il s'agit de la restriction du lemme faible de König aux arbres de mesure positive. Dans cette section, nous ferons appel à des notions d'aléatoire algorithmique introduites dans la partie II, et plus précisément aux propriétés calculatoires des aléatoires de Martin-Löf. Il n'est cependant pas nécessaire de comprendre cette section pour continuer le chapitre.

Définition 7.6. Un arbre $T \subseteq 2^{<\mathbb{N}}$ est de *mesure positive* si

$$\liminf_n \frac{|\{\sigma \in T : |\sigma| = n\}|}{2^n} > 0.$$

◇

Nous avons vu qu'il existait une correspondance entre classes Π_1^0 et arbres binaires. La notion d'arbre de mesure positive décrit les arbres dont les classes Π_1^0 correspondantes sont de mesure positive.

Exercice 7.7. (★) Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre. Montrer que

$$\lambda([T]) = \liminf_n \frac{|\{\sigma \in T : |\sigma| = n\}|}{2^n}.$$

◇

La notion d'arbre de mesure positive s'avère plus pratique à manipuler que la notion de classe Π_1^0 de mesure positive dans l'arithmétique du second ordre, car elle évite de devoir formaliser la notion de mesure qui est une fonction sur les classes, et donc d'ordre supérieur. Nous utiliserons donc la formulation en termes d'arbres de mesure positive pour définir le système $WWKL_0$.

Notation

$WWKL$ est l'énoncé « Tout arbre binaire de mesure positive admet un chemin infini. »

$WWKL_0$ est le système RCA_0 augmenté de l'énoncé $WWKL$.

L'acronyme $WWKL$ vient de l'anglais « Weak weak König's lemma », et nous l'appellerons « Lemme très faible de König ». Il découle directement du lemme faible de König, car tout arbre binaire de mesure positive est infini. Les classes Π_1^0 de mesure positive possèdent des liens très forts avec l'aléatoire de Martin-Löf. En particulier, tout MLR est, à préfixe près, un membre de chaque classe Π_1^0 de mesure positive, comme le montre le lemme suivant.

Lemme 7.8 (Kučera [128])

Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre X -calculable tel que $\lambda([T]) > 0$. Alors, $[T]$ contient un suffixe de n'importe quel ensemble $MLR(X)$. Autrement dit, si l'ensemble Z est $MLR(X)$, il existe alors $Y \in [T]$ et σ tels que $Z = \sigma Y$. ★

PREUVE. Considérons la classe $\Sigma_1^0(X) \mathcal{U} = 2^{\mathbb{N}} \setminus [T]$. En particulier, $\lambda(\mathcal{U}) < 1$. Soit W un ensemble X -c.e. sans préfixe tel que $\mathcal{U} = \bigcup_{\sigma \in W} [\sigma]$. On définit alors $\mathcal{U}^1 = \mathcal{U}$ et $W^1 = W$, puis par induction sur n on définit la classe $\mathcal{U}^{n+1}$ comme étant celle décrite par l'ensemble $W^{n+1} = \{\sigma\tau : \sigma \in W \text{ et } \tau \in W^n\}$. On a dès lors $\lambda([W^{n+1}]) = \sum_{\sigma \in W} 2^{-|\sigma|} \lambda([W^n]) = \lambda([W]) \times \lambda([W^n])$.

On en déduit $\lambda(\mathcal{U}^n) = (\lambda(\mathcal{U}))^n$. Comme $\lambda(\mathcal{U}) < 1$, alors $f : n \rightarrow \lambda(\mathcal{U}^n)$ tend vers 0 de manière effective, c'est-à-dire bornée par une fonction calculable. En particulier, $\bigcap_n \mathcal{U}^n$ est un X -test de Martin-Löf.

Considérons à présent un ensemble Z qui n'a pas de suffixe dans $2^{\mathbb{N}} \setminus \mathcal{U}$. Donc, pour tout σ tel que $Z = \sigma Y$, on a $Y \in \mathcal{U}$. En particulier, $Z \in \mathcal{U} = \mathcal{U}^1$. Soit $\sigma \prec Z$ tel que $\sigma \in W^1$. Alors, Z amputé du préfixe σ appartient aussi à $\mathcal{U}$, et donc par définition Z a aussi un préfixe dans W^2 , et appartient donc à $\mathcal{U}^2$. On montre en continuant ainsi que $Z \in \bigcap_n \mathcal{U}^n$, et donc que Z n'est pas $\text{MLR}(X)$. ■

Le lemme 7.8 nous permet de donner une belle caractérisation des ω -modèles de WWKL_0 .

Proposition 7.9. Une ω -structure $\mathcal{M}$ est un modèle de WWKL_0 si, et seulement si, son second ordre est un idéal Turing $\mathcal{I}$ tel que pour tout $X \in \mathcal{I}$, il existe un $\text{MLR}(X)$ $Z \in \mathcal{I}$. ★

PREUVE. Soit $\mathcal{M}$ un ω -modèle de WWKL_0 et soit $\mathcal{I}$ son second ordre. En particulier, $\mathcal{M} \models \text{RCA}_0$, donc par la proposition 5.9, $\mathcal{I}$ est un idéal Turing. Soit $X \in \mathcal{I}$, et soit $T \subseteq 2^{<\mathbb{N}}$ un arbre X -calculable de mesure positive tel que tous les chemins infinis soient $\text{MLR}(X)$. Par exemple, nous pouvons fixer $c \in \mathbb{N}$ et définir $T = \{\sigma \in 2^{<\mathbb{N}} : \forall \tau \preceq \sigma \ K^X(\tau) \geq |\tau| - c\}$, où $K^X(\sigma)$ est la complexité de Kolmogorov sans préfixe, relativisée à X (voir le théorème 18-2.1). Comme $\mathcal{M} \models \text{WWKL}$, il existe $Z \in [T]$ tel que $Z \in \mathcal{I}$. En particulier, Z est $\text{MLR}(X)$.

Réciproquement, soit $\mathcal{I}$ un idéal Turing tel que pour tout $X \in \mathcal{I}$, il existe un $\text{MLR}(X)$ $Y \in \mathcal{I}$. Soit $\mathcal{M}$, l' ω -structure induite par $\mathcal{I}$. Par la proposition 5.9, $\mathcal{M} \models \text{RCA}_0$. Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre de mesure positive codé par un ensemble $X \in \mathcal{I}$. Par hypothèse, il existe un $\text{MLR}(X)$ $Z \in \mathcal{I}$. En particulier, $\lambda([T]) > 0$, donc il existe $Y \in [T]$ et $\sigma \in 2^{<\mathbb{N}}$ tels que $Z = \sigma Y$. Comme $Y \leq_T Z$ et $Z \in \mathcal{I}$, il vient $Y \in \mathcal{I}$, et donc $\mathcal{M} \models \text{WWKL}$. ■

Le système WWKL_0 peut être considéré comme capturant les mathématiques faisant appel à des arguments probabilistes. Cette vision est cependant à relativiser, car certains arguments probabilistes nécessitent des notions d'aléatoire plus fortes. Comme aucun MLR n'est calculable, l' ω -modèle minimal $\mathcal{M}_{\text{CALC}}$ de RCA_0 n'est pas un modèle de WWKL_0 , ce qui implique que RCA_0 ne prouve pas WWKL_0 .

Montrons maintenant que WWKL_0 est strictement plus faible de WKL_0 , en construisant un modèle de WWKL_0 qui n'est pas modèle de WKL_0 à l'aide des outils de l'aléatoire algorithmique.

Proposition 7.10. WWKL_0 n'implique pas WKL_0 dans RCA_0 . ★

PREUVE. Soit Z un ensemble 2-aléatoire. Par le théorème 19-1.7 et le corollaire 19-1.8, Z n'est pas de degré PA. Soit

$$Y_0 \oplus Z_0 = Z, \quad Y_1 \oplus Z_1 = Z_0, \quad Y_2 \oplus Z_2 = Z_1,$$

et ainsi de suite. Par le théorème de van Lambalgen relativisé (voir le théorème 19-2.5), Y_0 est MLR et Z_0 est $\text{MLR}(Y_0)$, Y_1 est $\text{MLR}(Y_0)$ et Y_2 est $\text{MLR}(Y_0 \oplus Y_1)$, et de manière générale, Y_{n+1} est $\text{MLR}(Y_0 \oplus \dots \oplus Y_n)$ et $Y_n \leq_T Z$ pour tout $n \in \mathbb{N}$. Soit $\mathcal{I} = \{X \in 2^{\mathbb{N}} : \exists n \ X \leq_T Y_0 \oplus \dots \oplus Y_n\}$. Soit $\mathcal{M}$ l' ω -structure induite par $\mathcal{I}$. La classe $\mathcal{I}$ est un idéal Turing par l'exercice 5.11, donc $\mathcal{M} \models \text{RCA}_0$ par la proposition 5.9. Soit $X \in \mathcal{I}$, et soit $n \in \mathbb{N}$ tel que $X \leq_T Y_0 \oplus \dots \oplus Y_n$. Comme $Y_{n+1} \in \mathcal{I}$ et Y_{n+1} est $\text{MLR}(Y_0 \oplus \dots \oplus Y_n)$, alors Y_{n+1} est $\text{MLR}(X)$, donc $\mathcal{M} \models \text{WWKL}$. L' ω -structure $\mathcal{M}$ est donc un modèle de WWKL_0 . Cependant, $\mathcal{I}$ ne contient aucun ensemble de degré PA car pour tout $X \in \mathcal{I}$, $X \leq_T Z$, et Z n'est pas de degré PA. Ainsi par la proposition 7.2, $\mathcal{M}$ n'est pas un modèle de WKL_0 . ■

8. Systèmes plus puissants

Nous avons pour l'instant défini cinq systèmes :

$$\text{RCA}_0, \quad \text{WWKL}_0, \quad \text{WKL}_0, \quad \text{ACA}_0 \quad \text{et} \quad Z_2,$$

listés par ordre strictement croissant de force logique. Le système RCA_0 capture les mathématiques calculables, ce qui couvre une grande partie des mathématiques ordinaires, mais il existe cependant de nombreux théorèmes classiques non calculables. Le système WWKL_0 ajoute les arguments probabilistes, tandis que le système WKL_0 ajoute les arguments de compacité à la boîte à outils du mathématicien. Le système ACA_0 représente quant à lui un saut dans la force logique des mathématiques. Il capture les mathématiques prédicatives, ce qui correspond à l'écrasante majorité des théorèmes. De très rares théorèmes classiques échappent à la puissance de ACA_0 , et font appel au système Z_2 .

La grande majorité des théorèmes se situant en dessous de ACA_0 , les sous-systèmes faibles de l'arithmétique du second ordre, ont reçu une attention particulière. En effet, la démarche des mathématiques à rebours est avant tout de comprendre des tendances des mathématiques et non pas de rechercher l'exhaustivité. Il convient cependant de mentionner quelques systèmes importants qui se situent au-delà de ACA_0 , et que nous étudierons plus en détail dans le chapitre 31 de la partie IV sur l'hypercalculabilité.

8.1. ACA'_0 ou la puissance de l'induction

Nous allons commencer par un système légèrement plus forts que ACA_0 , et qui n'est pas considéré comme faisant partie du phénomène du Club des cinq.

Nous avons vu que ACA_0 était équivalent à l'énoncé $\forall X \exists Y Y = X'$ (voir l'exercice 6.2). En itérant l'énoncé de l'existence du saut Turing, il s'ensuit immédiatement que ACA_0 est équivalent à l'énoncé $\forall X \exists Y Y = X^{(n)}$ pour tout $n \in \mathbb{N}$. Ici, il convient de faire bien attention à la quantification de n , qui est une quantification externe. L'énoncé $\forall n \forall X \exists Y Y = X^{(n)}$ fait appel à un axiome d'induction qui sort du cadre de ACA_0 .

Beaucoup de théorèmes que nous étudierons dans cette partie s'expriment sous la forme $\forall X (F(X) \rightarrow \exists Y G(X, Y))$, où F et G sont des formules arithmétiques éventuellement avec des variables libres. On peut alors voir un théorème de ce type comme un problème mathématique, formulé en termes d'instances et de solutions. On dira qu'un ensemble X est une instance du problème si $F(X)$ est vrai, et Y est une solution de l'instance X si $G(X, Y)$ est vrai. Par exemple, une instance du lemme faible de König est un arbre binaire infini, et une solution de cette instance est un chemin infini. Nous avons vu que par itérations du saut Turing, il était possible de prouver dans ACA_0 l'existence de tous les ensembles de la hiérarchie arithmétique. En revanche, ACA_0 garde une forme de faiblesse, dans le sens où il n'est pas capable de prouver l'existence de solutions à un problème, qui soient arbitrairement élevées dans la hiérarchie arithmétique. Plus précisément, le théorème suivant a été prouvé indépendamment par Jockusch et Solovay (voir Wang [230]).

Théorème 8.1 (Jockusch et Solovay)

Soient $F(X)$ et $G(X, Y)$ des formules arithmétiques n'ayant que des variables libres explicites. Si

$$\text{ACA}_0 \vdash \forall X (F(X) \rightarrow \exists Y G(X, Y)),$$

alors il existe un $k \in \mathbb{N}$ tel que

$$\text{ACA}_0 \vdash \forall X (F(X) \rightarrow \exists Y \leq_T X^{(k)} G(X, Y)).$$

PREUVE COMMUNIQUÉE À WANG PAR JOCKUSCH

Soient $F(X)$ et $G(X, Y)$ des formules arithmétiques n'ayant que des variables libres explicites. Soit C un symbole de constante du second ordre. Soit $G_k(X)$ la formule $(F(X) \rightarrow \exists Y \leq_T X^{(k)} G(X, Y))$. Soit T la théorie dans le langage $\mathcal{L}_{\text{PA}}$ augmenté de la constante C , composée des axiomes de PA, du schéma d'induction pour les formules arithmétiques avec C comme paramètre, et des axiomes $\neg G_k(C)$ pour tout $k \in \mathbb{N}$. Supposons

que $\forall X G_k(X)$ ne soit prouvable par ACA_0 pour aucun $k \in \mathbb{N}$. En appliquant le théorème 9-2.20 et en utilisant $\neg G_{k+1}(X) \rightarrow \neg G_k(X)$, on en déduit que pour tout k le système $\text{ACA}_0 \cup \{\neg G_k(C)\}_{i \leq k}$ a un modèle. Par compacité, on en déduit que $\text{ACA}_0 \cup \{\neg G_k(C)\}_{k \in \mathbb{N}}$ a un modèle. Comme $\text{ACA}_0 \vdash \text{PA}$, la théorie T a un modèle $\mathcal{M}$. Plus précisément, $\mathcal{M}$ est spécifié par un premier ordre M et une interprétation $C^M \subseteq M$ du symbole C . Soit $\mathcal{I} \subseteq \mathcal{P}(M)$ la classe des sous-ensembles de M définissables dans $\mathcal{M}$ par une formule arithmétique avec comme unique paramètre C^M . En particulier, $C^M \in \mathcal{I}$. Considérons la structure $\mathcal{M}'$ dans le langage de $\mathcal{L}_{Z_2}$ dont le premier ordre est M et le second ordre est $\mathcal{I}$.

Montrons que $\mathcal{M}' \models \text{ACA}_0$. Comme $\mathcal{M} \models \text{PA}$, alors $\mathcal{M}'$ satisfait les axiomes de l'arithmétique de Robinson. Soit $H(x)$ une formule de l'arithmétique avec paramètres dans $\mathcal{M}'$. Comme tout paramètre de H est définissable de manière arithmétique dans $\mathcal{M}$, alors H peut être transformée en une formule $\widehat{H}$ avec comme unique paramètre C^M , telle que

$$\{x \in M : M \models \widehat{H}(x)\} = \{x \in M : \mathcal{M}' \models H(x)\}.$$

Comme $\{x \in M : M \models \widehat{H}(x)\} \in \mathcal{I}$, alors $\mathcal{M}'$ satisfait le schéma de compréhension arithmétique avec paramètres. Montrons enfin que $\mathcal{M}'$ satisfait l'axiome d'induction (11). Soit $X \in \mathcal{I}$. En particulier, X est définissable par une formule arithmétique avec comme unique paramètre C^M ; or, T prouve le schéma d'induction pour ces formules, donc l'axiome d'induction (11) est satisfait par $\mathcal{M}'$.

Ainsi, $\mathcal{M}' \models \text{ACA}_0$, mais comme $\mathcal{M} \models \neg G_k(C^M)$ pour tout k et que tout ensemble de $\mathcal{M}'$ est arithmétiquement définissable en C^M , alors $\mathcal{M}'$ est modèle de $\neg F(C^M) \vee \forall Y \neg G(C^M, Y)$. Donc, $\forall X (F(X) \rightarrow \exists Y G(X, Y))$ n'est pas un théorème de ACA_0 . ■

Il découle directement du théorème 8.1 que l'énoncé $\forall n \forall X \exists Y Y = X^{(n)}$ n'est pas prouvable dans ACA_0 . Souvenons-nous que les ω -modèles de ACA_0 sont clos par saut Turing, et sont donc en particulier des modèles de $\forall n \forall X \exists Y Y = X^{(n)}$. Les modèles de ACA_0 dans lesquels cet énoncé est faux auront donc nécessairement un premier ordre non standard, au sein duquel il manque le niveau d'induction nécessaire pour montrer par exemple $(\exists X X = \emptyset^{(n)} \rightarrow \exists X X = \emptyset^{(n+1)}) \rightarrow \forall n \exists X X = \emptyset^{(n)}$. Cela nous conduit à définir le système suivant.

Notation

On note ACA'_0 le système RCA_0 augmenté de l'axiome « Pour tout ensemble X et tout entier $n \in \mathbb{N}$, il existe une suite $Y_0, Y_1, \dots, Y_n$ telle que $Y_0 = X$ et $Y_{s+1} = Y'_s$. »

Les ω -modèles de ACA_0 et ACA'_0 coïncident, sachant que tout idéal de saut satisfait l'énoncé ci-dessus. Toute preuve de séparation entre ACA_0 et ACA'_0 se fait donc nécessairement dans des modèles non standard. Nous verrons dans le chapitre 25 un exemple de théorème classique prouvable dans ACA'_0 , mais pas dans ACA_0 .

8.2. ACA_0^+ et l' ω -saut Turing

Le système ACA_0^+ est un renforcement de ACA_0 qui apparaît naturellement dans l'analyse des preuves de théorèmes combinatoires dénombrables, et en particulier le théorème 8.2 de Hindman.

Notation

ACA_0^+ est le système RCA_0 augmenté de l'énoncé « Pour tout ensemble X , l' ω -saut Turing $X^{(\omega)} = \bigoplus_{n \in \mathbb{N}} X^{(n)}$ de X existe. »

Le système ACA_0^+ est strictement plus fort que ACA_0 , car l' ω -structure dont le second ordre est composé des ensembles arithmétiques est un modèle de ACA_0 (et de ACA'_0), mais ne contient pas l' ω -saut Turing de $\emptyset$, donc n'est pas un modèle de ACA_0^+ . Ce système a été introduit par Blass, Hirst, et Simpson [21] pour mesurer la puissance logique du théorème de Hindman. Il s'agit d'un théorème des mathématiques combinatoires, et plus particulièrement de la théorie de Ramsey, qui sera abordée dans le chapitre 25. Nous n'aurons toutefois pas l'occasion de revenir sur le théorème de Hindman, dont nous mentionnons tout de même ici l'énoncé pour le lecteur qui souhaite en mesurer la difficulté. Étant donné un ensemble $X \subseteq \mathbb{N}$, on note $\text{FS}(X)$ l'ensemble

$$\left\{ \sum_{n \in F} n : F \subseteq X \text{ non vide et fini} \right\}.$$

Théorème 8.2 (Hindman [86])

Pour tout $k \in \mathbb{N}$ et toute fonction $f : \mathbb{N} \rightarrow \{0, \dots, k\}$, il existe un ensemble infini $H \subseteq \mathbb{N}$ tel que $|f(\text{FS}(H))| = 1$.

Blass, Hirst, et Simpson [21] ont prouvé que le théorème de Hindman impliquait ACA_0 et qu'il était prouvable dans ACA_0^+ . Le théorème de Hindman est l'exemple typique, en mathématiques à rebours, de théorème pour lequel l'analyse calculatoire a nécessité de trouver de nouvelles preuves plus élémentaires. Il existe à ce jour plusieurs preuves, dont celle originale de Hindman [86] formalisable dans ACA_0^+ , une preuve courte due à Baumgartner [11], mais dont l'analyse calculatoire est plus complexe, une preuve de

Galvin-Glazer à l'aide d'ultrafiltres [39] qui sont des objets du troisième ordre, et donc plus difficiles à étudier en mathématiques à rebours, et plus récemment une preuve simple due à Towsner [224], mais dont la complexité logique est la même que la preuve originale de Hindman. La question suivante reste l'une des plus grandes questions ouvertes en mathématiques à rebours.

Question 8.3. Quelle est la puissance calculatoire exacte du théorème de Hindman ? ★

On ignore à ce jour si le théorème de Hindman est équivalent dans RCA_0 à ACA_0 , à ACA_0^+ , ou même strictement entre les deux systèmes. On ne connaît pour l'instant aucun théorème classique équivalent au système ACA_0^+ , ce qui relativise l'importance de ce système.

8.3. ATR_0 et l'induction transfinie

Les systèmes ATR_0 et $\Pi_1^1\text{-CA}_0$ sont respectivement les quatrième et cinquième grands systèmes du phénomène du Club des cinq étudiés en mathématiques à rebours. Leur étude ainsi que celle de leurs modèles fait appel à des outils plus puissants que ceux de la calculabilité classique, à savoir l'hypercalculabilité, qui est un domaine d'étude à part entière et que nous présenterons dans la partie IV. Les systèmes ATR_0 et $\Pi_1^1\text{-CA}_0$ seront étudiés plus en détail dans le chapitre 31. Nous en donnons cependant un rapide aperçu dans cette section pour compléter le panorama des systèmes des mathématiques à rebours.

Le système ATR_0 affirme informellement l'existence de l' α -saut Turing, pour un ordinal α dans le modèle considéré. Plus précisément, un ensemble *bien ordonné* est un ensemble totalement ordonné $(A, <_A)$ n'admettant pas de suite infinie décroissante (cette notion sera formellement présentée et étudiée dans les moindres détails dans le chapitre 27). Soit $\theta(x, X)$ une formule arithmétique, contenant notamment les variables libres x et X , et potentiellement d'autres variables libres. Cette formule induit sur les ensembles un opérateur $\Theta : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ défini par $\Theta(X) = \{n \in \mathbb{N} : \theta(n, X)\}$.

Définition 8.4. Le *schéma de récursion transfinie* énonce, pour toute formule arithmétique $\theta(x, X)$ avec des variables libres, et tout ensemble bien ordonné $(A, <_A)$, l'existence d'un ensemble $Y = \bigoplus_{a \in A} Y_a$ défini pour tout $a \in A$ par

$$Y_a = \Theta\left(\bigoplus_{b <_A a} Y_b\right).$$

◇

Par exemple, si l'on considère la formule $\theta(x, X) = \Phi_x^X(x) \downarrow$ et l'ensemble bien ordonné $(\mathbb{N}, <_{\mathbb{N}})$, alors l'opérateur Θ est le saut Turing défini par $\Theta(X) = X'$.

On a

$$Y_0 = \Theta(\emptyset) = \emptyset', \quad Y_1 = \Theta(Y_0) = \emptyset'', \quad Y_2 = \Theta(Y_0 \oplus Y_1) = (\emptyset' \oplus \emptyset'')',$$

et ainsi de suite. L'ensemble Y résultant est de même degré que l' ω -saut Turing de $\emptyset$.

Notation

ATR_0 est le système RCA_0 augmenté du schéma de récursion transfinie.

Au vu de l'exemple précédent, nous voyons que ATR_0 implique ACA_0^+ , et donc ACA_0 . Nous verrons dans le chapitre 31 que ATR_0 est un système strictement plus puissant.

8.4. $\Pi_1^1\text{-CA}_0$ et l'imprédictivité

Il est possible d'étendre la classification des formules au-delà de la hiérarchie arithmétique, en se basant sur l'alternance des quantificateurs sur les ensembles. On obtient alors la *hiérarchie analytique*. Nous ne définirons que le niveau le plus bas. Une formule de l'arithmétique du second ordre est Π_1^1 (resp. Σ_1^1) si elle est de la forme $\forall X F(X)$ (resp. $\exists X F(X)$) où F est une formule arithmétique.

Notation

$\Pi_1^1\text{-CA}_0$ est le système RCA_0 augmenté du schéma de compréhension (9) pour les formules Π_1^1 .

Le système $\Pi_1^1\text{-CA}_0$ est strictement plus puissant que ATR_0 , et de très rares théorèmes lui sont équivalents. Mentionnons le théorème de Cantor-Bendixson, qui énonce que toute classe fermée indénombrable de $\mathbb{R}^n$ est la réunion d'une classe fermée parfaite et d'une classe finie ou dénombrable. Le théorème de Cantor-Bendixson affirme en particulier que l'hypothèse du continu est prouvable pour les classes fermées de $\mathbb{R}^n$. Nous verrons la démonstration d'une certaine forme de ce théorème au sein de l'espace de Baire $\mathbb{N}^{\mathbb{N}}$ avec le théorème 30-3.2.

Contrairement au système ATR_0 , le système $\Pi_1^1\text{-CA}_0$ est foncièrement imprédictatif. Rappelons qu'une définition imprédictative est en substance une définition circulaire dans laquelle l'objet qui est défini est lui-même susceptible d'être utilisé dans la définition. Considérons l'exemple suivant.

Exemple 8.5. Les ensembles suivants sont définis de manière imprédictative :

(1) $A = \{n \in \mathbb{N} : \forall X \, n \notin X\};$

(2) $W = \{e \in \mathbb{N} : W_e \text{ énumère un arbre } T \subseteq \mathbb{N}^{<\mathbb{N}} \text{ tel que } \forall Y \, Y \notin [T]\}$.

Les ensembles A et W sont définis via une quantification sur tous les ensembles, et sont donc définis en fonction d'eux-mêmes.

Il va de soi que l'ensemble A de l'exemple précédent peut se définir autrement — $A = \emptyset$ —, mais nous verrons dans la section 29-3 que ce n'est pas le cas pour l'ensemble W . Notons que l'imprédictivité de Z_2 fait uniquement appel au schéma de compréhension (9) pour une formule Π_1^1 , ce qui fait bien de $\Pi_1^1\text{-CA}_0$ un système imprédictif.

8.5. Résumé

Dans ce chapitre, nous avons présenté une hiérarchie strictement croissante de sous-systèmes de l'arithmétique du second ordre :

$$\text{RCA}_0 < \text{WKL}_0 < \text{ACA}_0 < \text{ACA}'_0 < \text{ACA}^+_0 < \text{ATR}_0 < \Pi_1^1\text{-CA}_0 < Z_2.$$

Les systèmes RCA_0 , WKL_0 , ACA_0 , ATR_0 et $\Pi_1^1\text{-CA}_0$ forment le Club des cinq. La plupart des théorèmes ordinaires sont soit prouvables dans RCA_0 , soit équivalents à l'un des quatre autres systèmes modulo RCA_0 . Parmi ces théorèmes, la très grande majorité d'entre eux est prouvable dans ACA_0 , voire WKL_0 .

L'étude des sous-systèmes de l'arithmétique du second ordre ne se cantonne cependant pas à cette hiérarchie minimaliste linéairement ordonnée. Nous allons étudier dans les chapitres suivants plusieurs sous-systèmes de puissance intermédiaire qui apparaissent naturellement dans l'étude des mathématiques. Le chapitre 25 en particulier se focalisera sur l'étude méta-mathématique du théorème de Ramsey, qui a profondément bouleversé le visage des mathématiques à rebours.

Chapitre 23

Induction et conservation

Dans ce chapitre nous nous concentrons sur les possibilités et les limites du système RCA_0 en ce qui concerne les énoncés du premier ordre. Rappelons que RCA_0 restreint le schéma d'induction aux formules Σ_1^0 .

La restriction de l'induction pour capturer les mathématiques calculables peut sembler étonnante à première vue, car il s'agit d'un principe régissant le comportement des entiers et non des ensembles d'entiers. Nous verrons cependant qu'il existe un lien entre le schéma d'induction et le schéma de compréhension bornée, autrement dit, l'existence des ensembles finis dans le modèle. La restriction de l'induction s'inscrit dans le programme des mathématiques à rebours qui vise à trouver les axiomes optimaux pour prouver les théorèmes ordinaires. Il est donc raisonnable de considérer l'induction comme une ressource que l'on cherche à minimiser.

Étant donné que les entiers standard sont un modèle de l'induction pour toutes les formules, il est utile pour bien appréhender les limites de RCA_0 d'appuyer son intuition sur les modèles non standard, qui rappelons-le contiennent des éléments plus grands que tous les entiers standard.

Notation : ω is the new $\mathbb{N}$

Dans le cadre de l'étude de modèles potentiellement non standard, il est d'usage de désigner les entiers standard (ceux que nous considérons comme étant les « vrais entiers » et notés $\mathbb{N}$ jusqu'alors dans ce livre) par la lettre ω . Notre notation habituelle $\mathbb{N}$ fera alors référence au sein de ce chapitre à l'ensemble syntaxique des entiers : par exemple, la phrase « $\forall n \in \mathbb{N} F(n)$ » signifie que tout entier du point de vue du modèle considéré vérifie la formule F . La notation $\mathbb{N}$ est alors simplement un symbole syntaxique qui est interprété par M au sein d'un modèle $\mathcal{M} = (M, S, +, \times, <, 0, 1)$.

1. Fonctions RCA_0 -prouvablement calculables

Afin de s'assurer de la validité d'une preuve dans RCA_0 , il faut notamment vérifier que seul l'axiome d'induction Σ_1^0 est utilisé. Il s'agit d'une restriction à ne pas prendre à la légère, et qu'il faut garder en tête en particulier avant d'appliquer l'axiome de compréhension pour les formules Δ_1^0 , que nous rappelons ici. Pour toute formule $\Sigma_1^0 F(x)$ et toute formule $\Pi_1^0 G(x)$, nous avons :

$$(\forall x (F(x) \leftrightarrow G(x))) \rightarrow \exists X \forall y (y \in X \leftrightarrow F(y)). \quad (12)$$

En particulier, il sera nécessaire de démontrer $(\forall x (F(x) \leftrightarrow G(x)))$ au sein de RCA_0 avant de pouvoir appliquer l'axiome de compréhension. Il est de fait envisageable que certains ensembles soient calculables, c'est-à-dire Δ_1^0 , sans pour autant que nous puissions le démontrer au sein de RCA_0 . Il apparaît donc important de mener une étude détaillée des énoncés arithmétiques qu'il est possible de démontrer dans RCA_0 , et notamment de comprendre quelles fonctions y sont prouvablement calculables.

Rappelons qu'une fonction $f : \omega \rightarrow \omega$ est codée sous la forme d'un ensemble d'entiers par son graphe $G_f = \{\langle m, n \rangle : f(m) = n\}$. Une fonction $f : \omega \rightarrow \omega$ est calculable si, et seulement si, son graphe est définissable par une formule Σ_1 de l'arithmétique (via l'ingénieux codage de Gödel, présenté dans la preuve du théorème 9-3.4), autrement dit s'il existe une formule $\Sigma_1 F(x, y)$ telle que $G_f = \{\langle m, n \rangle : \omega \models F(m, n)\}$, où ω dénote de modèle standard de PA.

Remarque

Les graphes des fonctions calculables correspondent à des formules Σ_1 (en particulier sans paramètres du second ordre). Pour les fonctions X -calculables avec oracle X , la correspondance tient toujours, mais pour une formule Σ_1^0 utilisant le paramètre X .

Définition 1.1. On dit qu'une formule $F(x, y)$ de l'arithmétique (du premier ou du second ordre) est *fonctionnelle* si $\omega \models \forall x \exists! y F(x, y)$. ◆

Rappelons que la notation $\exists! y$ signifie « il existe un unique y ». Notons que si une formule $\Sigma_1^0 F(x, y)$ est fonctionnelle, la fonction $f : \omega \rightarrow \omega$ qu'elle définit est bien de graphe Δ_1^0 , car pour toute fonction $f : \omega \rightarrow \omega$ (totale), on a $f(x) = y$ ssi $f(x) \neq z$ pour tout $z \neq y$. Formellement, la formule $F(x, y)$ s'exprime de manière Π_1^0 via la formule $\forall z \neq y \neg F(x, z)$. Attention toutefois, ce n'est pas parce que $F(x, y)$ est fonctionnelle que le système RCA_0 peut le démontrer ; il se pourrait que dans certains modèles la relation $F(x, y)$ ne soit pas fonctionnelle à cause d'éléments x non standard. Cela nous conduit à la définition suivante.

Définition 1.2. Une formule $F(x, y)$ de l'arithmétique (du premier ou du second ordre) est *T-prouvablement fonctionnelle* pour une théorie T si $T \vdash \forall x \exists! y F(x, y)$, c'est-à-dire :

$$T \vdash \forall x \exists y F(x, y) \quad \text{et} \quad T \vdash \forall x \forall y_0 \forall y_1 (F(x, y_0) \wedge F(x, y_1) \rightarrow y_0 = y_1).$$

Une fonction $f : \omega \rightarrow \omega$ est *T-prouvablement totale* s'il existe une formule $F(x, y)$ *T-prouvablement fonctionnelle* telle que $F(n, f(n))$ est vraie pour tout $n \in \omega$. Une fonction $f : \omega \rightarrow \omega$ est *T-prouvablement calculable* si elle est *T-prouvablement totale* via une formule Σ_1 . $\diamond$

Formules Σ_1 vs Σ_1^0

Notons que si $T \vdash \forall x \exists! y F(x, y)$ où F est une formule Σ_1^0 avec une variable libre Z , cela équivaut à une preuve de $T \vdash \forall Z \forall x \exists y! F(x, y)$. La fonction correspondante sera ainsi une fonctionnelle prouvablement totale sur tous les oracles Z .

La définition ci-dessus se généralise sans problème aux fonctions à plusieurs paramètres. La question principale de ce chapitre est donc la caractérisation des fonctions RCA_0 -prouvablement calculables. Dans un premier temps, nous allons notamment montrer le résultat qui suit.

Théorème 1.3

Les fonctions primitives récursives sont RCA_0 -prouvablement calculables.

Il s'agit d'un théorème qui va permettre de nous abstraire un peu du système d'axiomes RCA_0 . Souvenons-nous de la caractérisation des fonctions primitives récursives comme celles que l'on peut calculer par des programmes structurés utilisant des boucles **for**, mais pas de boucles **while** (voir le théorème 6-3.22). Nous pouvons garder grâce à cela une certaine légèreté dans le formalisme, tout en restant rigoureux : pour montrer que l'utilisation d'une fonction calculable est valide dans RCA_0 , il suffit de montrer qu'elle est primitive récursive.

Le théorème 1.3 va en fait plus loin, et l'on a la superbe caractérisation suivante.

Théorème 1.4 (Parikh, voir Hajek et Pudlak [81])

Les fonctions RCA_0 -prouvablement calculables sont exactement les fonctions primitives récursives.

La réciproque est hélas bien plus complexe à démontrer et dépasse le cadre de cette ouvrage ; nous donnerons toutefois dans la deuxième partie de ce chapitre (voir la section 7) certains éléments de la preuve.

Nous consacrons donc le début de ce chapitre à la preuve du théorème 1.3. Nous allons en fait montrer un résultat un peu plus fort : les fonctions primitives récursives définissables à l'aide d'un oracle Z quelconque sont prouvablement calculables quel que soit l'oracle dans la théorie RCA_0 .

Définition 1.5. La classe des fonctions *Z -primitives récursives* est la plus petite classe de fonctions contenant les fonctions de base (successeur, projection, constantes), la fonction caractéristique de Z , et qui est close par le schéma de composition et de récursion primitive. $\diamond$

Dans le cas d'une fonction primitive récursive sans oracle, la partie du premier ordre de RCA_0 — notée $\Sigma_1\text{-PA}$, c'est-à-dire l'arithmétique de Robinson et le schéma d'induction pour les formules Σ_1 — suffit pour montrer qu'elle est calculable. Il ne s'agit pas de quelque chose de bien surprenant, et nous verrons dans la deuxième partie de ce chapitre que tout énoncé du premier ordre démontrable dans RCA_0 est déjà démontrable dans $\Sigma_1\text{-PA}$.

2. Sous-systèmes faibles de PA

Afin de montrer le théorème 1.3, nous allons procéder à un codage des fonctions primitives récursives (éventuellement avec oracle) par des formules fonctionnelles Σ_1^0 , similaire à celui qui fut présenté dans la preuve du théorème 9-3.4, à ceci près que nous allons cette fois devoir nous assurer que tout ce que nous faisons se formalise bien dans RCA_0 . Le caractère fastidieux des développements à venir sera compensé par une compréhension détaillée des principaux axiomes de sous-systèmes faibles de l'arithmétique, menée notamment dans la section 3.

Même si nous étudions ici le premier ordre, nous nous plaçons dans le cadre de l'arithmétique du second ordre. Les différentes formules utilisées sont susceptibles de comporter des variables libres du second ordre. Pour la majorité des preuves qui suivent, cela n'a pas d'incidence.

2.1. L'arithmétique de Robinson

L'arithmétique de Robinson (notée $\mathbf{Q}$) est, rappelons-le, un fragment de l'arithmétique de Peano à laquelle on a retiré le schéma d'induction. Ses axiomes sont précisés dans la section 9-2.3. Les axiomes de l'arithmétique de Robinson seuls forment un système très faible, au sein duquel à peu près aucun énoncé du type $\forall x F(x)$ n'est démontrable.

Voici, à titre d'exemple, un modèle non standard de $\mathbf{Q}$ au sein duquel l'énoncé $\forall x \neg(x < x)$ est faux.

Exemple 2.1. Soit $\mathcal{M} = (\omega \cup \{\infty\}, +, \times, <, 0, 1)$ la structure où $+$, $\times$, $<$ ont leur sens usuel sur ω , et où

- ▷ $n + \infty = \infty + n = \infty$ pour $n \in \omega \cup \{\infty\}$,
- ▷ $n \times \infty = \infty \times n = \infty$ pour $n \in (\omega \setminus \{0\}) \cup \{\infty\}$,
- ▷ $0 \times \infty = \infty \times 0 = 0$.

La structure $\mathcal{M}$ est un modèle de $\mathbf{Q}$. On laisse au lecteur le soin de vérifier que les axiomes de $\mathbf{Q}$ sont tous vérifiés dans ce modèle.

Notons que l'arithmétique de Robinson $\mathbf{Q}$ pourra tout de même démontrer des énoncés du type $\neg(\dot{n} < \dot{n})$ où $\dot{n}$ est le terme $(\dots(1 + 1) + \dots + 1)$ — avec 1 répété n fois —, mais sans être capable de faire une généralisation à tous les entiers.

Exercice 2.2. (★) Montrer que les énoncés

$$\forall x, y \ x + y = y + x \quad \text{et} \quad \forall x, y \ x \times y = y \times x$$

ne sont pas démontrables dans $\mathbf{Q}$. ◇

Afin de pouvoir utiliser les faits même les plus basiques sur les entiers, comme la commutativité de l'addition ou de la multiplication, il apparaît alors nécessaire d'utiliser un minimum d'induction.

2.2. Le schéma d'induction ouvert

Voyons pour commencer le schéma d'induction le plus basique possible, qui permet de démontrer les faits élémentaires concernant les entiers.

Notation

On dénote par $\mathbf{I}_{\text{ouvert}}$ le schéma d'axiomes d'induction

$$(F(0) \wedge \forall x (F(x) \rightarrow F(x + 1))) \rightarrow \forall x F(x)$$

restreint aux formules sans quantificateur.

Proposition 2.3. La commutativité, l'associativité et l'injectivité de l'addition sont prouvables dans $\mathbf{Q} + \mathbf{I}_{\text{ouvert}}$. Formellement,

$$\begin{aligned} \mathbf{Q} + \mathbf{I}_{\text{ouvert}} &\vdash \forall x, y \ x + y = y + x, \\ \mathbf{Q} + \mathbf{I}_{\text{ouvert}} &\vdash \forall x, y, z \ (x + y) + z = x + (y + z), \\ \mathbf{Q} + \mathbf{I}_{\text{ouvert}} &\vdash \forall x, y, z \ x + y = x + z \rightarrow y = z. \end{aligned}$$

★

PREUVE

Commençons par la commutativité. Montrons d'abord $0 + x = x$ pour tout x (F_1). On a $0 + 0 = 0$ par (4) de $\mathbf{Q}$.

Supposons $0 + x = x$. Alors, $0 + (x + 1) = (0 + x) + 1$, par (5) de $\mathbf{Q}$. Par

l'hypothèse d'induction $(0 + x) + 1 = x + 1$. Par I_{ouvert} , on obtient $0 + x = x$ pour tout x . Montrons à présent $(x + 1) + y = (x + y) + 1$ pour tous x, y (F_2). On a bien $(x + 1) + 0 = x + 1 = (x + 0) + 1$, par (4) de I_{ouvert} . Supposons $(x + 1) + y = (x + y) + 1$. Alors,

$$(x + 1) + (y + 1) = ((x + 1) + y) + 1 = ((x + y) + 1) + 1 = (x + (y + 1)) + 1,$$

par (5) de I_{ouvert} et par l'hypothèse d'induction.

Par I_{ouvert} , on obtient $(x + 1) + y = (x + y) + 1$ pour tous x, y . Montrons finalement la commutativité de l'addition. On a $x + 0 = 0 + x$ pour tout x , par (4) de Q et (F_1). Supposons $x + y = y + x$. Alors, $x + (y + 1) = (x + y) + 1$, par (5) de Q . Par hypothèse d'induction, $(x + y) + 1 = (y + x) + 1$. Par (F_2), $(y + x) + 1 = (y + 1) + x$. Par I_{ouvert} , on obtient $x + y = y + x$ pour tous x, y .

Passons à l'associativité. On a bien $(x + y) + 0 = x + y = x + (y + 0)$, par (4) de Q . Supposons $(x + y) + z = x + (y + z)$. Alors,

$$\begin{aligned} (x + y) + (z + 1) &= ((x + y) + z) + 1 = (x + (y + z)) + 1 \\ &= x + ((y + z) + 1) = x + (y + (z + 1)), \end{aligned}$$

par (5) de Q et par hypothèse d'induction. Par I_{ouvert} , on obtient l'associativité pour tous x, y, z .

Montrons pour finir l'injectivité. Comme on a $0 + x = x$ pour tout x , alors $0 + y = 0 + z \rightarrow y = z$. Supposons $x + y = x + z \rightarrow y = z$. Puis supposons $(x + 1) + y = (x + 1) + z$. Alors, $(x + y) + 1 = (x + z) + 1$, par associativité et commutativité de l'addition. Par (2) de Q on a par conséquent $(x + y) = (x + z)$. Donc, $y = z$ par hypothèse d'induction, ce qui implique $y + 1 = z + 1$. Par I_{ouvert} , on obtient l'injectivité pour tous x, y, z . ■

Exercice 2.4. (★) Montrer que les trois énoncés suivants sont prouvables dans $Q + I_{\text{ouvert}}$.

- (1) La commutativité de la multiplication : $x \times y = y \times x$.
- (2) La distributivité de la multiplication sur l'addition :

$$x \times (y + z) = x \times y + x \times z.$$

- (3) L'associativité de la multiplication : $(x \times y) \times z = x \times (y \times z)$. ◇

L'injectivité de la multiplication sera plus facile à montrer en utilisant $<$.

Notation

On écrit $x \leq y$ comme raccourci pour l'énoncé $x = y \vee x < y$.

Lemme 2.5. Les énoncés suivants sont prouvables dans $\mathbf{Q} + \mathbf{I}_{\text{ouvert}}$.

- (1) Totalité de l'ordre : $x = y \vee x < y \vee y < x$. Chacun de ces cas est, de plus, mutuellement exclusif.
- (2) Transitivité de la comparaison : $(x < y \wedge y \leq z) \rightarrow x < z$.
- (3) Addition d'inégalité : $(x_1 < y_1 \wedge x_2 \leq y_2) \rightarrow x_1 + x_2 < y_1 + y_2$.
- (4) Croissance de la multiplication : $x < x \times y$ pour $x \neq 0$ et $y \neq 0, 1$.
- (5) L'injectivité de la multiplication : $(x \times y = x \times z \wedge x \neq 0) \rightarrow y = z$. ★

PREUVE. (1) Montrons $x = y \vee x < y \vee y < x$. Rappelons que par (8) de $\mathbf{Q}$ on a $x < y$ ssi $x + z = y$ pour $z \neq 0$. Pour $x = 0$ et y quelconque, soit $y = 0$ auquel cas $x = y$, soit $y \neq 0$ auquel cas $x + y = y$ par (4) de $\mathbf{Q}$ et la commutativité de l'addition, et donc $x < y$.

Supposons $x = y \vee x < y \vee y < x$ et montrons

$$(x + 1) = y \vee (x + 1) < y \vee y < (x + 1).$$

Si $x = y$, alors $y + 1 = x + 1$, et donc $y < x + 1$. Si $x < y$, alors $x + z = y$ pour $z \neq 0$. Si $z = 1$, alors $x + 1 = y$. Si $z \neq 1$, alors $z = z' + 1$ par (3) de $\mathbf{Q}$, pour $z' \neq 0$, car $z \neq 1$. En particulier, $x + (z' + 1) = y$, et donc $(x + 1) + z' = y$, par commutativité et associativité de l'addition. Donc, $x + 1 < y$, par (8) de $\mathbf{Q}$. Finalement, si $y < x$, alors $y + z = x$ pour $z \neq 0$. Par suite, $(y + z) + 1 = y + (z + 1) = x + 1$ par associativité de l'addition. Dès lors, $y < x + 1$, par (8) de $\mathbf{Q}$. Par $\mathbf{I}_{\text{ouvert}}$, on a donc $x = y \vee x < y \vee y < x$ pour tous x, y .

Montrons à présent que chaque cas est mutuellement exclusif. Si $x = y$, on ne peut avoir $x + z = y$ ou $y + z = x$ pour $z \neq 0$. En effet, par injectivité de l'addition, cela donnerait $z = 0$. Si $x < y$, alors $x + z = y$ pour $z \neq 0$. Supposons $y \leq x$. Alors, $y + z' = x$ pour $z' \neq 0$, ce qui donne $y + z' + z = y = y + 0$, et donc $z' + z = 0$ par injectivité de l'addition, ce qui est absurde.

- (2) Si $x < y$ et $y \leq z$, alors $x + a = y$ pour $a \neq 0$ et $y + b = z$ pour $b \neq 0$. Donc, $x + (a + b) = (x + a) + b = z$, par associativité de l'addition. Ainsi, $x < z$.
- (3) Si $x_1 < y_1$ et $x_2 \leq y_2$, alors $x_1 + z_1 = y_1$ pour $z_1 \neq 0$ et $x_2 + z_2 = y_2$. Donc, $(x_1 + x_2) + (z_1 + z_2) = (x_1 + z_1) + (x_2 + z_2) = y_1 + y_2$, par associativité et commutativité de l'addition. Ainsi, $x_1 + x_2 < y_1 + y_2$.
- (4) Montrons $x < x \times y$ pour tout $x \neq 0$ et tout $y \neq 0, 1$. Pour $y = 2$, on a bien $x < x + x$, par (8) de $\mathbf{Q}$. Supposons à présent $x < x \times y$. Alors, $x < x \times y < x \times y + x = x \times (y + 1)$. Par $\mathbf{I}_{\text{ouvert}}$, on a bien $x < x \times y$ pour tout $x \neq 0$ et tout $y \neq 0, 1$.
- (5) Montrons la contraposée de $(x \times y = x \times z \wedge x \neq 0) \rightarrow y = z$. Supposons $y \neq z$. Par (1) de ce lemme, $y < z$ ou $z < y$. Supposons $y < z$,

l'autre cas étant symétrique. Montrons alors par induction que pour tout $x \neq 0$, $x \times y < x \times z$. Pour $x = 1$, on a bien $x \times y = y < z = x \times z$. Soit $x \neq 0$ tel que $y < z$ implique $x \times y < x \times z$ pour tous y, z . Alors, $(x + 1) \times y = x \times y + y$. Or, $x \times y < x \times z$ par l'hypothèse d'induction et $y < z$. Donc, par (3) de ce lemme,

$$(x + 1) \times y = x \times y + y < x \times z + z = (x + 1) \times z.$$

Par I_{ouvert} , on a bien $y < z \rightarrow x \times y < x \times z$ pour tout $x \neq 0$ et tous y, z .

Par (1) de ce lemme, on a $x \times y < x \times z$ implique $x \times y \neq x \times z$. ■

Exercice 2.6. Montrer la multiplication d'inégalité dans $\mathbb{Q} + I_{\text{ouvert}}$:

$$(x < y \wedge z \neq 0) \rightarrow x \times z < y \times z.$$

◇

2.3. Le schéma d'induction Δ_0^0

En dehors de faits basiques sur les entiers, le système I_{ouvert} reste relativement faible. La définition de certaines fonctions simples nécessitera l'utilisation de quantificateurs bornés. Nous introduisons pour cela un système légèrement plus puissant.

Notation

On dénote par $I\Delta_0^0$ le schéma d'axiomes d'induction

$$(F(0) \wedge \forall x (F(x) \rightarrow F(x + 1))) \rightarrow \forall x F(x)$$

restreint aux formules Δ_0^0 .

Notation

On écrit $x|y$ pour signifier que x divise y , ce qui se formalise par l'énoncé Δ_0 suivant : $\exists m \leq y, x \times m = y$.

Exercice 2.7. (★) Montrer la division euclidienne dans $\mathbb{Q} + I\Delta_0^0$: pour tous a, b avec $b \neq 0$, il existe un unique couple q, r avec $0 \leq r < b$ tel que $a = qb + r$. ◇

Lemme 2.8. Les énoncés suivants sont prouvables dans $\mathbb{Q} + I\Delta_0^0$:

- (1) $z|x \wedge z|y \rightarrow z|(x + y)$;
- (2) $2|x \vee 2|(x + 1)$.

PREUVE. (1) Si $z \times m_1 = x$ et $z \times m_2 = y$, alors $z \times (m_1 + m_2) = x + y$.

- (2) On a $2 \times 0 = 0$, donc $2|0$. Supposons $2|x \vee 2|(x + 1)$. Si $2|(x + 1)$, alors $2|(x + 1) \vee 2|(x + 2)$. Si $2|x$, comme aussi $2|2$, alors $2|(x + 2)$. Par $I\Delta_0^0$, on en conclut $2|x \vee 2|(x + 1)$ pour tout x . ■

Lemme 2.9

La fonction $\dot{\div} : \omega^2 \rightarrow \omega$ définie par $a \dot{\div} b$ égal à z tel que $b + z = a$ si $b < a$, et par $a \dot{\div} b = 0$ sinon, est prouvablement totale dans $\mathbf{Q} + \mathbf{I}\Delta_0^0$ via une formule Δ_0 . ★

PREUVE

La formule Δ_0 qui définit $a \dot{\div} b$ est $F(a, b, r) \equiv (b < a \wedge b + r = a) \vee r = 0$. Montrons l'existence. Pour tous a, b , si $b < a$, alors par (8) de $\mathbf{Q}$ il existe $r \neq 0$ tel que $b + r = a$. Si $b \geq a$, alors la formule est vérifiée pour $r = 0$. Montrons l'unicité. Supposons $F(a, b, r_1)$ et $F(a, b, r_2)$ vérifiées. Si $\neg(b < a)$, alors $r_1 = r_2 = 0$. Si $b < a$, alors $b + r_1 = a$ et $b + r_2 = a$, donc $b + r_1 = b + r_2$, et donc $r_1 = r_2$, par injectivité de l'addition. ■

Lemme 2.10. Les énoncés suivants sont prouvables dans $\mathbf{Q} + \mathbf{I}\Delta_0^0$:

- (1) $x \times (a \dot{\div} b) = x \times a \dot{\div} x \times b$;
- (2) si $z|x$ et $z|y$, alors $z|(x \dot{\div} y)$. ★

PREUVE. (1) Supposons $a \leq b$. Alors, $x \times a \leq x \times b$, par l'exercice 2.6. En particulier, $x \times (a \dot{\div} b) = 0 = x \times a \dot{\div} x \times b$. Supposons à présent $b < a$. Alors, $x \times b \leq x \times a$, par l'exercice 2.6. Soit r_0 tel que $r_0 + b = a$. Soit r_1 tel que $r_1 + x \times b = x \times a$. Alors, $x \times r_0 + x \times b = x \times a$ ce qui nous donne $x \times r_0 = r_1$, ce qui donne bien $x \times (a \dot{\div} b) = x \times a \dot{\div} x \times b$.

- (2) Soient m_1, m_2 tels que $z \times m_1 = x$ et $z \times m_2 = y$. Par (1),

$$z \times m_1 \dot{\div} z \times m_2 = x \dot{\div} y.$$

Donc, $z \times (m_1 \dot{\div} m_2) = x \dot{\div} y$. Ainsi, $z|(x \dot{\div} y)$. ■

Lemme 2.11. La bijection de Cantor $\langle \rangle : \omega^2 \rightarrow \omega$ et ses projections sont prouvablement totales dans $\mathbf{Q} + \mathbf{I}\Delta_0^0$ via une formule Δ_0 . ★

PREUVE. La fonction de couplage est définie par la formule

$$F(x, y, r) \equiv r \times 2 = (x + y)(x + y + 1) + 2y.$$

Montrons l'existence. Par le lemme 2.8, soit $2|(x + y)$, soit $2|(x + y + 1)$. Donc, toujours par le lemme 2.8, $2|(x + y)(x + y + 1) + 2y$. Donc, $\exists r F(x, y, r)$. L'unicité est claire par injectivité de la multiplication.

Nous avons montré dans l'exercice 2-3.7 que la fonction de couplage ainsi définie était bijective et croissante sur chacun de ses paramètres. On laisse au lecteur le soin de vérifier que cette preuve se formalise dans $\mathbf{I}\Delta_0^0$. Les projections sont définies par les deux formules suivantes :

$$P_1(z, r) \equiv \exists x \leq z \langle x, r \rangle = z, \quad P_2(z, r) \equiv \exists y \leq z \langle r, y \rangle = z.$$

L'existence et l'unicité découlent du caractère bijectif de la fonction de couplage. ■

3. Hiérarchies d'induction

Afin de continuer plus loin notre démonstration du fait que les fonctions primitives récurrentes sont prouvablement calculables dans RCA_0 , nous avons besoin d'examiner quelques conséquences de l'induction, utilisées couramment en mathématique sans que l'on y réfléchisse.

1. Le principe de collection bornée : tout ensemble fini d'entiers possède une borne supérieure.
2. Le principe de minimum : tout ensemble non vide d'entiers admet un plus petit élément.

En présence d'une induction restreinte, chacun des deux principes ci-dessus l'est lui aussi. Nous allons comprendre de quelle manière la restriction de l'induction impacte ces principes, et étudier leurs différents liens.

Notation

Soit $n > 0$. On dénote par $\text{I}\Sigma_n^0$ (resp. $\text{II}\Pi_n^0$) le schéma d'axiomes d'induction $(F(0) \wedge \forall x (F(x) \rightarrow F(x+1))) \rightarrow \forall x F(x)$ restreint aux formules Σ_n^0 (resp. Π_n^0).

3.1. Le schéma de collection bornée

Le principe de collection bornée — tout ensemble fini d'entiers admet une borne supérieure — est évident quand on définit par « ensemble fini » les ensembles d'éléments qui justement sont bornés : il s'agit alors d'une tautologie. Le principe devient moins évident quand on définit les ensembles finis comme étant ceux que l'on peut mettre en bijection avec $\{0, 1, \dots, n\}$ pour un certain n . On appelle *axiome de collection bornée* pour une formule $F(x, y)$ l'énoncé :

$$\forall n ((\forall x < n \exists y F(x, y)) \rightarrow \exists b \forall x < n \exists y < b F(x, y)). \quad (13)$$

Notation

Soit $n > 0$. On désigne par $\text{B}\Sigma_n^0$ (resp. $\text{BII}\Pi_n^0$) le schéma d'axiomes de collection bornée restreint aux formules Σ_n^0 (resp. Π_n^0).

Dans les modèles non standard. Soit M un modèle non standard, et soit un entier non standard $a \in M$. Nous pouvons définir une fonction f de $\{b : b < a\}$ vers M qui ne soit pas bornée dans M (on peut même définir une bijection, M étant dénombrable). Bien entendu, si M est un modèle du schéma de collection bornée, une telle fonction ne peut être définie par une formule de l'intérieur de M . À l'inverse, si M n'est pas modèle du

schéma de collection bornée pour les formules Σ_n^0 , cela signifie qu'il existe un élément a et une formule $\Sigma_n^0 F(x, y)$ tels que pour tout $x < a$ il existe au moins un élément y pour lequel $F(x, y)$ est vrai, et tel que ces éléments y sont non bornés dans M .

Clôture des formules Σ_n^0 et Π_n^0 par quantifications bornées. De nombreuses preuves font appel au fait que les différents niveaux de la hiérarchie arithmétique sont clos par quantification bornée.

Ainsi, pour une formule Σ_1^0 de la forme $\exists y F(x, y)$ avec $F(x, y) \Delta_0^0$, la formule $\exists a \forall y < a \exists y F(x, y)$ est elle aussi considérée comme étant Σ_1^0 . Cela a été démontré via la proposition 9-3.3 et vient du fait que l'on peut la récrire sous la forme $\exists a \exists b \forall y < a \exists y < b F(x, y)$. L'équivalence entre les deux formules nécessite cependant l'axiome de collection bornée, et n'est donc pas garantie dans les modèles ne le vérifiant pas pour la formule F . Nous montrons ici que cet axiome est suffisant pour garantir la proposition 9-3.3.

Proposition 3.1 (Parsons [171]). Soient $F_1(\bar{a}, x), F_2(\bar{a}, x)$ et $F(\bar{a}, x)$ des formules Σ_n^0 (resp Π_n^0) pour $n > 0$. Alors, chacune des trois formules suivantes est prouvablement équivalente dans $\mathbf{Q} + \mathbf{I}\Delta_0^0 + \mathbf{B}\Sigma_n^0$, ainsi que dans $\mathbf{Q} + \mathbf{I}\Delta_0^0 + \mathbf{B}\Pi_n^0$, à une formule Σ_n^0 (resp. Π_n^0) :

- (i) $F_1(\bar{a}, x) \wedge F_2(\bar{a}, x), F_1(\bar{a}, x) \vee F_2(\bar{a}, x)$;
- (ii) $\exists x < b F(\bar{a}, x), \forall x < b F(\bar{a}, x)$;
- (iii) $\exists x F(\bar{a}, x)$ (resp. $\forall x F(\bar{a}, x)$). ★

PREUVE. Soient

$$F(\bar{a}, x) \equiv \exists y G(\bar{a}, x, y), \quad F_1(\bar{a}, x) \equiv \exists y G_1(\bar{a}, x, y) \quad \text{et} \quad F_2(\bar{a}, x) \equiv \exists y G_2(\bar{a}, x, y).$$

Considérons les énoncés suivants :

- $F_1(\bar{a}, x) \wedge F_2(\bar{a}, x) \leftrightarrow \exists y \exists y_1, y_2 < y (G_1(\bar{a}, x, y_1) \wedge G_2(\bar{a}, x, y_2))$; (a)
- $F_1(\bar{a}, x) \vee F_2(\bar{a}, x) \leftrightarrow \exists y (G_1(\bar{a}, x, y) \vee G_2(\bar{a}, x, y))$; (b)
- $\exists x < b F(\bar{a}, x) \leftrightarrow \exists y \exists x < b G(\bar{a}, x, y)$; (c)
- $\forall x < b F(\bar{a}, x) \leftrightarrow \exists z \forall x < b \exists y < z G(\bar{a}, x, y)$; (d)
- $\exists x F(\bar{a}, x) \leftrightarrow \exists z \exists x < z \exists y < z G(\bar{a}, x, y)$. (e)

À présent, si F, F_1, F_2 sont Σ_1^0 avec $G, G_1, G_2 \Delta_0^0$, alors les quatre équivalences (a) (b) (c) (e) sont aisément démontrables dans $\mathbf{I}\Delta_0^0$. L'équivalence (d) découle quant à elle directement de l'axiome de collection bornée pour G qui est Δ_0^0 . Donc, les formules de (i), (ii), (iii) sont prouvablement équivalentes à des formules Σ_1^0 dans $\mathbf{B}\Sigma_1^0$ et dans $\mathbf{B}\Pi_1^0$. Par passage à la négation, les formules de (i), (ii), (iii) sont prouvablement équivalentes à des formules Π_1^0 dans $\mathbf{B}\Sigma_1^0$ et dans $\mathbf{B}\Pi_1^0$, pour le cas où F, F_1, F_2 sont Π_1^0 .

Supposons la proposition vraie pour les cas Σ_n^0 et Π_n^0 . Alors, si F, F_1, F_2 sont des formules Σ_{n+1}^0 avec $G, G_1, G_2 \Pi_n^0$, de la même manière $\mathbf{B}\Sigma_{n+1}^0$ ou $\mathbf{B}\Pi_{n+1}^0$

montrent les équivalences (a) (b) (c) (d) (e). Comme $\mathbf{B}\Sigma_{n+1}^0$ ou $\mathbf{B}\Pi_{n+1}^0$ démontrent $\mathbf{B}\Sigma_n^0$, et par hypothèse d'induction sur G, G_1, G_2 , les formules de (i), (ii), (iii) sont prouvablement équivalentes à des formules Σ_{n+1}^0 dans $\mathbf{B}\Sigma_{n+1}^0$ ou dans $\mathbf{B}\Pi_{n+1}^0$. Par passage à la négation, les trois formules de (i), (ii), (iii) sont prouvablement équivalentes à des formules Π_{n+1}^0 dans $\mathbf{B}\Sigma_{n+1}^0$ ou dans $\mathbf{B}\Pi_{n+1}^0$, dans le cas où F, F_1, F_2 sont Π_{n+1}^0 . ■

L'induction implique le schéma de collection. Le schéma de collection bornée ne fait pas partie des axiomes usuels de l'arithmétique. Nous voyons ici comment il découle de l'induction. Nous commençons d'abord par montrer que le schéma de collection pour les formules Π_n^0 permet de le montrer pour les formules Σ_{n+1}^0 .

Proposition 3.2 (Paris et Kirby [170]). Soit $n > 0$. Alors,

$$\mathbf{Q} + \mathbf{I}\Delta_0^0 \vdash \mathbf{B}\Pi_n^0 \leftrightarrow \mathbf{B}\Sigma_{n+1}^0. \quad \star$$

PREUVE. L'implication $\mathbf{B}\Sigma_{n+1}^0 \rightarrow \mathbf{B}\Pi_n^0$ est immédiate. Supposons $\mathbf{B}\Pi_n^0$. Soit $\exists z F(x, y, z)$ une formule Σ_{n+1}^0 avec $F \Pi_n^0$, et soit $a \in \mathbb{N}$ tel que

$$\forall x < a \exists y (\exists z F(x, y, z)).$$

On a alors $\forall x < a \exists v \exists y, z < v F(x, y, z)$. En utilisant $\mathbf{B}\Pi_n^0$, par la proposition 3.1 soit $G(x, v)$ une formule Π_n^0 équivalente à $\exists y, z < v F(x, y, z)$. Alors, par $\mathbf{B}\Pi_n^0$ on a $\exists b \forall x < a \exists v < b G(x, v)$, qui est alors équivalent à $\exists b \forall x < a \exists v < b \exists y, z < v F(x, y, z)$, ce qui implique

$$\exists b \forall x < a \exists y < b (\exists z F(x, y, z)). \quad \blacksquare$$

Voyons à présent comment montrer le schéma de collection à l'aide du schéma d'induction.

Théorème 3.3 (Paris et Kirby [170])

Soit $n > 0$. Alors, $\mathbf{Q} \vdash \mathbf{I}\Sigma_n^0 \rightarrow \mathbf{B}\Sigma_n^0$.

PREUVE. Par induction sur n . Comme $\mathbf{Q} + \mathbf{I}\Delta_0^0 \vdash \mathbf{B}\Sigma_{n+1}^0 \leftrightarrow \mathbf{B}\Pi_n^0$, montrons $\mathbf{Q} \vdash \mathbf{I}\Sigma_{n+1}^0 \rightarrow \mathbf{B}\Pi_n^0$. Soit $a \in \mathbb{N}$ et soit $F(x, y)$ une formule. Considérons les formules

$$G \equiv \forall x < a \exists y F(x, y) \quad \text{et} \quad H(a') \equiv \exists b \forall x < a' \exists y < b (a' \leq a \rightarrow F(x, y)).$$

Supposons G . Dans le cas où $F(x, y)$ est Δ_0^0 , alors $H(a')$ est Σ_1^0 . Or, $H(0)$ est trivialement vrai, et l'on déduit par G que $H(a') \rightarrow H(a' + 1)$ (il y a ici un argument que nous laissons au lecteur). Par $\mathbf{I}\Sigma_1^0$, on a donc $\forall a' H(a')$, et donc $H(a)$, ce qui implique $\exists b \forall x < a \exists y < b F(x, y)$.

Supposons à présent $\mathbf{Q} \vdash \mathbf{I}\Sigma_n^0 \rightarrow \mathbf{B}\Sigma_n^0$ avec $F(x, y)$ une formule Π_n^0 . Alors, H est équivalente à une formule Σ_{n+1}^0 par $\mathbf{B}\Sigma_n^0$. On procède alors comme dans le paragraphe précédent. Ainsi, pour tout $n \in \omega$, on a $\mathbf{Q} \vdash \mathbf{I}\Sigma_n^0 \rightarrow \mathbf{B}\Sigma_n^0$. ■

3.2. Le schéma de minimum

Le schéma de minimum nous dit informellement que tout ensemble d'entiers non vide a un plus petit élément. Formellement, on appelle *axiome de minimum* pour une formule $F(x)$ l'énoncé :

$$\exists x F(x) \rightarrow \exists x (F(x) \wedge \forall y < x \neg F(y)). \quad (14)$$

Notation

Soit $n > 0$. On dénote par $\mathbf{L}\Sigma_n^0$ (resp. $\mathbf{L}\Pi_n^0$) le schéma d'axiomes de minimum restreint aux formules Σ_n^0 (resp. Π_n^0).

Dans les modèles non standard. Dans un modèle non standard M , une *coupure* est un segment initial strict de M , non vide et clos par successeur. Par exemple, la partie standard ω d'un modèle non standard est toujours une coupure. Si un élément a est dans le complémentaire d'une coupure, c'est aussi nécessairement le cas pour $a - 1$, car une coupure est close par successeur. On en déduit que si une coupure est définissable par une formule F , cela contredit directement le schéma de minimum pour la formule $\neg F$. Il s'agit en fait d'une condition nécessaire et suffisante. En effet, si $A = \{x : F(x)\}$ est un ensemble non vide qui n'a pas de plus petit élément, alors l'ensemble des éléments strictement plus petits que tout élément de A est nécessairement une coupure. On en déduit le principe suivant.

Principe de débordement (overspill)

L'ensemble des entiers standard n'est jamais définissable dans un modèle qui respecte le schéma de minimum. En particulier, si une formule est vérifiée par tous les entiers standard, elle le sera nécessairement aussi pour un entier non standard.

L'induction équivaut au schéma de minimum. Nous voyons à présent que le schéma de minimum est, à peu de chose près, une simple reformulation de l'induction : si l'ensemble $\{x : F(x)\}$ est non vide et n'a pas de plus petit élément, c'est que l'induction rate pour la formule qui définit la coupure que l'on peut créer à partir de $F(x)$.

Proposition 3.4. Soit $n > 0$. Alors, $\mathbf{Q} \vdash \mathbf{I}\Sigma_n^0 \leftrightarrow \mathbf{L}\Pi_n^0$ et $\mathbf{Q} \vdash \mathbf{I}\Pi_n^0 \leftrightarrow \mathbf{L}\Sigma_n^0$. $\star$

PREUVE. On montre l'équivalence $\mathbf{I}\Sigma_n^0 \leftrightarrow \mathbf{L}\Pi_n^0$, l'autre se montrant de manière identique.

Supposons $\neg \mathbf{I}\Sigma_n^0$. Soit $F(x)$ une formule Σ_n^0 telle que $F(0), F(x) \rightarrow F(x+1)$, mais telle que l'ensemble $A = \{x : \neg F(x)\}$ est non vide. Par contraposée, $\neg F(x) \rightarrow \neg F(x-1)$. Donc, A n'a pas de plus petit élément. On a ainsi $\neg \mathbf{L}\Pi_n^0$.

Supposons à présent $\neg \mathbf{L}\Pi_n^0$, et supposons $\mathbf{I}\Sigma_n^0$ par l'absurde. Soit $F(x)$ une formule Π_n^0 telle que l'ensemble $A = \{x : F(x)\}$ est non vide et n'a pas de plus petit élément. Soit $G(x) \equiv \forall y \leq x \neg F(y)$. Par $\mathbf{I}\Sigma_n^0$ qui implique $\mathbf{B}\Sigma_n^0$, $G(x)$ est prouvablement équivalente à une formule Σ_n^0 . Notons que $\neg F(0)$, car sinon A aurait un plus petit élément. Donc, $G(0)$. Supposons $G(x)$. Si l'on avait $F(x+1)$, alors A aurait un plus petit élément. Par suite, $\neg F(x+1)$, ainsi que $G(x+1)$. On a ainsi $G(0) \wedge (G(x) \rightarrow G(x+1))$, mais $G(x)$ n'est vérifié pour aucun élément de A , qui est non vide. On a donc $\neg \mathbf{I}\Sigma_n^0$. On a là une contradiction. $\blacksquare$

Le théorème suivant est particulièrement intéressant : en utilisant les propriétés des entiers, on montre que l'induction pour les formules Σ_n^0 est équivalente à l'induction pour les formules Π_n^0 .

Théorème 3.5 (Paris et Kirby [170])

Soit $n > 0$. Alors, $\mathbf{Q} \vdash \mathbf{I}\Sigma_n^0 \leftrightarrow \mathbf{I}\Pi_n^0$.

PREUVE. Montrons $\mathbf{Q} \vdash \mathbf{I}\Sigma_n^0 \rightarrow \mathbf{I}\Pi_n^0$. Supposons que $\mathbf{I}\Pi_n^0$ échoue, mais pas $\mathbf{I}\Delta_0^0$ (afin de pouvoir utiliser les faits basiques sur les entiers). Soit $F(x)$ une formule Π_n^0 telle que $F(0)$ et $\forall x(F(x) \rightarrow F(x+1))$, mais $\neg F(a)$ pour un entier $a > 0$. Soit $G(y)$ la formule $\exists x (a = x + y \wedge \neg F(x))$. Remarquons que la formule $G(y)$ est équivalente à une formule Σ_n^0 , en décalant $\ll a = x + y \gg$ vers sa partie Δ_0^0 . De plus, $G(0)$ est vrai et $G(a)$ est faux. Soit y tel que $G(y)$ est vrai. En particulier, il existe un x tel que $a = x + y$ et $\neg F(x)$. Nécessairement, $y < a$, donc $x > 0$; or, $a = (x-1) + (y+1)$ et par hypothèse, $\neg F(x) \rightarrow \neg F(x-1)$, donc $G(y+1)$ est vrai. Comme $G(0)$ et $\forall y (G(y) \rightarrow G(y+1))$ et $\neg G(a)$, alors $\mathbf{I}\Sigma_n^0$ échoue.

Montrons $\mathbf{Q} \vdash \mathbf{I}\Pi_n^0 \rightarrow \mathbf{I}\Sigma_n^0$. Supposons que $\mathbf{I}\Sigma_n^0$ échoue mais pas $\mathbf{I}\Delta_0^0$. Soit $F(x)$ une formule Σ_n^0 telle que $F(0)$ et $\forall x(F(x) \rightarrow F(x+1))$, mais $\neg F(a)$ pour un entier $a > 0$. Soit $H(y)$ la formule $\forall x (a = x + y \rightarrow \neg F(x))$. Comme précédemment, $H(y)$ est équivalente à une formule Π_n^0 . De plus, $H(0)$ est vrai et $H(a)$ est faux. On montre également $H(y) \rightarrow H(y+1)$. Ainsi, $H(0)$ et $\forall y (H(y) \rightarrow H(y+1))$ et $\neg H(a)$, donc $\mathbf{I}\Pi_n^0$ échoue. $\blacksquare$

Exercice 3.6. (★) (Hájek et Pudlák [81]). On appelle *schéma d'induction ordonnée* pour toute formule $F(x)$ l'énoncé

$$\forall x((\forall y < x F(y)) \rightarrow F(x)) \rightarrow \forall x F(x).$$

Montrer que pour tout $n \in \omega$, $\mathbf{Q}$ prouve l'équivalence entre $\mathbf{I}\Sigma_n^0$ (resp. $\mathbf{I}\Pi_n^0$) et le schéma d'induction ordonnée pour les formules Σ_n^0 (resp. Π_n^0). $\diamond$

3.3. Collection et induction Δ_n^0

Nous avons vu que $\mathbf{B}\Sigma_n^0$ se déduisait de $\mathbf{I}\Sigma_n^0$. Nous voyons à présent que sous réserve d'avoir un minimum d'induction ($\mathbf{I}\Delta_0^0$), le schéma de collection pour toutes les formules implique le schéma d'induction pour toutes les formules. Plus précisément, on dispose du résultat qui suit.

Théorème 3.7 (Paris et Kirby [170])

Soit $n > 0$. Alors, $\mathbf{Q} + \mathbf{I}\Delta_0^0 \vdash \mathbf{B}\Sigma_{n+1}^0 \rightarrow \mathbf{I}\Sigma_n^0$.

PREUVE. Par induction sur n .

Supposons $\mathbf{B}\Sigma_{n+1}^0$. Soit $F(x) \equiv \exists y G(x, y)$ pour G une formule Π_{n-1}^0 (ou Δ_0^0 si $n = 1$). Supposons $F(0)$ et $\forall x(F(x) \rightarrow F(x+1))$. Soit $a \in \mathbb{N}$. Montrons $F(a)$. Nous avons

$$\forall x \leq a (\exists y G(x, y) \rightarrow \exists u G(x+1, u)).$$

Autrement dit,

$$\forall x \leq a \exists u (\exists y G(x, y) \rightarrow G(x+1, u)).$$

Par $\mathbf{B}\Pi_n^0$ (équivalent à $\mathbf{B}\Sigma_{n+1}^0$), il existe $b \in \mathbb{N}$ tel que

$$\forall x \leq a \exists u \leq b (\exists y G(x, y) \rightarrow G(x+1, u)),$$

ce qui implique

$$\forall x \leq a \exists u \leq b (\exists y \leq b G(x, y) \rightarrow G(x+1, u)),$$

et donc

$$\forall x \leq a (\exists y \leq b G(x, y) \rightarrow \exists u \leq b G(x+1, u)).$$

Soit $H(x) \equiv x \leq a \rightarrow (\exists u \leq b G(x, u))$. Si $n = 1$, la formule $G(x, u)$ est Δ_0^0 et donc aussi la formule $H(x)$. Par $\mathbf{I}\Delta_0^0$, on a donc $H(x)$ pour tout x . Si $n > 1$, la formule $G(x, u)$ est Π_{n-1}^0 . Par $\mathbf{B}\Sigma_{n+1}^0$, la formule $H(x)$ est donc équivalente à une formule Π_{n-1}^0 . Par hypothèse d'induction, on a $\mathbf{I}\Sigma_{n-1}^0$ (qui implique $\mathbf{I}\Pi_{n-1}^0$ par le théorème 3.5), et l'on a donc $H(x)$ pour tout x . Dans tous les cas, on a $H(x)$ pour tout x . En particulier, $H(a)$ est vrai, donc $\exists u \leq b G(a, u)$, et donc $F(a)$. ■

Il est possible d'affiner la seconde implication, et montrer que $\mathbf{B}\Sigma_n^0$ implique l'induction pour les formules prouvablement Δ_n^0 .

Notation

Soit $n > 0$. On note $\mathsf{I}\Delta_n^0$ le schéma d'axiomes

$$\forall x (F(x) \leftrightarrow G(x)) \rightarrow ((F(0) \wedge F(n) \rightarrow F(n+1)) \rightarrow \forall n F(n))$$

restreint aux formules F, G avec $F \in \Sigma_n^0$ et $G \in \Pi_n^0$.

Théorème 3.8 (Hájek et Pudlák [81])

Soit $n > 0$. Alors, $\mathsf{Q} + \mathsf{I}\Delta_0^0 \vdash \mathsf{B}\Sigma_n^0 \rightarrow \mathsf{I}\Delta_n^0$.

PREUVE. Supposons $\mathsf{B}\Sigma_n^0$. Soient $\exists y F(x, y)$ et $\forall x G(x, y)$ des formules respectivement Σ_n^0 et Π_n^0 telles que $\forall x (\exists y F(x, y) \leftrightarrow \forall y G(x, y))$. Supposons $\exists y F(0, y)$ et $\forall x (\exists y F(x, y) \rightarrow \exists y F(x+1, y))$.

Soit $a \in \mathbb{N}$. Montrons $\exists y F(a, y)$. On a

$$\forall x \exists y (F(x, y) \vee \neg G(x, y)).$$

Donc, par $\mathsf{B}\Sigma_n^0$, il existe $b \in \mathbb{N}$ tel que

$$\forall x \leq a \exists y \leq b (F(x, y) \vee \neg G(x, y)).$$

Soit $H(x) \equiv (x \leq a \rightarrow (\exists y \leq b F(x, y)))$. Par hypothèse sur F et G , on a $H(0)$ et $H(x) \rightarrow H(x+1)$. Par $\mathsf{B}\Sigma_n^0$, la formule $H(x)$ est équivalente à une formule Π_{n-1}^0 . Comme $\mathsf{B}\Sigma_n^0 \rightarrow \mathsf{I}\Sigma_{n-1}^0$ et $\mathsf{I}\Sigma_{n-1}^0 \leftrightarrow \mathsf{I}\Pi_{n-1}^0$, alors on a $H(x)$ pour tout x . En particulier, $\exists y F(a, y)$. ■

Ainsi, $\mathsf{B}\Sigma_n^0$ se situe entre $\mathsf{I}\Sigma_n^0$ et $\mathsf{I}\Delta_n^0$. La question de savoir si $\mathsf{I}\Delta_n^0$ implique en retour $\mathsf{B}\Sigma_n^0$ est restée ouverte pendant plus d'une décennie, avant d'être résolue par Slaman en 2002, en utilisant toutefois la théorie $\mathsf{Q} + \mathsf{I}\Sigma_1^0$ comme théorie de base.

Théorème 3.9 (Slaman [204])

Soit $n > 0$. Alors, $\mathsf{Q} + \mathsf{I}\Sigma_1^0 \vdash \mathsf{B}\Sigma_n^0 \leftrightarrow \mathsf{I}\Delta_n^0$.

La preuve de l'implication $\mathsf{I}\Delta_n^0 \rightarrow \mathsf{B}\Sigma_n^0$ fait appel à des techniques avancées de codage dans les modèles non standard de l'arithmétique. Il n'existe pas à ce jour de preuve syntaxique de cette implication.

3.4. Schéma récapitulatif

Nous avons donc pour l'instant dans $\mathsf{Q} + \mathsf{I}\Delta_0^0$ (et donc dans RCA_0) la hiérarchie d'induction suivante, visible sur la figure 3.10.

Il est possible de montrer que les implications réciproques de la figure ne tiennent pas. Cela se fait via la construction de modèles non standard, au

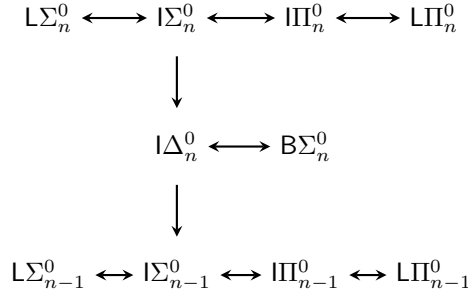


FIGURE 3.10 – *Hiérarchie d'induction. Les flèches indiquent une implication dans $\mathbf{Q} + \text{I}\Sigma_1^0$.*

sein desquels par exemple $\text{B}\Sigma_n^0$ est valide, mais pas $\text{I}\Sigma_n^0$. Le lecteur qui souhaite en apprendre davantage pourra consulter [109] ou [81].

4. Fonctions primitives récursives et RCA_0

Nous avons à présent les éléments nécessaires pour démontrer que les fonctions primitives récursives sont prouvablement calculables dans RCA_0 . La preuve passe par un codage des suites finies par des entiers, via une technique similaire à celle utilisée pour la fonction β de Gödel (voir le lemme 9-3.6), mais ne reposant pas sur le théorème des restes chinois. L'unicité de la décomposition des nombres en facteurs premiers sera un élément clef de notre codage, plus particulièrement le lemme de Gauss. On rappelle que deux entiers naturels a, b sont premiers entre eux si 1 est leur seul diviseur commun. Il s'agit d'une formule Δ_0 .

Lemme 4.1 (Lemme de Gauss). Le système $\mathbf{Q} + \text{I}\Delta_0^0$ montre que pour tous nombres p, a, b tels que p et b sont premiers entre eux, si $p|ab$, alors $p|a$.★

PREUVE. Fixons p, b premiers entre eux. Supposons par l'absurde que le lemme de Gauss est faux, i.e. que l'ensemble $\{a \neq 0 : p|ab \text{ et } \neg p|a\}$ est non vide. Par schéma de minimum pour les formules Δ_0 , soit a le plus petit élément de cet ensemble. Montrons $a < p$. Supposons par l'absurde $a \geq p$. Alors, par la division euclidienne (voir l'exercice 2.7), il existe q, r avec $r < p$ tels que $a = pq + r$. Comme $\neg p|a$, alors $r \neq 0$. Comme $p|(pq + r)b$ alors $p|pqb + rb$. Or, $p|pqb$, donc $p|rb$. Cependant, comme $r < p$, on a $\neg p|r$, ce qui contredit la minimalité de a . Donc, $a < p$. À présent, soit $p = aq + r$ avec $0 \leq r < a$. Montrons $r \neq 0$. Supposons $p = aq$. Comme $\neg p|a$,

alors $q > 1$. Comme $p|ab$, alors $aq|ab$, et donc $q|b$. Donc, q est un diviseur commun différent de 1 de b et p , ce qui contredit nos hypothèses. Ainsi, $r \neq 0$. On a alors $bp = baq + br$, donc $br = bp \dot{-} baq$. Comme $p|bp$ et $p|baq$, alors $p|br$. Or, $0 < r < a < p$, donc $\neg p|r$, ce qui contredit la minimalité de a . ■

Notre codage des listes repose sur la capacité à générer des suites de nombres tous deux à deux premiers, de la manière suivante.

Lemme 4.2. Soit x fixé, et soit a un entier multiple de tous les entiers i pour $1 \leq i \leq x$. Le système $\mathbf{Q} + \mathbf{I}\Delta_0^0$ montre que pour $i, j \leq x$ avec $i \neq j$ les nombres $a(i+1) + 1$ et $a(j+1) + 1$ sont premiers entre eux. ★

PREUVE. Supposons par l'absurde qu'un nombre premier p divise les deux entiers $a(i+1) + 1$ et $a(j+1) + 1$ avec $i < j$. Alors aussi, p divise

$$a(j+1) + 1 - a(i+1) + 1 = a(j \dot{-} i).$$

Comme p est premier, d'après le lemme de Gauss, p divise a ou p divise $j \dot{-} i$. Par hypothèse sur a , dans tous les cas, p divise a . Donc, p divise $a(i+1)$. Comme p divise également $a(i+1) + 1$, alors p divise $a(i+1) + 1 \dot{-} a(i+1) = 1$, ce qui est une contradiction. Donc, les nombres $a(i+1) + 1$ et $a(j+1) + 1$ sont premiers entre eux. ■

L'idée de codage de nos listes est la suivante. Étant donné $x_0, x_1, \dots, x_{n-1}$, nous commençons par choisir a multiple de tous les entiers z compris (au sens large) entre 1 et $\max\{\langle i, x_i \rangle : i < n\}$. Notre liste est alors codée par le couple (a, b) tel que $b = \prod_{i < n} (a(\langle i, x_i \rangle + 1) + 1)$. Étant donné (a, b) , le i -ième élément de notre liste est donné par l'unique $z \leq a$ tel que $\langle i, z \rangle \leq a$ et tel que $(a(\langle i, x_i \rangle + 1) + 1) | b$.

Notation

Pour un entier a , on note p_i^a l'entier $a(i+1) + 1$.

Définition 4.3. Un couple (a, b) code pour une suite de taille n si

$$\forall i < n \exists! x \leq a \langle i, x \rangle \leq a \wedge p_{\langle i, x \rangle}^a | b.$$

◇

Notation

Pour un couple (a, b) qui code pour une suite de taille n , et pour $i < n$, on note $(a, b)_i$ l'unique entier $x \leq a$ tel que $\langle i, x \rangle \leq a \wedge p_{\langle i, x \rangle}^a | b$.

Notre première utilisation de l'induction Σ_1^0 sera pour montrer l'existence d'un entier a multiple de suffisamment d'éléments pour coder nos suites.

Lemme 4.4. Le système RCA_0 montre que, pour tout $x \in \mathbb{N}$, il existe un plus petit entier $y \in \mathbb{N}$ qui est multiple de tous les entiers i pour $1 \leq i \leq x$.★

PREUVE. Montrons l'existence. C'est clair pour $x = 0$. Supposons l'existence pour x . Soit y un multiple de tous les entiers plus petits que x . On montre dans $\mathbf{Q} + \mathbf{I}\Delta_0^0$ que les diviseurs de y sont aussi des diviseurs de $y \times (x + 1)$, et que $x + 1$ est diviseur de $y \times (x + 1)$. Par $\mathbf{I}\Sigma_1^0$, l'existence est assurée pour tout x . En particulier, pour x fixé, l'ensemble

$$\{y : \forall 1 \leq i \leq x \ i|y\}$$

est non vide. On applique enfin le schéma de minimum pour les formules Δ_0 pour obtenir le plus petit élément. ■

Nous pouvons à présent passer au codage des suites finies.

Lemme 4.5

Supposons que (a, b) code pour une suite de taille n . Alors, RCA_0 montre que pour tout r il existe un couple (a', b') qui code pour une suite de taille $n + 1$, tel que $(a', b')_n = r$ et tel que $\forall i < n \ (a', b')_i = (a, b)_i$. ★

PREUVE. Montrons d'abord l'énoncé suivant : si (a, b) code pour une suite de taille n , alors pour tout a' tel que les diviseurs de a sont tous des diviseurs de a' , il existe b' tel que $(a', b')_i = (a, b)_i$ pour tout $i < n$.

Supposons que ce soit le cas pour n . Soit (a, b) codant pour une suite de taille $n + 1$ et a' tel que les diviseurs de a sont tous des diviseurs de a' . Alors, (a, b) code aussi pour une suite de taille n . Donc, par hypothèse d'induction, il existe b' tel que (a, b) et (a', b') codent pour la même suite de taille n . Soit b' le plus petit entier de la sorte (schéma de minimisation Δ_0). Alors, (a', b') ne code pas pour une suite de taille $n + 1$ car si $m \times p'_{\langle n, x \rangle} = b'$, alors (a', m) code pour la même suite de taille n que (a, b) — par application du lemme de Gauss et le fait que $p'_{\langle n, x \rangle}$ et $p'_{\langle i, y \rangle}$ soient premiers entre eux pour tout $i < n$ —, ce qui contredit la minimalité de b' . À présent, soit $r = (a, b)_n$ et $b'' = b' p'_{\langle n, r \rangle}$. Toujours en utilisant le lemme de Gauss, on montre que (a, b) et (a', b'') codent pour la même suite de taille $n + 1$.

Montrons à présent le lemme. Soit (a, b) codant pour une suite de taille n , et soit $r \in \mathbb{N}$. D'après la proposition 4.4, soit a' tel que tout $i \leq \max(a, \langle n, r \rangle)$ divise a' . D'après le paragraphe précédent, il existe b' tel que (a', b') code pour la même suite de taille n que (a, b) . Soit $b'' = b' p'_{\langle n, r \rangle}$. Il est clair que $(a', b'')_n = r$. Toujours en utilisant le lemme de Gauss, on montre que $(a', b'')_i = (a', b')_i = (a, b)_i$ pour $i < n$. ■

Passons finalement à la preuve que les fonctions primitives récursives sont prouvablement calculables.

Théorème 4.6

Les fonctions primitives récursives sont prouvablement calculables dans le système RCA_0 .

PREUVE

On laisse au lecteur le soin de montrer que le théorème est vérifié pour les fonctions de base. Le schéma de composition ne présente pas de difficulté particulière : soit $f(\bar{x}) = g(h_1(\bar{x}), \dots, h_k(\bar{x}))$ pour des fonctions $g, h_1, \dots, h_k$ prouvablement calculables via par des formules $G, H_1, \dots, H_k$. Alors, f est prouvablement calculable par la formule

$$F(\bar{x}, r) \equiv \exists y_1, \dots, y_k \ H_1(\bar{x}, y_1) \wedge \dots \wedge H_k(\bar{x}, y_k) \wedge G(y_1, \dots, y_k, r).$$

Nous passons à présent au schéma de récursion primitive. Soit

$$\begin{aligned} f(\bar{x}, 0) &= g(\bar{x}) \\ f(\bar{x}, n+1) &= h(\bar{x}, n, f(\bar{x}, n)), \end{aligned}$$

pour des fonctions g, h prouvablement calculables par des formules G, H . Nous allons créer une formule $F(\bar{x}, n, r)$ telle que :

$$\begin{aligned} \text{RCA}_0 &\vdash \forall \bar{x}, n \ \exists! r \ F(\bar{x}, n, r); \\ \text{RCA}_0 &\vdash \forall \bar{x}, \exists r \ F(\bar{x}, 0, r) \wedge G(\bar{x}, r); \\ \text{RCA}_0 &\vdash \forall \bar{x}, n \ \exists r_1, r_2 \ F(\bar{x}, n, r_1) \wedge F(\bar{x}, n+1, r_2) \wedge H(\bar{x}, n, r_1, r_2). \end{aligned}$$

La formule en question est donnée par

$$F(\bar{x}, n, r) \equiv \begin{aligned} &\exists a, b \ (a, b) \text{ code pour une suite de taille } n+1 \text{ et} \\ &r = (a, b)_n \text{ et } G(\bar{x}, (a, b)_0) \text{ et } \forall i < n \ H(\bar{x}, i, (a, b)_i, (a, b)_{i+1}). \end{aligned}$$

Notons que par BS_1^0 , qui est prouvable dans IS_1^0 , la formule ci-dessus est équivalente à une formule Σ_1^0 .

Montrons l'existence par induction sur n . Il est clair que l'on a $\exists r \ F(\bar{x}, 0, r)$. Supposons $\exists r \ F(\bar{x}, n, r)$ via le couple (a, b) codant pour une suite ayant pour taille $n+1$. Soit r l'unique entier tel que $H(\bar{x}, n, (a, b)_n, r)$.

D'après le lemme 4.5, il existe (a', b') qui code pour une suite ayant pour taille $n+2$, tel que $(a', b')_{n+1} = r$ et tel que $(a', b')_i = (a, b)_i$ pour $i \leq n$. Il est alors clair que l'on a $G(\bar{x}, (a', b')_0) \wedge \forall i < n \ H(\bar{x}, i, (a', b')_i, (a', b')_{i+1})$. De plus, $H(\bar{x}, n, (a', b')_n, (a', b')_{n+1})$. Donc, $F(\bar{x}, n+1, (a', b')_{n+1})$. Par IS_1^0 , l'existence est assurée pour tout n .

Montrons à présent l'unicité. Là encore, elle est claire pour $n=0$. Supposons l'unicité pour n , et supposons que $F(\bar{x}, n+1, r_1)$ est vérifiée via (a_1, b_1) et que $F(\bar{x}, n+1, r_2)$ est vérifiée via (a_2, b_2) . Soient $v_1 = (a_1, b_1)_n$ et $v_2 = (a_2, b_2)_n$. Alors, $F(\bar{x}, n, v_1)$ et $F(\bar{x}, n, v_2)$. Par hypothèse d'induction, on a $v_1 = v_2$. Donc, $H(\bar{x}, n, v_1, (a, b)_{n+1})$ et $H(\bar{x}, n, v_1, (a', b')_{n+1})$. Donc, $(a, b)_{n+1} = (a', b')_{n+1}$, et donc $r_1 = r_2$. ■

5. Le schéma de compréhension bornée

Nous voyons à présent un autre schéma d'axiomes sur lequel il convient de s'arrêter, et que l'on pourrait résumer ainsi : « les ensembles finis existent ». Nous allons voir que lorsque l'induction est restreinte, la formulation précise de cet énoncé, ainsi que sa validité, deviennent non triviales.

5.1. Codage des ensembles finis

À présent que nous pouvons utiliser les fonctions primitives récursives dans RCA_0 , il est d'usage de faire appel à la fonction $x \mapsto 2^x$ pour coder les ensembles finis, en codant l'ensemble $x_0 < x_1 < \dots < x_n$ par l'entier $n = 2^{x_0} + 2^{x_1} + \dots + 2^{x_n}$. Via ce codage, on récupère aisément à partir d'un entier n l'ensemble qu'il code via le lemme suivant.

Lemme 5.1 (Hájek et Pudlák [81])

Le système RCA_0 prouve que pour tous $n, x \in \mathbb{N}$, il existe un unique triplet (u, v, w) avec $u \leq y$, $v < 1$ et $w < 2^x$ tel que :

$$y = 2^{x+1}u + 2^xv + w. \quad \star$$

Le lemme 5.1 se prouve à l'aide de deux applications de la division euclidienne.

Définition 5.2. On appelle *x-ième bit de y* l'unique entier $v \in \{0, 1\}$ de la décomposition précédente. On notera $x \in y$ si $v = 1$, et $x \notin y$ sinon. Un entier y est le *code canonique* de l'ensemble A si pour tout $x \in \mathbb{N}$, $x \in A$ ssi $x \in y$. Un ensemble est *codé* s'il possède un code canonique. $\diamond$

Notons que la relation « $x \in y$ » est Δ_0^0 , car elle s'écrit

$$\exists u \leq y \exists w < 2^x (y = 2^{x+1}u + 2^x + w).$$

Le code canonique d'un ensemble, s'il existe, est unique par le lemme 5.1. Il est aisé de montrer dans RCA_0 qu'un ensemble X est fini si, et seulement si, il possède un code canonique (voir l'exercice 5.4 à venir). Attention toutefois, par « fini », il faut comprendre fini dans le modèle.

Il est possible d'extraire dans RCA_0 toutes les informations standard des codes canoniques, comme leur taille, leur maximum ou leur minimum. On laisse au lecteur le soin de montrer que les fonctions correspondantes sont bien primitives récursives.

5.2. Notions d'infini et RCA_0

Qu'est-ce qu'un ensemble infini ? La définition la plus intuitive, et que nous adopterons par défaut, est celle de l'absence de majorant.

Définition 5.3. Un ensemble A est *infini* si pour tout x il existe $y > x$ tel que $y \in A$. Un ensemble est *fini* s'il n'est pas infini. $\diamond$

Dans les modèles non standard. Attention ! La notion de fini et d'infini est toujours relative au modèle. Si M est un modèle non standard, et si $a \in M$ est un entier non standard, l'ensemble $\{x \in M : x < a\}$ est infini de l'extérieur du modèle, mais fini dans le modèle. Attention également, de l'extérieur du modèle, on peut construire des ensembles bornés dans le modèle (et donc finis), mais qui n'y appartiennent pas ! C'est le cas par exemple du segment initial constitué des entiers standard.

Exercice 5.4. Montrer dans RCA_0 qu'un ensemble est fini si, et seulement si, il est canoniquement codé :

- (i) $\text{RCA}_0 \vdash \forall y \exists X (y \text{ est le code canonique de } X)$;
- (ii) $\text{RCA}_0 \vdash \forall X \forall b (\forall x (x \in X \rightarrow x < b) \rightarrow X \text{ est canoniquement codé})$. $\diamond$

Définitions équivalentes. On pourrait considérer qu'un ensemble $A \subseteq \mathbb{N}$ est infini s'il est en bijection avec $\mathbb{N}$. On pourrait également considérer qu'un ensemble est infini s'il contient des « blocs » de taille arbitraire, et RCA_0 est suffisamment puissant pour montrer l'équivalence de ces définitions.

Proposition 5.5. Le système RCA_0 prouve les équivalences suivantes, pour tout A :

- (1) l'ensemble A est infini ;
- (2) il existe une fonction bijective croissante $f : \mathbb{N} \rightarrow A$;
- (3) pour tout z , le segment initial de A de cardinalité exactement z possède un code canonique. $\star$

PREUVE. (1) $\rightarrow$ (2). Soit $g(x)$ le plus petit $y > x$ tel que $y \in X$. La fonction est Σ_1^0 , et totale car A est infini, donc est Δ_1^0 . La fonction g existe par compréhension Δ_1^0 . Soit a le plus petit élément de A (qui existe par la proposition 3.4). Alors, la fonction $f : \mathbb{N} \rightarrow A$ définie par $f(0) = a$ et $f(n+1) = g(f(n))$ existe par récursion primitive, et satisfait (2).

(2) $\rightarrow$ (3). Soit $f : \mathbb{N} \rightarrow A$ une fonction bijective croissante, et soit $z \in \mathbb{N}$. Par compréhension Δ_1^0 , l'ensemble $A_z = \{f(x) : x < z\}$ existe, et par l'exercice 5.4, l'ensemble A_z possède un code canonique.

(3) $\rightarrow$ (1). Soit $x \in \mathbb{N}$. Par (3), le code canonique q du segment initial de A de cardinalité $x+1$ existe. L'élément maximal de q est un élément de A plus grand que x . Ainsi, A est infini. $\blacksquare$

5.3. Schéma de compréhension bornée

Nous avons vu que dans RCA_0 un ensemble est fini si, et seulement si, il a un code canonique. Attention toutefois, il s'agit des ensembles dont on peut montrer l'existence dans le modèle, via le schéma de compréhension ! En revanche, étant donné une formule $F(x)$ arbitraire et une borne b , l'ensemble $\{x < b : F(x)\}$ n'existe pas nécessairement si F est trop complexe. L'ensemble existera bien sûr quand F est prouvablement Δ_1^0 . Il est en fait possible de montrer, grâce à l'induction Σ_1^0 , que ce sera également le cas si F est Σ_1^0 . Il s'agit en fait d'une condition équivalente : $\text{Q} + \text{ID}_0^0$ montre que IS_1^0 est équivalent au fait que l'ensemble $\{x < b : F(x)\}$ est canoniquement codé pour toute formule $F \in \Sigma_1^0$. Cette équivalence se propage à tous les niveaux de la hiérarchie.

Souvenons-nous de la notation $y \in n$ de la définition 5.2. On appelle *axiome de compréhension bornée* pour une formule $F(x)$ l'énoncé

$$\forall t \exists n \forall y (y \in n \leftrightarrow (y < t \wedge F(y))). \quad (15)$$

Comme pour le cas du schéma d'induction, la hiérarchie de compréhension bornée s'étend aux prédicats Δ_n^0 . Pour des formules $F(x), G(x)$ avec $F(x) \in \Sigma_n^0$ et $G(x) \in \Pi_n^0$, l'axiome de compréhension bornée est l'énoncé

$$\forall x (F(x) \leftrightarrow G(x)) \rightarrow (\forall t \exists n \forall y (y \in n \leftrightarrow (y < t \wedge F(y)))). \quad (16)$$

Notation

Soit $n > 0$. On note $\text{BC}\Sigma_n^0$ (resp. $\text{BC}\Pi_n^0$) le schéma de compréhension bornée (15) restreint aux formules Σ_n^0 (resp. Π_n^0) et l'on note $\text{BC}\Delta_n^0$ le schéma de compréhension bornée (16) restreint aux formules Δ_n^0 .

Dans RCA_0 , si un ensemble existe, son complémentaire existe également. Ainsi, pour tout $n > 0$, $\text{RCA}_0 \vdash \text{BC}\Sigma_n^0 \leftrightarrow \text{BC}\Pi_n^0$.

Théorème 5.6 (Hájek et Pudlák [81])

Soit $n > 0$. Alors, $\text{RCA}_0 \vdash \text{BC}\Sigma_n^0 \leftrightarrow \text{IS}_n^0$.

PREUVE. Supposons $\text{BC}\Sigma_n^0$. Soit $F(x)$ une formule Σ_n^0 telle que $F(0)$ est vraie et telle que $F(n) \rightarrow F(n+1)$ pour tout $n \in \mathbb{N}$. Soit $a \in \mathbb{N}$. Montrons que $F(a)$ est vraie. Par $\text{BC}\Sigma_n^0$, l'ensemble $X = \{x \leq a : F(x)\}$ est codé par un entier z . Soit $G(n) \equiv (n \leq a \wedge n \in z) \vee n > a$. On a bien $G(0)$ et $G(n) \rightarrow G(n+1)$. Par ID_0^0 , on a $G(x)$ pour tout x et donc $F(a)$.

Supposons IS_n^0 . Soit $F(x)$ une formule Σ_n^0 et soit z fixé. Soit $G(q)$ la formule $\Pi_n^0 \forall x < z (F(x) \rightarrow x \in q)$. L'entier $2^z - 1$ code pour l'ensemble $\{x \in \mathbb{N} : x < z\}$. En particulier, $G(2^z - 1)$ est vraie. Donc, par $\text{L}\Pi_n^0$,

qui est équivalent à $I\Sigma_n^0$, il y a un plus petit élément q qui vérifie G . Supposons $x \in q \wedge \neg F(x)$. Alors, $q - 2^x < q$ vérifie aussi G , ce qui contredit la minimalité de q . Donc, $\forall x < z \ (F(x) \leftrightarrow x \in q)$. ■

Exercice 5.7. (★★) Montrer que pour tout $n \in \omega$,

$$\text{RCA}_0 \vdash I\Delta_n^0 \leftrightarrow \text{BC}\Delta_n^0.$$

◇

La figure 5.8 résume les relations entre les différents schémas rencontrés jusqu'à maintenant.

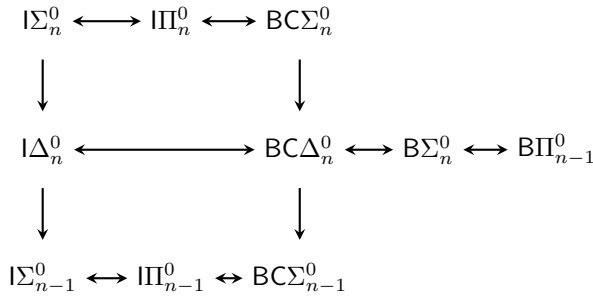


FIGURE 5.8 – Hiérarchies d'induction, de collection et de compréhension bornée. Les flèches indiquent une implication dans RCA_0 .

6. Théorèmes de conservation

Lorsque l'on étend une théorie S en une théorie T , il est naturel de se demander dans quelle mesure la nouvelle théorie T est plus forte que S : quels énoncés indémontrables dans S deviennent des théorèmes de T ? Nous avons étudié jusqu'ici des énoncés individuels, comme le théorème de Bolzano-Weierstrass, qui est prouvable dans ACA_0 mais pas dans WKL_0 . Dans ce chapitre, nous allons adopter une approche plus systématique, et étudier la prouvabilité relative pour des collections de formules. On s'intéressera aux trois grands types de questions suivantes.

- (1) *Dans quelle mesure l'arithmétique du second ordre prouve-t-elle de nouveaux énoncés de l'arithmétique du premier ordre ?* Pour y apporter une réponse plus complète, nous allons nous employer à déterminer la *partie du premier ordre* de sous-systèmes de l'arithmétique du second ordre comme RCA_0 , WKL_0 ou ACA_0 . Autrement dit, nous allons trouver,

pour chacun de ces systèmes T , une théorie du premier ordre S telle que les énoncés du premier ordre prouvables dans T sont exactement ceux prouvables dans S .

- (2) *Quelle est la puissance relative des systèmes de l'arithmétique du second ordre ?* Plus précisément, nous allons montrer que pour certaines classes Γ de formules et pour des sous-systèmes $S \subseteq T$ de l'arithmétique du second ordre, si T prouve un énoncé dans Γ , alors cet énoncé est déjà prouvable dans S . Par exemple, nous verrons que tout énoncé arithmétique prouvable dans WKL_0 est déjà prouvable dans RCA_0 .
- (3) *Dans quelle mesure les mathématiques infinitaires prouvent-elles des énoncés qui ne sont pas prouvables par les mathématiques finitaires ?* Cette question, au cœur de la crise des fondements des mathématiques, est à l'origine des mathématiques à rebours. Nous verrons comment ces dernières apportent une réponse partielle au programme de Hilbert.

Nous allons apporter à ces trois questions des réponses de même nature, en prouvant des *théorèmes de conservation*.

Définition 6.1. Soient T et S des théories dénombrables dans les langages $\mathcal{L}_T \supseteq \mathcal{L}_S$ et soit Γ un ensemble de formules closes de $\mathcal{L}_S$. La théorie T est une *extension conservative* de S pour les formules de Γ si pour toute formule $F \in \Gamma$, $T \vdash F$ ssi $S \vdash F$. Si Γ est l'ensemble des formules de $\mathcal{L}_S$, alors on dit que T est une *extension conservative* de S . $\diamond$

La question (1) demandera de prouver des théorèmes de conservation dans le cas où S est une théorie dans $\mathcal{L}_{\text{PA}}$ et T dans $\mathcal{L}_{\text{Z}_2}$, tandis que la question (2) fera appel à des théorèmes de conservation où S et T sont toutes les deux des théories dans $\mathcal{L}_{\text{Z}_2}$.

Les théories que nous allons étudier sont toutes sujettes au théorème de complétude, qui dit qu'un énoncé A est prouvable dans une théorie T ssi tout modèle de T est un modèle de A . Dans ce cas, pour prouver qu'une théorie T est une extension conservative de S pour une collection de formules closes Γ , il suffit de montrer que pour tout modèle $\mathcal{M}$ de S , il existe un modèle $\mathcal{N}$ de T , tel que pour toute formule $F \in \Gamma$, si $\mathcal{N} \models F$ alors $\mathcal{M} \models F$. La structure $\mathcal{N}$ est généralement obtenue à partir de $\mathcal{M}$ en lui ajoutant des éléments pour satisfaire T , en faisant attention à ne pas prouver de nouvelles formules de Γ .

Dans ce chapitre, nous allons nous restreindre à des théories dans des langages dénombrables, comme $\mathcal{L}_{\text{PA}}$ ou $\mathcal{L}_{\text{Z}_2}$. La proposition suivante va nous permettre de restreindre la technique décrite ci-dessus aux structures dénombrables.

Théorème 6.2 (Löwenheim-Skolem dénombrable)

Soit T une théorie cohérente dans un langage $\mathcal{L}$ dénombrable. Alors, il existe un modèle dénombrable de T .

PREUVE. La preuve du théorème 9-2.22 de complétude de Gödel produit un modèle dénombrable dans le cas où le langage $\mathcal{L}$ est dénombrable. ■

La proposition précédente est un cas particulier du théorème plus général dit de *Löwenheim-Skolem*, et nous y ferons parfois référence de cette manière. La proposition suivante donne une approche très générale pour prouver des résultats de conservation.

Proposition 6.3. Soient S et $T \supseteq S$ des théories dans les langages dénombrables $\mathcal{L}_T \supseteq \mathcal{L}_S$, et soit Γ un ensemble de formules closes de $\mathcal{L}_S$. Supposons que pour tout modèle dénombrable $\mathcal{M}$ de S , il existe un modèle $\mathcal{N}$ de T tel que

($\star$) pour toute formule $F \in \Gamma$, si $\mathcal{M} \not\models F$, alors $\mathcal{N} \not\models F$.

Alors, T est une extension conservative de S pour les formules de Γ . ★

PREUVE. Soit $F \in \Gamma$ une formule telle que $S \not\models F$. Alors, $S \cup \{\neg F\}$ est une théorie cohérente, donc par le théorème 6.2 de Löwenheim-Skolem, il existe un modèle dénombrable $\mathcal{M}$ de $S \cup \{\neg F\}$. Par hypothèse, il existe un modèle $\mathcal{N}$ de T satisfaisant ($\star$). En particulier, $\mathcal{N} \not\models F$, or $\mathcal{N} \models T$, donc $T \not\models F$ (voir le théorème 9-2.20). ■

6.1. Parties du premier ordre

Nous allons concentrer notre attention sur la partie du premier ordre des sous-systèmes de l'arithmétique du second ordre. Commençons par définir formellement ce que l'on entend par *partie du premier ordre* pour une théorie et pour une structure.

Définition 6.4. La *partie du premier ordre* d'une théorie T de $\mathcal{L}_{Z_2}$ est l'ensemble des énoncés de $\mathcal{L}_{PA}$ prouvables dans T . La *partie du premier ordre* d'une $\mathcal{L}_{Z_2}$ -structure $(M, S^M, +^M, \times^M, <^M, 0^M, 1^M)$ est la $\mathcal{L}_{PA}$ -structure

$$\mathcal{M} \upharpoonright_{\mathcal{L}_{PA}} = (M, +^M, \times^M, <^M, 0^M, 1^M).$$

◆

Notons que si $\mathcal{M} \models T$ pour une théorie T de l'arithmétique du second ordre, alors $\mathcal{M} \upharpoonright_{\mathcal{L}_{PA}} \models S$, où S est la partie du premier ordre de T . Caractériser la partie du premier ordre d'un système T de l'arithmétique du second ordre consiste en deux étapes : tout d'abord, il faut identifier un ensemble S d'axiomes dans $\mathcal{L}_{PA}$ tels que $T \vdash S$, puis prouver que T est une extension conservative de S . Pour ce faire, nous allons utiliser une version plus spécifique de la proposition 6.3 où $\mathcal{N} \upharpoonright_{\mathcal{L}_{PA}} = \mathcal{M}$.

Définition 6.5. Une $\mathcal{L}_{\mathbb{Z}_2}$ -structure $\mathcal{N}$ est une ω -extension d'une $\mathcal{L}_{\text{PA}}$ -structure $\mathcal{M}$ si $\mathcal{N} \upharpoonright_{\mathcal{L}_{\text{PA}}} = \mathcal{M}$. On note alors $\mathcal{M} \subseteq_{\omega} \mathcal{N}$ et l'on dit aussi que $\mathcal{M}$ est une ω -sous-structure de $\mathcal{N}$. $\diamond$

Attention ! La notion d' ω -extension n'a rien à voir avec celle d' ω -structure de la définition 22-3.4. En particulier, $\mathcal{N}$ peut être une ω -extension de $\mathcal{M}$ sans être une ω -structure.

Proposition 6.6. Soient S une $\mathcal{L}_{\text{PA}}$ -théorie et T une $\mathcal{L}_{\mathbb{Z}_2}$ -théorie. Si tout modèle dénombrable de S est une ω -sous-structure d'un modèle de T , alors T est une extension conservative de S . $\star$

PREUVE. Soit $\mathcal{M}$ un modèle dénombrable de S , et soit $\mathcal{N}$ une ω -extension de $\mathcal{M}$ telle que $\mathcal{N} \models T$. Soit F une formule close de $\mathcal{L}_{\text{PA}}$ telle que $\mathcal{M} \not\models F$. Alors, comme la satisfaction d'une formule du premier ordre sur $\mathcal{N}$ ne fait intervenir que sa partie du premier ordre, on a $\mathcal{N} \not\models F$. Ainsi, par la proposition 6.3, T est une extension conservative de S pour les formules closes de $\mathcal{L}_{\text{PA}}$. $\blacksquare$

Premier ordre vs formule arithmétique

Attention à distinguer les formules dans $\mathcal{L}_{\text{PA}}$, et les formules arithmétiques dans $\mathcal{L}_{\mathbb{Z}_2}$. En effet, dans le second cas, les variables libres peuvent être des ensembles d'entiers. On distingue donc le principe IS_n^0 qui est une théorie du second ordre, où les formules peuvent avoir des variables libres d'ensembles, de IS_n qui est une théorie du premier ordre.

Notre premier résultat de conservation montre que la partie du premier ordre de ACA_0 est l'arithmétique de Peano. Autrement dit, les seules formules de $\mathcal{L}_{\text{PA}}$ prouvables à l'aide de ACA_0 sont les théorèmes de PA. Nous aurons besoin de la définition suivante.

Définition 6.7. Soit $\mathcal{M} = (M, S^{\mathcal{M}})$ une $\mathcal{L}_{\mathbb{Z}_2}$ -structure et Γ une collection de formules de $\mathcal{L}_{\mathbb{Z}_2}$ à paramètres dans $\mathcal{M}$, sans variable libre d'ensemble, et n'ayant que x pour variable libre d'entier. Un ensemble $X \subseteq M$ est Γ -définissable dans $\mathcal{M}$ si $X = \{n \in M : \mathcal{M} \models F(n)\}$ pour une formule $F \in \Gamma$. Un ensemble X est Δ_n^0 -définissable dans $\mathcal{M}$ s'il est à la fois Σ_n^0 -définissable et Π_n^0 -définissable. $\diamond$

La définition précédente s'applique également aux structures du premier ordre en considérant leur ω -extension triviale dont la partie du second ordre est vide.

Théorème 6.8 (Friedman [65])

Le système ACA_0 est une extension conservative de PA.

PREUVE

Par la proposition 6.6, il suffit de prouver que pour tout modèle dénombrable $\mathcal{M} \models \text{PA}$, il existe une ω -extension $\mathcal{N}$ de $\mathcal{M}$ telle que $\mathcal{N} \models \text{ACA}_0$. Soit $\mathcal{M} = (M, +^{\mathcal{M}}, \times^{\mathcal{M}}, <^{\mathcal{M}}, 0^{\mathcal{M}}, 1^{\mathcal{M}})$ un modèle dénombrable de PA. Soit $S^{\mathcal{N}}$ la classe des ensembles $X \subseteq M$ définissables par une formule de $\mathcal{L}_{\text{PA}}$ avec paramètres dans $\mathcal{M}$. Notons que les paramètres ne sont que du premier ordre. Soit $\mathcal{N}$ l' ω -extension de $\mathcal{M}$ dont la partie du second ordre est $S^{\mathcal{N}}$. Montrons que $\mathcal{N} \models \text{ACA}_0$. Par l'exercice 22-2.2, il suffit de montrer que $\mathcal{N}$ satisfait l'arithmétique de Robinson (Q) augmentée de l'axiome d'induction sur les ensembles (11) et du schéma de compréhension pour les formules arithmétiques avec paramètres.

Comme $\mathcal{N}$ est une ω -extension de $\mathcal{M}$ et que $\mathcal{M} \models \text{PA}$, alors $\mathcal{N} \models \text{PA}$, donc $\mathcal{N} \models \text{Q}$. Soit $X \in S^{\mathcal{N}}$, et soit F la formule de $\mathcal{L}_{\text{PA}}$ définissant X dans $\mathcal{M}$. Comme $\mathcal{N} \models \text{PA}$, alors le schéma d'induction est satisfait pour F , donc l'axiome d'induction est satisfait pour X .

Montrons que $\mathcal{N}$ satisfait le schéma de compréhension pour les formules arithmétiques avec paramètres dans $\mathcal{N}$. Soit $G(x)$ une formule arithmétique de $\mathcal{L}_{\mathbb{Z}_2}$ avec paramètres dans $\mathcal{N}$. Soient $X_0, \dots, X_{k-1}$ les paramètres du second ordre de G , et soient $F_0, \dots, F_{k-1}$ les formules de $\mathcal{L}_{\text{PA}}$ les définissant. Soit $H(x)$ la formule de $\mathcal{L}_{\mathbb{Z}_2}$ où l'on a remplacé toutes les occurrences de $y \in X_s$ par $F_s(y)$, de telle sorte que H n'ait plus que des paramètres du premier ordre. En particulier, H est une formule de $\mathcal{L}_{\text{PA}}$ avec paramètres dans $\mathcal{M}$, donc $Y = \{n \in M : \mathcal{M} \models H(n)\} \in S^{\mathcal{N}}$, or $Y = \{n \in M : \mathcal{N} \models G(n)\}$, donc $\mathcal{N} \models \exists X \forall n (n \in X \leftrightarrow G(n))$. ■

Le second résultat de conservation concerne la partie du premier ordre de RCA_0 . On note $\Sigma_1\text{-PA}$ la théorie du premier ordre composée de l'arithmétique de Robinson Q augmentée du schéma d'induction Σ_1 pour les formules de $\mathcal{L}_{\text{PA}}$ avec paramètres du premier ordre ($\text{I}\Sigma_1$). Nous aurons besoin du lemme suivant, laissé en exercice.

Exercice 6.9. Soit $\mathcal{M} = (M, S^{\mathcal{M}})$ une structure telle que $\mathcal{M} \models \text{B}\Sigma_1^0$, et soit $X \subseteq M$ un ensemble Δ_1^0 -définissable avec paramètres dans $\mathcal{M}$. Pour toute formule $\Sigma_1^0 F$ à paramètres dans $\mathcal{M} \cup \{X\}$, il existe une formule $\Sigma_1^0 G$ à paramètres dans $\mathcal{M}$, ayant les mêmes variables libres que F , et telle que $\mathcal{M} \cup \{X\} \models \forall x (F(x) \leftrightarrow G(x))$. ♦

Nous sommes maintenant prêts à prouver notre résultat de conservation.

Théorème 6.10 (Friedman [65])

Le système RCA_0 est une extension conservative de $\Sigma_1\text{-PA}$.

PREUVE

Par la proposition 6.6, il suffit de prouver que pour tout modèle dénombrable $\mathcal{M} \models \Sigma_1\text{-PA}$, il existe une ω -extension $\mathcal{N}$ de $\mathcal{M}$ telle que $\mathcal{N} \models \text{RCA}_0$. Soit $\mathcal{M} = (M, +^{\mathcal{M}}, \times^{\mathcal{M}}, <^{\mathcal{M}}, 0^{\mathcal{M}}, 1^{\mathcal{M}})$ un modèle dénombrable de $\Sigma_1\text{-PA}$. Soit $S^{\mathcal{N}}$ la classe des ensembles $X \subseteq M$ définissables par une formule Δ_1 de $\mathcal{L}_{\text{PA}}$ avec paramètres dans $\mathcal{M}$. Notons que les paramètres ne sont que du premier ordre. Soit $\mathcal{N}$ l' ω -extension de $\mathcal{M}$ dont la partie du second ordre est $S^{\mathcal{N}}$.

En itérant l'exercice 6.9, ci-dessus, pour retirer à chaque itération un paramètre du second ordre, on peut montrer que si $F(x)$ est une formule Σ_1^0 à paramètres dans $\mathcal{N}$ et sans variable libre d'ensemble, il existe une formule Σ_1 $G(x)$ à paramètres dans $\mathcal{M}$, avec les mêmes variables libres, et telle que $\mathcal{N} \models \forall x (F(x) \leftrightarrow G(x))$. En effet, si $F(x)$ a comme paramètres du second ordre $X_1, \dots, X_k$, on peut construire successivement des formules Σ_1^0 $F_1(x), \dots, F_{k-1}(x)$ telles que $F_i(x)$ a comme paramètres du second ordre $X_1, \dots, X_{k-i-1}$, et telles que $\mathcal{N} \models \forall x (F(x) \leftrightarrow F_i(x))$. La dernière formule $F_{k-1}(x)$ est la formule $G(x)$ désirée.

Montrons que $\mathcal{N} \models \text{RCA}_0$. Comme $\mathcal{N}$ est une ω -extension de la structure $\mathcal{M}$ et $\mathcal{M} \models \Sigma_1\text{-PA}$, alors $\mathcal{N} \models \Sigma_1\text{-PA}$, donc $\mathcal{N} \models \text{Q}$. Montrons que $\mathcal{N} \models \text{I}\Sigma_1^0$. Soit $F(x)$ une formule Σ_1^0 close avec paramètres dans $\mathcal{N}$. Soit $G(x)$ une formule Σ_1 avec paramètres dans $\mathcal{M}$ telle que $\mathcal{N} \models \forall x (F(x) \leftrightarrow G(x))$. Comme $\mathcal{M} \models \text{I}\Sigma_1$, alors le schéma d'induction s'applique dans $\mathcal{M}$ pour G , donc s'applique dans $\mathcal{N}$ pour G , donc dans $\mathcal{N}$ pour F . Ainsi, $\mathcal{N} \models \text{I}\Sigma_1^0$. Montrons que $\mathcal{N}$ satisfait le schéma de compréhension pour les formules Δ_1^0 closes avec paramètres dans $\mathcal{N}$ et comme unique variable libre x . Soit $F_{\exists}(x)$ une formule Σ_1^0 à paramètres dans $\mathcal{N}$ et soit $F_{\forall}$ une formule Π_1^0 à paramètres dans $\mathcal{N}$, telles que $\mathcal{N} \models \forall x (F_{\exists}(x) \leftrightarrow F_{\forall}(x))$. Soient $G_{\exists}$ et $G_{\forall}$ les formules Σ_1 et Π_1 à paramètres dans $\mathcal{M}$ correspondantes en utilisant l'exercice précédent. Alors, $\mathcal{M} \models \forall x (G_{\exists}(x) \leftrightarrow G_{\forall}(x))$, donc l'ensemble $X = \{n \in M : G_{\exists}(n)\}$ est définissable par une formule Δ_1 avec paramètres dans $\mathcal{M}$, donc $X \in S^{\mathcal{N}}$. Ainsi, $\mathcal{N} \models \exists X \forall z (z \in X \leftrightarrow F_{\exists}(z))$, donc $\mathcal{N}$ satisfait le schéma de compréhension pour les formules Δ_1^0 closes avec paramètres dans $\mathcal{N}$. Ainsi, $\mathcal{N} \models \text{RCA}_0$. ■

6.2. Conservation Π_1^1

Nous avons étudié jusqu'ici la partie de premier ordre des théories ACA_0 et RCA_0 , en montrant qu'elles étaient des extensions conservatives, respectivement de PA et $\Sigma_1\text{-PA}$. Notons qu'il s'agit d'extensions de théories du premier ordre en théories du second ordre.

Nous allons dans cette section changer de stratégie. Supposons que l'on dispose de théories $S \subseteq T$ toutes deux dans le langage de l'arithmétique

du second ordre. Supposons aussi que la partie du premier ordre de S soit connue. Afin de connaître la partie du premier ordre de T , il suffit de montrer que cette dernière est une extension conservative de S pour les formules arithmétiques. Elle aura ainsi la même partie du premier ordre que S . La technique que nous exposerons fera en fait le travail directement pour toutes les formules Π_1^1 , c'est-à-dire les formules de la forme $\forall XF(X)$ où F est une formule arithmétique et X une variable du second ordre (nous étudierons les formules et classes Π_1^1 en détail dans le chapitre 29).

Nous allons prouver en particulier que WKL_0 est une extension conservative de RCA_0 pour les formules Π_1^1 , et donc que la partie du premier ordre de WKL_0 est $\Sigma_1\text{-PA}$. Commençons par étendre la notion d' ω -extension aux structures du second ordre.

Définition 6.11. Soient $\mathcal{M} = (M, S^{\mathcal{M}})$ et $\mathcal{N} = (N, S^{\mathcal{N}})$ deux structures de $\mathcal{L}_{Z_2}$. On dit que $\mathcal{N}$ est une ω -extension de $\mathcal{M}$ si $\mathcal{M} \upharpoonright_{\mathcal{L}_{\text{PA}}} = \mathcal{N} \upharpoonright_{\mathcal{L}_{\text{PA}}}$ et $S^{\mathcal{M}} \subseteq S^{\mathcal{N}}$. On dira alors aussi que $\mathcal{M}$ est une ω -sous-structure de $\mathcal{N}$. $\diamond$

Nous voyons à présent comment utiliser la notion d' ω -extension pour prouver que deux théories montrent les mêmes formules Π_1^1 , et donc les mêmes formules arithmétiques.

Proposition 6.12. Soient S et $T \supseteq S$ des $\mathcal{L}_{Z_2}$ -théories telles que tout modèle dénombrable $\mathcal{M}$ de S est une ω -sous-structure d'un modèle dénombrable $\mathcal{N}$ de T . Alors, T est une extension conservative de S pour les formules closes Π_1^1 . $\star$

PREUVE. Soit $\mathcal{M}$ un modèle dénombrable de S , et soit $\mathcal{N}$ une ω -extension de $\mathcal{M}$ telle que $\mathcal{N} \models T$. Soient $S^{\mathcal{M}}$ et $S^{\mathcal{N}}$ les parties du second ordre de respectivement $\mathcal{M}$ et $\mathcal{N}$. Soit F une formule close Π_1^1 de la forme $\forall XG(X)$, où $G(X)$ est une formule arithmétique telle que $\mathcal{M} \not\models F$.

Alors, $\mathcal{M} \models \exists X \neg G(X)$, donc il existe un $X \in S^{\mathcal{M}}$ tel que $\mathcal{M} \models G(X)$. Comme $S^{\mathcal{M}} \subseteq S^{\mathcal{N}}$, alors $X \in S^{\mathcal{N}}$, et comme $\mathcal{M} \upharpoonright_{\mathcal{L}_{\text{PA}}} = \mathcal{N} \upharpoonright_{\mathcal{L}_{\text{PA}}}$, alors $\mathcal{N} \models G(X)$, donc $\mathcal{N} \models \exists X \neg G(X)$, et $\mathcal{N} \models \neg F$. Ainsi, par la proposition 6.3, T est une extension conservative de S pour les formules closes Π_1^1 . $\blacksquare$

Comme exprimé dans la section 22-8.1, un certain nombre d'énoncés s'expriment sous la forme $\forall X (F(X) \rightarrow \exists Y H(X, Y))$, où F et H sont des formules arithmétiques éventuellement avec des variables libres. Par exemple, pour le lemme faible de König, $F(X)$ est le prédicat « X code pour un arbre binaire infini » et $H(X, Y)$ signifie « Y est un chemin de l'arbre codé par X ». Nous allons développer des techniques générales pour créer des modèles d'énoncés de la forme ci-dessus, tout en garantissant la conservation de formules Π_1^1 .

Définition 6.13. Soit $\mathcal{M} = (M, S^{\mathcal{M}})$ une structure de l'arithmétique du second ordre, et soit $G \subseteq M$. On note $\mathcal{M} \cup \{G\}$ l' ω -extension de $\mathcal{M}$ dont la partie du second ordre est $S^{\mathcal{M}} \cup \{G\}$, et $\mathcal{M}[G]$ l' ω extension de $\mathcal{M}$ dont la partie du second ordre est la collection des ensembles qui sont définissables par un prédicat Δ_1^0 à paramètres dans $\mathcal{M} \cup \{G\}$. $\diamond$

Reprenons l'exemple d'énoncé de la forme $\forall X (F(X) \rightarrow \exists Y H(X, Y))$. Intuitivement, en partant d'une structure $\mathcal{M} \models \text{RCA}_0$ et d'un ensemble $X \in \mathcal{M}$ tel que $F(X)$, nous voulons créer une ω -extension $\mathcal{M}_1$ contenant un ensemble $G \subseteq M$ tel que $\mathcal{M}_1 \models H(X, G)$, de telle sorte que l'énoncé $F(X) \rightarrow \exists Y H(X, Y)$ soit vrai dans $\mathcal{M}_1$. En itérant le processus, nous allons obtenir une suite d' ω -extensions $\mathcal{M} \subseteq \mathcal{M}_1 \subseteq \mathcal{M}_2 \subseteq \dots$ telle que $\bigcup_n \mathcal{M}_n$ est un modèle de RCA_0 et de $\forall X (F(X) \rightarrow \exists Y H(X, Y))$.

Pour satisfaire une étape, nous allons donc construire un ensemble $G \subseteq M$ tel que $\mathcal{M} \cup \{G\} \models H(X, G)$ et l'ajouter à $\mathcal{M}$. La structure $\mathcal{M} \cup \{G\}$ obtenue ne satisfait pas le schéma de compréhension Δ_1^0 en général. Nous devons donc clore la structure $\mathcal{M} \cup \{G\}$ par ce schéma, ce qui donne $\mathcal{M}[G]$. Tout le « travail » va consister à s'assurer que le modèle $\mathcal{M}[G]$ ainsi obtenu ne va pas saboter RCA_0 , et en particulier l'induction Σ_1^0 . Il est donc nécessaire de s'assurer que $\mathcal{M}[G] \models \text{IS}_1^0$. La proposition suivante montre qu'il suffit de prouver que $\mathcal{M} \cup \{G\} \models \text{IS}_1^0$ pour obtenir $\mathcal{M}[G] \models \text{IS}_1^0$, et donc $\mathcal{M}[G] \models \text{RCA}_0$.

Proposition 6.14. Soit $T \equiv \forall X (F(X) \rightarrow \exists Y H(X, Y))$, où F et H sont des formules arithmétiques. Supposons que pour tout modèle dénombrable $\mathcal{M}$ de RCA_0 , et tout ensemble $X \in \mathcal{M}$ tel que $\mathcal{M} \models F(X)$, il existe un ensemble $G \subseteq M$ tel que $\mathcal{M} \cup \{G\} \models \text{IS}_1^0$ et $\mathcal{M} \cup \{G\} \models H(X, G)$. Alors, $\text{RCA}_0 + T$ est une extension conservative de RCA_0 pour les formules Π_1^1 . $\star$

PREUVE. Soit $\mathcal{M}$ un modèle dénombrable de RCA_0 . Nous allons définir une suite $\mathcal{M} = \mathcal{M}_0 \subseteq \mathcal{M}_1 \subseteq \mathcal{M}_2 \subseteq \dots$ d' ω -extensions dénombrables de telle sorte que, pour tout $n \in \mathbb{N}$,

- (1) $\mathcal{M}_n \models \text{RCA}_0$;
- (2) pour tout $X \in \mathcal{M}_n$ tel que $\mathcal{M}_n \models F(X)$, il existe $m \in \mathbb{N}$ et $Y \in \mathcal{M}_m$ tel que $\mathcal{M}_m \models H(X, Y)$.

Montrons que si c'est le cas, alors $\text{RCA}_0 + T$ est une extension conservative de RCA_0 pour les formules Π_1^1 . Soit $\mathcal{N} = \bigcup_n \mathcal{M}_n$. Par (1), $\mathcal{N} \models \text{RCA}_0$, et par (2), comme F et H sont des formules arithmétiques et que $\mathcal{N}$ est une ω -extension de $\mathcal{M}_n$ pour tout $n \in \mathbb{N}$, $\mathcal{N} \models \forall X (F(X) \rightarrow \exists Y H(X, Y))$. Par la proposition 6.12, $\text{RCA}_0 + T$ est une extension conservative de RCA_0 pour les formules Π_1^1 .

Supposons que l'on a construit $\mathcal{M}_n$, et soit $X \in \mathcal{M}_n$ un ensemble tel que $\mathcal{M}_n \models F(X)$. Soit $G \subseteq M$ tel que

$$\mathcal{M}_n \cup \{G\} \models I\Sigma_1^0 \text{ et } \mathcal{M}_n \cup \{G\} \models H(X, G).$$

Montrons que $\mathcal{M}_{n+1} = \mathcal{M}_n[G]$ satisfait (1) et (2) ci-dessus. Par construction, $\mathcal{M}_n[G] \models H(X, G)$ et $\mathcal{M}_n[G]$ satisfait le schéma de compréhension Δ_1^0 . Montrons que $\mathcal{M}_n[G] \models I\Sigma_1^0$. Soit $P(x)$ une formule Σ_1^0 à paramètres dans $\mathcal{M}_n[G]$ sans autre variable libre que x . En itérant l'exercice 6.9 comme dans la preuve du théorème 6.10, il existe une formule $\Sigma_1^0 Q(x)$ à paramètres dans $\mathcal{M}_n \cup \{G\}$ telle que $\mathcal{M}_n[G] \models \forall x (P(x) \leftrightarrow Q(x))$. Comme

$$\mathcal{M}_n \cup \{G\} \models I\Sigma_1^0,$$

alors le schéma d'induction est satisfait pour Q , donc également pour P . Ainsi, $\mathcal{M}_n[G] \models I\Sigma_1^0$. Donc, $\mathcal{M}_n[G] \models \text{RCA}_0$. Cela termine la construction.

Notons qu'il faut faire attention au choix de X pour que (2) soit satisfait. Plus précisément, si l'on fixe une énumération $\mathcal{M}_n = \{X_0^n, X_1^n, \dots\}$ de tous les ensembles de $\mathcal{M}_n$, à l'étape n , on choisit X_m^n pour le plus petit $\langle n, m \rangle$ qui n'a pas encore été satisfait. ■

Nous avons à présent les éléments nécessaires pour montrer le théorème annoncé : WKL_0 est une extension conservative de RCA_0 pour les formules Π_1^1 . On peut en particulier en déduire que la partie du premier ordre de WKL_0 est la même que celle de RCA_0 : Σ_1 -PA.

Théorème 6.15 (Harrington (voir Simpson [203]))

Le système WKL_0 est une extension conservative de RCA_0 pour les formules Π_1^1 .

Avant de prouver le théorème 6.15, nous avons besoin d'un lemme technique (donné en exercice) sur les formules Σ_1^0 . Par le théorème 9-3.4, un ensemble $A \subseteq \mathbb{N}$ est X -c.e. s'il est définissable par une formule $\Sigma_1^0(X)$ de $\mathcal{L}_{Z_2}$. En particulier, pour toute formule $\Sigma_1^0 F(X, x)$, il existe une fonctionnelle Turing Φ_e tel que $\{n : F(X, n)\} = \{n : \Phi_e^X(n) \downarrow\}$. Il s'ensuit par la propriété de l'usage que $\{n : F(X, n)\} = \{n : \exists k \Phi_e^{X \upharpoonright k}(n)[k] \downarrow\}$. Autrement dit, si l'on définit la formule $\Delta_0^0 H(\sigma, x) \equiv \Phi_e^\sigma(x)[|\sigma|]$, on a

$$\{n : F(X, n)\} = \{n : \exists k H(G \upharpoonright_k, n)\}.$$

L'exercice suivant montre que cette équivalence peut se formaliser dans $\text{BS}\Sigma_1^0$.

Exercice 6.16. (★) (Simpson [203]). Soit $\mathcal{M} = (M, S^\mathcal{M})$ une $\mathcal{L}_{Z_2}$ -structure telle que $\mathcal{M} \models \text{BS}\Sigma_1^0$. Soit $F(X)$ une formule Σ_1^0 à paramètres dans $\mathcal{M}$, et sans autre variable libre d'ensemble que X . Alors, il existe une formule $\Delta_0^0 H(\sigma)$ avec les mêmes paramètres que F , sans variable

libre d'ensemble, et avec les même variables libres d'entiers que F plus une variable libre d'entier σ , telle que pour tout ensemble $G \subseteq M$,

$$\mathcal{M} \cup \{G\} \models F(G) \leftrightarrow \exists k H(G \upharpoonright_k). \quad \diamond$$

Venons-en à la preuve du théorème 6.15. Soit $\mathcal{M} = (M, S^{\mathcal{M}})$ un modèle dénombrable de RCA_0 , et soit $T \in S^{\mathcal{M}}$ un arbre binaire $\mathcal{M}$ -infini (voir la section 5.2). Nous allons construire un chemin $\mathcal{M}$ -infini $G \in [T]$ par un forcing $\mathbb{P}$ sur les sous-arbres $\mathcal{M}$ -infinis de T dans $S^{\mathcal{M}}$. Commençons par montrer que tout filtre $\mathcal{F}$ suffisamment générique pour $\mathbb{P}$ induit un unique chemin $G \in \bigcap_{T \in \mathcal{F}} [T]$.

Lemme 6.17. Pour tout entier $n \in M$, l'ensemble D_n des arbres $S \in \mathbb{P}$ ne contenant qu'une chaîne de longueur n est dense dans $\mathbb{P}$. ★

PREUVE. Soit $T \in \mathbb{P}$ un arbre binaire $\mathcal{M}$ -infini.

Pour tout $\sigma \in 2^n$, soit $S_\sigma = \{\tau \in T : \sigma \preceq \tau \vee \sigma \succ \tau\}$. Montrons qu'il existe $\sigma \in 2^n$ tel que S_σ est $\mathcal{M}$ -infini.

En effet, si tous les ensembles S_σ sont $\mathcal{M}$ -finis, on a :

$$\forall \sigma \in 2^n \exists m \forall \tau \in 2^m (\sigma \prec \tau \rightarrow \tau \notin T).$$

On utilise $\text{B}\Sigma_1^0$ (qui est une conséquence de RCA_0) pour obtenir :

$$\exists m \forall \sigma \in 2^n \exists m' < m \forall \tau \in 2^{m'} (\sigma \prec \tau \rightarrow \tau \notin T).$$

Et, par clôture de T par le bas, on obtient $\exists m \forall \tau \in 2^m \tau \notin T$, donc T est $\mathcal{M}$ -fini, ce qui contredit notre hypothèse sur $T \in \mathbb{P}$. ■

Rappelons qu'une condition T force une formule $F(G)$, formule supposée Σ_1^0 ou Π_1^0 , si $\mathcal{M} \models F(P)$ pour tout $P \in [T]$. Notre but est de « réduire » l'induction Σ_1^0 de $\mathcal{M} \cup \{G\}$ à celle de $\mathcal{M}$ à l'aide de la relation de forcing. En effet, la relation de forcing pour les propriétés Σ_1^0 est Σ_1^0 sur $\mathcal{M}$. Plus précisément, nous allons utiliser le principe $\text{L}\Pi_1^0$, équivalent à $\text{I}\Sigma_1^0$, qui affirme que tout sous-ensemble Π_1^0 non vide de $\mathcal{N}$ admet un élément minimum. Pour ce faire, supposons que $F(x, G)$ soit une formule Σ_1^0 telle que $T \Vdash \neg F(b, G)$ pour une condition $T \in \mathbb{P}$ dans notre filtre et un $b \in \mathcal{M}$. Il s'ensuit que l'ensemble $\{a \in \mathcal{M} : \neg F(a, G)\}$ sera non vide, où G est l'ensemble générique construit. Pour que F satisfasse le principe $\text{L}\Pi_1^0$, il faudra s'assurer qu'il existe un élément b_1 minimum dans $\mathcal{M}$ tel que $\neg F(b_1, G)$ soit vrai. Une manière de s'en assurer consiste à trouver une extension $S \in \mathbb{P}$ telle que pour tout $a <^{\mathcal{M}} b$, $S \Vdash F(a, G)$ ou $S \Vdash \neg F(a, G)$. En effet, l'ensemble $E = \{a \leq^{\mathcal{M}} b : S \Vdash \neg F(a, G)\}$ est Π_1^0 non vide dans $\mathcal{M}$ et doit donc satisfaire le principe $\text{L}\Pi_1^0$ dans le modèle de départ $\mathcal{M}$. On en déduit donc

qu'il existe un b_1 minimum dans $\mathcal{M}$ tel que $S \Vdash \neg F(b_1, G)$, donc que $\text{L}\Pi_1^0$ est satisfait pour F .

Étant donné une condition $T \in \mathbb{P}$ et une formule $\Sigma_1^0 F(G)$, il est aisé de trouver une extension S forçant soit $F(G)$, soit $\neg F(G)$. L'approche naturelle pour préserver $\text{L}\Pi_1^0$ consisterait à essayer de créer une suite décroissante d'extensions $T \supseteq T_1 \supseteq T_2 \supseteq T_3 \supseteq \dots \supseteq T_b$ telle que $T_{a+1} \Vdash F(a, G)$ ou $T_{a+1} \Vdash \neg F(a, G)$ pour tout $a <^{\mathcal{M}} b$. Nous travaillons cependant dans un modèle dénombrable arbitraire de RCA_0 , qui peut contenir des entiers b non standard, et notamment tels que l'ensemble $\{a \in \mathcal{M} : a <^{\mathcal{M}} b\}$ est infini. Le but du lemme suivant est de montrer que l'on arrive à forcer simultanément $F(a, G)$ ou $\neg F(a, G)$ pour tout $a <^{\mathcal{M}} b$ avec une même condition de forcing, quand bien même l'ensemble $\{a \in \mathcal{M} : a <^{\mathcal{M}} b\}$ serait infini.

Lemme 6.18. Pour tout $b \in M$ et toute formule $\Sigma_1^0 F(x, G)$, l'ensemble $D_{b, F}$ des arbres $S \in \mathbb{P}$ tels que pour tout $a <^{\mathcal{M}} b$, $S \Vdash F(a, G)$ ou $S \Vdash \neg F(a, G)$ est dense dans $\mathbb{P}$. ★

PREUVE

Par l'exercice 6.16, $F(x, G) \equiv \exists z H(x, z, G \upharpoonright_z)$ pour une formule $\Delta_0^0 H$. Pour tout $\sigma \in 2^{\leq b}$, définissons $S_\sigma \subseteq T$ inductivement comme suit : $S_\epsilon = S$. Supposons que S_σ est défini. Alors, $S_{\sigma 0} = \{\rho \in S_\sigma : \forall z < |\rho| \neg H(a, z, \rho \upharpoonright_z)\}$ et $S_{\sigma 1} = S_\sigma$. Soit

$$U = \{\sigma \in 2^b : S_\sigma \text{ est } \mathcal{M}\text{-infini}\}.$$

Notons que U est non vide, car $S_{111\dots 1} = S$ est $\mathcal{M}$ -infini. De plus, U est Π_1^0 dans $\mathcal{M}$, car S_σ est clos par préfixe pour tout σ :

$$U = \{\sigma \in 2^b : \forall n \in \mathcal{M} \exists \tau \in 2^n \tau \in S_\sigma\}.$$

Par $\text{L}\Pi_1^0$, U contient un élément minimum σ pour l'ordre lexicographique.

Montrons que $S_\sigma \in D_{b, F}$. Soit $a <^{\mathcal{M}} b$.

- ▷ Si $\sigma(a) = 0$, on a alors $\forall \rho \in S_\sigma \forall z < |\rho| \neg H(a, z, \rho \upharpoonright_z)$. Ainsi, pour tout $G \in [S_\sigma]$, on a $\forall z \neg H(a, z, G \upharpoonright_z)$, autrement dit $\neg F(a, G)$. Donc, $S_\sigma \Vdash \neg F(a, G)$.
- ▷ Si $\sigma(a) = 1$, alors par minimalité de σ dans U , la chaîne $\sigma \upharpoonright_a 0111\dots 1$ de longueur b n'est pas dans U , donc $S_{\sigma \upharpoonright_a 0}$ est $\mathcal{M}$ -fini. Il s'ensuit qu'il existe un $k \in M$ tel que pour toute chaîne $\rho \in 2^k$, si $\rho \in S_{\sigma \upharpoonright_a}$ alors $\exists z < k H(a, z, \rho \upharpoonright_z)$. Ainsi, pour tout $G \in [S_{\sigma \upharpoonright_a}]$, il existe $z < k$ tel que $H(a, z, G \upharpoonright_z)$, donc $S_{\sigma \upharpoonright_a} \Vdash F(a, G)$, or $S_\sigma \subseteq S_{\sigma \upharpoonright_a}$, de sorte que $S_\sigma \Vdash F(a, G)$. ■

Nous sommes maintenant prêts à assembler les pièces du puzzle pour prouver le théorème 6.15.

PREUVE DU THÉORÈME 6.15. Par la proposition 6.14, il suffit de montrer que pour tout modèle dénombrable $\mathcal{M} = (M, S^{\mathcal{M}})$ de RCA_0 et tout arbre binaire $\mathcal{M}$ -infini $T \in \mathcal{M}$, il existe un ensemble $G \subseteq M$ tel que $G \in [T]$ et $\mathcal{M} \cup \{G\} \models \text{L}\Pi_1^0$. Fixons $\mathcal{M}$, et $T \in \mathcal{M}$. Soit $\mathbb{P}$ la collection des sous-arbres $\mathcal{M}$ -infinis de T dans $S^{\mathcal{M}}$, ordonnés par l'inclusion, et soit $\mathcal{F}$ un filtre suffisamment générique pour $\mathbb{P}$. Par le lemme 6.17, il existe un unique ensemble $G \subseteq M$ tel que $G \in \bigcap_{S \in \mathcal{F}} [S]$. En particulier, $T \in \mathcal{F}$, donc $G \in [T]$.

Montrons que $\mathcal{M} \cup \{G\} \models \text{L}\Pi_1^0$. Soit donc $F(x, G)$ une formule Σ_1^0 telle que $\mathcal{M} \cup \{G\} \models \neg F(b, G)$ pour un élément $b \in M$. Par le lemme 6.18, il existe une condition $S \in \mathcal{F}$ telle que pour tout $a \leq^{\mathcal{M}} b$, $S \Vdash F(a, G)$ ou $S \Vdash \neg F(a, G)$. On a ainsi

$$\{a \leq^{\mathcal{M}} b : \mathcal{M} \cup \{G\} \models F(a, G)\} = \{a \leq^{\mathcal{M}} b : S \Vdash F(a, G)\},$$

et le second prédicat est Σ_1^0 dans $\mathcal{M}$. Comme $\neg F(b, G)$, alors $S \Vdash \neg F(b, G)$, donc par $\text{L}\Pi_1^0$ appliqué au prédicat Σ_1^0 $S \Vdash F(x, G)$ avec pour variable libre x , il existe un plus petit $a \leq^{\mathcal{M}} b$ tel que $S \Vdash \neg F(a, G)$. En particulier, a est le plus petit élément tel que $\neg F(a, G)$ est satisfait. Cela termine la preuve du théorème 6.15. ■

Corollaire 6.19

Le système WKL_0 est une extension conservative de $\Sigma_1\text{-PA}$.

PREUVE. Soit F une formule close de $\mathcal{L}_{\text{PA}}$ telle que $\Sigma_1\text{-PA} \not\models F$. Par le théorème 6.10, $\text{RCA}_0 \not\models F$, et par le théorème 6.15, $\text{WKL}_0 \not\models F$. ■

7. Programme de Hilbert

Revenons aux motivations originelles des mathématiques à rebours. Comme nous l'avons mentionné dans le chapitre 9, les mathématiques ont connu une grande crise des fondements au début du XX^e siècle, avec le développement d'une théorie de l'infini par Cantor marquant une étape supplémentaire dans l'abstraction mathématique et son éloignement d'avec le réel. La difficulté, voire l'impossibilité, de raccrocher l'infini à une réalité sensible, ainsi que les paradoxes sur lesquels ont trébuché les mathématiciens ont naturellement suscité la méfiance. Ainsi la crise des fondements a-t-elle favorisé le développement de pensées finitistes. À cette époque, les avancées scientifiques tendaient à montrer la finitude du monde réel, à travers la découverte de l'atome (l'infiniment petit n'existerait donc pas) ou de la finitude de l'univers (l'infiniment grand non plus). Des mathématiciens influents comme Poincaré, Brouwer ou Kronecker ont alors prôné la restriction aux mathématiques finitaires.

7.1. Interprétation formelle du programme de Hilbert

David Hilbert proposa en 1900 un programme destiné à rétablir la confiance dans les fondements des mathématiques, dont l'objectif était de fonder les mathématiques dans toute leur généralité, sur des bases purement finitaires. Ce programme peut se décomposer en trois étapes :

- (1) formaliser les mathématiques infinitaires T_∞ ;
- (2) identifier le fragment finitaire T_{fin} des mathématiques ;
- (3) montrer que tout énoncé, ou tout du moins tout énoncé dans une certaine classe $\mathcal{C}$, prouvable dans T_∞ est prouvable dans T_{fin} .

Hilbert ayant laissé des points d'incertitude quant à la teneur formelle de son programme, Steve Simpson [202] en a proposé l'interprétation suivante.

La théorie T_∞ . Simpson propose le choix de Z_2 comme théorie de référence pour les mathématiques infinitaires. Ce choix est justifié par les travaux de Hilbert et Bernays, *Grundlagen der Mathematik* (1934-1936), qui montrent que la grande majorité des mathématiques usuelles peut être formalisée dans l'arithmétique du second ordre.

La théorie T_{fin} . Une première idée serait de choisir RCA_0 comme fragment finitaire de référence. Toutefois, le schéma d'axiomes d'induction, même restreint aux formules Σ_1^0 , constitue une hypothèse infinitaire. Elle est malheureusement nécessaire pour formaliser la notion de fonction calculable au sein de l'arithmétique. Un système plus élémentaire est cependant possible : celui dit d'*arithmétique primitive réursive* (PRA), lequel fait généralement consensus (voir Tait [220]). L'idée est la suivante, on restreint l'induction à son strict minimum, c'est-à-dire aux formules sans quantificateurs (le I_{ouvert} défini dans la section 2.2). Afin de garder malgré tout quelques outils de travail, on rajoute des symboles de fonctions dans notre langage permettant de manipuler directement d'autres fonctions usuelles que l'addition et la multiplication, et en premier lieu la fonction exponentielle. On s'autorisera en fait toutes les fonctions primitives récursives, telles que définies dans la section 6-1.

7.2. Quelques intuitions sur le système PRA

Rappelons ici la définition de la classe des fonctions primitives récursives que l'on dénote par $\mathcal{PR}$. Il s'agit de la plus petite classe de fonctions de $\mathbb{N}^n$ vers $\mathbb{N}$ pour $n \in \mathbb{N}$ quelconque, contenant les fonctions de base suivantes.

- (a) La fonction successeur : $\text{succ}(x) = x + 1$.
 - (b) Les fonctions constantes : $c_m^n(x_1, \dots, x_n) = m$ pour tous $n, m \geq 0$.
 - (c) Les projections : $p_i^n(x_1, \dots, x_n) = x_i$ pour tout n et tout $i \in 1, \dots, n$
- et qui est close par les opérations suivantes :

- ▷ *composition* : si $g_1, \dots, g_m \in \mathcal{PR}$ sont des fonctions de n variables et que $h \in \mathcal{PR}$ soit une fonction de m variables, alors la fonction

$$f(\bar{x}) = h(g_1(\bar{x}), \dots, g_m(\bar{x}))$$

est dans $\mathcal{PR}$;

- ▷ *réursion primitive* : si $g, h \in \mathcal{PR}$ sont des fonctions de respectivement n et $n + 2$ variables, la fonction f définie par $f(\bar{x}, 0) = g(\bar{x})$ et $f(\bar{x}, m + 1) = h(\bar{x}, m, f(\bar{x}, m))$ est une fonction de $n + 1$ variables dans $\mathcal{PR}$.

Le lecteur peut se reporter à la section 6-1 pour une argumentation du fait que les fonctions primitives récurrentes sont calculables (et par définition totales) et à la section 6-3 pour une étude formelle de ces fonctions et enfin au théorème 4.6 pour une preuve que ces fonctions sont toutes prouvables totales dans RCA_0 . Il ne s'agit toutefois que d'une sous-classe stricte de la classe de toutes les fonctions calculables totales, et il est nécessaire de rajouter la clôture par le schéma de minimisation afin de les obtenir toutes.

Si la classe $\mathcal{PR}$ ne contient pas toutes les fonctions calculables, elle contient cependant un grand nombre de fonctions usuelles, comme l'addition, la multiplication, la fonction exponentielle, mais également des fonctions de codage et décodage, comme celle des paires d'entiers de Cantor. Elle forme donc une classe robuste de fonctions calculables avec de bonnes propriétés de clôture.

Le système PRA se place donc dans un langage contenant un symbole pour chaque fonction primitive récurrente. Ses axiomes comportent le schéma d'induction I_{ouvert} , et de la même manière que les axiomes de $\mathbf{Q}$ régulent le comportement de l'addition et de la multiplications, ceux de PRA vont réguler chacune des fonctions primitives récurrentes.

Ainsi, si $f, h, g_1, \dots, g_m$ sont, à titre d'exemple, des symboles de fonctions tels que f doit être interprété par $\bar{x} \mapsto h(g_1(\bar{x}), \dots, g_m(\bar{x}))$, alors

$$\forall \bar{x} \ f(\bar{x}) = h(g_1(\bar{x}), \dots, g_m(\bar{x}))$$

sera un axiome.

Les résultats suivants, dont les preuves dépassent le cadre de cet ouvrage, montrent que le système PRA est suffisant pour montrer tous les énoncés Π_2 prouvables dans WKL_0 .

Théorème 7.1 (Friedman (voir Simpson [203]))

Le système RCA_0 est une extension conservative de PRA pour les énoncés Π_2 de PA.

Notons que si l'on est tout à fait formel, le théorème ci-dessus n'entre pas dans le cadre des énoncés de conservation, pour les deux raisons suivantes.

Tout d'abord, le langage de l'arithmétique du second ordre n'est pas une extension de celui de l'arithmétique primitive réursive. En effet, PRA comporte des symboles de fonction pour chaque fonction primitive réursive. Cependant, comme nous l'avons vu avec le théorème 4.6, toute fonction primitive réursive est définissable par une formule de l'arithmétique du premier ordre. On peut donc traduire toute formule de PRA en une formule de PA, et *a fortiori* en une formule de l'arithmétique du second ordre.

Ensuite, les énoncés Π_2 de l'arithmétique du premier ordre ne sont pas des énoncés de l'arithmétique primitive réursive : en effet, PA possède un symbole de fonction pour l'addition, un pour la multiplication, et un symbole de relation pour l'ordre. Cependant, ces deux opérations et la relation d'ordre sont primitives récursives, et il existe donc une traduction naturelle des énoncés de l'arithmétique vers le langage de PRA.

Il existe donc une double traduction des énoncés de PA vers PRA et réciproquement, de telle sorte que les formules obtenues par va-et-vient sont prouvablement équivalentes à la formule de départ. Le théorème 7.1 nous donne le corollaire suivant.

Corollaire 7.2 (Friedman)

Le système WKL_0 est une extension conservative de PRA pour les énoncés Π_2 .

PREUVE. Supposons que $\text{WKL}_0 \vdash F$ pour F un énoncé Π_2 . En particulier, F est Π_1^1 , donc $\text{RCA}_0 \vdash F$ par le théorème 6.15. Il vient alors, par le théorème 7.1, $\text{PRA} \vdash F$. ■

La connexion entre PRA et RCA_0 va plus loin : les fonctions primitives récursives sont exactement les fonctions totales qui sont prouvablement calculables dans RCA_0 .

Théorème (1.4 Parikh, voir Hajek et Pudlak [81])

Les fonctions RCA_0 -prouvablement calculables sont exactement les fonctions primitives récursives.

Ce dernier résultat est à mettre en perspective avec l'assertion selon laquelle RCA_0 capture les mathématiques calculables. Du point de vue de la calculabilité, qui se préoccupe des modèles standard, il existe un ω -modèle minimal (pour l'inclusion) de RCA_0 qui contient exactement les ensembles calculables au second ordre. Ainsi, du point de vue sémantique, RCA_0 capture les mathématiques calculables. En revanche, du point de vue de RCA_0 , les seuls prédicats que la théorie peut prouver comme étant Δ_1^0 sont les prédicats primitifs récursifs, et ainsi RCA_0 ne peut prouver que l'existence

des ensembles primitifs récursifs. Cette différence vient du fait que la notion de Δ_1^0 est relative à une théorie ou à un modèle, contrairement à la notion de Σ_1^0 qui est syntaxique et donc absolue.

Chapitre 24

Réductions calculatoires

Comme mentionné dans la section 22-8.1, la plupart des théorèmes que nous étudierons dans cette partie s'expriment sous la forme

$$\forall X (F(X) \rightarrow \exists Y G(X, Y)),$$

où F et G sont des formules arithmétiques. On peut alors voir le théorème comme un problème mathématique, formulé en termes d'instances et de solutions. On dira qu'un ensemble X est une instance du problème si $F(X)$ est vrai, et que Y est une solution de l'instance X si $G(X, Y)$ est vrai.

Exemple 1. Le lemme de König est le problème dont les instances sont les arbres infinis à branchement fini, et pour lequel les solutions d'un arbre sont ses chemins infinis.

Ce genre de théorème porte sur l'existence d'objets du second ordre, et l'écrasante majorité des séparations de la forme $\text{RCA}_0 + T_1 \not\vdash T_2$, où T_1, T_2 sont deux théorèmes de ce type, se fait en construisant un ω -modèle du système $\text{RCA}_0 + T_1$ qui n'est pas un modèle de T_2 . C'est ce que nous allons étudier dans ce chapitre. Nous nous recentrons donc sur les ω -modèles afin de nous concentrer sur la puissance calculatoire des théorèmes. Définissons ce que nous entendons par problème en toute généralité.

Définition 2. Un *problème* est une relation $P \subseteq 2^{\mathbb{N}} \times 2^{\mathbb{N}}$. Une *instance* de P est un élément de $\text{dom } P = \{X \in 2^{\mathbb{N}} : \exists Y (X, Y) \in P\}$. Étant donné une instance X de P , on notera $P(X) = \{Y : (X, Y) \in P\}$ la classe des *solutions* de X . ◇

Pour toute formule $\forall X(F(X) \rightarrow \exists Y G(X, Y))$ vraie, où F et G sont des formules arithmétiques, on peut associer le problème

$$P_{F,G} = \{(X, Y) \in 2^{\mathbb{N}} \times 2^{\mathbb{N}} : F(X) \wedge G(X, Y)\}.$$

On a alors $\text{dom}(P_{F,G}) = \{X \in 2^{\mathbb{N}} : F(X)\}$ et pour tout $X \in \text{dom}(P_{F,G})$, la classe $P_{F,G}(X) = \{Y \in 2^{\mathbb{N}} : G(X, Y)\}$. On appelle *problème* Π_2^1 un problème de cette forme, et on l'identifiera à sa formule correspondante.

Exemple 3. Le lemme de König est un problème Π_2^1 . Voyons un autre exemple. Le théorème ADS — acronyme de « Ascending-Descending Sequence » — affirme pour tout ordre linéaire infini $R \subseteq \mathbb{N} \times \mathbb{N}$, l'existence d'une suite infinie strictement croissante, ou celle d'une suite infinie strictement décroissante. Ce théorème est donc le problème Π_2^1 $\text{ADS} \subseteq 2^{\mathbb{N}} \times 2^{\mathbb{N}}$ dont les instances sont les ordres infinis linéaires sur $\mathbb{N}$ et, pour chaque instance $R \in \text{dom}(\text{ADS})$, les solutions $\text{ADS}(R)$ sont les suites infinies strictement croissantes ou strictement décroissantes d'éléments.

Nous allons étudier différentes notions de réductions entre les problèmes. Informellement, une réduction $P \leq Q$ entre deux problèmes P et Q signifie que le problème Q est au moins aussi difficile à résoudre que P , au sens où si l'on disposait d'une boîte noire permettant de calculer des solutions des instances de Q , il serait possible de calculer des solutions des instances de P . Voyons tout de suite notre première réduction, qui découle directement des techniques de séparations usuelles de théorèmes dans RCA_0 .

1. ω -réduction

L'implication dans RCA_0 induit une réduction purement calculatoire en se restreignant aux ω -structures. Considérons un problème Π_2^1 P , et sa formule correspondante $\forall X(F(X) \rightarrow \exists Y G(X, Y))$. Soit $\mathcal{M} = (M, S, +, \times, <)$ une $\mathcal{L}_{Z_2}$ -structure. Alors, $\mathcal{M} \models P$ si, et seulement si, pour tout $X \in S$ tel que $\mathcal{M} \models F(X)$, il existe $Y \in S$ tel que $\mathcal{M} \models G(X, Y)$. Rappelons qu'une ω -structure $\mathcal{M} = (\omega, \mathcal{I}, +, \times, <)$ est un modèle de RCA_0 ssi $\mathcal{I}$ est un idéal Turing. Cela nous conduit à la définition suivante.

Définition 1.1. Un idéal Turing $\mathcal{I}$ *satisfait* un problème P (noté $\mathcal{I} \models P$) si pour toute instance X de P dans $\mathcal{I}$, il existe une solution de X dans $\mathcal{I}$. $\diamond$

Notons que pour un problème Π_2^1 P , un idéal Turing $\mathcal{I}$ satisfait P si, et seulement si, l' ω -structure $\mathcal{M} = (\omega, \mathcal{I}, +, \times, <)$ satisfait sa formule correspondante. La restriction aux ω -structures permet de définir une notion de satisfiabilité sur des problèmes arbitraires, au-delà des problèmes Π_2^1 .

Définition 1.2. Un problème P est ω -réductible à Q (noté $P \leq_\omega Q$) si tout idéal Turing qui satisfait Q satisfait aussi P . $\diamond$

En particulier, si P et Q sont des problèmes Π_2^1 , alors $P \leq_\omega Q$ si, et seulement si, tout ω -modèle de $\text{RCA}_0 + Q$ est un ω -modèle de P .

Exemple 1.3. Considérons les problèmes KL et J , où KL est le problème associé au lemme de König (distinct de WKL , le lemme faible de König), et où J est le problème du saut Turing, dont les instances sont les éléments de $2^{\mathbb{N}}$, et dont l'unique solution de l'instance X est X' . Les idéaux Turing satisfaisant J sont par définition les idéaux de saut (voir la section 22-6.1), et caractérisent les ω -modèles de ACA_0 .

Montrons que $\text{KL} \leq_\omega \text{J}$. Soit $\mathcal{I} \models \text{J}$, et soit $T \in \mathcal{I}$ une instance de KL . Alors, T est un arbre calculatoirement borné relativement à T' . Par le corollaire 8-7.9 relativisé à T' , tout degré PA relativement à T' calcule un chemin de T . En particulier, le double saut Turing de T calcule un chemin de T . Comme $\mathcal{I} \models \text{J}$ et $T \in \mathcal{I}$, alors $T' \in \mathcal{I}$, et donc $T'' \in \mathcal{I}$. Par clôture de $\mathcal{I}$ par réduction Turing, $\mathcal{I}$ contient un chemin de T . Ainsi, $\mathcal{I} \models \text{KL}$.

Montrons que $\text{J} \leq_\omega \text{KL}$. Soit $\mathcal{I} \models \text{KL}$, et soit $X \in \mathcal{I}$ une instance de J . Par la proposition 8-7.4, il existe un arbre X -calculable T à branchement fini dont l'unique chemin est Turing équivalent à X' . Comme $\mathcal{I}$ est un idéal Turing, $T \in \mathcal{I}$. Comme $\mathcal{I} \models \text{KL}$, l'unique chemin de T appartient à $\mathcal{I}$, et comme $\mathcal{I}$ est un idéal Turing, $X' \in \mathcal{I}$. Donc, $\mathcal{I} \models \text{J}$.

Notons que dans l'exemple précédent, nous avons utilisé deux instances de J pour « résoudre » une instance de KL , tandis que dans le sens contraire, une seule instance de KL a suffi. Nous reviendrons sur ce point dans la section suivante.

Exercice 1.4. Rappelons que WKL et KL dénotent respectivement le lemme de König pour les arbres binaires infinis et les arbres infinis à branchement fini. Montrer que $\text{WKL} \leq_\omega \text{KL}$, mais $\text{KL} \not\leq_\omega \text{WKL}$. $\diamond$

1.1. Preuves de séparation

Soient P et Q des problèmes Π_2^1 . Comme déjà expliqué, si $\text{RCA}_0 + Q \vdash P$, alors en particulier $P \leq_\omega Q$ ¹. Par contraposée, la réduction $\leq_\omega$ permet de séparer des théorèmes : si l'on arrive à montrer $P \not\leq_\omega Q$, alors $\text{RCA}_0 + Q \not\vdash P$. La grande majorité des preuves de séparation en mathématiques à rebours sont de fait des preuves de séparation pour l' ω -réduction.

1. Il est à noter que l'on n'a pas nécessairement $P \leq_\omega Q$ implique $\text{RCA}_0 \vdash Q \rightarrow P$.

Exemple 1.5. La séparation de WKL_0 et ACA_0 de la proposition 22-7.3 s'est faite en montrant $\text{J} \not\leq_\omega \text{WKL}$.

Afin de prouver une séparation $\text{P} \not\leq_\omega \text{Q}$, il est nécessaire de créer un idéal Turing $\mathcal{I}$ tel que $\mathcal{I} \models \text{Q}$ mais $\mathcal{I} \not\models \text{P}$. Pour cela, on fixe en général une instance X_P de P dont les solutions sont « calculatoirement compliquées », et qui en particulier ne sont pas X_P -calculables. La classe

$$\mathcal{I}_0 = \{Z \in 2^{\mathbb{N}} : Z \leq_T X_P\}$$

est un idéal Turing contenant l'instance X_P , mais aucune de ses solutions. Ainsi, $\mathcal{I}_0 \not\models \text{P}$. Il s'agit ensuite d'étendre $\mathcal{I}_0$ en un idéal Turing $\mathcal{I} \models \text{Q}$, tout en s'assurant que $\mathcal{I}$ ne possède pas de solution pour X_P .

La démarche consiste à fixer une instance X_0 de Q dans $\mathcal{I}_0$, à choisir une solution Y_0 « calculatoirement faible » — suffisamment pour que $X_P \oplus Y_0$ ne calcule aucune solution de X_P — et à définir

$$\mathcal{I}_1 = \{Z \in 2^{\mathbb{N}} : Z \leq_T X_P \oplus Y_0\}.$$

L'idéal Turing $\mathcal{I}_1$ a fait une étape de progression pour satisfaire Q , en ajoutant une solution de X_0 , sans pour autant en ajouter une de X_P . En revanche, cette solution a aussi ajouté de nouvelles instances de Q dans $\mathcal{I}_1$, qu'il faudra également satisfaire. Il s'agit alors d'itérer intelligemment la construction, pour définir une suite croissante d'idéaux Turing $\mathcal{I}_0 \subseteq \mathcal{I}_1 \subseteq \mathcal{I}_2 \subseteq \dots$ de manière à prendre en compte toutes les instances de Q qui apparaissent dans $\mathcal{I}_n$ pour un certain n , tout en s'assurant d'éviter de rajouter des solutions de X_P . On obtient un idéal Turing $\mathcal{I} = \bigcup_n \mathcal{I}_n$ satisfaisant Q , sans pour autant satisfaire P .

La notion de solution « calculatoirement faible » correspondra à une propriété de faiblesse, telle que définie dans la section 4-8 sur la méthode des extensions finies :

Définition 1.6. Une *propriété de faiblesse* est une classe $\mathcal{W} \subseteq 2^{\mathbb{N}}$ close par le bas pour la réduction Turing : si $X \in \mathcal{W}$ et $Y \leq_T X$, alors $Y \in \mathcal{W}$. $\diamond$

En particulier, tout idéal Turing est une propriété de faiblesse. De nombreuses notions de calculabilité sont des propriétés de faiblesse sans toutefois forcément former des idéaux Turing.

Exemple 1.7. Les classes suivantes sont des propriétés de faiblesse :

1. les ensembles de degré low ;
2. les ensembles calculatoirement dominés ;
3. pour tout A non calculable, la classe $\mathcal{W} = \{Z \in 2^{\mathbb{N}} : A \not\leq_T Z\}$;
4. les ensembles arithmétiques.

Soit $\mathcal{W}$ une propriété de faiblesse non vide. Supposons que $\mathbf{P}$ possède une instance $X_{\mathbf{P}} \in \mathcal{W}$ telle que $\mathcal{W}$ ne contient aucune solution de $X_{\mathbf{P}}$. On va alors chercher à construire notre idéal $\mathcal{I} \subseteq \mathcal{W}$ avec $X_{\mathbf{P}} \in \mathcal{I}$ et tel que $\mathcal{I} \models \mathbf{Q}$. Au cours de la construction de notre idéal, on se retrouvera avec des instances de $\mathbf{Q}$ calculables relativement à des solutions d'instances précédemment considérées. Cela nous conduit à la définition suivante.

Définition 1.8. Soit $\mathcal{W}$ une propriété de faiblesse. Un problème $\mathbf{Q}$ *préserve* $\mathcal{W}$ si pour tout $Z \in \mathcal{W}$, toute instance $X \leqslant_T Z$ de $\mathbf{Q}$ admet une solution Y telle que $Z \oplus Y \in \mathcal{W}$. $\diamond$

Nous avons déjà vu un certain nombre de preuves de préservation en calculabilité.

Exemple 1.9. Le système WKL préserve la classe des ensembles low. Pour tout Z de degré low, pour tout arbre $T \subseteq 2^{<\mathbb{N}}$ infini Z -calculable, par le théorème 8-4.3 relativisé à Z , il existe $P \in [T]$ tel que $Z \oplus P$ est low relativement à Z . Or, si A est low, et si B est low relativement à A , alors B est low. En effet, $B' \leqslant_T A' \leqslant_T \emptyset'$. Il s'ensuit que $Z \oplus P$ est low. De la même manière, par le théorème 8-4.5, WKL préserve la classe des ensembles calculatoirement dominés.

La notion de préservation d'une propriété de faiblesse est un outil très puissant pour prouver des séparations pour l' ω -réduction.

Lemme 1.10. Supposons que $\mathbf{Q}$ préserve une propriété de faiblesse $\mathcal{W}$. Alors, pour tout $X \in \mathcal{W}$, il existe un idéal Turing $\mathcal{I} \subseteq \mathcal{W}$ contenant X tel que $\mathcal{I} \models \mathbf{Q}$. $\star$

PREUVE. Définissons une suite croissante d'ensembles $Z_0 \leqslant_T Z_1 \leqslant_T \dots$ telle que $Z_0 = X$, telle que pour tout $n \in \mathbb{N}$ on a $Z_n \in \mathcal{W}$, et telle que pour tout $n, e \in \mathbb{N}$, si $\Phi_e^{Z_n}$ est une $\mathbf{Q}$ -instance, alors $Z_{\langle n, e \rangle}$ calcule une solution de $\Phi_e^{Z_n}$.

Supposons que $Z_0, \dots, Z_n$ est défini. Soit $n+1 = \langle p, e \rangle$, avec $p, e \leqslant n$. Si $\Phi_e^{Z_p}$ n'est pas une instance de $\mathbf{Q}$, alors $Z_{n+1} = Z_n$. Sinon, comme $\mathbf{Q}$ préserve $\mathcal{W}$ et $Z_n \in \mathcal{W}$, il existe une solution $\tilde{Y}$ de $\Phi_e^{Z_p}$ — car, en particulier, $\tilde{Y}$ est aussi calculable en Z_n — telle que $Z_n \oplus \tilde{Y} \in \mathcal{W}$. Soit $Z_{n+1} = Z_n \oplus \tilde{Y}$.

Par l'exercice 22-5.11, $\mathcal{I} = \{Y \in 2^{\mathbb{N}} : \exists n Y \leqslant_T Z_n\}$ est un idéal Turing. Comme $Z_0 = X$, alors $X \in \mathcal{I}$. Enfin, montrons que $\mathcal{I} \models \mathbf{Q}$. Soit $\tilde{X}$ une instance de $\mathbf{Q}$ dans $\mathcal{I}$. Soit $n \in \mathbb{N}$ tel que $\tilde{X} \leqslant_T Z_n$, et soit $e \in \mathbb{N}$ tel que $\Phi_e^{Z_n} = \tilde{X}$. Alors, $Z_{\langle n, e \rangle}$ calcule une solution de $\tilde{X}$, $Z_{\langle n, e \rangle} \in \mathcal{I}$, et $\mathcal{I}$ contient donc une solution de $\tilde{X}$. $\blacksquare$

Le théorème suivant généralise de nombreuses constructions des mathématiques à rebours.

Théorème 1.11

Soit $\mathcal{W}$ une propriété de faiblesse. Supposons que Q préserve $\mathcal{W}$, contrairement à P . Alors, $P \not\leq_{\omega} Q$.

PREUVE. Comme P ne préserve pas $\mathcal{W}$, il existe un élément $Z \in \mathcal{W}$ et une P -instance $X \leq_T Z$ telle que pour toute P -solution Y , $Z \oplus Y \notin \mathcal{W}$. Comme Q préserve $\mathcal{W}$, par le lemme 1.10, il existe un idéal Turing $\mathcal{I} \subseteq \mathcal{W}$ tel que $Z \in \mathcal{I}$ et $\mathcal{I} \models Q$. Montrons que $\mathcal{I} \not\models P$. En effet, par clôture de $\mathcal{I}$ par la réduction Turing, $X \in \mathcal{I}$. Soit Y une P -solution de X . Alors, $Z \oplus Y \notin \mathcal{W}$, donc $Z \oplus Y \notin \mathcal{I}$; or, $Z \in \mathcal{I}$, donc par clôture de $\mathcal{I}$ par jointure, $Y \notin \mathcal{I}$. Ainsi, $\mathcal{I}$ est un idéal Turing satisfaisant Q mais non P , et donc $P \not\leq_{\omega} Q$. ■

Corollaire 1.12

Soit $\mathcal{W}$ une propriété de faiblesse, et soient P, Q des problèmes Π_2^1 . Supposons que Q préserve $\mathcal{W}$, contrairement à P . Alors, $\text{RCA}_0 + Q \not\models P$.

PREUVE. Par le théorème 1.11, $P \not\leq_{\omega} Q$, donc il existe un ω -modèle $\mathcal{M}$ de $\text{RCA}_0 + P$ qui n'est pas un modèle de Q , et donc $\text{RCA}_0 + Q \not\models P$. ■

Par exemple, WKL préserve la classe des ensembles low, tandis que KL ne la préserve pas, donc $\text{KL} \not\leq_{\omega} \text{WKL}$, et $\text{RCA}_0 + \text{WKL} \not\models \text{KL}$.

1.2. Préservation de cônes

Parmi les propriétés de faiblesse, on comptera plusieurs grandes familles récurrentes en mathématiques à rebours. La plus emblématique est celle de *préservation de cône*, qui exprime l'incapacité d'un problème de « coder » un ensemble non calculable dans chacune de ses solutions.

Définition 1.13. Soit $k \in \mathbb{N}$. Un problème P *préserve k cônes* (resp. ω cônes) si pour tout $Z \in 2^{\mathbb{N}}$, toute suite $(C_j)_{j < k}$ (resp. $(C_j)_{j \in \mathbb{N}}$) d'ensembles non Z -calculables, et pour toute instance $X \leq_T Z$ de P , il existe une solution Y de X telle C_j n'est $Z \oplus X$ -calculable pour aucun $j < k$ (resp. aucun $j \in \mathbb{N}$). ◇

Formulé dans le langage des préservations, pour tout $k \in \mathbb{N}$ et toute suite d'ensembles $\bar{C} = (C_j)_{j < k}$, on associe la propriété de faiblesse

$$\mathcal{W}_{\bar{C}} = \{Z : \forall j < k \ C_j \not\leq_T Z\}.$$

Notons que $\mathcal{W}_{\bar{C}} = \emptyset$ si l'un des ensembles C_j est calculable, auquel cas tout problème préserve $\mathcal{W}_{\bar{C}}$. En déroulant les définitions, P préserve k cônes ssi P

préserve $\mathcal{W}_{\overline{C}}$ pour toute suite d'ensembles $\overline{C} = (C_j)_{j < k}$. En particulier, par le théorème 1.11, si Q préserve k cônes, contrairement à P , alors $P \not\leq_{\omega} Q$.

Exemple 1.14. Le système WKL préserve ω cônes, au vu du théorème 8-4.7. En revanche, KL ne préserve pas de cône, car il existe un arbre calculable à branchement fini dont tous les chemins calculent le problème de l'arrêt. Il s'ensuit par le théorème 1.11 que $KL \not\leq_c WKL$.

La notion de préservation de cône est notamment utilisée pour séparer des théorèmes de ACA_0 . Nous en verrons un exemple non trivial avec le théorème de Ramsey pour les paires. Outre son utilisation pour les théorèmes de séparation, la notion fut étudiée pour elle-même, notamment par Downey, Greenberg, Harrison-Trainor, Patey et Turetsky [46], qui en ont trouvé plusieurs caractérisations équivalentes. Mentionnons en particulier le joli résultat suivant.

Théorème 1.15 ([46])

Un problème P préserve un cône si, et seulement si, il préserve ω cônes.

2. Réduction calculatoire

L'implication dans RCA_0 et l' ω -réduction sont grossières d'un point de vue calculatoire, car elles ne prennent pas en compte le nombre d'applications de chaque problème. Pour prouver $RCA_0 \vdash Q \rightarrow P$, on peut utiliser la prémisse Q autant de fois que l'on veut dans notre démonstration. De la même manière, pour prouver $P \leq_{\omega} Q$, on est autorisé à faire usage d'un nombre arbitraire d'instances de Q pour résoudre P .

Dès lors que l'on suppose l'existence du saut Turing dans RCA_0 , le système obtenu implique ACA_0 , et implique l'existence de toutes les itérations finies du saut Turing. Dzhamalov [52] a introduit une notion de réduction plus fine, sensible au nombre d'applications de chaque problème.

Définition 2.1. Un problème P est *calculatoirement réductible* à Q (ce que l'on note $P \leq_c Q$) si pour toute instance X de P , il existe une instance $\tilde{X}$ de Q calculable en X , telle que pour toute Q -solution $\tilde{Y}$ de $\tilde{X}$, l'ensemble $X \oplus \tilde{Y}$ calcule une P -solution de X . $\diamond$

La réduction calculatoire est plus proche des notions de réduction étudiées en théorie de la complexité. Notons que la Q -solution $\tilde{Y}$ de $\tilde{X}$ a le droit de faire appel à l'instance X de P pour calculer une P -solution de X . La réduction calculatoire est un raffinement de l' ω -réduction.

Exercice 2.2. Montrer que si $P \leq_c Q$, alors $P \leq_\omega Q$. $\diamond$

L'implication inverse n'est pas vraie en général. Intuitivement, la réduction calculatoire est une ω -réduction qui n'autorise qu'une seule application du problème de droite. Elle permet par exemple de donner un cadre formel pour signifier qu'une seule application du problème J ne suffit pas pour prouver la réduction $KL \leq_\omega J$ dans l'exemple 1.3.

Exercice 2.3. Avec toujours KL et J les problèmes définis dans l'exemple 1.3. Montrer que $KL \not\leq_c J$. $\diamond$

La réduction calculatoire est donc strictement plus fine que l' ω -réduction.

Remarque

La réduction $RCA_0 \vdash Q \rightarrow P$ n'implique pas $P \leq_c Q$, car plusieurs instances de Q sont peut-être nécessaires dans la preuve. Il est à noter aussi que la réduction $P \leq_c Q$ n'implique pas non plus $RCA_0 \vdash Q \rightarrow P$, car les réductions Turing utilisées dans $P \leq_c Q$ ne sont pas nécessairement prouvablement totales à l'aide seule de l'induction Σ_1^0 .

Cependant, beaucoup de preuves de $RCA_0 \vdash Q \rightarrow P$ sont en réalité des preuves de $P \leq_c Q$ qui se formalisent dans RCA_0 .

Comme nous l'avons vu, $P \leq_\omega Q$ n'implique pas en général $P \leq_c Q$. C'est cependant le cas lorsque $Q = WKL$. Ici, WKL est le problème dont les instances sont les arbres binaires infinis, et les solutions sont les chemins de ces arbres. La raison de cette implication provient du résultat suivant, dû à Scott.

Proposition 2.4 (Scott [194]). Soit P un degré PA. Il existe un idéal de Scott dont tous les éléments sont P -calculables. $\star$

PREUVE. Soit T une fonctionnelle d'arbre telle que pour tout $X \in 2^\mathbb{N}$, T^X est un arbre binaire dont tous les chemins sont de degré PA relativement à X . Par exemple, $T^X = \{\sigma \in 2^{<\mathbb{N}} : \forall e < |\sigma| \Phi_e^X(e)[\sigma] \neq \sigma(e)\}$. Soit alors $\mathcal{C} = \{\bigoplus_i X_i \in 2^\mathbb{N} : \forall i X_{i+1} \in [T^{X_i}]\}$. Notons que $\mathcal{C}$ est une classe Π_1^0 non vide. En particulier, P calcule un membre $\bigoplus_i X_i$ de $\mathcal{C}$. Notons que X_{i+1} est de degré PA relativement à X_i , et en particulier $X_{i+1} \geq_T X_i$ (voir l'exercice 8-7.6).

Soit $\mathcal{I} = \{Z \in 2^\mathbb{N} : \exists i Z \leq_T X_i\}$. Par l'exercice 22-5.11, $\mathcal{I}$ est un idéal Turing. De plus, pour tout $Z \in \mathcal{I}$, il existe i tel que $Z \leq_T X_i$; or, $X_{i+1} \in \mathcal{I}$ est de degré PA relativement à X_i , donc à Z . Ainsi, $\mathcal{I}$ est un idéal de Scott. Enfin, tous les éléments de $\mathcal{I}$ sont P -calculables. $\blacksquare$

La proposition suivante est une conséquence directe du résultat de Scott.

Proposition 2.5 (Hirschfeldt et Jockusch [88]). Soit P un problème. Si $P \leq_\omega \text{WKL}$, alors $P \leq_c \text{WKL}$. ★

PREUVE. Soit X une instance de P . Soit T un arbre X -calculable dont tous les chemins sont de degré PA relativement à X . L'arbre T est une instance X -calculable de WKL . Soit $\tilde{Y}$ une WKL -solution de T , autrement dit $\tilde{Y} \in [T]$. Montrons que $\tilde{Y}$ calcule une P -solution de X . Par la proposition 2.4 relativisée à X , il existe un idéal de Scott $\mathcal{I}$ contenant X et dont tous les éléments sont $X \oplus \tilde{Y}$ -calculables (donc $\tilde{Y}$ -calculables, car $\tilde{Y} \geq_T X$). Comme $\mathcal{I}$ est un idéal de Scott, $\mathcal{I} \models \text{WKL}$; or, $P \leq_\omega \text{WKL}$, donc $\mathcal{I} \models P$. Comme $X \in \mathcal{I}$, il existe une P -solution Y de X dans $\mathcal{I}$, et il existe donc une P -solution $\tilde{Y}$ -calculable de X . ■

Notons que dans la preuve précédente, $\tilde{Y}$ n'a pas besoin de l'instance X en oracle pour calculer une P -solution de X . Nous verrons dans la section 5 que cela correspond à une notion plus forte que la réduction calculatoire.

Terminons en introduisant une dernière notion qui formalise le fait que certains problèmes sont triviaux du point de vue de la réduction calculatoire.

Définition 2.6. Un problème P est *calculatoirement vrai* si toute instance X de P calcule une P -solution de X . ◇

Nous verrons dans la proposition 25-2.4 que c'est par exemple le cas du principe des tiroirs.

Exercice 2.7. Montrer que P est calculatoirement vrai si, et seulement si, $P \leq_c \text{Id}$, où Id est le problème identité dont les solutions sont égales aux instances. ◇

3. Réduction Weihrauch

Considérons la preuve dans RCA_0 du théorème des valeurs intermédiaires (cf. la proposition 22-5.13). Étant donné une fonction continue $f : [0, 1] \rightarrow \mathbb{R}$ satisfaisant $f(0) < 0 < f(1)$, la preuve procède par analyse de cas : soit il existe un nombre rationnel $q \in]0, 1[$ tel que $f(q) = 0$, auquel cas tout rationnel étant calculable, f possède une solution calculable, soit l'on obtient un réel r tel que $f(r) = 0$ par une recherche dichotomique calculable. Peut-on se passer de cette analyse de cas ? De même que la réduction calculatoire raffine l' ω -réduction en prenant en compte le nombre d'applications, la réduction de Weihrauch permet de donner un sens formel à la question précédente en prenant en compte l'uniformité des preuves.

Définition 3.1

Un problème P est *Weihrauch réductible* à Q (noté $P \leq_W Q$) s'il existe deux fonctionnelles Turing Φ et Ψ telles que pour toute instance X de P , l'ensemble Φ^X est une instance de Q , telle que pour toute Q -solution $\tilde{Y}$ de Φ^X , l'ensemble $\Psi^{X \oplus \tilde{Y}}$ est une P -solution de X . On dit que les fonctionnelles Φ et Ψ *témoignent* de $P \leq_W Q$. $\diamond$

Il résulte directement des définitions que si $P \leq_W Q$, alors $P \leq_c Q$. La réduction de Weihrauch a été introduite par Klaus Weihrauch dans les années 80 comme un outil de l'analyse calculable, avant de connaître un renouveau avec les travaux de Gherardi et Marcone [73], en 2009, montrant l'utilité de cette réduction pour analyser le contenu uniforme des preuves en mathématiques à rebours. Depuis, les degrés Weihrauch ont connu un essor et se présentent en formalisme concurrent de l'implication dans RCA_0 pour l'analyse des théorèmes.

3.1. Théorème des valeurs intermédiaires

Afin de formuler plus précisément notre question sur l'uniformité du théorème des valeurs intermédiaires, nous introduisons un analogue pour la réduction de Weihrauch de la notion de calculatoirement vrai.

Définition 3.2. Un problème P est *uniformément vrai* s'il existe une fonctionnelle Turing Φ telle que pour toute instance X de P , l'ensemble Φ^X est une P -solution de X . $\diamond$

Exercice 3.3. Montrer que P est uniformément vrai si, et seulement si, la réduction $P \leq_W Id$ est vraie — où Id désigne le problème identité dont les solutions sont égales aux instances. $\diamond$

Nous avons maintenant les outils nécessaires pour répondre à la question formulée en début de section, à savoir existe-t-il une preuve du théorème des valeurs intermédiaires sans analyse de cas ?

Proposition 3.4 (Weihrauch [233]). Le théorème des valeurs intermédiaires n'est pas uniformément vrai. $\star$

PREUVE

Soit Φ une fonctionnelle Turing telle que pour toute fonction $f : [0, 1] \rightarrow \mathbb{R}$ continue satisfaisant $f(0) < 0 < f(1)$, le réel $\Phi(f)$ est une racine de f . Le concept de fonctionnelle calculable $\Phi : \mathcal{C}([0, 1], \mathbb{R}) \rightarrow \mathbb{R}$ se définit de la manière suivante. La fonctionnelle Φ lance son calcul sur un élément de $2^{\mathbb{N}}$ qui représente une fonction continue comme expliqué dans la section 22-4, et renvoie un élément de $2^{\mathbb{N}}$ qui représente — toujours comme expliqué dans la section 22-4 — un réel. On demande une seule chose, en l'occurrence que,

sur toute représentation de f , notre processus calcule une représentation de $\Phi(f)$.

Soit $f : [0, 1] \rightarrow \mathbb{R}$ une fonction continue strictement croissante sur l'intervalle $[0, 0.25]$, strictement décroissante sur l'intervalle $[0.25, 0.75]$, et strictement croissante sur $[0.75, 1]$, telle que

$$f(0) = -1, \quad f(0.25) = 0.5, \quad f(0.5) = 0, \quad f(0.75) = -0.5 \quad \text{et} \quad f(1) = 1$$

(voir la figure 3.5). Soit $g : [0, 1] \rightarrow \mathbb{R}$ définie par $g(a) = \Phi(x \mapsto f(x) + a)$. Le lecteur pourra vérifier que g est une fonction continue, en particulier car Φ est calculable dans le sens donné ci-dessus. Aussi, pour $a \in]0.5, 1[$, la fonction $x \mapsto f(x) + a$ n'a qu'une seule racine (celle de gauche). Par continuité, g continuera de renvoyer la racine la plus à gauche sur $[0, 0.5]$. À l'inverse, pour $a \in]-1, -0.5[$, la fonction $x \mapsto f(x) + a$ n'a que la racine de droite. Par continuité, g renverra la racine la plus à droite sur $[-0.5, 0]$. On a donc deux valeurs différentes pour $g(0)$, d'où une contradiction. ■

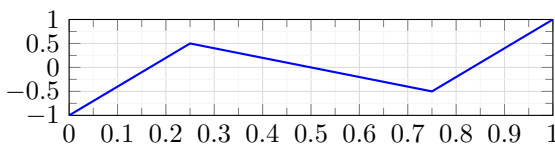


FIGURE 3.5 – Graphe de la fonction f de la proposition 3.4

La structure des degrés induits par la réduction Weihrauch se révèle extrêmement riche. Le lecteur qui voudrait en savoir plus trouvera une introduction détaillée dans *Weihrauch Complexity in Computable Analysis* de Brattka, Gherardi et Pauly [24].

3.2. Problèmes de choix et parallélisation

La réduction Weihrauch fournit un panel de problèmes très faibles qui permettent de mesurer des puissances calculatoires qui ne sont pas analysables en termes de degrés Turing. Parmi ces problèmes, on compte les principes d'omniscience.

Définition 3.6

- (1) Le *principe limité d'omniscience* (LPO) est le problème dont les instances sont des suites $X \in 2^{\mathbb{N}}$ et qui décide s'il existe $n \in \mathbb{N}$ tel que $X(n) = 0$.
- (2) Le *principe très limité d'omniscience* (LLPO) est le problème dont les instances sont des suites $X \in 2^{\mathbb{N}}$ telles que $X(n) = 1$ pour au plus un n , et dont les solutions sont les $i < 2$ tels que $\forall n \, X(2n+i) = 0$. ◇

Notons que le co-domaine de ces problèmes n'est pas dans $2^{\mathbb{N}}$ mais dans $\mathbb{N}$. Il est cependant possible de représenter tout entier n comme la suite $1^n 0^\infty$. Les acronymes viennent de l'anglais *Limited Principle of Omniscience* et *Lesser Limited Principle of Omniscience*. Chaque instance du principe LPO possède une solution unique, tandis que certaines instances de LLPO possèdent deux solutions.

Exercice 3.7. (★) Montrer que la réduction $\text{LLPO} \leq_W \text{LPO}$ est vraie, mais que $\text{LPO} \leq_W \text{LLPO}$ ne l'est pas. $\diamond$

Exercice 3.8. (★) Montrer que le principe très limité d'omniscience LLPO n'est pas uniformément vrai. $\diamond$

Les problèmes LPO et LLPO peuvent être vus comme des versions atomiques de phénomènes calculatoires bien connus à travers l'opération de parallélisation.

Définition 3.9. La *parallélisation* d'un problème P est le problème $\widehat{P}$ dont les instances sont de la forme $\bigoplus_n X_n$, où $X_n \in \text{dom } P$ pour tout n . Une $\widehat{P}$ -solution de $\bigoplus_n X_n$ est un ensemble de la forme $\bigoplus_n Y_n$, où Y_n est une P -solution de X_n pour tout n . $\diamond$

Il est aisé de montrer que l'opération de parallélisation est stable par réduction Weihrauch, et correspond donc à une opération sur les degrés Weihrauch.

Exercice 3.10. Montrer que si $P \leq_W Q$, alors $\widehat{P} \leq_W \widehat{Q}$. $\diamond$

Rappelons que le problème du saut Turing est le problème J , dont les instances sont les ensembles $X \in 2^{\mathbb{N}}$, chacune ayant pour unique solution son saut Turing X' . La proposition suivante montre que le problème LPO peut être vu comme un bit d'information du saut Turing.

Proposition 3.11. On a $J \equiv_W \widehat{\text{LPO}}$. $\star$

PREUVE. Montrons que $J \leq_W \widehat{\text{LPO}}$. Pour tout e , soit Γ_e la fonctionnelle Turing définie par $\Gamma_e^X(s) = 0$ ssi $\Phi_e^X(e)[s] \downarrow$. Notons que pour tout X , Γ_e^X vu comme une instance de LPO a pour solution 1 ssi $\Phi_e^X(e) \downarrow$, autrement dit ssi $e \in X'$. Soit Φ la fonctionnelle Turing définie par $\Phi^X = \bigoplus_e \Gamma_e^X$. Soit Ψ la fonctionnelle définie par $\Psi(X \oplus Y) = Y$. Les fonctionnelles Φ et Ψ témoignent de $J \leq_W \widehat{\text{LPO}}$.

Montrons que $\widehat{\text{LPO}} \leq_W J$. Soit Ψ définie comme suit : $\Psi^{X \oplus Y}(n) = Y(e_n)$, où e_n est l'indice Turing de la machine telle que $\Phi_{e_n}^X(e_n) \downarrow$ ssi il existe $s \in \mathbb{N}$ tel que $\langle n, s \rangle \notin X$. Montrons que si $\bigoplus_r X_r$ est une instance de $\widehat{\text{LPO}}$, alors $\Psi((\bigoplus_r X_r) \oplus (\bigoplus_r X_r)')$ est une $\widehat{\text{LPO}}$ -solution de $\bigoplus_r X_r$.

Soit $n \in \mathbb{N}$; on a $\Psi((\bigoplus_r X_r) \oplus (\bigoplus_r X_r)', n) = (\bigoplus_r X_r)'(e_n)$, où $\Phi_{e_n}^{\bigoplus_r X_r}(e_n) \downarrow$, ssi il existe $s \in \mathbb{N}$ tel que $s \notin X_n$. Ainsi, $(\bigoplus_r X_r)'(e_n) = 1$ ssi il existe un $s \in \mathbb{N}$ tel que $s \notin X_n$, donc pour tout n , $\Psi((\bigoplus_r X_r) \oplus (\bigoplus_r X_r)', n)$ est une LPO-solution de X_n . Ainsi, $\Psi((\bigoplus_r X_r) \oplus (\bigoplus_r X_r)')$ est une $\widehat{\text{LPO}}$ -solution de $\bigoplus_r X_r$. La fonctionnelle identité ainsi que Ψ témoignent donc de $\widehat{\text{LPO}} \leq_W J$. ■

De la même manière, le problème LLPO peut être vu comme une étape du lemme faible de König.

Exercice 3.12. (★) Montrer que $\text{WKL} \equiv_W \widehat{\text{LLPO}}$. ◇

Remarque

L'opérateur de parallélisation permet d'établir des ponts entre la réduction Weihrauch et la réduction calculatoire. Par l'exercice 3.10, si $P \leq_W Q$, alors $\widehat{P} \leq_W \widehat{Q}$, donc en particulier $\widehat{P} \leq_c \widehat{Q}$. La contraposée nous permet de prouver un certain nombre de résultats de séparation dans les degrés Weihrauch à l'aide de la calculabilité. Par exemple, par le théorème de base low pour les classe Π_1^0 , $J \not\leq_c \text{WKL}$. Il s'ensuit, par la proposition 3.11 et l'exercice 3.12, que $\text{LPO} \not\leq_W \text{LLPO}$. De la même manière, si P est uniformément vrai, alors $\widehat{P}$ est calculatoirement vrai. On en déduit donc que, WKL n'étant pas calculatoirement vrai, LLPO n'est pas uniformément vrai.

4. Jeux de réduction

Il est souvent plus intuitif de penser aux réductions calculatoires comme à des jeux à deux : le premier joueur tente de convaincre que la réduction est vraie, tandis que le second joueur met la réduction à l'épreuve en tentant de faire les pires choix de solutions. Hirschfeldt et Jockusch [88] ont donné une caractérisation de l' ω -réduction en termes de jeux.

Définition 4.1. Soient P et Q des problèmes. Le *jeu de réduction*

$$G(Q \rightarrow P)$$

est un jeu à deux joueurs qui procède comme suit.

- (1) Si au cours du jeu, un joueur se retrouve bloqué, l'autre joueur gagne.
- (2) Si un joueur gagne, le jeu se termine.
- (3) Au premier tour, le joueur 1 joue une instance X_0 de P . Ensuite, le joueur 2 joue soit une solution X_0 -calculable de X_0 et gagne, ou joue une instance X_0 -calculable Y_1 de Q .

- (4) Au tour $n > 2$, le joueur 1 joue une solution X_{n-1} de la $\mathbf{Q}$ -instance Y_{n-1} . Ensuite, le joueur 2 joue soit une $\mathbf{P}$ -solution $(X_0 \oplus \cdots \oplus X_{n-1})$ -calculable de X_0 et gagne, ou joue une instance $(X_0 \oplus \cdots \oplus X_{n-1})$ -calculable Y_n de $\mathbf{Q}$.
- (5) Si le jeu ne se termine pas, le joueur 1 gagne. ◇

Notons que le joueur 1 ne peut jamais se retrouver bloqué, car par définition, toute instance admet une solution. En revanche, le joueur 2 peut se retrouver bloqué car $\mathbf{Q}$ peut n'avoir aucune instance $(X_0 \oplus \cdots \oplus X_{n-1})$ -calculable. La plupart des problèmes considérés en mathématiques à rebours ont des instances calculables, et le jeu ne se retrouve jamais bloqué.

Proposition 4.2 (Hirschfeldt et Jockusch [88]). On a $\mathbf{P} \leq_\omega \mathbf{Q}$ si, et seulement si, le joueur 2 a une stratégie gagnante pour $G(\mathbf{Q} \rightarrow \mathbf{P})$. ★

PREUVE. Supposons $\mathbf{P} \leq_\omega \mathbf{Q}$. Définissons une stratégie gagnante pour le joueur 2. Au tour n , quand le joueur 2 peut jouer, $X_0, \dots, X_{n-1}$ et $Y_1, \dots, Y_{n-1}$ ont déjà été définis. Si notre joueur 2 peut gagner en jouant une $\mathbf{P}$ -solution $(X_0 \oplus \cdots \oplus X_{n-1})$ -calculable de X_0 , alors il la joue et gagne. Sinon, soit $n = \langle m, e \rangle$. Si $\Phi_e(X_0 \oplus \cdots \oplus X_{n-1})$ est une instance de $\mathbf{Q}$, alors le joueur 2 la joue. Sinon, le joueur 2 joue une instance X_0 -calculable de $\mathbf{Q}$. Notons que $\mathbf{Q}$ admet nécessairement une telle instance, car dans le cas contraire $\mathcal{I} = \{Z \in 2^{\mathbb{N}} : Z \leq_T X_0\}$ serait un idéal Turing satisfaisant $\mathbf{Q}$, donc satisfaisant $\mathbf{P}$, donc X_0 aurait une solution X_0 -calculable, et le joueur 2 aurait gagné au tour 1.

Supposons que le joueur 2 ne se retrouve jamais dans le cas où il trouve une solution de X_0 . Le jeu ne se termine pas, donc le joueur 1 gagne. Soit $\mathcal{I} = \{Z : \exists n \ Z \leq_T X_0 \oplus \cdots \oplus X_n\}$. Par l'exercice 22-5.11, $\mathcal{I}$ est un idéal Turing. De plus, $X_0 \in \mathcal{I}$, mais comme le joueur 2 n'a à aucun moment réussi à calculer une solution de X_0 , l'idéal $\mathcal{I}$ ne contient pas de solution de X_0 , donc $\mathcal{I} \not\models \mathbf{P}$. Enfin, pour toute $\mathbf{Q}$ -instance $Z \in \mathcal{I}$, soit n tel que $Z \leq_T X_0 \oplus \cdots \oplus X_n$, et soit alors e tel que $\Phi_e(X_0 \oplus \cdots \oplus X_n) = Z$. Par suite, $Y_{\langle n, e \rangle} = Z$, et $X_{\langle n, e \rangle}$ est une $\mathbf{Q}$ -solution de Z dans $\mathcal{I}$, donc $\mathcal{I} \models \mathbf{Q}$. Il s'ensuit que $\mathbf{P} \not\leq_\omega \mathbf{Q}$. Contradiction ! Le joueur 2 a donc une stratégie gagnante.

Supposons maintenant que $\mathbf{P} \not\leq_\omega \mathbf{Q}$. Définissons une stratégie gagnante pour le joueur 1. Soit $\mathcal{I}$ un idéal Turing tel que $\mathcal{I} \models \mathbf{Q}$, mais $\mathcal{I} \not\models \mathbf{P}$. Soit alors $X_0 \in \mathcal{I}$ une instance de $\mathbf{P}$ sans solution dans $\mathcal{I}$. La stratégie du joueur 1 consiste à maintenir le jeu dans $\mathcal{I}$ pour empêcher le joueur 2 de calculer une $\mathbf{P}$ -solution de X_0 . Au tour 1, le joueur 1 joue X_0 . Au tour $n > 1$, $X_0, \dots, X_{n-2}$ et $Y_0, \dots, Y_{n-1}$ ont déjà été joués, tous dans $\mathcal{I}$.

Comme $\mathcal{I} \models Q$, c'est que Y_{n-1} possède une solution X_{n-1} dans $\mathcal{I}$. Cette solution est jouée par le joueur 1. À tout instant de la construction, le joueur 2 ne possède que des oracles dans $\mathcal{I}$, et ne peut donc que jouer des instances de Q dans $\mathcal{I}$. Le jeu ne s'arrête jamais, sauf si le joueur 2 se retrouve bloqué sans instance $(X_0 \oplus \dots \oplus X_{n-1})$ -calculable de Q . Dans les deux cas, le joueur 1 gagne. ■

Le formalisme des jeux permet également de compter le nombre d'instances nécessaires pour une réduction.

Définition 4.3 (Hirschfeldt et Jockusch [88]). On note $P \leq_\omega^n Q$ si le joueur 2 possède une stratégie gagnante pour le jeu $G(Q \rightarrow P)$ qui lui garantit une victoire en au plus $n + 1$ tours. ◇

En particulier, $P \leq_\omega^0 Q$ si, et seulement si, P est calculatoirement vrai. Si $P \leq_c Q$, alors $P \leq_\omega^1 Q$. De plus, si Q admet des instances calculables, alors la réciproque est vraie.

Exercice 4.4. (★) Rappelons que KL est le problème du lemme de König pour les arbres à branchement fini, et que J est le problème du saut Turing. Montrer que $KL \leq_\omega^2 J$, mais qu'en revanche $KL \not\leq_\omega^1 J$. ◇

Il existe des variantes du jeu de réduction pour formaliser des généralisations de la réduction Weihrauch (voir Hirschfeldt et Jockusch [88]) ainsi que des variantes orientées théorie des modèles qui formalisent l'implication dans RCA_0 . Le lecteur intéressé par cette approche trouvera une analyse approfondie dans les travaux de Dzhamalov, Hirschfeldt et Reitzes [53].

5. Réductions fortes

Nous avons vu jusqu'à présent trois grandes réductions qui coexistent en calculabilité. L' ω -réduction est le pendant calculatoire de l'implication dans le système RCA_0 . La réduction calculatoire raffine cette réduction en n'autorisant qu'une application du problème, tandis que la réduction de Weihrauch étudie la dimension uniforme de la réduction. Pour chacune des réductions $P \leq Q$, la solution de l'instance $\tilde{X}$ de Q est autorisée à utiliser l'instance X de P en oracle pour calculer une solution de X . Il arrive cependant fréquemment que $\tilde{X}$ calcule directement une solution de X .

Définition 5.1. Un problème P est *fortement calculatoirement réductible* à Q (noté $P \leq_{sc} Q$) si pour toute instance X de P , il existe une instance $\tilde{X}$ de Q calculable en X , telle que toute Q -solution $\tilde{Y}$ de $\tilde{X}$ calcule une P -solution de X . ◇

De la même manière, on définit la *réduction forte de Weihrauch*, que l'on note $\leq_{sW}$. Les réductions fortes traduisent un lien plus étroit entre les solutions de P et de Q , dans le sens où l'on est capable de coder l'intégralité de l'information d'une P -solution dans une Q -solution.

Exemple 5.2. La plupart des preuves de réduction que nous avons vues sont fortes. Par exemple, $WKL \leq_{sW} J \leq_{sW} KL$. En revanche, la réduction $LLPO \leq_W LPO$, vue dans l'exercice 3.7, n'est pas forte.

Certains problèmes, comme le lemme faible de König, possèdent la capacité de coder des ensembles dans leurs solutions. On les appelle des cylindres.

Définition 5.3

Le *produit* de deux problèmes P et Q , est le problème $P \times Q$ dont le domaine est $\{X_0 \oplus X_1 : X_0 \in \text{dom } P \wedge X_1 \in \text{dom } Q\}$, et tel que pour toute instance $X_0 \oplus X_1$, les solutions sont de la forme $Y_0 \oplus Y_1$, où Y_0 est une P -solution de X_0 et Y_1 une Q -solution de X_1 . Un problème P est un *cylindre* pour $\leq_{sc}$ (resp. $\leq_{sW}$) si $\text{Id} \times P \leq_{sc} P$ (resp. $\text{Id} \times P \leq_{sW} P$). $\diamond$

Trivialement, si P est un cylindre pour $\leq_{sW}$, alors c'est un cylindre pour $\leq_{sc}$. La notion de cylindre capture exactement les problèmes qui sont insensibles à la réduction forte, au sens suivant.

Proposition 5.4. Un problème Q est un cylindre pour $\leq_{sc}$ si, et seulement si, pour tout problème P , on a l'équivalence $P \leq_c Q \leftrightarrow P \leq_{sc} Q$. $\star$

PREUVE. Notons que pour tout problème Q , si $P \leq_{sc} Q$, alors $P \leq_c Q$, car il suffit de ne pas tenir compte de l'instance X de P .

Supposons que Q est un cylindre pour $\leq_{sc}$ et que $P \leq_c Q$. Soit X une instance de P . Soit $\tilde{X}$ l'instance X -calculable de Q telle que pour toute Q -solution $\tilde{Y}$, l'ensemble $X \oplus \tilde{Y}$ calcule une P -solution de X . Comme Q est un cylindre, $\text{Id} \times Q \leq_{sc} Q$. Soit X_1 l'instance $(X \oplus \tilde{X})$ -calculable de Q telle que pour toute Q -solution Y_1 de X_1 , Y_1 calcule une $(\text{Id} \times Q)$ -solution de $X \oplus \tilde{X}$, autrement dit Y_1 calcule $X \oplus \tilde{Y}$, où $\tilde{Y}$ est une Q -solution de $\tilde{X}$. Comme $X \geq_T \tilde{X}$, X_1 est X -calculable, et comme $X \oplus \tilde{Y}$ calcule une P -solution de X , alors Y_1 calcule une P -solution de X . Donc, $P \leq_{sc} Q$.

Supposons maintenant que pour tout problème P , si l'on a $P \leq_c Q$, on a aussi $P \leq_{sc} Q$. En particulier, si l'on prend pour P le problème $\text{Id} \times Q$, on a bien $\text{Id} \times Q \leq_c Q$, donc $\text{Id} \times Q \leq_{sc} Q$. Autrement dit, Q est un cylindre pour $\leq_{sc}$. $\blacksquare$

Parmi les cylindres, on comptera le lemme faible de König, mais également le problème du saut J .

Proposition 5.5. Le problème WKL est un cylindre pour $\leq_{sW}$. ★

PREUVE. Soit Φ la fonctionnelle telle que pour tout ensemble X et tout arbre binaire T , $\Phi(X \oplus T)$ est un arbre binaire satisfaisant

$$[\Phi(X \oplus T)] = \{X \oplus P : P \in [T]\}.$$

Plus précisément, on peut définir

$$\Phi(X \oplus T) = \left\{ \sigma \in 2^{<\mathbb{N}} : \exists \tau \in T^{[|\sigma|]} \forall i < |\sigma| \begin{cases} i \text{ pair} \rightarrow \sigma(i) = \tau(i/2) \\ i \text{ impair} \rightarrow \sigma(i) = X((i-1)/2) \end{cases} \right\},$$

où $T^{[n]}$ est l'ensemble des nœuds de T de taille n . Alors, Φ et la fonctionnelle identité témoignent de la réduction $\text{Id} \times \text{WKL} \leq_{sW} \text{WKL}$. ■

Rappelons que par la proposition 2.5, si $P \leq_{\omega} \text{WKL}$, alors $P \leq_c \text{WKL}$. Grâce à la proposition 5.5 et la proposition 5.4, nous pouvons en déduire que $P \leq_{sc} \text{WKL}$.

Exercice 5.6. Montrer l'équivalent de la proposition 5.4 pour la réduction de Weihrauch. ◇

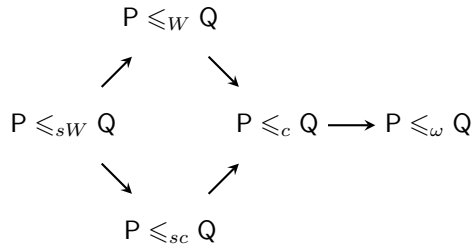


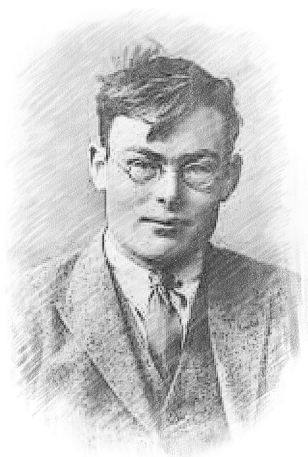
FIGURE 5.7 – *Diagramme récapitulatif des réductions calculatoires.*
Chacune des implications est stricte.

Chapitre 25

Théorème de Ramsey

De l'avis de ses professeurs et contemporains, Frank Ramsey était un esprit extrêmement brillant. Il est malheureusement mort trop jeune, à vingt-six ans, des suites d'une maladie. Malgré son jeune âge, il avait déjà beaucoup accompli, et son nom passera à la postérité notamment à travers deux travaux. Le premier en économie : il publie en 1928 un article sur le taux d'épargne, où il développe ce que l'on appellera le modèle de Ramsey. Un de ses professeurs, l'économiste mondialement célèbre John Maynard Keynes qualifiera ce travail « d'une des contributions les plus remarquables jamais apportées aux mathématiques économiques ». Le deuxième est en logique mathématique.

Ramsey s'intéresse à la philosophie, à la logique, aux probabilités et aux fondements des mathématiques. Il publiera en 1930 un article sur ce thème, dans lequel il étudie la cohérence de certaines formules logiques. Ce n'est toutefois pas l'objet initial de l'article que l'on retiendra, mais un simple lemme prouvé au détour d'une page, dont Erdős et Szekeres devaient prendre cinq ans plus tard toute la mesure. Ce lemme sera le point de départ de ce que l'on appellera la théorie de Ramsey, un champ de recherche important en mathématique combinatoire.



Frank Ramsey, 1903–1930

1. Aperçu général

La grande observation historique des mathématiques à rebours est son phénomène de structure : la plupart des théorèmes ordinaires sont soit prouvables dans RCA_0 , soit équivalents modulo RCA_0 à l'un des quatre grands systèmes d'axiomes WKL_0 , ACA_0 , ATR_0 , $\Pi^1_1\text{-CA}_0$, linéairement ordonnés par l'implication dans RCA_0 . Il s'agit d'un phénomène avant tout empirique, et qui porte sur les théorèmes « naturels » ; mais il est facile de concevoir des énoncés artificiels qui échappent à cette structure. Cette dernière relève-t-elle alors simplement d'un biais humain dans la pratique des mathématiques, ou bien en traduit-elle malgré tout quelque chose de plus profond ? Ce chapitre aidera peut-être le lecteur à se faire une opinion sur la question.

Le théorème de Ramsey pour les paires d'entiers a reçu une attention particulière, car c'est historiquement l'un des premiers théorèmes naturels échappant à cette classification. La communauté s'est donc penchée dans les moindres détails sur ses aspects méta-mathématiques, apportant peut-être ainsi quelques éléments de réponses à la question posée dans le paragraphe ci-dessus. Il s'agit d'une aventure mathématique passionnante à l'origine de la découverte de nombreuses techniques en calculabilité et en théorie de la preuve.

Commençons par fixer quelques notations et autres conventions, après quoi nous donnerons l'énoncé du théorème de Ramsey dans le cas général.

Notation

Soit $X \subseteq \mathbb{N}$. On écrit $[X]^n$ pour désigner l'ensemble des sous-ensembles de X de taille n .

Les ensembles non ordonnés de taille n sont en correspondance bijective avec les n -uplets $(a_1, \dots, a_n)$, où $a_i < a_{i+1}$. On identifiera donc parfois $[X]^n$ avec l'ensemble des n -uplets strictement croissants et l'on parlera par abus de langage de n -uplets pour désigner les éléments de $[X]^n$. Un *coloriage* est une fonction $f : [N]^n \rightarrow \mathbb{N}$. Si l'image de f est bornée, de cardinal au plus k , il s'agit d'un *k -coloriage* ou *coloriage fini*. Étant donné un coloriage f sur $[N]^n$, un ensemble $X \subseteq \mathbb{N}$ est *homogène* pour f si la couleur $f(\{x_1, \dots, x_n\})$ est la même pour tout $\{x_1, \dots, x_n\} \in [X]^n$.

On abrégera $f(\{x_1, \dots, x_n\})$ par $f(x_1, \dots, x_n)$ — où l'on supposera alors $x_1 < \dots < x_n$ — suivant notre identification de $[X]^n$ avec l'ensemble des n -uplets strictement croissants. Enfin, on identifiera l'entier k avec l'ensemble $\{0, 1, \dots, k-1\}$.

Théorème 1.1 (Théorème de Ramsey généralisé)
Soient $n, k \in \mathbb{N}^*$. Tout coloriage $f : [\mathbb{N}]^n \rightarrow k$ admet un ensemble homogène infini.

Le théorème établi à l'origine par Ramsey est en fait la restriction du théorème précédent au cas $n = 2$, c'est-à-dire aux paires d'entiers. À partir de $n > 1$, la preuve est non triviale et sera étudiée dans la section 2.

Notation

On désignera le théorème de Ramsey pour les n -uplets d'entiers et k couleurs par RT_k^n .

L'énoncé RT_k^n peut être vu comme le problème dont les instances sont des coloriages $f : [\mathbb{N}]^n \rightarrow k$, et dont les solutions sont les ensembles infinis homogènes. L'énoncé du théorème de Ramsey généralisé peut paraître assez abstrait à première vue. Arrêtons-nous sur sa signification pour des petites valeurs de n .

Cas $n = 1$. Il s'agit tout simplement du principe infini des tiroirs, qui dit que si l'on colorie un ensemble infini à l'aide d'un nombre fini de couleurs, alors une infinité d'éléments se voit attribuer la même couleur. Plus formellement, RT_k^1 énonce, pour toute k -partition $A_0 \sqcup \dots \sqcup A_{k-1} = \mathbb{N}$, l'existence d'un ensemble infini $H \subseteq A_i$ pour une couleur $i < k$.

0	1	3		8	...
	2		5	6	...
		4		7	...

FIGURE 1.2 – Instance de RT_3^1 où chaque ligne correspond à une couleur. Si la première ligne est infinie, $H = \{0, 1, 3, 8, \dots\}$ est une solution, mais également tout sous-ensemble infini de H en est une.

Le principe infini des tiroirs est combinatoirement trivial. Du point de vue calculatoire et de la théorie de la preuve, la situation est plus délicate. De la même manière que combinatoirement, le principe des tiroirs est le principe le plus fondamental à l'origine de la théorie de Ramsey, nous verrons que la plus grande partie des phénomènes calculatoires du théorème de Ramsey se retrouvent dans le principe des tiroirs.

Cas $n = 2$. Le théorème de Ramsey pour les paires est le théorème historique prouvé par Frank Ramsey. Il peut se formuler en termes de cliques comme suit : une clique est un graphe dont les sommets sont reliés deux à deux. Le théorème de Ramsey pour les paires et k couleurs (RT_k^2) énonce,

pour tout k -coloriage des arêtes d'une clique infinie, l'existence d'un sous-ensemble infini de sommets dont le sous-graphe induit est monochrome. Dans le cas où $k = 2$, on peut également considérer la présence et l'absence d'arête comme chacune des deux couleurs. RT_2^2 énonce donc, pour tout graphe infini, l'existence d'un sous-ensemble de sommets dont le sous-graphe induit est soit une clique, soit une anti-clique (sans aucune arête).

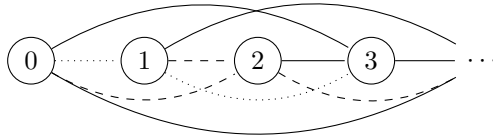


FIGURE 1.3 – Instance de RT_3^2 . Les motifs des arêtes correspondent aux couleurs. Une solution est un sous-ensemble infini de sommets dont les arêtes ont le même motif.

Le théorème de Ramsey pour les paires est combinatoirement beaucoup moins trivial que celui pour les singletons, et cette complexité se traduit du point de vue calculatoire. Nous verrons que RT_k^2 échappe au phénomène de structure calculatoire des mathématiques à rebours. La démonstration du théorème de Ramsey pour les paires est un exercice intéressant qui requiert une manipulation un peu poussée des ensembles infinis. Nous encourageons le lecteur à en chercher une preuve par lui-même.

Exercice 1.4. (★) Démontrer le théorème de Ramsey pour les paires. $\diamond$

Le théorème de Ramsey possède une version finie, qui découle de sa version infinie par compacité :

Théorème 1.5 (Théorème de Ramsey fini)

Pour tous $n, k, p \in \mathbb{N}^*$ il existe m suffisamment grand tel que tout k -coloriage de $[\{1, \dots, m\}]^n$ admet un ensemble homogène de taille p .

PREUVE. Supposons que le théorème de Ramsey fini soit faux. Alors, il existe n, k, p tels que pour tout m il existe un k -coloriage de $[\{1, \dots, m\}]^n$ sans ensemble homogène de taille p . On peut alors construire un arbre dont les nœuds de taille m sont les k -coloriages de $[\{1, \dots, m\}]^n$ n'admettant aucun ensemble homogène de taille supérieure à p . Comme il n'y a qu'un nombre fini de k -coloriages possibles de $[\{1, \dots, m\}]^n$, notre arbre est à branchement fini.

Par le lemme de König, l'arbre contient donc un chemin infini, qui est un coloriage de $[\mathbb{N}]^n$ sans aucun ensemble homogène de taille supérieure à p , ce qui contredit le théorème de Ramsey infini. ■

La borne optimale m en fonction de n, k et p est appelée *nombre de Ramsey*. La détermination des nombres de Ramsey en fonction des paramètres est encore un sujet de recherche en combinatoire.

Théorème de Ramsey et degrés Weihrauch

Il existe deux traductions possibles du théorème de Ramsey en problème mathématique.

Dans tous les cas, une instance est un coloriage $f : [\mathbb{N}]^n \rightarrow k$. En revanche, dans le premier cas, une solution est un ensemble H homogène pour l'instance f , tandis que dans le second cas, la solution est $i \cap H$, où H est homogène de couleur i pour l'instance f . On appellera ces problèmes respectivement RT_k^n et cRT_k^n . Cette distinction n'a pas d'importance pour la réduction Weihrauch, car en ayant accès à l'instance et l'ensemble homogène, il est possible de retrouver la couleur d'homogénéité. Ainsi, $\text{RT}_k^n \equiv_W \text{cRT}_k^n$ pour tous $n, k \geq 1$. En revanche, si l'on considère la réduction forte de Weihrauch où l'on perd l'accès à l'instance, la couleur d'homogénéité ne peut pas être récupérée en général, et l'on peut prouver que $\text{cRT}_k^n \not\leq_{sW} \text{RT}_k^n$ pour tous $n, k \geq 1$.

Le théorème de Ramsey du point de vue des degrés Weihrauch a été étudié de manière systématique par Brattka et Rakotoniaina [25].

2. Théorème de Ramsey dans la hiérarchie arithmétique

Notre aventure commence avec Specker [213], qui montre que RCA_0 ne suffit pas à démontrer le théorème de Ramsey pour les paires, en construisant un coloriage calculable qui n'admet pas d'ensemble homogène calculable infini. Un an plus tard, en 1972, Carl Jockusch [99] améliore ce résultat dans un article fondateur, qui marque le point de départ d'une étude approfondie du théorème de Ramsey, et c'est essentiellement ce premier travail de Jockusch que nous présentons dans cette section, résumée par la figure 2.1.

Comme illustré dans la figure, l'article de 1972 de Jockusch laissa ouvertes deux questions : la prouvabilité de ACA_0 ou celle de WKL_0 dans le système $\text{RCA}_0 + \text{RT}_2^2$. Il faudra attendre vingt-trois ans pour que David Seetapun ne résolve la première par la négative (voir le corollaire 4.15), et ce ne sera qu'en 2012 que la dernière sera résolue également par la négative, par Lu Liu (voir le corollaire 4.17), jeune étudiant alors encore en licence cette année-là.

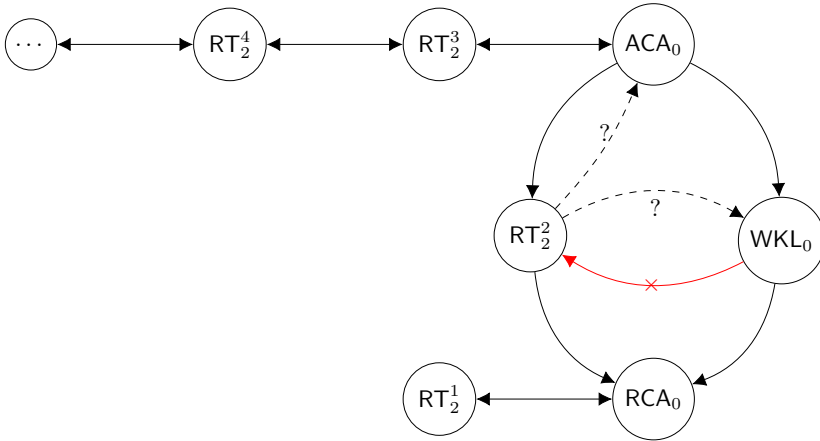


FIGURE 2.1 – Le travail de Jockusch. Les implications non représentées et qui ne peuvent être déduites des autres ne tiennent pas.

2.1. Étude préliminaire

Avant de nous lancer dans la preuve du théorème de Ramsey, nous allons établir deux propositions utiles pour le reste de notre analyse. La première proposition est un renforcement du théorème de Ramsey qui exprime d'une certaine manière que c'est un énoncé qui ne dépend pas de la nature de l'ensemble, mais uniquement de sa cardinalité.

Proposition 2.2. Le système $\text{RCA}_0 + \text{RT}_k^n$ prouve que pour tout ensemble infini $X \subseteq \mathbb{N}$ et tout k -coloriage f de $[X]^n$, il existe un ensemble $H \subseteq X$ infini homogène pour f . ★

PREUVE. Soit X un ensemble infini, et soit $f : [X]^n \rightarrow k$ fixés. Considérons alors $g : \mathbb{N} \rightarrow X$ la bijection canonique qui à m associe le $(m + 1)$ -ième élément de X . On peut alors définir un coloriage $h : [\mathbb{N}]^n \rightarrow k$ par $h(\{x_1, \dots, x_n\}) = f(\{g(x_1), \dots, g(x_n)\})$. Soit $Y \subseteq \mathbb{N}$ un ensemble infini homogène pour h . Alors, l'ensemble $H = \{g(x) : x \in Y\} \subseteq X$ est un ensemble infini homogène pour f . On vérifiera bien que sous l'hypothèse que X est infini, l'induction Σ_1^0 sur le paramètre X est suffisante pour montrer que g est totale, et donc que H est Δ_1^0 (voir la section 23-3). ■

La seconde proposition montre que le nombre de couleurs k du coloriage a peu d'impact en mathématiques à rebours. En effet, RT_{k+1}^n implique trivialement RT_k^n , car tout k -coloriage est un $(k + 1)$ -coloriage. Inversement, un argument de « daltonisme » permet de prouver RT_{k+1}^n à partir de RT_k^n .

Proposition 2.3. On a $\text{RCA}_0 \vdash \forall n \forall k \geq 2 (\text{RT}_k^n \rightarrow \text{RT}_{k+1}^n)$. ★

PREUVE. Supposons RT_k^n vrai. Considérons un coloriage $f : [\mathbb{N}]^n \rightarrow k + 1$. Définissons alors le k -coloriage $g : [\mathbb{N}]^n \rightarrow k$ qui à un n -uplet $\{x_1, \dots, x_n\}$ associe la couleur $f(\{x_1, \dots, x_n\})$ si $f(\{x_1, \dots, x_n\}) < k$, et la couleur $k - 1$ si $f(\{x_1, \dots, x_n\}) = k$. Par RT_k^n , il existe un ensemble X infini homogène pour g . Si la couleur de X est parmi $\{0, \dots, k - 2\}$, alors X est également homogène pour f . Supposons que X est homogène de couleur $k - 1$ pour g . Alors, f restreint à $[X]^n$ est un 2-coloriage. Par la version locale de RT_k^n (voir la proposition 2.2) appliquée à $f : [X]^n \rightarrow \{k - 1, k\}$, il existe un ensemble infini $H \subseteq X$ homogène pour f . ■

Notons que la preuve de la proposition 2.3 fait appel à deux instances de RT_k^n pour prouver RT_{k+1}^n . Il s'agit d'applications multiples nécessaires : on peut montrer $\text{RT}_{k+1}^n \not\leq_c \text{RT}_k^n$ pour tout $n, k \geq 1$ (voir Patey [172]). Dans la suite on s'autorisera à utiliser la version locale du théorème de Ramsey par la proposition 2.2, et l'on considérera principalement le cas $k = 2$ en vertu de la proposition 2.3.

Le théorème de Ramsey se prouve de manière inductive sur la taille des n -uplets coloriés. Le cas de base est le principe infini des tiroirs RT_2^1 .

Proposition 2.4. $\text{RCA}_0 \vdash \text{RT}_2^1$. ★

PREUVE. Soit $f : \mathbb{N} \rightarrow 2$ un coloriage, et soit $C_i = \{x \in \mathbb{N} : f(x) = i\}$. Notons que C_i existe pour tout $i < 2$ par Δ_0^0 -compréhension paramétrée par f . Si C_i est infini pour un $i < 2$, alors C_i est un ensemble infini homogène pour f . Sinon, soient n_0 et n_1 des bornes respectives de C_0 et C_1 . Alors, $\max(n_0, n_1)$ est une borne de $C_0 \cup C_1 = \mathbb{N}$, contradiction. ■

Il s'ensuit que RT_k^1 est prouvable dans RCA_0 pour tout $k \in \mathbb{N}$, et que RT_k^1 est un énoncé calculatoirement vrai, au sens où il a toujours une solution calculable en son instance. Nous étudierons plus en détail le principe infini des tiroirs — et ses richesses insoupçonnées — dans la section 3.

2.2. Théorème de Ramsey pour les paires

Nous rentrons à présent dans le vif du sujet en commençant par le théorème de Ramsey pour les paires, et en établissant des bornes très précises sur la complexité des solutions du point de vue de la hiérarchie arithmétique.

Théorème 2.5 (Jockusch [99])

Pour tout $k \geq 1$, toute instance f de RT_k^2 possède une solution $\Pi_2^0(f)$.

PREUVE. Explications préliminaires. Pour simplifier les notations, nous allons prouver le cas $k = 2$. Le cas général est une simple adaptation de la construction. Soit $f : [\mathbb{N}]^2 \rightarrow 2$ un coloriage. Le but est de construire une suite $\Pi_2^0(f)$ d'éléments $x_0 < x_1 < x_2 < \dots$ telle que pour tout x_i la couleur $f(\{x_i, x_j\})$ soit la même pour tout $x_j > x_i$. L'ensemble que l'on construit n'est donc pas directement un ensemble homogène. En revanche, on voit aisément comment calculer à partir d'une telle suite un sous-ensemble homogène : soit $g : \mathbb{N} \rightarrow 2$ la fonction qui à i associe $f(\{x_i, x_{i+1}\})$. Par le principe infini des tiroirs, il existe une couleur $c < 2$ qui apparaît infiniment souvent. Alors, l'ensemble $\{x_i : g(i) = c\}$ est un sous-ensemble $\Pi_2^0(f)$ homogène infini.

Chaque élément x_i sera approximé par une suite $(x_i^s)_{s \in \mathbb{N}}$. Au départ, chaque x_i^0 est indéfini. On va également utiliser une approximation d'ensembles $(X_i^s)_{s \in \mathbb{N}}$ qui sont eux aussi au départ tous indéfinis. Chaque ensemble X_i^s qui est défini sera au départ égal à $\{x \in X_{i-1}^s : f(x_i^s, x) = 0\}$ en supposant au début qu'il s'agit d'un ensemble infini. Puis, si ce n'est pas le cas, on le changera pour qu'il devienne égal à $\{x \in X_{i-1}^s : f(x_i^s, x) = 1\}$, et ainsi de suite. On utilisera également une borne inférieure b_s sur les nouveaux x_i^s que l'on souhaite définir. L'utilité de cette borne est essentiellement de pouvoir rendre notre construction $\Pi_2^0(f)$.

Construction. À l'étape 0, on définit $x_0^0 = 0$, $X_0^0 = \{x : f(x_0^0, x) = 0\}$ et $b_0 = 0$. Supposons qu'à une étape s , on ait défini $x_0^s < \dots < x_n^s$ pour un certain n (les x_{n+1}^s suivants étant indéfinis), avec des ensembles calculables $X_0^s \supseteq \dots \supseteq X_n^s$. À l'étape $s + 1$, on cherche à l'aide de $\emptyset'$ le plus grand $i \leq n$ tel que X_i^s possède au moins un élément strictement supérieur à b_s . Si un tel $i \geq 0$ n'existe pas, on pose $i = -1$.

Cas 1. Si $i = n$, alors on pose $X_j^{s+1} = X_j^s$ et $x_j^{s+1} = x_j^s$ pour $j \leq n$, et l'on définit x_{n+1}^{s+1} comme étant le plus petit élément de X_n^{s+1} strictement supérieur à b_s , et aussi $X_{n+1}^{s+1} = \{x \in X_n^{s+1} : f(x_{n+1}^{s+1}, x) = 0\}$. On définit enfin $b_{s+1} = x_{n+1}^{s+1}$.

Cas 2. Si $i < n$, alors pour tout $j \leq i$ on définit $X_j^{s+1} = X_j^s$ et $x_j^{s+1} = x_j^s$. Puis, on définit $x_{i+1}^{s+1} = x_{i+1}^s$ et $X_{i+1}^{s+1} = \{x \in X_i^{s+1} : f(x_{i+1}^{s+1}, x) = 1\}$. Pour tout $j > i + 1$, les valeurs x_j^{s+1} et X_j^{s+1} redeviennent indéfinies si elles ne l'étaient pas déjà. On définit enfin $b_{s+1} = b_s$. Cela conclut la construction.

Vérification. Montrons que chaque suite $(x_n^s)_{s \in \mathbb{N}}$ est finalement définie et converge vers une valeur x_n , ainsi que chaque suite $(X_n^s)_{s \in \mathbb{N}}$, qui converge vers un ensemble X_n tel que $|X_n| = \infty$. Il est clair que $(x_0^s)_{s \in \mathbb{N}}$ est toujours égale à 0. Si l'ensemble X_0^0 est infini, alors il est clair aussi qu'il ne changera jamais. S'il est fini, alors par construction il changera une seule fois pour

devenir co-fini, et ne changera donc plus jamais à partir de ce moment. Supposons à présent qu'à l'étape s les variables $x_0^s, \dots, x_n^s$ et $X_0^s, \dots, X_n^s$ aient atteint leurs valeurs de convergence $x_0, \dots, x_n$ et $X_0, \dots, X_n$ avec X_n infini. Là encore, par construction, la suite $(x_{n+1}^t)_{t>s}$ sera définie et constante à chaque étape t . Il s'ensuit que si $X_{n+1}^t \subseteq X_n$ est infini pour $t > s$, il ne changera jamais, sinon il changera une fois pour devenir co-fini.

Montrons enfin que la suite $x_0 < x_1 < \dots$ est $\Pi_2^0(f)$. On montre pour cela qu'elle est co-énumérée à l'aide de f' . Il suffit pour cela d'utiliser le marqueur b_s . En effet, à l'étape $s+1$, l'élément que l'on ajoute est forcément supérieur à b_s . Les éléments qui passent de indéfini à défini à l'étape s sont forcément supérieurs à b_s . La co-énumération se fait donc à chaque étape s en supprimant tous les éléments plus petits que b_s et qui sont différents d'un certain x_n^s pour x_n^s défini. ■

Notons que le théorème précédent constitue une preuve du théorème de Ramsey pour les paires. Le fait que la solution exhibée soit calculable à l'aide du double saut suggère que la preuve a été faite dans ACA_0 . Pour s'en assurer, il faut vérifier que la démonstration du fait que l'ensemble $\Pi_2^0(f)$ qui est construit est bien infini et homogène pour f , n'utilise pas plus que l'induction arithmétique. Le lecteur pourra constater que c'est bien le cas, et nous verrons à défaut que c'est une conséquence du théorème 2.17.

La borne supérieure $\Pi_2^0(f)$ du théorème précédent est-elle optimale du point de vue de la hiérarchie arithmétique ? Jockusch a répondu positivement, via le résultat suivant.

Théorème 2.6 (Jockusch [99])

Il existe une instance calculable de RT_2^2 qui ne possède aucune solution Σ_2^0 .

PREUVE. Montrons d'abord qu'il existe un coloriage calculable $f : [\mathbb{N}]^2 \rightarrow 2$ sans ensemble homogène infini Δ_2^0 .

Pour tous $e, s \in \mathbb{N}$, soit $A_{e,s} = \{x < s : \Phi_e(\emptyset'_s, x)[s] \downarrow = 1\}$. Notons que $(A_{e,s})_{e,s \in \mathbb{N}}$ est uniformément calculable. De plus, si $x \mapsto \Phi_e(\emptyset'_s, x)$ est une fonction totale, alors pour tout x , $\Phi_e(\emptyset'_s, x) = 1$ ssi $\forall^\infty s \ x \in A_{e,s}$. Notons que si $\Phi_e(\emptyset'_s, x) \uparrow$ pour x, e , alors x peut entrer et sortir de $A_{e,s}$ pour une infinité de s .

On définit $f : [\mathbb{N}]^2 \rightarrow 2$ de la manière suivante : à l'étape de calcul s , pour tout $e < s$, si jamais $|A_{e,s}| \geq 2(e+1)$, alors on choisit les deux plus petits éléments distincts $a, b \in A_{e,s}$ tels que $f(\{a, s\})$ et $f(\{b, s\})$ ne sont pas encore définis, et l'on définit $f(\{a, s\}) = 0$ et $f(\{b, s\}) = 1$.

Notons que si $|A_{e,s}| \geq 2(e+1)$, il y a forcément toujours deux éléments $a, b \in A_{e,s}$ sur lesquels $f(\{a, s\})$ et $f(\{b, s\})$ ne sont pas encore définis : par induction sur $e < s$, $f(\{x, s\})$ est défini pour au plus $2e$ valeurs de x avant de choisir les éléments de $A_{e,s}$. Ainsi, si $|A_{e,s}| \geq 2(e+1)$, il y a toujours au moins deux valeurs disponibles.

Montrons que f n'admet aucun ensemble homogène infini Δ_2^0 .

Soit A un ensemble Δ_2^0 infini. Alors, il existe un entier e tel que pour tout x , $x \in A$ si, et seulement si, $\forall^\infty s \ x \in A_{e,s}$. Soit $n \in \mathbb{N}$ le plus petit entier tel que $A \upharpoonright_n$ possède $2(e+1)$ éléments, et soit $s \in A$ suffisamment grand tel que $A_{e,s} \upharpoonright_n = A \upharpoonright_n$. En particulier, $|A_{e,s}| \geq 2(e+1)$, et l'on choisira deux éléments $a, b \in A \upharpoonright_n$ tels que $f(\{a, s\}) \neq f(\{b, s\})$. Comme $a, b, s \in A$, l'ensemble A n'est pas homogène pour f .

Pour finir, il suffit de montrer que tout ensemble Σ_2^0 infini contient un ensemble Δ_2^0 infini.

Étant donné $A = \{n : \exists x \forall y R(x, y, n)\}$ pour un prédicat calculable R , à l'aide de $\emptyset'$ on cherche n_0 et x tels que $\forall y R(x, y, n_0)$, puis on cherche $n_1 > n_0$ et x tels que $\forall y R(x, y, n_1)$, etc. Comme tout sous-ensemble homogène d'un ensemble homogène est lui aussi homogène, un coloriage qui n'a aucun ensemble homogène $\emptyset'$ -calculable n'a aussi aucun ensemble homogène Σ_2^0 . ■

En particulier, le simple saut n'est pas suffisant pour calculer un ensemble homogène infini pour un coloriage calculable des paires. Cela implique notamment que RT_2^2 n'est pas prouvable à partir du lemme faible de König.

Corollaire 2.7

On a $\text{RT}_2^2 \not\leq_\omega \text{WKL}$. En particulier, $\text{WKL}_0 \not\vdash \text{RT}_2^2$.

PREUVE. Par l'exercice 22-7.5, il existe un idéal de Scott $\mathcal{I}$ ne contenant que des ensembles low, et donc Δ_2^0 . Par la proposition 22-7.2, $\mathcal{I} \models \text{WKL}$. En revanche, par le théorème 2.6, il existe une instance calculable $f : [\mathbb{N}]^2 \rightarrow 2$ de RT_2^2 n'admettant aucune solution Δ_2^0 . En particulier, $f \in \mathcal{I}$, mais f n'admet pas de solution dans $\mathcal{I}$, donc $\mathcal{I} \not\models \text{RT}_2^2$. Il vient alors $\text{RT}_2^2 \not\leq_\omega \text{WKL}$, de sorte que $\text{WKL}_0 \not\vdash \text{RT}_2^2$. ■

2.3. Théorème de Ramsey généralisé

Nous allons maintenant étendre l'analyse précédente au théorème de Ramsey généralisé aux n -uplets. La preuve inductive du théorème de Ramsey se fait à l'aide de la notion d'ensemble pré-homogène, qui permet de réduire un coloriage des $(n+1)$ -uplets en un coloriage des n -uplets.

Définition 2.8. Soient $n, k \geq 1$, et soit $f : [\mathbb{N}]^{n+1} \rightarrow k$ un coloriage. Un ensemble $X \subseteq \mathbb{N}$ est *pré-homogène* pour la fonction f si la couleur d'un $(n+1)$ -uplet de X dépend uniquement des n premiers éléments du $(n+1)$ -uplet. Formellement, pour $x_1 < \dots < x_n \in X$ fixé, pour tout $y_1, y_2 \in X$ avec $y_1, y_2 > x_n$, on a

$$f(\{x_1, \dots, x_n, y_1\}) = f(\{x_1, \dots, x_n, y_2\}). \quad \diamond$$

Les ensembles pré-homogènes forment une passerelle entre les instances de RT_k^{n+1} et celles de RT_k^n dans le sens suivant.

Fait

Fixons $n, k \geq 1$, puis un coloriage $f : [\mathbb{N}]^{n+1} \rightarrow k$ et enfin un ensemble infini X pré-homogène pour f . Soit $g : [\mathbb{N}]^n \rightarrow k$ le coloriage qui à $\{x_1, \dots, x_n\} \in [X]^n$ associe $f(\{x_1, \dots, x_n, y\})$ pour n'importe quel $y \in X$ satisfaisant $y > x_n$. Alors, tout ensemble homogène $Y \subseteq X$ pour g est un ensemble homogène pour f .

Il s'agit donc de prouver, pour tout coloriage $f : [\mathbb{N}]^{n+1} \rightarrow k$, l'existence d'un ensemble infini pré-homogène pour f .

Proposition 2.9. Étant donné un coloriage $f : [\mathbb{N}]^{n+1} \rightarrow k$, il existe un arbre $T \subseteq \mathbb{N}^{<\mathbb{N}}$ f -calculable, infini à branchement fini, tel que tout chemin est un ensemble infini pré-homogène pour f . En particulier, tout degré PA relativement à f' calcule un ensemble infini pré-homogène pour f . ★

PREUVE. L'arbre que nous allons construire est connu sous le nom d'*arbre d'Erdős-Rado*. Soit T_0 l'arbre ne contenant que le nœud 0. Nous allons définir inductivement à chaque étape $s+1$ un arbre T_{s+1} qui contiendra les nœuds de T_s , auquel on rajoutera le nœud $\sigma \frown (s+1)$ pour un certain $\sigma \in T_s$.

Aux étapes $0 < s < n$, on met dans l'arbre T_s l'entier s comme fils du nœud $0 \frown 1 \frown 2 \frown \dots \frown s-1 \in T_{s-1}$. Soit T_s le résultat du calcul à une étape $s \geq n-1$. À l'étape $s+1$, nous allons parcourir l'arbre T_s depuis sa racine pour trouver le nœud $\sigma \in T_s$ auquel nous rajouterons $s+1$ comme fils de σ dans T_{s+1} . On définit $\sigma_0 = 0 \frown 1 \frown 2 \frown \dots \frown n-1 \in T_s$, puis à chaque étape i , en supposant σ_i définie, on cherche a tel que $\sigma_i a \in T_s$ et tel que $f(\sigma(m_1), \dots, \sigma(m_n), s+1) = f(\sigma(m_1), \dots, \sigma(m_n), a)$ pour tous entiers $m_1 < \dots < m_n < |\sigma_i|$. Si l'on trouve un tel entier a , on définit alors $\sigma_{i+1} = \sigma_i a$. Sinon, notre recherche s'arrête là, et l'on définit T_{s+1} comme étant $T_s \cup \{\sigma_i \frown (s+1)\}$.

L'arbre $T = \bigcup_s T_s$ est calculable, et infini car $|T_{s+1}| = |T_s| + 1$ pour tout s . Il est aussi à branchement fini, car pour tout nœud $\sigma a \in T$, il n'y a qu'un nombre fini de possibilités pour les combinaisons $f(\sigma(m_1), \dots, \sigma(m_n), a)$ de valeurs, pour tous $m_1 < \dots < m_n < |\sigma|$. Enfin, par construction, il est clair que tout chemin constitue un ensemble infini pré-homogène. ■

La proposition 2.9 permet de transférer la borne supérieure de Jockusch du théorème de Ramsey pour les paires vers le théorème généralisé de Ramsey.

Corollaire 2.10 (Jockusch [99])

Soient $n, k \geq 1$. Toute instance f de RT_k^n possède une solution $\Pi_n^0(f)$.

PREUVE. On procède par induction sur n . Pour $n = 1$, toute instance f -calculable de RT_k^1 admet une solution f -calculable, donc en particulier $\Pi_1^0(f)$. Pour $n = 2$, il s'agit du théorème 2.5. Supposons le théorème vrai pour n et montrons qu'il est vrai pour $n + 1$. Soit $f : [\mathbb{N}]^{n+1} \rightarrow k$ un coloriage. Par le théorème 8-4.3, il existe un ensemble P de degré PA relativement à f' tel que $P' \leq_T f''$. Par la proposition 2.9, P calcule un ensemble infini X pré-homogène pour f . Soit $g : [X]^n \rightarrow k$ le coloriage P -calculable qui à $\{x_1, \dots, x_n\}$ associe $f(\{x_1, \dots, x_n, y\})$ pour n'importe quel $y \in X$ tel que $y > x_n$. Par hypothèse d'induction relativisée à P , il existe un ensemble $\Pi_n^0(g)$ infini $Y \subseteq X$ homogène pour g , donc homogène pour f . En particulier, Y est $\Pi_{n-1}^0(g')$, donc $\Pi_{n-1}^0(f'')$, et donc $\Pi_{n+1}^0(f)$. ■

La notion d'ensemble pré-homogène donne la possibilité, via la proposition 2.9, de transformer un coloriage sur les $(n + 1)$ -uplets en un coloriage sur les n -uplets calculables en un degré PA relativement à $\emptyset'$. La proposition suivante est une sorte de réciproque partielle.

Proposition 2.11. Pour tout coloriage $\emptyset'$ -calculable $f : [\mathbb{N}]^n \rightarrow k$, il existe un coloriage calculable $g : [\mathbb{N}]^{n+1} \rightarrow k$ tel que tout ensemble infini homogène pour g est homogène pour f . ★

PREUVE. Soit $f : [\mathbb{N}]^n \rightarrow k$ un coloriage $\emptyset'$ -calculable. Par le lemme de limite de Shoenfield (voir le lemme 4-7.2), il existe une approximation Δ_2^0 de la fonction f . On peut voir cette approximation Δ_2^0 comme une fonction calculable $g : [\mathbb{N}]^{n+1} \rightarrow k$ dont le dernier paramètre correspond au temps d'approximation. Ainsi, pour $\{x_1, \dots, x_n\} \in [\mathbb{N}]^n$,

$$\lim_z g(\{x_1, \dots, x_n, z\}) = f(\{x_1, \dots, x_n\}).$$

Soit H un ensemble infini homogène pour g , de couleur $i < k$. Montrons que H est homogène pour f de couleur i . Soit $\{x_1, \dots, x_n\} \in [H]^n$. Soit $y \in H$ suffisamment grand pour que l'on ait

$$g(\{x_1, \dots, x_n, y\}) = \lim_z g(\{x_1, \dots, x_n, z\}).$$

En particulier, $\lim_z g(\{x_1, \dots, x_n, z\}) = i$, et ainsi $f(\{x_1, \dots, x_n\}) = i$. ■

Notons que la fonction g construite dans la proposition précédente peut posséder des ensembles homogènes finis qui ne sont pas homogènes pour f .

De même que la notion d'ensemble pré-homogène a permis d'obtenir une borne supérieure de la complexité du théorème de Ramsey généralisé, la proposition 2.11 permet de transférer la borne inférieure de Jockusch aux coloriage sur les n -uplets.

Corollaire 2.12 (Jockusch [99])

Pour tout $n \geq 2$, il existe une instance calculable de RT_2^n sans solution Σ_n^0 .

PREUVE. Montrons par induction sur $n \geq 2$ que pour tout Z , il existe une instance Z -calculable de RT_2^n qui ne possède aucune solution $\Sigma_n^0(Z)$. Le cas $n = 2$ est une relativisation du théorème 2.6. Supposons vrai le cas n et montrons le cas $n + 1$. Soit Z fixé. Par hypothèse d'induction appliquée à Z' , il existe une instance Z' -calculable f de RT_2^n qui ne possède aucune solution $\Sigma_n^0(Z')$, donc $\Sigma_{n+1}^0(Z)$. Par la proposition 2.11 relativisée, il existe une instance Z -calculable g de RT_2^{n+1} telle que tout ensemble infini homogène pour g est homogène pour f . En particulier, g ne possède pas de solution $\Sigma_{n+1}^0(Z)$. ■

La situation du théorème de Ramsey vis-à-vis de la hiérarchie arithmétique est donc la suivante.

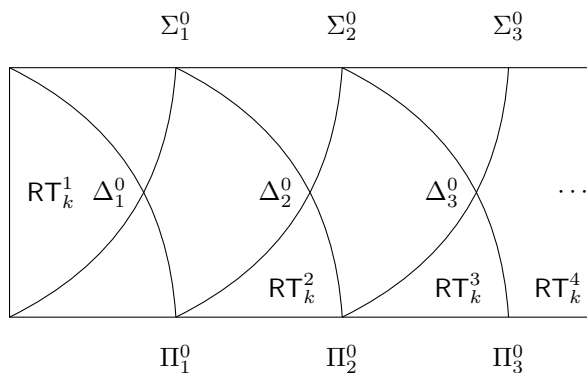


FIGURE 2.13 – Théorème de Ramsey dans la hiérarchie arithmétique

Les théorèmes de Jockusch cisèlent avec une précision d'horloger les contours de la complexité arithmétique des solutions possibles pour le théorème de Ramsey généralisé : pour tout n , toute instance calculable de RT_2^n a une solution Π_n^0 , et l'une de ces instances n'a pas de solution Σ_n^0 . Pour autant, ces théorèmes ne nous renseignent pas sur la « force calculatoire » du théorème de Ramsey : nous savons créer des instances calculables qui demandent beaucoup de puissance de calcul pour en trouver des solutions,

mais nous ne savons pas créer d'instances calculables dont toutes les solutions possibles ont beaucoup de puissance de calcul. C'est typiquement le genre de chose que nous devons faire pour montrer par exemple l'implication $\text{RCA}_0 \vdash \text{RT}_k^n \rightarrow \text{ACA}_0$.

Prenons l'exemple de RT_2^2 , dont les instances calculables admettent toujours des solutions Π_2^0 . Le double saut peut donc toujours calculer des solutions de RT_2^2 . Serait-il possible de construire une instance dont toutes les solutions calculent le double saut ? Jockusch a répondu à cette première question par la négative, en montrant qu'un degré $\text{PA}(\emptyset')$ était en fait suffisant.

Théorème 2.14 (Jockusch [99])

Pour tout $n, k \geq 1$ et tout coloriage $f : [\mathbb{N}]^n \rightarrow k$, tout degré PA relativement à $f^{(n-1)}$ calcule un ensemble infini homogène pour f .

PREUVE. Par induction sur $n \geq 1$. Pour $n = 1$, f calcule un ensemble infini homogène pour f . Tout degré PA relativement à f calcule f , donc calcule un ensemble infini homogène pour f .

Supposons que cela soit le cas pour $n \geq 1$. Soit $f : [\mathbb{N}]^{n+1} \rightarrow k$ un coloriage, et soit P_0 un ensemble $\text{PA}(f')$ tel que $P'_0 \leq_T f''$ et soit aussi P_1 un ensemble $\text{PA}(f^{(n)})$. Par la proposition 2.9, P_0 calcule un ensemble infini X pré-homogène pour f . Soit $g : [X]^n \rightarrow k$ le coloriage P_0 -calculable qui à $\{x_1, \dots, x_n\}$ associe $f(\{x_1, \dots, x_n, y\})$ pour n'importe quel $y \in X$ tel que $y > x_n$. En particulier, $P_0^{(n-1)} \leq_T f^{(n)}$, donc P_1 est $\text{PA}(P_0^{(n-1)})$ et donc $\text{PA}(g^{(n-1)})$. Par hypothèse d'induction, P_1 calcule un ensemble infini $H \subseteq X$ homogène pour g , donc homogène pour f . ■

En particulier, un degré $\text{PA}(\emptyset')$ permet de calculer une solution de toute instance calculable de RT_2^2 . À l'inverse, est-il possible de trouver une instance de RT_2^2 dont toutes les solutions sont de degré $\text{PA}(\emptyset')$? Jockusch n'a pu le montrer que pour les instances de RT_2^3 et non RT_2^2 . Nous verrons de fait dans la section 4 qu'il est impossible « d'encoder » de la puissance de calcul dans les solutions possibles d'une instance calculable de RT_2^2 . C'est en revanche le cas pour celles de RT_2^3 :

Exercice 2.15. (★) Construire une instance $\emptyset'$ -calculable de RT_2^2 dont toutes les solutions calculent $\emptyset'$. En déduire qu'il existe des instances calculables de RT_2^3 dont toutes les solutions calculent $\emptyset'$.

Indication.— Utiliser la notion de modulus (voir la définition 4-7.7). ◇

Il est possible de renforcer l'exercice précédent, et d'encoder dans les instances calculables de RT_2^3 , la puissance nécessaire au calcul d'ensembles pré-homogènes.

Théorème 2.16 (Hirschfeldt et Jockusch [88])

Il existe un coloriage calculable $f : [\mathbb{N}]^3 \rightarrow 2$ tel que tout ensemble infini pré-homogène pour f est de degré PA relativement à $\emptyset'$.

PREUVE

Soit $\Phi_0, \Phi_1, \dots$ une énumération de toutes les fonctionnelles Turing à valeurs dans $\{0, 1\}$. On peut supposer sans perte de généralité que $\Phi_e(\emptyset'_s, e)[s]$ renvoie une valeur pour n'importe quel temps de calcul s (en assignant par exemple 0 si $\Phi_e(\emptyset'_s, e)[s] \uparrow$). Soit $f : [\mathbb{N}]^3 \rightarrow 2$ défini pour tous $m < s < t$ par $f(\{m, s, t\}) = 1$ si pour tout $e < m$, $\Phi_e(\emptyset'_s, e)[s] = \Phi_e(\emptyset'_t, e)[t]$. Sinon, $f(\{m, s, t\}) = 0$. Soit X un ensemble infini pré-homogène pour f . Soit $g : \mathbb{N} \rightarrow 2$ la fonction qui, sur e , cherche un entier $m \in X$ tel que $e < m$, et des entiers $s, t \in X$ tels que $m < s < t$ et $\Phi_e(\emptyset'_s, e)[s] = \Phi_e(\emptyset'_t, e)[t]$, et renvoie $1 - \Phi_e(\emptyset'_s, e)[s]$. Notons qu'une telle recherche aboutit forcément, car il n'y a que deux valeurs possibles pour chaque Φ_e .

Par pré-homogénéité de X , $g(e) = 1 - \Phi_e(\emptyset'_t, e)[t]$ pour une infinité de $t \in X$. La fonction g est totale X -calculable. Montrons que g est une fonction DNC relativement à $\emptyset'$. Soit e tel que $\Phi_e(\emptyset', e) \downarrow$. Alors, $\lim_{t \in X} \Phi_e(\emptyset'_t, e)[t]$ existe, donc $g(e) = 1 - \Phi_e(\emptyset', e)$. Toute fonction DNC relativement à $\emptyset'$ à valeurs dans $\{0, 1\}$ est de degré PA relativement à $\emptyset'$. ■

Le théorème précédent se reformule par $\text{KL} \leq_c \text{RT}_2^3$ dans le langage des réductions calculatoires.

2.4. Mathématiques à rebours

L'analyse purement calculatoire effectuée par Jockusch [99] a été formalisée par Simpson [203] dans l'arithmétique du second ordre pour en déduire la puissance logique du théorème de Ramsey en mathématiques à rebours. Les preuves du théorème 2.17 sont essentiellement les constructions vues précédemment, mais analysées avec soin pour s'assurer que l'on n'utilise que les axiomes disponibles dans RCA_0 .

Théorème 2.17 (Jockusch [99] et Simpson [203])

Pour tout $n \geq 3$, on a l'équivalence $\text{RCA}_0 \vdash \forall k \text{RT}_k^n \leftrightarrow \text{ACA}_0$.

PREUVE. Montrons par induction sur $n \in \mathbb{N}$ que $\text{ACA}_0 \vdash \forall k \text{RT}_k^n$.

On a $\text{ACA}_0 \vdash \forall k \text{RT}_k^1$, comme nous le verrons dans la proposition 3.1. Supposons que $\text{ACA}_0 \vdash \text{RT}_k^n$. Soit $f : [\mathbb{N}]^{n+1} \rightarrow k$ une instance de RT_k^{n+1} . Soit $T \subseteq \mathbb{N}^{<\mathbb{N}}$ l'arbre d'Erdős-Rado de la proposition 2.9. Il est clair que la définition de T peut être faite de manière primitive récursive relativement à f . D'après le théorème 23-4.6, l'induction Σ_1^0 suffit pour montrer que la fonction qui sur n renvoie le nœud $\sigma \frown n \in T$ est totale, et

donc que l'arbre admet une définition Δ_1^0 : il existe donc par le schéma de compréhension Δ_1^0 . Par définition, il est infini, et RCA_0 suffit à prouver qu'il est infini à branchement fini. Par le lemme de König, prouvable dans ACA_0 , il existe un chemin $P \in [T]$, donc un ensemble infini pré-homogène pour f . Soit $g : [P]^n \rightarrow k$ le coloriage qui à $\{x_1, \dots, x_n\}$ associe $f(\{x_1, \dots, x_n, y\})$ pour n'importe quel $y \in X$ tel que $y > x_n$. Le coloriage g existe par le schéma de compréhension Δ_1^0 . Par hypothèse d'induction, et par la proposition 2.2, ACA_0 montre l'existence d'un ensemble infini $H \subseteq P$ homogène pour g . En particulier, H est homogène pour f .

Montrons que $\text{RCA}_0 + \text{RT}_2^3 \vdash \forall X \exists Y Y = X'$.

Soit X un ensemble, et soit $f : [\mathbb{N}]^3 \rightarrow 2$ la fonction définie pour $x < s < t$ par $f(\{x, s, t\}) = 1$ si, et seulement si,

$$\forall e < x ((\Phi_e(X, e)[s] \downarrow \wedge \Phi_e(X, e)[t] \downarrow) \vee (\Phi_e(X, e)[s] \uparrow \wedge \Phi_e(X, e)[t] \uparrow)).$$

Soit H un ensemble infini homogène pour f . Montrons que H est homogène de couleur 1. Soit $x \in H$, et soit $g : H \rightarrow 2^x$ définie par $g(s) = \langle a_e : e < x \rangle$ où $a_e = 1$ si $\Phi_e(X, e)[s] \downarrow$, et $a_e = 0$ sinon. RCA_0 prouve que g n'est pas injective, donc il existe $s < t \in H$ tels que $g(s) = g(t)$. En particulier, $f(\{x, s, t\}) = 1$. Soit $p_H : \mathbb{N} \rightarrow \mathbb{N}$ la fonction principale de H . Alors, l'ensemble $Y = \{e \in \mathbb{N} : \Phi_e(X, e)[p_H(e)] \downarrow\}$ existe par le schéma de compréhension Δ_1^0 , et $Y = X'$. ■

Notons que l'induction sur n dans la preuve du théorème 2.14 se fait à l'extérieur de ACA_0 . Il s'agit en effet d'une induction sur une formule du second ordre. Le corollaire 2.12, combiné au théorème de Jockusch et Solovay (voir le théorème 22-8.1), nous permet de montrer que l'énoncé $\forall n \text{RT}_2^n$ n'est pas prouvable dans ACA_0 .

Proposition 2.18. On a $\text{ACA}_0 \not\vdash \forall n \text{RT}_2^n$. ★

PREUVE. Supposons que $\text{ACA}_0 \vdash \forall n \text{RT}_2^n$. Par le théorème de Jockusch et Solovay (voir le théorème 22-8.1), il existe une borne $k \in \mathbb{N}$ telle que ACA_0 prouve que pour tout $n \in \mathbb{N}$, toute instance $f : [\mathbb{N}]^n \rightarrow 2$ admet une solution $\Sigma_k^0(f)$. En particulier, pour $n = k$, par le corollaire 2.12, il existe une instance calculable $f : [\mathbb{N}]^k \rightarrow 2$ sans solution Σ_k^0 . Contradiction ! ■

Comme expliqué dans la section 22-8.1, une telle séparation se fait nécessairement dans les modèles non standard puisque ACA'_0 et ACA_0 partagent les mêmes ω -modèles. En adaptant la preuve du théorème 2.14, McAloon [153] a prouvé que l'énoncé $\forall n \text{RT}_2^n$ était équivalent à ACA'_0 .

L'équivalence du théorème 2.17 n'est pas satisfaisante du point de vue de la calculabilité, dans la mesure où elle n'est pas sensible aux niveaux de la

hiérarchie arithmétique. Le corollaire 2.12 montre que plus la taille des n -uplets coloriés est grande, plus il faut de la puissance calculatoire pour trouver des solutions. La preuve dans RCA_0 de $\text{RT}_2^n \rightarrow \text{RT}_2^{n+1}$ lorsque $n \geq 3$ s'effectue en plusieurs étapes : une première instance de RT_2^n permet de calculer un ensemble infini pré-homogène pour l'instance de RT_2^{n+1} , ce qui la réduit en une seconde instance de RT_2^n . La réduction calculatoire, sensible au nombre d'applications, permet de rendre compte de cette différence.

Proposition 2.19. Pour tout $n \geq 1$, on a $\text{RT}_2^{n+1} \not\leq_c \text{RT}_2^n$. ★

PREUVE. Par le corollaire 2.12, il existe une instance f calculable de RT_2^{n+1} sans solution Σ_{n+1}^0 . Pour toute instance g de RT_2^n calculable en f , par le corollaire 2.10, g admet une solution $H \Pi_n^0(f)$, donc Π_n^0 . Tout ce que calcule H est Δ_{n+1}^0 , et donc aussi Σ_{n+1}^0 . Donc cette solution, H de g ne calcule (et donc ne f -calcule) pas de solution de f . ■

Du point de vue des mathématiques à rebours, le théorème de Ramsey pour les singletons est prouvable dans RCA_0 , et le théorème de Ramsey pour les n -uplets est équivalent à ACA_0 lorsque $n \geq 3$. Le cas du théorème de Ramsey pour les paires est plus intrigant. Par le théorème 2.17, RT_2^2 est prouvable dans ACA_0 , et par le corollaire 2.7 il n'est pas prouvable dans WKL_0 . Qu'en est-il des réciproques ? Une manière de prouver que le théorème de Ramsey pour les paires implique ACA_0 serait d'adapter la preuve du théorème 2.16 pour construire un coloriage calculable $f: [\mathbb{N}]^2 \rightarrow 2$ dont tous les ensembles homogènes calculent le problème de l'arrêt, et de vérifier que l'argument se formalise dans RCA_0 . David Seetapun [195] réussit à montrer en 1995 qu'une telle chose est impossible. En 2012, Lu Liu [146] montrera qu'il est également impossible de construire une instance calculable de RT_2^2 dont toutes les solutions sont de degré PA. Ces deux résultats majeurs passent par une bonne compréhension du principe infini des tiroirs.

3. Principe infini des tiroirs

La notion d'ensemble pré-homogène P permet de réduire une instance calculable de RT_k^{n+1} en une instance P -calculable de RT_k^n . En particulier, le théorème de Ramsey pour les paires est réduit à une instance non calculable du principe infini des tiroirs. Il est donc naturel d'étudier les instances non calculables de RT_k^1 .

Nous marquons donc une pause dans l'analyse calculatoire du théorème de Ramsey pour nous pencher sur son cas de base : le principe infini des tiroirs. Nous allons voir que ce principe, si élémentaire soit-il, présente tout de même quelques subtilités, et que l'analyse d'instances arbitraires de RT_k^1 est loin d'être triviale.

3.1. Nombre de couleurs et uniformité

Nous avons vu avec la proposition 2.3 et la proposition 2.4 que $\text{RCA}_0 \vdash \text{RT}_k^1$ pour tout entier standard k . En particulier, RT_k^1 est calculatoirement vrai, au sens où toute instance calcule sa propre solution.

En revanche, l'induction restreinte de RCA_0 n'est pas suffisante pour en déduire $\text{RCA}_0 \vdash \forall k \text{RT}_k^1$. Nous le montrons formellement avec la proposition suivante, via l'équivalence dans RCA_0 de $\forall k \text{RT}_k^1$ et du principe de collection pour les formules Π_1^0 , qui comme nous l'avons vu n'est pas prouvable dans RCA_0 .

Proposition 3.1 (Hirst [91]). On a l'équivalence

$$\text{RCA}_0 \vdash \forall k \text{RT}_k^1 \leftrightarrow \text{B}\Pi_1^0. \quad \star$$

PREUVE

$\text{B}\Pi_1^0 \rightarrow \forall k \text{RT}_k^1$. Soit $f : \mathbb{N} \rightarrow k$ une instance de $\forall k \text{RT}_k^1$. Soit $F(x, y)$ la formule $\Pi_1^0 \forall z > y \ f(z) \neq x$. S'il existe $x < k$ tel que pour tout y , $F(x, y)$ est faux, alors l'ensemble $\{z \in \mathbb{N} : f(z) = x\}$ est infini, et existe par le schéma de compréhension Δ_1^0 . Sinon, pour tout $x < k$, il existe y tel que $F(x, y)$ est vrai. Par $\text{B}\Pi_1^0$, il existe $p \in \mathbb{N}$ tel que pour tout $x < k$, il existe $y < p$ tel que $F(x, y)$ est vrai. Soit x la couleur de $f(p+1)$. Alors, $p+1$ contredit $F(x, p)$.

$\forall k \text{RT}_k^1 \rightarrow \text{B}\Pi_1^0$. Soit $F(x, y) = \forall z G(x, y, z)$ une formule Π_1^0 , et soit $k \in \mathbb{N}$ tels que $\forall x < k \ \exists y \ F(x, y)$. Soit $f : \mathbb{N} \rightarrow k$ définie par $f(t) = n$ pour le plus petit entier n tel que $\forall x < k \ \exists y < n \ \forall z < t \ G(x, y, z)$, si un tel n existe ; sinon, $f(t) = t$. S'il existe un ensemble infini H homogène pour f de couleur ℓ pour un entier $\ell \in \mathbb{N}$, alors $\forall x < k \ \exists y < \ell \ \forall z \ G(x, y, z)$.

S'il n'existe pas d'ensemble infini H homogène pour f , alors par $\forall k \text{RT}_k^1$, l'image de f est non bornée. Par compréhension Δ_1^0 , il est possible de construire une suite $(t_i)_{i \in \mathbb{N}}$ strictement croissante telle que $f(t_i) < f(t_{i+1})$ pour tout $i \in \mathbb{N}$. Soit $g : \mathbb{N} \rightarrow \mathbb{N}$ défini comme suit : $g(i)$ est le plus petit $x < k$ tel que $\forall y < f(t_i) - 1 \ \exists z < t_i \ \neg G(x, y, z)$. Soit S un ensemble infini homogène pour g , de couleur x_0 . Soit $y \in \mathbb{N}$. Comme S est infini, il existe $i \in S$ tel que $f(t_i) - 1 > y$, donc $\exists z < t_i \ \neg G(x_0, y, z)$. Ainsi, $\forall y \ \exists z \ \neg G(x_0, y, z)$, d'où contradiction. ■

Notons que la preuve de $\text{RCA}_0 \vdash \forall k \text{RT}_k^1 \rightarrow \text{B}\Pi_1^0$ fait intervenir deux instances de $\forall k \text{RT}_k^1$.

Nous voyons à présent que le théorème de Ramsey pour les paires est suffisant pour montrer $\forall k \text{RT}_k^1$ dans RCA_0 . Nous sommes ici dans un cas typique où un énoncé qui porte sur le second ordre a des conséquences sur le premier ordre. Cela vient du fait que les solutions d'une instance de RT_2^2 peuvent être utilisées comme paramètres dans le schéma d'induction Σ_1^0 .

Proposition 3.2. Pour tout $n \geq 1$, $\text{RCA}_0 \vdash \text{RT}_2^{n+1} \rightarrow \forall k \text{RT}_k^n$. En particulier, $\text{RCA}_0 \vdash \text{RT}_2^2 \rightarrow \text{B}\Pi_1^0$. ★

PREUVE. Soit $f : [\mathbb{N}]^n \rightarrow k$ une instance de RT_k^n pour un $k \in \mathbb{N}$, et soit $g : [\mathbb{N}]^{n+1} \rightarrow 2$ définie pour tout $F \in [\mathbb{N}]^{n+1}$ par $g(F) = 1$ ssi F est homogène pour f . Soit H un ensemble infini homogène pour g .

Montrons que G est homogène pour g de couleur 1. Par le théorème fini de Ramsey (prouvable dans RCA_0 , voir le théorème 1.10 dans Hájek et Pudlák [81]), il existe un sous-ensemble $A \in [H]^{n+1}$ homogène pour f . Donc, $g(A) = 1$.

Montrons que H est homogène pour f . Soient $A, B \in [H]^n$. Alors, il existe une suite finie d'ensembles $A_0, \dots, A_\ell \in [H]^{n+1}$ telle que $A \subseteq A_0, B \subseteq A_\ell$ et pour tout $i < \ell - 1$, $|A_i \cap A_{i+1}| \geq n$. Comme $A_0, \dots, A_\ell \in [H]^{n+1}$, alors ils sont tous f -homogènes, et comme $|A_i \cap A_{i+1}| \geq n$, alors A_i et A_{i+1} sont f -homogènes pour la même couleur. Il s'ensuit que $f(A) = f(B)$. Pour $n = 1$, $\text{RCA}_0 \vdash \text{RT}_2^2 \rightarrow \forall k \text{RT}_k^1$, donc $\text{RCA}_0 \vdash \text{RT}_2^2 \rightarrow \text{B}\Pi_1^0$ par 3.1. ■

La question de savoir si $\text{RCA}_0 + \text{RT}_2^2$ permet de montrer d'autres énoncés du premier ordre que $\text{B}\Pi_1^0$ est toujours ouverte.

Question 3.3. RT_2^2 est-il une extension conservative de $\text{RCA}_0 + \text{B}\Pi_1^0$ pour les formules arithmétiques? ★

Un progrès a été fait dans cette direction par Patey et Yokoyama, qui ont montré que $\text{RCA}_0 + \text{RT}_2^2$ ne pouvait montrer aucun énoncé Π_3^0 qui n'était pas déjà démontrable dans RCA_0 .

Théorème 3.4 (Patey, Yokoyama [173])

Le système RT_2^2 est une extension conservative de RCA_0 pour les formules Π_3^0 .

Le lecteur pourra vérifier qu'il n'y a pas de contradiction avec le fait que $\text{RCA}_0 + \text{RT}_2^2$ implique $\text{B}\Pi_1^0$.

Exercice 3.5. (★) Montrer qu'une instance de $\text{B}\Pi_1^0$ est un énoncé Σ_4^0 , et prouvablement équivalent à un énoncé Σ_3^0 dans RCA_0 . ◇

Recentrons-nous à présent sur les ω -modèles. La preuve de $\text{RCA}_0 \vdash \text{RT}_2^1$ procède par analyse de cas sur les couleurs : soit la couleur 0 apparaît infiniment souvent, soit non auquel cas la couleur 1 apparaît infiniment souvent. En particulier, le choix de la couleur n'est pas uniforme de manière calculable, car il résulte de la réponse à une question Π_2^0 . La réduction Weihrauch permet de traduire cette non-uniformité. Brattka et Rakotoniaina [25] et Hirschfeldt et Jockusch [88] ont prouvé de manière indépendante la proposition suivante.

Proposition 3.6 ([25],[88]). Pour tout $k \geq 1$, $\text{RT}_{k+1}^1 \not\leq_W \text{RT}_k^1$. ★

PREUVE. Supposons que $\text{RT}_{k+1}^1 \leq_W \text{RT}_k^1$ via des fonctionnelles Φ et Ψ . Soit $P(n)$ le prédicat « il existe une suite $\sigma_0 \prec \dots \prec \sigma_n \in (k+1)^{<\mathbb{N}}$ de segments initiaux de l'instance de RT_{k+1}^1 , une suite d'ensembles finis non vides $F_0, \dots, F_{n-1}$ de solutions partielles pour les instances partielles $\Phi(\sigma_0), \dots, \Phi(\sigma_{n-1})$ de RT_k^1 , de couleurs respectives et deux à deux distinctes $i_0, \dots, i_{n-1} \in \{0, \dots, k-1\}$, telles que pour tout $s < n$, il existe x pour lequel $\sigma_{s+1}(x) = s$ et $\Psi(\sigma_{s+1} \oplus F_s, x) \downarrow = 1$ ». On a $P(0)$ avec la chaîne $\sigma_0 = \epsilon$. Montrons que l'on a $P(n) \rightarrow P(n+1)$ pour tout $n \leq k$, ce qui implique une contradiction pour $P(k+1)$, car on a alors une suite de couleurs deux à deux différentes $i_0, \dots, i_k \in \{0, \dots, k-1\}$.

Supposons que $P(n)$ est vérifié avec $\sigma_0, \dots, \sigma_n, F_0, \dots, F_{n-1}$ et $i_0, \dots, i_{n-1}$. Soit H une solution pour l'instance $\Phi(\sigma_n \frown n^\infty)$ de RT_k^1 , et soit i sa couleur. On peut supposer que $\min H > |\sigma_n|$.

Si $i = i_s$ pour un $s < n$, alors $F_s \cup H$ est toujours homogène. En particulier, $\Psi((\sigma_n \frown n^\infty) \oplus (F_s \cup H))$ est infini, si bien qu'il doit être homogène pour l'ensemble $\sigma_n \frown n^\infty$ de couleur n . Cependant, par hypothèse d'induction, il existe x tel que $\sigma_{s+1}(x) = s$ et $\Psi(\sigma_{s+1} \oplus F_s, x) \downarrow = 1$, donc $\Psi((\sigma_n \frown n^\infty) \oplus (F_s \cup H))$ contient à la fois des éléments de couleur n et s , contradiction. Donc, i est différent de i_s pour tout $s < n$.

Soit $i_n = i$. Soit F_n un ensemble fini non vide homogène pour $\Phi(\sigma_n \frown n^\infty)$ de couleur i , et soit $x \geq |\sigma_n|$ tel que $\Psi((\sigma_n \frown n^\infty) \oplus F_n, x) \downarrow = 1$. Soit enfin σ_{n+1} un segment initial de $\sigma_n \frown n^\infty$ suffisamment long pour que F_n soit solution partielle de $\Phi(\sigma_{n+1})$ et pour que $\Psi(\sigma_{n+1} \oplus F_n, x) \downarrow = 1$. On a bien $P(n+1)$. ■

Exercice 3.7. Vérifier que la preuve de la proposition 3.2 est aussi une preuve de $\text{RT}_k^n \leq_W \text{RT}_2^{n+1}$ pour tous $n, k \geq 1$. ◇

3.2. Principe infini des tiroirs et évitement de cônes

Nous nous attaquons à présent aux instances non calculables de RT_2^1 . Le but de cette section est de démontrer le théorème suivant.

Théorème 3.8 (Dzhafarov et Jockusch [54])

Pour tout ensemble A et tout ensemble C non calculable, il existe un ensemble infini $H \subseteq A$ ou $H \subseteq \mathbb{N} \setminus A$ tel que $C \not\leq_T H$.

Nous verrons que ce théorème constitue une étape clef pour montrer que RT_2^2 préserve la propriété de faiblesse donnée par $\{X : X \not\leq_T C\}$ pour tout C non calculable (voir la définition 24-1.6). Via les développements de la section 24-1.1, cela nous permettra en particulier de construire un ω -modèle de $\text{RCA}_0 + \text{RT}_2^2$ ne contenant pas $\emptyset'$, et donc d'établir le théorème 4.14 de Seetapun à venir, qui sépare RT_2^2 de ACA_0 . Notons que nous ne suivons pas la preuve originale de Seetapun, qui est plus directe, mais aussi moins informative, notamment sur la faiblesse combinatoire des instances arbitraires du problème RT_2^1 .

Nous invitons le lecteur à réfléchir à la signification du théorème ci-dessus, en commençant peut-être par le mettre en parallèle avec quelque chose que nous avons déjà utilisé dans la preuve du lemme 14-4.6, et que nous donnons ci-après en exercice.

Exercice 3.9. (★) Soit C un ensemble quelconque. Montrer qu'il existe un ensemble infini A dont tous les sous-ensembles infinis permettent de calculer C . $\diamond$

La preuve du théorème 3.8 se fait via une notion de forcing définie par Dzhafarov et Jockusch, qui consiste en une variante effective du *forcing de Mathias*, utilisé au départ en théorie des ensembles pour construire des modèles de ZFC dans lesquels pour n'importe quelle classe $\mathcal{C} \subseteq 2^{\mathbb{N}}$, il existe un ensemble infini dont tous les sous-ensembles infinis sont dans $\mathcal{C}$ ou dans $2^{\mathbb{N}} \setminus \mathcal{C}$ ([151] et [152], voir [106]). Commençons par définir quelques notations qui en simplifieront la présentation.

Notation

On considérera les chaînes $\sigma \in 2^{<\mathbb{N}}$ comme des ensemble finis d'entiers via les notations suivantes.

1. La chaîne $\sigma \cup \tau$ est la chaîne de taille $\max(|\sigma|, |\tau|)$ vérifiant que $\sigma \cup \tau(n) = 1$ ssi $\sigma(n) = 1$ ou $\tau(n) = 1$.
2. La chaîne $\tau - \sigma$ est la chaîne de taille $|\tau|$ à laquelle on enlève les 1 de σ . On a donc :
 $(\tau - \sigma)(n) = 1$ ssi $\tau(n) = 1$ et $\sigma(n) = 0$, pour $n < \min(|\sigma|, |\tau|)$,
et $(\tau - \sigma)(n) = 1$ ssi $\tau(n) = 1$, pour n tel que $|\tau| > n \geq \min(|\sigma|, |\tau|)$.
3. L'ensemble $X - \sigma$ est l'ensemble X auquel on enlève les $n < |\sigma|$ tels que $\sigma(n) = 1$.
4. Le prédicat $\sigma \subseteq X$ signifie $\sigma(n) = 1$ implique $X(n) = 1$.

Le forcing de Mathias, tel qu'utilisé en combinatoire, est défini comme suit.

Définition 3.10. Une *condition de Mathias* est un couple (σ, X) tel que

1. $\sigma \in 2^{<\mathbb{N}}$;
2. X est un sous-ensemble infini de $\mathbb{N}$;
3. $X \cap \{0, \dots, |\sigma|\} = \emptyset$.

On dira pour deux conditions de Mathias (τ, Y) et (σ, X) que (τ, Y) *étend* (σ, X) , et l'on écrira $(\tau, Y) \leq (\sigma, X)$ si $\tau \succeq \sigma$, $Y \subseteq X$ et $\tau - \sigma \subseteq X$. L'ordre partiel des conditions de Mathias peut être enrichi en un forcing de Cantor en interprétant tout filtre maximal F par $\dot{F}$ comme étant l'unique élément de $\bigcap_{(\sigma, X) \in F} [\sigma]$. $\diamond$

Étant donné une condition de Mathias (σ, X) , on note $[\sigma, X]_\infty$ l'ensemble des filtres maximaux contenant (σ, X) , et $[\sigma, X] = \{\dot{F} : F \in [\sigma, X]_\infty\}$. On a alors

$$[\sigma, X] = \{G \in \mathcal{P}(\mathbb{N}) : \sigma \prec G \text{ et } G - \sigma \subseteq X\}.$$

Ainsi, une condition de Mathias (σ, X) peut être considérée comme un raffinement du forcing de Cohen. En effet, la partie finie σ est un segment initial de l'élément construit, ce qui permet de fixer des propriétés Σ_1^0 . On y ajoute un réservoir infini X des valeurs que l'on autorise à ajouter au segment initial. Le réservoir ne contient que de l'information négative, au sens où si $G \in [\sigma, X]$ et $n \in X$, alors il n'est pas garanti que $n \in G$. En revanche, si $n \notin X$, alors $n \notin G$.

Définition 3.11. Soit (σ, X) une condition de Mathias et soit Φ_e une fonctionnelle Turing.

1. $(\sigma, X) \Vdash \Phi_e(G, n) \downarrow = m$ si $\Phi_e(\sigma, n) \downarrow = m$.
2. $(\sigma, X) \Vdash \Phi_e(G, n) \uparrow$ si $\forall \tau \subseteq X \ \Phi_e(\sigma \cup \tau, n) \uparrow$. $\diamond$

Nous sommes maintenant prêts à définir la notion de forcing de Dzhafarov-Jockusch, qui présente une certaine difficulté conceptuelle supplémentaire par rapport à un forcing classique : étant donné A quelconque et C non calculable, on ne saura pas à l'avance si l'ensemble générique $G \not\leq_T C$ que l'on cherche à construire sera un sous-ensemble infini de A ou de $\mathbb{N} \setminus A$. On va donc construire en parallèle deux génériques G^0, G^1 , l'un d'eux étant inclus dans A et l'autre dans $\mathbb{N} \setminus A$.

Définition 3.12. Soit $A^0 \sqcup A^1$ une 2-partition de $\mathbb{N}$ et soit $\mathcal{I}$ un idéal de Scott. Une *condition de Dzhafarov-Jockusch* pour $A^0 \sqcup A^1$ et $\mathcal{I}$ est un triplet (σ_0, σ_1, X) tel que

- (1) (σ_i, X) est une condition de Mathias pour tout $i < 2$;
- (2) $\sigma_i \subseteq A^i$ pour tout $i < 2$;

(3) $X \in \mathcal{I}$.

Une condition (τ_0, τ_1, Y) *étend* (σ_0, σ_1, X) si (τ_i, Y) étend (σ_i, X) au sens de Mathias pour chaque $i < 2$. $\diamond$

Une condition de Dzhafarov-Jockusch dépendra par conséquent d'une partition $A^0 \sqcup A^1$ et d'un idéal de Scott $\mathcal{I}$.

Notation

Dans la suite de la preuve, on représentera les 2-partitions selon le contexte par $A, \mathbb{N} \setminus A$ pour un ensemble A , ou bien sous la forme de deux ensembles A^0, A^1 , où A^0 est une notation pour A et A^1 une notation pour $\mathbb{N} \setminus A$.

Pour construire un générique $G \not\leq_T C$ pour un certain ensemble C non calculable fixé, on utilisera un idéal de Scott ne contenant pas C .

Un filtre maximal F s'interprète en deux ensembles $\dot{F} = (G^0, G^1)$ définis pour tout $i < 2$ par $\{G^i\} = \bigcap_{(\sigma_0, \sigma_1, X) \in F} [\sigma_i]$. Pour une condition (σ_0, σ_1, X) , on définit

$$[\sigma_0, \sigma_1, X] = \{(G^0, G^1) : \forall i < 2 \ \sigma_i \preceq G^i \wedge G^i - \sigma_i \subseteq X\}.$$

Ainsi, pour tout filtre maximal F , on a

$$\{\dot{F}\} = \bigcap_{(\sigma_0, \sigma_1, X) \in F} [\sigma_0, \sigma_1, X].$$

Relation de forcing

Notons ici une subtilité : pour une notion de forcing de Cantor $(\mathbb{P}, \leq)$ et pour $c \in \mathbb{P}$, nous avons défini $[c]_\leq$ comme la classe des filtres maximaux contenant c , et $[c] = \{\dot{F} : F \in [c]\}$. La définition de $[c]$ pour le forcing de Dzhafarov-Jockusch est différente. En effet, pour tout filtre maximal F , on a $\dot{F} \in [\sigma_0, \sigma_1, X]$ mais l'inverse n'est pas vrai, et il existe des paires $(G^0, G^1) \in [\sigma_0, \sigma_1, X]$ telles que $G^i \not\subseteq A^i$, tandis que c'est nécessairement le cas pour $\dot{F}$, par (2) de la définition d'une condition.

Nous imposerons également l'hypothèse suivante qu'il faudra justifier pour toute application du forcing de Dzhafarov-Jockusch.

Pour tout $i < 2$, il n'existe pas d'ensemble infini $H \subseteq A^i$ dans $\mathcal{I}$. (H1)

L'idée est que les éléments de $\mathcal{I}$ vont satisfaire la propriété que l'on veut montrer, typiquement ne pas calculer un ensemble C fixé. Il s'ensuit que si un ensemble infini $H \subseteq A^i$ est dans $\mathcal{I}$, alors H est la solution désirée, et il n'y a plus besoin de faire de construction. La supposition (H1) permet de prouver que les deux ensembles produits par un filtre suffisamment

générique pour le forcing de Dzhafarov-Jockusch sont infinis, comme le montre le lemme suivant.

Lemme 3.13. Supposons (H1). Soit $c = (\sigma_0, \sigma_1, X)$ une condition et soit $i < 2$. Il existe une extension (τ_0, τ_1, Y) de c et un entier $n > |\sigma_i|$ tels que $n \in \tau_i$. ★

PREUVE. Si $X \cap A^i$ est vide, alors $X \subseteq A^{1-i}$; or, $X \in \mathcal{I}$, ce qui contredit (H1). Ainsi, il existe $n \in X \cap A^i$. Soient $\tau_i = \sigma_i \cup \{n\}$ et $\tau_{1-i} = \sigma_{1-i}$. Alors, $(\tau_0, \tau_1, X \setminus \{0, \dots, n-1\})$ est une extension de (σ_0, σ_1, X) telle que $n \in \tau_i$. ■

Nous avons vu dans la section 11-4 que les propriétés calculatoires d'un ensemble produit par un filtre suffisamment générique étaient très liées à l'existence d'une *question de forcing* avec de bonnes propriétés définitionnelles. Rappelons qu'une question de forcing pour un forcing de Cantor $\mathbb{P}$ est une relation $c ? \vdash \mathcal{R}$, où $c \in \mathbb{P}$ et $\mathcal{R}$ est un contrat, telle que si $c ? \vdash \mathcal{R}$ est vraie, alors il existe une extension $d \leq c$ pour laquelle $d \Vdash \mathcal{R}$, et sinon, il existe une extension $d \leq c$ pour laquelle $d \Vdash \neg \mathcal{R}$. La situation du forcing de Dzhafarov-Jockusch est un peu plus complexe, car deux ensembles distincts sont créés parallèlement. Il serait naturel de définir une question de forcing pour chaque ensemble construit :

$$(\sigma_0, \sigma_1, X) ? \vdash^i \exists n \Phi_e(G, n) \downarrow$$

s'il existe une chaîne $\rho \subseteq X \cap A^i$ et un entier $n \in \mathbb{N}$ tels que $\Phi_e(\sigma_i \cup \rho, n) \downarrow$. Ainsi, si $(\sigma_0, \sigma_1, X) ? \vdash^i \exists n \Phi_e(G, n) \downarrow$, il existe une extension (τ_0, τ_1, Y) telle que $(\tau_i, Y \cap A^i) \Vdash \exists n \Phi_e(G, n) \downarrow$, et il existe sinon une extension (τ_0, τ_1, Y) telle que $(\tau_i, Y \cap A^i) \Vdash \forall n \Phi_e(G, n) \uparrow$. Cependant, cette définition de question de forcing ne satisfait pas les bonnes propriétés définitionnelles, car elle est $\Sigma_1^0(X \oplus A)$, or A peut être arbitrairement complexe.

Nous allons donc définir une variante de la question de forcing pour les contrats Σ_1^0 et Π_1^0 qui s'émancipe de A , au prix d'une disjonction.

Définition 3.14. Soit $c = (\sigma_0, \sigma_1, X)$ une condition et soient $e_0, e_1 \in \mathbb{N}$.

$$c ? \vdash \exists n \Phi_{e_0}(G^0, n) \downarrow \vee \exists n \Phi_{e_1}(G^1, n) \downarrow$$

si pour toute 2-partition $Z^0 \sqcup Z^1 = \mathbb{N}$, il existe un entier $n \in \mathbb{N}$, un $i < 2$ et enfin une chaîne $\rho \subseteq Z^i \cap X$ tels que $\Phi_{e_i}(\sigma_i \cup \rho, n) \downarrow$. ◆

À première vue, nous nous sommes abstraits de l'ensemble A en le remplaçant par une formule autrement plus compliquée qui quantifie universellement sur les 2-partitions. Cependant, par un argument de compacité,

la relation de la définition 3.14 se simplifie comme par magie en une formule $\Sigma_1^0(X)$.

Lemme 3.15. Soit (σ_0, σ_1, X) une condition et soient $e_0, e_1 \in \mathbb{N}$. Alors, la relation

$$(\sigma_0, \sigma_1, X) \text{ ?} \vdash \exists n \Phi_{e_0}(G^0, n) \downarrow \vee \exists n \Phi_{e_1}(G^1, n) \downarrow$$

est $\Sigma_1^0(X)$ uniformément en e_0 et e_1 . ★

PREUVE. Soit $\mathcal{C}$ la classe des ensembles $Z^0 \oplus Z^1$ tels que $Z^0 \sqcup Z^1 = \mathbb{N}$ et tels que pour tout $i < 2$, tout $\rho \subseteq Z^i \cap X$ et tout $n \in \mathbb{N}$, $\Phi_{e_i}(\sigma_i \cup \rho, n) \uparrow$. Alors, $(\sigma_0, \sigma_1, X) \text{ ?} \vdash \exists n \Phi_{e_0}(G, n) \downarrow \vee \exists n \Phi_{e_1}(G, n) \downarrow$ ssi $\mathcal{C} = \emptyset$. Comme $\mathcal{C}$ est une classe $\Pi_1^0(X)$, la relation $\mathcal{C} = \emptyset$ est $\Sigma_1^0(X)$. ■

La définition 3.14 peut être faite de manière explicitement $\Sigma_1^0(X)$ comme suit.

Définition 3.16 (Définition alternative)

Soit $c = (\sigma_0, \sigma_1, X)$ une condition et soient $e_0, e_1 \in \mathbb{N}$.

$$c \text{ ?} \vdash \exists n \Phi_{e_0}(G^0, n) \downarrow \vee \exists n \Phi_{e_1}(G^1, n) \downarrow$$

s'il existe $r \in \mathbb{N}$ tel que pour toute 2-partition $Z^0 \sqcup Z^1 = \{0, \dots, r-1\}$, il existe $i < 2$, une chaîne $\rho \subseteq Z^i \cap X$ et un entier $n \in \mathbb{N}$ tels que $\Phi_{e_i}(\sigma_i \cup \rho, n) \downarrow$. ◇

Arrêtons-nous sur la signification de la définition 3.14. Cette définition-là évite de mentionner l'ensemble A en faisant une « sur-approximation ». La question de forcing demande si pour *toutes* les instances de RT_2^1 , il est possible de trouver une extension finie qui force d'un côté le contrat Σ_1^0 . Il est donc clair que si la réponse est oui, alors en particulier pour $Z^i = A^i$, il existe une extension (τ_0, τ_1, Y) telle que $(\tau_i, Y) \vdash \exists n \Phi_{e_i}(G, n)$ pour un certain $i < 2$. En revanche, si la réponse à la question de forcing est non, alors cet échec peut être dû à une instance $Z^0 \sqcup Z^1 = \mathbb{N}$ de RT_2^1 qui n'a rien à voir avec l'instance originelle $A^0 \sqcup A^1 = \mathbb{N}$. C'est là que la nature combinatoire de la théorie de Ramsey entre en jeu : il suffit de construire un ensemble qui est à la fois solution de l'instance $Z^0 \sqcup Z^1 = \mathbb{N}$ et de l'instance $A^0 \sqcup A^1 = \mathbb{N}$ pour forcer la propriété.

Proposition 3.17. Soit $c = (\sigma_0, \sigma_1, X)$ une condition et soient $e_0, e_1 \in \mathbb{N}$.

- (1) Si $c \text{ ?} \vdash \exists n \Phi_{e_0}(G^0, n) \downarrow \vee \exists n \Phi_{e_1}(G^1, n) \downarrow$, alors il existe un entier $i < 2$ et une extension (τ_0, τ_1, Y) tels que $(\tau_i, Y) \vdash \exists n \Phi_{e_i}(G, n) \downarrow$.
- (2) Si $c \text{ ?} \not\vdash \exists n \Phi_{e_0}(G^0, n) \downarrow \vee \exists n \Phi_{e_1}(G^1, n) \downarrow$, alors il existe un entier $i < 2$ et une extension (τ_0, τ_1, Y) tels que $(\tau_i, Y) \vdash \forall n \Phi_{e_i}(G, n) \uparrow$. ★

PREUVE. (1) Si $c \text{ ?} \vdash \exists n \Phi_{e_0}(G^0, n) \downarrow \vee \exists n \Phi_{e_1}(G^1, n) \downarrow$, alors en particulier, pour $Z^i = A^i$, il existe $i < 2$, une chaîne $\rho \subseteq A^i \cap X$ et un entier $n \in \mathbb{N}$

tels que $\Phi_{e_i}(\sigma_i \cup \rho, n) \downarrow$. Supposons $i = 0$, l'autre cas étant symétrique. La condition $(\sigma_0 \cup \rho, \sigma_1, X \setminus \{0, \dots, |\rho|\})$ est une extension de c telle que $(\sigma_0 \cup \rho, X \setminus \{0, \dots, |\rho|\}) \Vdash \Phi_{e_0}(G, n) \downarrow$.

(2) Cas où $c \not\vdash \exists n \Phi_{e_0}(G^0, n) \downarrow \vee \exists n \Phi_{e_1}(G^1, n) \downarrow$. Soit alors $\mathcal{C}$ la classe des ensembles $Z^0 \oplus Z^1$ tels que $Z^0 \sqcup Z^1 = \mathbb{N}$ et tels que pour tout $i < 2$, tout $\rho \subseteq Z^i \cap X$ et tout $n \in \mathbb{N}$, on ait $\Phi_{e_i}(\sigma_i \cup \rho, n) \uparrow$. Par hypothèse, $\mathcal{C} \neq \emptyset$. De plus, $\mathcal{C}$ est une classe $\Pi_1^0(X)$ avec $X \in \mathcal{I}$; donc, comme $\mathcal{I}$ est un idéal de Scott, il existe $Z^0 \oplus Z^1 \in \mathcal{I} \cap \mathcal{C}$. Soit $i < 2$ tel que $X \cap Z^i$ est infini. Alors, la condition $(\sigma_0, \sigma_1, X \cap Z^i)$ est une extension de la condition c telle que $(\sigma_i, X \cap Z^i) \Vdash \forall n \Phi_{e_i}(G, n) \uparrow$. Notons que $X \in \mathcal{I}$ et $Z^i \in \mathcal{I}$, donc $X \cap Z^i \in \mathcal{I}$. ■

Notons que dans chacun des deux cas précédents, il aurait été suffisant de s'assurer que $(\tau_i, Y \cap A^i)$ force la propriété, mais c'est également le cas pour (τ_i, Y) , ce qui est un résultat plus fort.

Nous avons à présent tous les éléments nécessaires pour montrer le théorème de Dzhabarov et Jockusch, que nous rappelons ci-après.

Théorème (3.8)

Pour tout ensemble A et tout ensemble C non calculable, il existe un ensemble infini $H \subseteq A$ ou $H \subseteq \mathbb{N} \setminus A$ tel que $C \not\leq_T H$.

PREUVE. Soient A et C fixés. Par la proposition 22-7.3, il existe un idéal de Scott $\mathcal{I}$ tel que $C \notin \mathcal{I}$. Supposons (H1), sinon nous avons déjà la solution désirée. Nous allons construire deux ensembles infinis $G^0 \subseteq A^0$, $G^1 \subseteq A^1$ satisfaisant pour tous $e_0, e_1 \in \mathbb{N}$ le contrat

$$\mathcal{R}_{e_0, e_1} : \Phi_{e_0}^{G^0} \neq C \vee \Phi_{e_1}^{G^1} \neq C.$$

Les ensembles G^0 et G^1 vont être construits par le forcing de Dzhabarov-Jockusch avec l'idéal $\mathcal{I}$.

Lemme 3.19. Soit $c = (\sigma_0, \sigma_1, X)$ une condition et soient $e_0, e_1 \in \mathbb{N}$. Il existe une extension (τ_0, τ_1, Y) de c forçant $\mathcal{R}_{e_0, e_1}$. ★

PREUVE. Soit $W = \{(x, v) \in \mathbb{N} \times \{0, 1\} : c \Vdash \Phi_{e_0}^{G^0}(x) \downarrow = v \vee \Phi_{e_1}^{G^1}(x) \downarrow = v\}$. Par le lemme 3.15, W est $\Sigma_1^0(X)$, avec $X \in \mathcal{I}$. Trois cas se présentent.

- ▷ Cas 1. On suppose $(x, 1 - C(x)) \in W$ pour un certain $x \in \mathbb{N}$. Par la proposition 3.17, il existe une extension (τ_0, τ_1, Y) de c et un entier $i < 2$ tels que $(\tau_i, Y) \Vdash \Phi_{e_i}^G(x) \downarrow = 1 - C(x)$, donc $(\tau_i, Y) \Vdash \Phi_{e_i}^G \neq C$.
- ▷ Cas 2. On suppose $(x, C(x)) \notin W$ pour un certain $x \in \mathbb{N}$. Par la proposition 3.17, il existe une extension (τ_0, τ_1, Y) de c et un entier $i < 2$ tels que $(\tau_i, Y) \Vdash \Phi_{e_i}^G(x) \uparrow \vee \Phi_{e_i}^G(x) \downarrow \neq C(x)$, donc $(\tau_i, Y) \Vdash \Phi_{e_i}^G \neq C$.

- ▷ Cas 3. Pour tout $(x, v) \in W$, $C(x) = v$, et pour tout $x \in \mathbb{N}$, il existe un $v < 2$ tel que $(x, v) \in W$. Alors, $C \leq_T X$, car pour $x \in \mathbb{N}$, il suffit d'attendre qu'un couple (x, v) soit X -énuméré dans W , et renvoyer v . Par hypothèse sur X , ce cas ne peut arriver. ■

Soit F un filtre suffisamment générique pour le forcing de Dzhanfarov-Jockusch, et soit $(G^0, G^1) = \dot{F}$. Par le lemme 3.13, G^0 et G^1 sont tous les deux infinis. Par définition d'une condition de forcing, $G^0 \subseteq A^0$ et $G^1 \subseteq A^1$. Par le lemme 3.19, (G^0, G^1) satisfait $\mathcal{R}_{e_0, e_1}$ pour tous $e_0, e_1 \in \mathbb{N}$. Il y a alors deux possibilités : soit $C \not\leq_T G^0$, soit il existe e_0 tel que $\Phi_{e_0}^{G^0} = C$. Dans ce cas-ci, pour tout e_1 on a $\Phi_{e_1}^{G^1} \neq C$, car $\mathcal{R}_{e_0, e_1}$ est satisfait, et $C \not\leq_T G^1$. ■

Exercice 3.20. (★★) Montrer que pour tout ensemble A et tout C non Σ_1^0 , il existe un ensemble infini $H \subseteq A$ ou $H \subseteq \bar{A}$ tel que C n'est pas $\Sigma_1^0(H)$. ◇

Exercice 3.21. (★★★) Montrer que pour tout ensemble A et toute fonction hyperimmune f , il existe un ensemble infini $H \subseteq A$ ou $H \subseteq \bar{A}$ tel que f est H -hyperimmune. ◇

Exercice 3.22. (★★) Adapter l'exercice précédent pour montrer que pour tout ensemble A et tout triplet de fonctions hyperimmunes f_0, f_1, f_2 , il existe un ensemble infini $H \subseteq A$ ou $H \subseteq \bar{A}$ tel que au moins deux parmi les trois fonctions sont H -hyperimmunes. ◇

Le théorème 3.8 a été généralisé par Monin et Patey [161] à toute la hiérarchie arithmétique.

Théorème 3.23 (Monin et Patey [161])

Soit $n \geq 1$. Pour tout ensemble A et tout C non Δ_n^0 , il existe un ensemble infini $H \subseteq A$ ou $H \subseteq \mathbb{N} \setminus A$ tel que C n'est pas $\Delta_n^0(H)$.

Une conséquence du théorème précédent est que pour tout ensemble A , il existe un ensemble infini $H \subseteq A$ ou $H \subseteq \mathbb{N} \setminus A$ qui n'est pas de degré high. La preuve se fait à l'aide d'une notion de forcing plus complexe qui sort du cadre introductif de ce livre.

3.3. Principe infini des tiroirs et degrés PA

Le théorème de Dzhanfarov et Jockusch sera utilisé pour montrer le théorème de Seetapun : RT_2^2 n'implique pas ACA_0 . Nous nous attaquons à présent au théorème de Liu, qui renforce celui de Seetapun en montrant que RT_2^2 n'implique pas WKL_0 . Là encore, le cœur de la preuve se situe dans les instances arbitraires de RT_2^1 .

Théorème 3.24 (Liu [146])

Pour tout ensemble A , il existe un ensemble infini $H \subseteq A$ ou $H \subseteq \mathbb{N} \setminus A$ qui n'est pas de degré PA .

La preuve que nous présentons ici de ce résultat est celle de Monin et Patey [159], plus courte que la preuve originale de Liu, grâce notamment à l'utilisation du concept de *classe large*.

Définition 3.25

Une classe $\mathcal{A} \subseteq 2^{\mathbb{N}}$ non vide est *large* si :

1. elle est *close par le haut* : $\forall X \in \mathcal{A} \forall Y \supseteq X \ Y \in \mathcal{A}$;
2. pour tout $k \in \mathbb{N}^*$, et pour tout $X_0 \sqcup \dots \sqcup X_{k-1} = \mathbb{N}$, il existe $i < k$ tel que $X_i \in \mathcal{A}$. ◇

Pour le reste de cette section on fixe un ensemble A , et l'on pose $A^0 = A$ et $A^1 = \mathbb{N} \setminus A$. On fixe également une énumération $(\mathcal{U}_e)_{e \in \mathbb{N}}$ de toutes les classes Σ_1^0 closes par le haut.

Définition 3.26

Soit $\mathbb{P}$ l'ensemble des conditions de forcing de la forme $\langle (\sigma_s, X_s)_{s < k}, C \rangle$ pour $k \in \mathbb{N}^*$ telles que :

- (a) pour tout $s < k$, il existe $i < 2$ tel que $\sigma_s \cup X_s \subseteq A^i$;
- (b) $X_0 \sqcup \dots \sqcup X_{k-1} = \mathbb{N} - \{0, \dots, \max_s |\sigma_s|\}$;
- (c) $\bigcap_{e \in C} \mathcal{U}_e$ est une classe large ne contenant que des ensembles infinis;
- (d) C est calculable. ◇

Notons de suite que nous n'avons aucune restriction d'effectivité sur les réservoirs $X_0 \sqcup \dots \sqcup X_{k-1}$.

Définition 3.27

L'ordre partiel sur $\mathbb{P}$ est défini par $\langle (\tau_s, Y_s)_{s < \ell}, D \rangle \leq \langle (\sigma_s, X_s)_{s < k}, C \rangle$ si $C \subseteq D$, si $\ell \geq k$, et s'il existe une surjection $f : \ell \rightarrow k$ telle que pour tout $s < \ell$ on a $\sigma_{f(s)} \preceq \tau_s$, $Y_s \subseteq X_{f(s)}$, et $\tau_s - \sigma_{f(s)} \subseteq X_{f(s)}$. ◇

Une condition $c = \langle (\sigma_s, X_s)_{s < k}, C \rangle$ doit être vue comme un ensemble de branches (σ_s, X_s) chacune étant une condition de Mathias, à ceci près que l'on n'exige pas que X_s soit infini. Une extension $d \leq c$ est une condition dont chaque branche sera une extension d'une des branches de c .

Définition 3.28

Soit $\langle (\sigma_s, X_s)_{s < k}, C \rangle \in \mathbb{P}$. On appelle *branche* $c^{[s]}$ de c la condition de

Mathias (σ_s, X_s) . Une telle branche est *valide* si $X_s \in \bigcap_{e \in C} \mathcal{U}_e$. $\diamond$

Notons que la notion de validité est cruciale : les branches de nos conditions $c_0 \geq c_1 \geq \dots$ vont former un arbre, et le générique final que nous utiliserons y proviendra d'un chemin infini de branches valides.

Notation

Pour deux conditions $c = \langle (\tau_s, Y_s)_{s < \ell}, D \rangle$ et $d = \langle (\sigma_s, X_s)_{s < k}, C \rangle$, on écrira $d \leq_f c$ si $d \leq c$ via la fonction $f : \ell \rightarrow k$. Si f est la fonction identité, alors d est une *extension simple*. Étant donné $s < k$, si $f^{-1}(t)$ est un singleton pour tout $t \neq s$, alors d est une *s-extension* de d .

Le lemme suivant sera utilisé pour montrer que le générique que nous construisons est infini.

Lemme 3.29. Soit $c \in \mathbb{P}$ une condition de branche valide $c^{[s]} = (\sigma, X)$. Alors, il y a une extension simple $d \leq c$ de branche $d^{[s]} = (\tau, Y)$ et un entier $n > |\sigma|$ tels que $n \in \tau$. $\star$

PREUVE

Soit $c = \langle (\sigma_s, X_s)_{s < k}, C \rangle$. Comme $c^{[s]}$ est valide on a $X_s \in \bigcap_{e \in C} \mathcal{U}_e$. Comme $\bigcap_{e \in C} \mathcal{U}_e$ ne contient que des ensembles infinis, alors X_s est infini. Soit $n \in X_s$ tel que $n > |\sigma_s|$. Alors, la condition

$$d = \langle (\tau_s, X_s \cap]n, \infty[)_{s < k}, C \rangle$$

définie par $\tau_s = \sigma_s \cup \{n\}$ et $\tau_t = \sigma_t$ pour $t \neq s$ est une extension simple de c qui satisfait le lemme. $\blacksquare$

La définition suivante sera nécessaire pour développer le cœur de l'argument.

Définition 3.30. Soit $\xi : \mathbb{N} \times 2^{<\mathbb{N}} \rightarrow \mathbb{N}$ une fonction calculable qui prend le code e d'une fonctionnelle Turing $\Phi_e(G, n)$, une chaîne σ , et qui renvoie le code de l'ouvert :

$$\{X : \exists \rho \subseteq X - \{0, \dots, |\sigma|\} \exists n \Phi_e(\sigma \cup \rho, n) \downarrow = \Phi_n(n)\}. \quad \diamond$$

Nous arrivons finalement au lemme clef de la preuve.

Lemme 3.31. Soit $c^{[s]}$ la branche d'une condition $c \in \mathbb{P}$. Soit $\Phi_e(G, n)$ une fonctionnelle Turing à valeurs dans $\{0, 1\}$ pour tous ses oracles. Il y a une *s-extension* $d \leq_f c$ telle que pour toute branche valide $d^{[t]}$ de d pour

laquelle $f(t) = s$, il existe n pour lequel :

$$d^{[t]} \Vdash \Phi_e(G, n) \downarrow = \Phi_n(n) \text{ ou } d^{[t]} \Vdash \Phi_e(G, n) \uparrow. \quad \star$$

PREUVE. Soit $c = \langle (\sigma_s, X_s)_{s < k}, C \rangle$, et soit $P(n, k, v)$ le prédicat :

$$\forall Z_0 \sqcup \dots \sqcup Z_{k-1} = \mathbb{N} \exists j < k$$

$$Z_j \in \bigcap_{e \in C, e < k} \mathcal{U}_e \text{ et } \exists \rho \subseteq Z_j - \{0, \dots, |\sigma_s|\} \Phi_e(\sigma_s \cup \rho, n) \downarrow = v.$$

Le prédicat $P(n, k, v)$ est vrai si, et seulement si, la classe Π_1^0 des éléments

$$\left\{ Z_0 \sqcup \dots \sqcup Z_{k-1} = \mathbb{N} : \forall j < k \begin{array}{l} Z_j \notin \bigcap_{e \in C, e < k} \mathcal{U}_e \\ \text{ou } \forall \rho \subseteq Z_j - \{0, \dots, |\sigma_s|\} \Phi_e(\sigma_s \cup \rho, n) \uparrow \neq v \end{array} \right\}$$

est vide. La notation $\Phi_e(\sigma_s \cup \rho, n) \uparrow \neq v$ ci-dessus signifie

$$\Phi_e(\sigma_s \cup \rho, n) \uparrow \vee \Phi_e(\sigma_s \cup \rho, n) \downarrow \neq v.$$

Par le lemme faible de König, le prédicat $P(n, k, v)$ est donc Σ_1^0 .

Supposons d'abord que, pour tout $k \in \mathbb{N}$, on ait $\forall n \exists v < 2 P(n, k, v)$.

Notons que pour tout $k \in \mathbb{N}$ l'ensemble $W_k = \{(n, v) : P(n, k, v)\}$ est c. e.

On doit donc avoir un entier $n \in \mathbb{N}$ tel que $(n, \Phi_n(n)) \in W_k$, sinon on calculerait une fonction DNC₂. Ainsi, en particulier :

$$\forall Z_0 \sqcup \dots \sqcup Z_{k-1} = \mathbb{N} \exists j < k Z_j \in \bigcap_{e \in C} \mathcal{U}_e \cap \mathcal{U}_{\xi(e, \sigma_s)}.$$

D'où, $\bigcap_{e \in C} \mathcal{U}_e \cap \mathcal{U}_{\xi(e, \sigma_s)}$ est une classe large. Si $X_s \notin \bigcap_{e \in C} \mathcal{U}_e \cap \mathcal{U}_{\xi(e, \sigma_s)}$, alors $d = \langle (\sigma_s, X_s)_{s < k}, C \cup \{\xi(e, \sigma_s)\} \rangle$ est une s -extension de c sur laquelle la branche $d^{[s]}$ n'est pas valide, et il n'y a rien de plus à vérifier.

Si $X_s \in \bigcap_{e \in C} \mathcal{U}_e \cap \mathcal{U}_{\xi(e, \sigma_s)}$, alors il existe n et il existe $\rho \subseteq X_s - \{0, \dots, |\sigma_s|\}$ tels que $\Phi_e(\sigma_s \cup \rho, n) \downarrow = \Phi_n(n)$. La condition $d = \langle (\tau_s, X_s \setminus \{0, \dots, |\rho|\})_{s < k}, C \rangle$ définie par $\tau_s = \sigma_s \cup \rho$ et $\tau_t = \sigma_t$ pour $t \neq s$ est une s -extension de c telle que $d^{[s]} \Vdash \Phi_e(G, n) \downarrow = \Phi_n(n)$.

Supposons à présent qu'il existe $k \in \mathbb{N}$ tel que

$$\exists n \forall v < 2 \neg P(n, k, v).$$

En particulier, pour $k, n \in \mathbb{N}$, on a des partitions $Z_0^0 \sqcup \dots \sqcup Z_{k-1}^0 = \mathbb{N}$ et $Z_0^1 \sqcup \dots \sqcup Z_{k-1}^1 = \mathbb{N}$ telles que pour $v < 2$ on a :

$$\forall j < k \left(Z_j^v \notin \bigcap_{e \in C, e < k} \mathcal{U}_e \text{ ou } \forall \rho \subseteq Z_j^v - \{0, \dots, |\sigma_s|\} \Phi_e(\sigma_s \cup \rho, n) \uparrow \neq v \right).$$

Soit $d = \langle (\tau_s, Y_s)_{s < \ell}, C \rangle$ la s -extension de c obtenue en dupliquant les branches $c^{[s]}$ en k^2 branches $c^{[s_{0,0}]}, \dots, c^{[s_{k-1,k-1}]}$, telles que $\tau_{s_{i,j}} = \sigma_s$ et $Y_{s_{i,j}} = X_s \cap Z_i^0 \cap Z_j^1$.

Chaque autre branche $c^{[t]}$ pour $t \neq s$ est identique dans d . Notons que si la branche $d^{[s_{i,j}]}$ est valide alors $X_s \cap Z_i^0 \cap Z_j^1 \in \bigcap_{e \in C} \mathcal{U}_e$ et donc

$$\forall \rho \subseteq X_s \cap Z_i^0 \cap Z_j^1 - \{0, \dots, |\sigma_s|\} \quad \Phi_e(\sigma_s \cup \rho, n) \uparrow \notin \{0, 1\}.$$

Comme Φ_e est à valeurs dans $\{0, 1\}$, on a donc $d^{[s_{i,j}]} \Vdash \Phi_e(G, n) \uparrow$ pour tous $i, j < k$ tels que $d^{[s_{i,j}]}$ est valide. Cela complète la preuve du lemme.

Lemme 3.32. Soit $c_0 \geq c_1 \geq \dots$ une suite décroissante de conditions. Alors, il existe une fonction $g : \mathbb{N} \rightarrow \mathbb{N}$ telle que $c_0^{[g(0)]} \geq c_1^{[g(1)]} \geq \dots$ et telle que pour tout n , $c_n^{[g(n)]}$ est une branche valide de c_n . ★

PREUVE. Montrons d'abord la chose suivante. Soit $c = \langle (\sigma_s, X_s)_{s < k}, C \rangle$, et soit $d = \langle (\tau_s, Y_s)_{s < \ell}, D \rangle$ avec $d \leq_f c$; si $d^{[s]}$ est valide, alors $c^{[f(s)]}$ est valide.

Supposons $d^{[s]}$ valide. Alors, $Y_s \in \bigcap_{e \in D} \mathcal{U}_e \subseteq \bigcap_{e \in C} \mathcal{U}_e$. Comme $Y_s \subseteq X_{f(s)}$ et comme chaque $\mathcal{U}_e$ est clos par le haut, alors $X_{f(s)} \in \bigcap_{e \in C} \mathcal{U}_e$, si bien que $c^{[f(s)]}$ est valide.

Les branches valides de chaque condition de la suite $(c_i)_{i \in \mathbb{N}}$ forment donc un arbre pour l'ordre $\leq$ entre les conditions de Mathias. L'arbre étant à branchement fini, il a un chemin infini si, seulement si, il est infini. Il est donc suffisant de montrer que chaque c_i possède une branche valide. Soit $c_i = \langle (\sigma_s, X_s)_{s < k}, C \rangle$, et soit $m = \sup_{s < k} |\sigma_s|$. Alors,

$$[0, m] \sqcup X_0 \sqcup \dots \sqcup X_{s-1} = \mathbb{N}.$$

Comme $\bigcap_{e \in C} \mathcal{U}_e$ est large et ne contient que des ensembles infinis, il existe nécessairement $s < k$ tel que $X_s \in \bigcap_{e \in C} \mathcal{U}_e$. La branche $c_i^{[s]}$ est donc valide.

La fonction g du lemme est donnée par le chemin infini de notre arbre de branches valides. ■

PREUVE DU THÉORÈME 3.24. Soit $c_0 \geq c_1 \geq \dots$ un filtre de conditions suffisamment générique. Notons que l'on peut démarrer avec la condition $\langle (\epsilon, A^0), (\epsilon, A^1), C \rangle$, où C est un ensemble calculable de codes tels que $\bigcap_{e \in C} \mathcal{U}_e$ est la classe des ensembles infinis.

D'après le lemme 3.32, soit $g : \mathbb{N} \rightarrow \mathbb{N}$ telle que $c_0^{[g(0)]} \geq c_1^{[g(1)]} \geq \dots$ et telle que pour tout i , $c_i^{[g(i)]}$ est une branche valide de c_i . D'après le lemme 3.29, l'ensemble générique G résultant est infini. D'après le lemme 3.31, pour toute fonctionnelle $\Phi_e(G, n)$, il existe n tel que $c_i^{[g(i)]} \Vdash \Phi_e(G, n) \downarrow = \Phi_n(n)$ ou tel que $c_i^{[g(i)]} \Vdash \Phi_e(G, n) \uparrow$. Il s'ensuit que notre générique n'est pas de degré PA. Enfin, par nature des conditions de forcing, on a $G \subseteq A^i$ pour $i = 0$ ou $i = 1$. ■

Le résultat de Liu est loin d'être évident, et la question de sa version itérée est encore ouverte à ce jour.

Question 3.33. Soit $n \geq 1$. Pour tout ensemble A , existe-t-il un ensemble infini $H \subseteq A$ ou $H \subseteq \mathbb{N} \setminus A$ tel que $H^{(n)}$ n'est pas de degré PA relativement à $\emptyset^{(n)}$? ★

Monin et Patey [160] ont donné une réponse positive partielle en montrant que c'était le cas pour les instances Δ_2^0 de RT_2^1 .

3.4. Instances Δ_2^0 de RT_2^1

Les instances Δ_2^0 sont particulièrement importantes pour l'étude du théorème de Ramsey pour les paires, car elles jouent un rôle clef dans la connexion entre RT_2^1 et RT_2^2 , comme nous le verrons dans la section 4. Le principe infini des tiroirs étant calculatoirement vrai, toute instance Δ_2^0 de RT_2^1 admet une solution Δ_2^0 . Existe-t-il une instance Δ_2^0 de RT_2^1 sans solution calculable ? La question peut se reformuler comme suit : existe-t-il un ensemble $\Delta_2^0 A$ à la fois immune, et de complémentaire immune (voir la définition 7-1.1) ? La proposition 3.34 donne un résultat plus fort, en montrant l'existence d'une instance Δ_2^0 de RT_k^1 dont les solutions ne peuvent même pas être approximées par blocs.

Proposition 3.34. Il existe un ensemble $A \in \Delta_2^0$ tel que A et $\mathbb{N} \setminus A$ sont tous les deux hyperimmunes. ★

PREUVE. Rappelons que la *fonction principale* d'un ensemble infini H est la fonction $p_H : \mathbb{N} \rightarrow \mathbb{N}$ qui à n associe le $(n+1)$ -ième élément de H . Rappelons également que H est hyperimmune si, et seulement si, p_H n'est bornée par aucune fonction calculable.

Nous définissons une suite $\emptyset'$ -calculable $\sigma_0 \prec \sigma_1 \prec \dots$ en alternant la concaténation de longues suites de 1 avec celle de longues suites de 0, afin de diagonaliser contre toutes les fonctions calculables.

Supposons σ_{2e} définie, alors $\sigma_{2e+1} = \sigma_{2e} 0^{k+1} 1$ si $\Phi_e(|\sigma_{2e}| + 1) \downarrow = k$, et $\sigma_{2e+1} = \sigma_{2e} 1$ sinon. Supposons σ_{2e+1} définie, alors $\sigma_{2e+2} = \sigma_{2e+1} 1^{k+1} 0$ si $\Phi_e(|\sigma_{2e+1}| + 1) \downarrow = k$, et $\sigma_{2e+2} = \sigma_{2e+1} 0$ sinon.

On vérifie sans peine que si Φ_e est totale, alors

$$\begin{aligned} &\text{pour } m = |\sigma_{2e}| + 1, \quad \text{on a } p_A(m) \geq |\sigma_{2e+1}| - 1 > \Phi_e(m), \\ &\text{et pour } m = |\sigma_{2e+1}| + 1, \quad \text{on a } p_{\mathbb{N} \setminus A}(m) \geq |\sigma_{2e+2}| - 1 > \Phi_e(m). \end{aligned}$$

■

Notons qu'un sous-ensemble infini d'un ensemble hyperimmune est hyperimmune. La proposition 3.34 nous donne donc une instance $\Delta_2^0 B_0 \sqcup B_1 = \mathbb{N}$

de RT_2^1 telle que toute solution est de degré hyperimmune. En particulier, il existe une instance Δ_2^0 de RT_2^1 sans solution calculable.

Exercice 3.35. (★) Montrer qu'il existe un ensemble Δ_2^0 A tel que A et $\mathbb{N} \setminus A$ sont effectivement immunes. En déduire que tout ensemble infini $H \subseteq A$ ou $H \subseteq \mathbb{N} \setminus A$ est de degré DNC. $\diamond$

Exercice 3.36. (★) Montrer que pour tout $k \geq 2$, il existe une k -partition Δ_2^0 , soit $A_0 \sqcup A_1 \sqcup \dots \sqcup A_{k-1} = \mathbb{N}$, telle que pour tout $i < k$, $\mathbb{N} \setminus A_i$ est hyperimmune. $\diamond$

Exercice 3.37. (★★) Montrer qu'il existe une instance g de RT_3^1 telle que pour toute instance h de RT_2^1 , il existe un ensemble h -homogène qui ne calcule pas d'ensemble g -homogène.

Indication. – Utiliser les exercices 3.36 et 3.22. $\diamond$

L'instance Δ_2^0 de RT_k^1 créée par la proposition 3.34 n'est cependant pas trop complexe, et possède en particulier une solution low, comme l'ont prouvé Hirschfeldt, Jockusch, Kjos-Hanssen et Lempp [89].

Proposition 3.38 ([89]). Soit A un ensemble Δ_2^0 tel que $\mathbb{N} \setminus A$ est hyperimmune. Alors, il existe un sous-ensemble infini $H \subseteq A$ de degré low. $\star$

PREUVE. Nous allons construire une suite Δ_2^0 $\sigma_0 \prec \sigma_1 \prec \dots \in 2^{<\mathbb{N}}$ de chaînes telles que pour tout n , $\sigma_n \subseteq A$, et telles que pour tout e ,

$$\Phi_e^{\sigma_{e+1}}(e) \downarrow \text{ ou } \forall \tau \succeq \sigma_{e+1} \Phi_e^\tau(e) \uparrow. \quad (1)$$

Notons que dans la seconde clause de (1), τ n'est pas nécessairement inclus dans A . Ainsi, $\emptyset'$ arrive à décider si $\Phi_e^H(e) \downarrow$ pour $\{H\} = \bigcap_n [\sigma_n]$ en effectuant la construction jusqu'à σ_{e+1} , et en décidant à l'aide de $\emptyset'$ si l'on est dans le premier cas ou le second cas.

Construction. Supposons que l'on a construit $\sigma_e \in 2^{<\mathbb{N}}$. Chercher $\emptyset'$ -calculatoirement une extension $\sigma_{e+1} \succeq \sigma_e$ telle que $\sigma_{e+1} \subseteq A$ et telle que (1) est satisfait. Montrons qu'il existe toujours une telle extension. Pour tout entier $n > |\sigma_e|$, on cherche τ_n , une extension de 0^n , telle que $\Phi_e^{\sigma_e \cup \tau_n}(e) \downarrow$. S'il existe un n tel que l'on ne trouve pas τ_n , alors $\sigma_{e+1} = \sigma_e \cup 0^n$ est une extension de σ_e satisfaisant la seconde clause de (1). Si τ_n est toujours défini, alors par hyperimmunité de $\mathbb{N} \setminus A$, il existe $n > |\sigma_e|$ tel que $\tau_n \subseteq A$. Alors, $\sigma_{e+1} = \sigma_e \cup \tau_n$ est une extension de σ_e satisfaisant la première clause de (1). Dans tous les cas, une telle extension existe. Cela conclut la construction et la preuve de la proposition 3.38. $\blacksquare$

Peut-on obtenir une solution low pour toute instance Δ_2^0 du principe infini des tiroirs? La réponse est non, et se prouve à l'aide d'un argument de

priorité à blessure infinie. Nous omettrons ici la preuve qui est longue et d'une grande complexité.

Théorème 3.39 (Downey, Hirschfeldt, Lempp et Solomon [47])

Il existe A un ensemble Δ_2^0 tel que ni lui ni son complémentaire ne contient de sous-ensemble infini low.

L'instance de RT_2^1 créée par le théorème 3.39 est très différente de celle créée par la proposition 3.34, puisque tout ensemble Δ_2^0 dont le complémentaire est hyperimmune admet un sous-ensemble infini de degré low.

Cholak, Jockusch et Slaman [33] ont cependant prouvé que toute instance Δ_2^0 de RT_2^1 admet une solution « presque low », dans le sens où tout degré PA relativement à $\emptyset'$ calcule son saut Turing.

Théorème 3.40 (Cholak, Jockusch et Slaman [33])

Soit A un ensemble Δ_2^0 et P un degré PA relativement à $\emptyset'$. Il existe un ensemble infini $H \subseteq A$ ou $H \subseteq \mathbb{N} \setminus A$ tel que $H' \leq_T P$.

PREUVE. La preuve du théorème 3.40 se fait à l'aide d'une construction effective d'un filtre générique pour le forcing de Dzhafarov-Jockusch. Rappelons qu'un *code de lowness* d'un ensemble low X est un entier e tel que $\{n : \Phi_e(\emptyset', n) \downarrow\} = X'$ (voir la section 5-7). Le théorème de base Π_1^0 low est uniforme, au sens où il existe une fonction calculable $g : \mathbb{N} \rightarrow \mathbb{N}$ telle que si Φ_e est un arbre binaire infini, alors $g(e)$ est le code de lowness d'un chemin de Φ_e . Par l'exercice 22-7.5, il existe un idéal de Scott $\mathcal{I}$ dont tous les éléments sont low. La construction est uniforme, et il existe une fonction calculable $h_{\text{WKL}} : \mathbb{N} \rightarrow \mathbb{N}$ telle que si e est le code de lowness d'un arbre binaire infini dans $\mathcal{I}$, alors $h_{\text{WKL}}(e)$ est le code de lowness d'un chemin infini de cet arbre dans $\mathcal{I}$. De plus, il existe fonction calculable $h_{\oplus} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ qui, étant donné deux indices de lowness e_0, e_1 d'ensembles X et Y , renvoie un code de lowness de l'ensemble $X \oplus Y$ dans $\mathcal{I}$. Un *code de condition* (σ_0, σ_1, X) est un triplet (σ_0, σ_1, e) tel que e est un code de lowness de X . Ainsi, toute condition de Dzhafarov-Jockusch dans l'idéal $\mathcal{I}$ peut être codée de manière finie.

Supposons (H1) (définie dans la section 3.2). En effet, il existe sinon un ensemble infini $H \subseteq A^0$ ou $H \subseteq A^1$ de degré low, et l'on a en particulier, $H' \leq_T \emptyset' \leq_T P$. Sachant que A est fixé et Δ_2^0 , le lemme 3.13 est uniforme en $\emptyset'$, au sens où il existe une procédure $\emptyset'$ -calculable qui prend en entrée un code (σ_0, σ_1, a) de condition (σ_0, σ_1, X) et un entier $i < 2$, et renvoie un code (τ_0, τ_1, b) de condition (τ_0, τ_1, Y) étendant (σ_0, σ_1, X) tel qu'il existe un entier $n \geq |\sigma_i|$ pour lequel $\tau_i(n) = 1$. Sachant que $\emptyset' \leq_T P$, alors P calcule deux ensembles infinis $G^0 \subseteq A^0$ et $G^1 \subseteq A^1$. Nous voulons

cependant décider le saut Turing de G^0 et G^1 , à l'aide de la question de forcing.

Considérons une condition $c = (\sigma_0, \sigma_1, X)$ et deux entiers $e_0, e_1 \in \mathbb{N}$. Par le lemme 3.15, le prédicat $c \Vdash \Phi_{e_0}^{G^0}(e_0) \downarrow \vee \Phi_{e_1}^{G^1}(e_1) \downarrow$ est $\Sigma_1^0(X)$. Ainsi, $\emptyset'$ peut décider, étant donné un code de la condition c , si le prédicat est vrai ou non. Si le prédicat est vrai, alors $\emptyset'$ peut chercher un entier $i < 2$, et une chaîne finie $\rho \subseteq X \cap A^i$ telle que $\Phi_{e_i}^{\sigma_i \cup \rho}(e_i) \downarrow$, et calculer un code de lowness de l'ensemble $X \setminus \{0, \dots, |\rho|\}$. Ainsi, si le prédicat est vrai, $\emptyset'$ peut calculer un code d'une extension d de c et trouver un $i < 2$ forçant $\Phi_{e_i}^{G^i}(e_i) \downarrow$. Si le prédicat est faux, alors $\emptyset'$ peut calculer un code de lowness d'un ensemble $Z^0 \oplus Z^1$ tel que $Z^0 \sqcup Z^1 = X$, et pour tout $i < 2$ et tout $\rho \subseteq Z^i \cup X$, on a $\Phi_{e_i}^{\sigma_i \cup \rho}(e_i) \uparrow$. Jusqu'ici, toutes les opérations pouvaient être effectuées à l'aide de $\emptyset'$. En revanche, il est nécessaire d'avoir recours à P pour trouver un entier $i < 2$ tel que $Z^i \cap X$ est infini, et donc pour calculer un code de l'extension $(\sigma_0, \sigma_1, Z^i \cap X)$. Un degré PA relativement à $\emptyset'$ est capable de trouver, parmi deux formules Π_2^0 F_0 et F_1 sachant que l'une des deux au moins est vraie, un entier $i < 2$ tel que F_i est vraie. Ainsi, si le prédicat est faux, P peut calculer un code d'une extension d de c et $i < 2$ tels que d force $\Phi_{e_i}^{G^i}(e_i) \uparrow$.

Construction. Construire une suite P -calculable de codes de conditions

$$c_0 \geq c_1 \geq \dots, \quad \text{avec } c_s = (\sigma_0^s, \sigma_1^s, X^s)$$

telle que, pour tout $s \in \mathbb{N}$,

- (1) il existe $n_0, n_1 > s$ tels que $\sigma_0^s(n_0) = 1$ et $\sigma_1^s(n_1) = 1$;
- (2) si $s = \langle e_0, e_1 \rangle$, alors il existe $i < 2$ tel que $(\sigma_i^{s+1}, X^{s+1}) \Vdash \Phi_{e_i}^G(e_i) \downarrow$ ou $(\sigma_i^{s+1}, X^{s+1}) \Vdash \Phi_{e_i}^G(e_i) \uparrow$.

Soient $\{G^0\} = \bigcap_s [\sigma_0^s]$ et $\{G^1\} = \bigcap_s [\sigma_1^s]$. Par (1), G^0 et G^1 sont tous les deux infinis. Par construction, $G^0 \subseteq A^0$ et $G^1 \subseteq A^1$. Par (2), il existe $i < 2$ tel que pour tout $e \in \mathbb{N}$, il existe s tel que $(\sigma_i^{s+1}, X^{s+1}) \Vdash \Phi_e^G(e) \downarrow$ ou $(\sigma_i^{s+1}, X^{s+1}) \Vdash \Phi_e^G(e) \uparrow$.

Montrons que le saut Turing de G^i est P -calculable. En effet, pour décider si $\Phi_e^{G^i}(e) \downarrow$, il suffit d'exécuter à l'aide de P la construction jusqu'à trouver un entier s tel que $(\sigma_i^{s+1}, X^{s+1}) \Vdash \Phi_e^G(e) \downarrow$ ou $(\sigma_i^{s+1}, X^{s+1}) \Vdash \Phi_e^G(e) \uparrow$. Cela termine la preuve du théorème 3.40. ■

Rappelons qu'un ensemble H est low_2 si $H'' \leq_T \emptyset''$. Le théorème de base $\text{low } \Pi_1^0$ (voir le théorème 8-4.3) nous permet de déduire le corollaire suivant.

Corollaire 3.41

Soit A un ensemble Δ_2^0 . Il existe un ensemble infini $H \subseteq A$ ou $H \subseteq \mathbb{N} \setminus A$ de degré low_2 .

PREUVE. Par le théorème 8-4.3 relativisé à $\emptyset'$, il existe un ensemble P de degré PA relativement à $\emptyset'$ tel que $P' \leq_T \emptyset''$. Par le théorème 3.40, il existe un ensemble infini $H \subseteq A$ ou $H \subseteq \mathbb{N} \setminus A$ tel que $H' \leq_T P$. En particulier, $H'' \leq_T P' \leq_T \emptyset''$, et H est donc de degré low_2 . ■

4. Théorème de Ramsey pour les paires

Le théorème 2.17 affirme que pour tout $n \geq 3$, $\forall k \text{RT}_k^n$ est équivalent à ACA_0 . Ce théorème se prouve en montrant que toute instance calculable de RT_k^n admet une solution arithmétique, tandis qu'il existe un coloriage calculable $f : [\mathbb{N}]^3 \rightarrow 2$ dont tout ensemble infini homogène calcule $\emptyset'$. Le principe des tiroirs RT_k^1 , quant à lui, est calculatoirement vrai, et prouvable dans RCA_0 pour tout entier k , tandis que l'énoncé $\forall k \text{RT}_k^1$ est équivalent au principe $\text{B}\Pi_1^0$ dans RCA_0 . Dans cette section, nous allons voir comme annoncé que les choses sont bien différentes pour le théorème de Ramsey pour les paires. En particulier, il existe une instance calculable de RT_2^2 dont aucune solution ne calcule $\emptyset'$, ce qui nous permettra de déduire le théorème de Seetapun. Nous verrons également qu'il existe une instance calculable de RT_2^2 dont aucune solution n'est de degré PA, ce qui nous amènera au théorème de Liu.

4.1. Principe de cohésion

La preuve du théorème de Ramsey se fait de manière inductive à l'aide de la notion d'ensemble pré-homogène. Dans le cas du théorème de Ramsey pour les paires, la bonne notion à considérer est celle d'ensemble cohésif, qui est une forme d'inversion de saut de la notion d'ensemble pré-homogène.

Définition 4.1. Étant donné deux ensembles X et Y , on note $X \subseteq^* Y$ si $X \setminus Y$ est un ensemble fini. Soit $(R_n)_{n \in \mathbb{N}}$ une suite d'ensembles. Un ensemble C est dit *cohésif* pour $(R_n)_{n \in \mathbb{N}}$ s'il est infini et si $C \subseteq^* R_n$ ou $C \subseteq^* \mathbb{N} \setminus R_n$ pour tout n . ◆

Notation

On dénote par **COH** le principe suivant : pour toute suite $(R_n)_{n \in \mathbb{N}}$, il existe un ensemble cohésif.

L'existence d'un ensemble cohésif pour tous les ensembles calculables se prouve aisément à l'aide de la restriction du forcing de Mathias aux conditions dont les réservoirs sont calculables.

Exercice 4.2. (★) Soit $\mathbb{P}$ l'ensemble des conditions de Mathias (σ, X) telles que X est calculable. Montrer que tout ensemble suffisamment générique pour $\mathbb{P}$ est cohésif pour tous les ensembles calculables. $\diamond$

Soit $f : [\mathbb{N}]^2 \rightarrow 2$ une instance calculable de RT_2^2 . Tandis qu'un ensemble pré-homogène P transforme f en une instance P -calculable de RT_2^1 , un ensemble cohésif C transforme f en une instance $\Delta_2^0(P)$ de RT_2^1 .

Fait

Avec $k \geq 1$, fixons un coloriage $f : [\mathbb{N}]^2 \rightarrow k$. Soit $(R_{n,i})_{n \in \mathbb{N}, i < k}$ la suite d'ensembles définie par $R_{n,i} = \{y \in \mathbb{N} : f(\{n, y\}) = i\}$. Soit C un ensemble infini cohésif pour $(R_{n,i})_{n \in \mathbb{N}, i < k}$. Alors, pour tout $x \in C$, $\lim_{y \in C} f(\{x, y\})$ existe. De plus, soit $g : C \rightarrow k$ le coloriage $\Delta_2^0(C \oplus f)$ défini par $g(x) = \lim_{y \in C} f(\{x, y\})$. Pour tout ensemble infini $Y \subseteq C$ homogène pour g , $Y \oplus f$ calcule un ensemble infini homogène pour f .

PREUVE. Soit $i < k$ la couleur d'homogénéité de Y pour g . Alors, pour tout $x \in Y$, $\lim_{y \in C} f(\{x, y\}) = i$. Ainsi, soit $(x_n)_{n \in \mathbb{N}}$ la suite d'éléments de Y définie comme suit : si x_s est défini pour tout $s < n$, alors x_n est le plus petit élément de Y tel que pour tout $s < n$, $f(\{x_s, x_n\}) = i$. Un tel x_n existe par l'hypothèse de limite. L'ensemble $\{x_n : n \in \mathbb{N}\}$ est homogène pour f . ■

Le fait 4.1 ramène donc l'étude des instances calculables de RT_2^2 à celle des instances Δ_2^0 de RT_2^1 et des instances calculables de COH. Nous avons déjà étudié de manière extensive le principe infini des tiroirs dans la section 3, ce qui nous ramène à l'étude de COH.

La proposition suivante nous indique que COH n'est pas calculatoirement plus compliqué que RT_2^2 . La réduction de COH à RT_2^2 est un peu subtile, et demande quelques précautions pour être formalisée dans RCA_0 .

Proposition 4.3 (Mileti [156]). On a :

$$\text{RCA}_0 \vdash \text{RT}_2^2 \rightarrow \text{COH} \quad \text{et} \quad \text{COH} \leq_W \text{RT}_2^2. \quad \star$$

PREUVE. Supposons RT_2^2 . Soit $(R_n)_{n \in \mathbb{N}}$ une suite d'ensembles. En rajoutant de manière calculable des ensembles à $(R_n)_{n \in \mathbb{N}}$, on peut supposer que pour tout $x < y$, il existe un plus petit $i \in \mathbb{N}$ tel que $R_i(x) \neq R_i(y)$;

soit $i_{x,y}$ cet entier. On définit alors le coloriage $f : [\mathbb{N}]^2 \rightarrow 2$ pour tout $x < y$ par $f(\{x, y\}) = 1$ si $x \in X_{i_{x,y}}$, et $f(\{x, y\}) = 0$ sinon.

Soit $C = \{a_0 < a_1 < a_2 < \dots\}$ un ensemble homogène pour f de couleur 0 (le cas 1 est symétrique). Soit $(H_{n,b}^i)$ la propriété « Pour tout ensemble fini F , si $R_n(a_s) = i$ et $R_n(a_{s+1}) = 1 - i$ pour tout $s \in F$, alors $|F| < b$. »

Montrons que $(H_{n,b}^1)$ implique $(H_{n,b+1}^0)$. Soit $F = \{s_0 < s_1 < \dots < s_{\ell-1}\}$ un ensemble fini tel que $R_n(a_s) = 0$ et $R_n(a_{s+1}) = 1$ pour tout $s \in F$. Alors, pour tout $t < \ell - 1$, il existe $s \in]s_t, s_{t+1}[$ tel que $R_n(a_s) = 1$ et $R_n(a_{s+1}) = 0$, donc par $(H_{n,b}^1)$, $\ell - 1 < b$, et donc $|F| < b + 1$. Ainsi, $(H_{n,b+1}^0)$ est vrai.

Montrons par induction sur n que $(H_{n,2^n}^1)$ est vrai. La formule $(H_{n,2^n}^1)$ étant Π_1^0 , cette induction est valide car $\text{RCA}_0 \vdash \text{I}\Pi_1^0$. Supposons que $(H_{m,2^m}^1)$ soit vrai pour tout $m < n$. Soit F un ensemble tel que $R_n(a_s) = 1$ et $R_n(a_{s+1}) = 0$ pour tout $s \in F$. Comme C est homogène pour f de couleur 0, pour tout $s \in F$, on a $i_{a_s, a_{s+1}} < n$, car sinon on aurait $i_{a_s, a_{s+1}} = n$, et donc $f(\{a_s, a_{s+1}\}) = 1$.

Pour tout $m < n$, soit $F_m = \{s \in F : i_{a_s, a_{s+1}} = m\}$. Notons que pour tout $s \in F_m$, $R_m(a_s) = 0$ et $R_m(a_{s+1}) = 1$, car C est homogène pour f de couleur 0. Donc, par $(H_{m,2^m}^1)$, $|F_m| \leq 2^m$. Comme $\bigcup_{m < n} F_m = F$, il vient $|F| \leq \sum_{m < n} 2^m < 2^n$. Ainsi, $(H_{n,2^n}^1)$ est vrai.

Il est clair que si $(H_{n,2^n}^1)$ est vrai pour tout entier n , alors C est cohésif pour $(R_n)_{n \in \mathbb{N}}$. ■

Jockusch et Stephan,[103] ont étudié la puissance calculatoire des ensembles cohésifs pour certaines suites d'ensembles particulières. La puissance calculatoire de COH est mise en lumière par la notion d'arbre de décision.

Notation

Étant donné une suite $(R_n)_{n \in \mathbb{N}}$ et $\sigma \in 2^{<\mathbb{N}}$, on note R^σ l'intersection

$$\bigcap_{\sigma(n)=0} \mathbb{N} \setminus R_n \quad \bigcap_{\sigma(n)=1} R_n$$

On note $\mathcal{C}((R_n)_{n \in \mathbb{N}}) = \{P \in 2^{\mathbb{N}} : \forall \sigma \prec P \ |R^\sigma| = \infty\}$.

Intuitivement, lors de la construction d'un ensemble infini C cohésif pour la suite $(R_n)_{n \in \mathbb{N}}$, on est confronté pour chaque n à la décision de s'assurer que $C \subseteq^* R_n$ ou $C \subseteq^* \mathbb{N} \setminus R_n$. Si l'on note $R_n^0 = \mathbb{N} \setminus R_n$ et $R_n^1 = R_n$, alors R^σ représente les décisions prises pour chaque $n < |\sigma|$. Une décision est mauvaise si R^σ est un ensemble fini. Notons que si la suite $(R_n)_{n \in \mathbb{N}}$ est calculable, alors la classe $\mathcal{C}((R_n)_{n \in \mathbb{N}})$ est $\Pi_1^0(\emptyset')$.

Les deux propositions suivantes clarifient les liens entre la puissance requise pour calculer un chemin dans un arbre Δ_2^0 et celle permettant de calculer un ensemble cohésif pour une suite calculable d'ensembles. Elles ont été prouvées de manière indépendante par Brattka et Rakotoniaina [25] et Patey [172].

Proposition 4.4. Soit $(R_n)_{n \in \mathbb{N}}$ une suite calculable d'ensembles. Un ensemble calcule un ensemble cohésif pour $(R_n)_{n \in \mathbb{N}}$ si, et seulement si, son saut Turing calcule un membre de $\mathcal{C}((R_n)_{n \in \mathbb{N}})$. ★

PREUVE. Soit C un ensemble cohésif pour $(R_n)_{n \in \mathbb{N}}$. Soit $P \in 2^{\mathbb{N}}$ l'unique suite telle que pour tout $\sigma \prec P$, $C \subseteq^* R^\sigma$. L'ensemble P est $\Delta_2^0(C)$, et pour tout $\sigma \prec P$, R^σ est infini, donc $P \in \mathcal{C}((R_n)_{n \in \mathbb{N}})$.

Réciproquement, soit X un ensemble dont le saut Turing calcule un membre P de $\mathcal{C}((R_n)_{n \in \mathbb{N}})$. Par le lemme de limite de Shoenfield, il existe une fonction X -calculable $f : \mathbb{N}^2 \rightarrow 2$ telle que $\forall x \lim_y f(x, y) = P(x)$. Soit alors $(x_n)_{n \in \mathbb{N}}$ la suite strictement croissante d'entiers définie X -calculatoirement comme suit. Initialement, $x_0 = 0$; à chaque étape $s > 0$, chercher une chaîne $\sigma \in 2^{<\mathbb{N}}$ de longueur s et un entier $x_s \in R^\sigma$ plus grand que x_{s-1} tel que $f(x, x_s) = \sigma(x)$ pour tout $x < s$. Montrons que de tels σ et x_s existent toujours. En effet, il existe un seuil $N_0 \in \mathbb{N}$ tel que pour tout $x_s > N_0$ et tout $x < s$, $f(x, x_s) = P(x)$. En particulier, pour $\sigma = P \upharpoonright_s$, R^σ est infini, donc tout $x_s \in R^\sigma$ tel que $x_s > N_0$, satisfait la propriété désirée.

Soit $C = \{x_n : n \in \mathbb{N}\}$. Montrons que C est cohésif pour $(R_n)_{n \in \mathbb{N}}$. Pour tout $x \in \mathbb{N}$, il existe un seuil $N_1 \in \mathbb{N}$ tel que pour tout $s > N_1$, $f(x, x_s) = P(x)$. En particulier, $x_s \in R_x^{P(x)}$ pour tout $s > N_1$, donc $C \subseteq^* R_x^{P(x)}$. ■

Corollaire 4.5

Soit $(R_n)_{n \in \mathbb{N}}$ une suite calculable d'ensembles. Tout degré high calcule un ensemble cohésif pour $(R_n)_{n \in \mathbb{N}}$.

PREUVE. À l'aide de $\emptyset''$, on peut calculer inductivement $\sigma_0 \prec \sigma_1 \prec \dots$, en cherchant à partir de σ_n une valeur $i \in \{0, 1\}$ telle que $|R^{\sigma_n i}| = \infty$. En particulier, si $X' \geq_T \emptyset''$, alors X' calcule un membre de $\mathcal{C}((R_n)_{n \in \mathbb{N}})$, et donc X calcule un ensemble cohésif pour $(R_n)_{n \in \mathbb{N}}$. ■

Il est possible d'améliorer le corollaire précédent pour obtenir une caractérisation de la capacité à calculer un ensemble $(R_n)_{n \in \mathbb{N}}$ -cohésif pour toute suite calculable $(R_n)_{n \in \mathbb{N}}$.

Proposition 4.6. Une classe non vide $\mathcal{D} \subseteq 2^{\mathbb{N}}$ est $\Pi_1^0(\emptyset')$ si, et seulement si, il existe une suite calculable d'ensembles $(R_n)_{n \in \mathbb{N}}$ telle

que $\mathcal{C}((R_n)_{n \in \mathbb{N}}) = \mathcal{D}$. ★

PREUVE. Soit $(R_n)_{n \in \mathbb{N}}$ une suite calculable d'ensembles. Alors, $\mathcal{C}((R_n)_{n \in \mathbb{N}})$ est la classe $\Pi_1^0(\emptyset')$ donnée par $\{P \in 2^{\mathbb{N}} : \forall \sigma \prec P \ \forall x \ |R^\sigma| > x\}$. Notons que l'on peut obtenir un code de l'ensemble calculable R^σ uniformément en σ , et donc que $\emptyset'$ peut répondre uniformément à la question de savoir si $|R^\sigma| > x$. Cela rend bien $\Pi_1^0(\emptyset')$ la classe $\mathcal{C}((R_n)_{n \in \mathbb{N}})$.

Pour la direction inverse, soit $T \subseteq 2^{<\mathbb{N}}$ un arbre Δ_2^0 tel que $[T] = \mathcal{D}$. Par le lemme de limite de Shoenfield, il existe une fonction $f : 2^{<\mathbb{N}} \times \mathbb{N} \rightarrow 2$ calculable et telle que pour tout $\sigma \in 2^{<\mathbb{N}}$, $\lim_s f(\sigma, s) = T(\sigma)$. $\mathcal{D}$ étant non vide, T est un arbre infini, et l'on peut donc supposer que

- (1) pour tout $s \in \mathbb{N}$, l'ensemble $U_s = \{\sigma \in 2^s : f(\sigma, s) = 1\}$ est non vide.
- (2) pour tout $\sigma \prec \tau$ et s , si $g(\tau, s) = 1$, alors $g(\sigma, s) = 1$.

Nous allons construire une suite calculable d'ensembles $(R_n)_{n \in \mathbb{N}}$ telle que $\mathcal{C}((R_n)_{n \in \mathbb{N}}) = \mathcal{D}$ comme suit. À l'étape 0, $R_n = \emptyset$ pour tout $n \in \mathbb{N}$, et supposons que l'on ait déjà décidé $\mathcal{R}_n \upharpoonright_{k_s}$ pour tout $n \in \mathbb{N}$. À l'étape $s+1$, soit $p = |U_{s+1}|$. Nous allons ajouter des éléments $[n_s, k_s + p[$ à $R_0, \dots, R_s$ de telle sorte que pour toute chaîne $\sigma \in 2^{<\mathbb{N}}$ de longueur $s+1$, $R^\sigma \cap [k_s, k_s + p[\neq \emptyset$ ssi $\sigma \in U_{s+1}$. Pour cela, soit $\{\sigma_0, \dots, \sigma_{p-1}\} = U_{s+1}$. Pour tout $j \leq s$, ajouter à R_j l'ensemble $\{k_s + i : \sigma_i(j) = 1, i < p\}$. Soit $k_{s+1} = k_s + p$. Cela termine la construction. Notons que par (1), la suite $(k_s)_{s \in \mathbb{N}}$ est strictement croissante, et les ensembles $(R_n)_{n \in \mathbb{N}}$ sont donc définis pour des segments initiaux arbitrairement grands.

Montrons que R^σ est infini ssi $\sigma \prec P$ pour un $P \in \mathcal{D}$. Si σ n'est préfixe d'aucun membre de la classe $\mathcal{D}$, alors par compacité, il existe un t tel qu'aucun $\tau \succeq \sigma$ de longueur t n'est dans T . Soit $N_0 \in \mathbb{N}$ un seuil tel que pour tout $s > N_0$, σ ne possède aucune extension dans U_{s+1} . Alors, par construction, pour tout $s > N_0$, $R^\sigma \cap [k_s, k_{s+1}[= \emptyset$, et R^σ est donc fini.

Inversement, si $\sigma \prec P$ pour un $P \in \mathcal{D}$, alors par (2), il existe un seuil N_1 tel que pour tout $s > N_0$, σ possède une extension dans U_{s+1} . Alors, par construction, pour tout $s > N_0$, $R^\sigma \cap [k_s, k_{s+1}[\neq \emptyset$, et R^σ est donc infini. ■

Corollaire 4.7 (Jockusch et Stephan [103])

Soit $X \in 2^{\mathbb{N}}$. Les deux énoncés suivants sont équivalents.

- (1) Pour toute suite calculable d'ensembles $(R_n)_{n \in \mathbb{N}}$, X calcule un ensemble cohésif pour $(R_n)_{n \in \mathbb{N}}$.
- (2) Le saut X' est $PA(\emptyset')$.

PREUVE. Immédiat, par la proposition 4.4 et la proposition 4.6. ■

Instance universelle pour COH

Nous avons vu dans le chapitre 8 que la classe Π_1^0 des complétions de l'arithmétique de Peano est *universelle*, au sens où tout degré PA calcule un membre de toute classe Π_1^0 non vide (voir la remarque 8-6). De la même manière, COH possède une instance $(R_n)_{n \in \mathbb{N}}$ universelle, dans le sens où tout ensemble cohésif pour $(R_n)_{n \in \mathbb{N}}$ calcule un ensemble cohésif pour toute suite calculable. En effet, par la proposition 4.4 et la proposition 4.6, il existe une suite calculable d'ensembles telle que tout ensemble cohésif a un saut Turing de degré PA relativement à $\emptyset'$. Par le corollaire 4.7, tout ensemble dont le saut Turing est PA relativement à $\emptyset'$ calcule un ensemble cohésif pour toute suite calculable d'ensembles.

Terminons l'étude calculatoire du principe de cohésion par la proposition suivante qui sera très utile dans la section 4.2.

Proposition 4.8. Soit A un ensemble et soit $\mathcal{B} \subseteq \mathbb{N}^{\mathbb{N}}$ un fermé non vide dans l'espace de Baire n'ayant pas de membre calculable. Il existe un ensemble X tel que $X' \geq_T A$ et X ne calcule pas de membre de $\mathcal{B}$. ★

PREUVE. La preuve est similaire à celle de la proposition 4-10.2. Considérons la notion de forcing suivante : une *condition* est un couple (g, n) pour $n \in \mathbb{N}$, où g est une fonction de type $\{0, \dots, p\}^2 \rightarrow 2$ pour $p \in \mathbb{N}$. Une condition (h, m) *étend* (g, n) si $g \subseteq h$, $m \geq n$, et pour tout (x, y) dans $\text{dom } h \setminus \text{dom } g$, si $x < n$, alors $y = A(x)$. Ainsi, pour toute condition (g, n) , les n premières colonnes de g sont stabilisées sur la valeur de A . Tout filtre maximal F peut être interprété en une fonction $G = \bigcup_{(g, n) \in F} g$. Il est aisé de vérifier que $\text{dom } G = \mathbb{N} \times \mathbb{N}$, et que si F est suffisamment générique, alors pour tout $x \in \mathbb{N}$, $\lim_y G(x, y) = A(x)$. Ainsi, A est $\Delta_2^0(G)$.

Lemme 4.9. Pour toute condition $c = (g, n)$ et tout $e \in \mathbb{N}$, il existe une extension d de c forçant $\Phi_e^G \notin \mathcal{B}$. ★

PREUVE. Trois cas se présentent.

Cas 1. Il existe $(h, n) \leq (g, n)$ et $\sigma \in \mathbb{N}^{<\mathbb{N}}$ tel que $\Phi_e^h \upharpoonright_{|\sigma|} = \sigma$ avec $[\sigma] \cap \mathcal{B} = \emptyset$. Dans ce cas, (h, n) force $\Phi_e^G \notin \mathcal{B}$.

Cas 2. Il existe $(h, n) \leq (g, n)$ et un entier x tels que $\Phi_e^f(x) \uparrow$ pour tout $(f, m) \leq (h, n)$. Alors, (h, n) force $\Phi_e^G(x) \uparrow$, donc $\Phi_e^G \notin \mathcal{B}$.

Cas 3. Les cas 1 et 2 sont faux. On obtient alors une contradiction en calculant une suite $(h, n) \geq (h_0, n) \geq (h_1, n) \geq \dots$ de conditions telle que $\Phi_e^{h_{x+1}} \succeq \sigma_x$ avec $|\sigma_x| \geq x$. D'après la négation du cas 2, on peut toujours continuer une telle suite (notons à ce propos que la recherche d'extensions de conditions (h_x, n) est calculable, car il n'y a besoin de connaître que les n premiers bits de A , et que si (h_x, m) est une extension de (h_{x-1}, n) , alors (h_x, n) est aussi une extension valide de (h_{x-1}, n)). D'après la négation

du cas 1, on a $[\sigma_x] \cap \mathcal{B} \neq \emptyset$ pour tout x . Soit $P \in \bigcap_x [\sigma_x]$. Comme $\mathcal{B}$ est clos, $P \in \mathcal{B}$, ce qui contredit le fait que $\mathcal{B}$ n'ait pas de membre calculable. ■

Soit F un filtre suffisamment générique pour cette notion de forcing, et soit $G = \bigcup_{(g,n) \in F} g$. Par le lemme 4.9, G ne calcule pas de membre de $\mathcal{B}$, et par définition du forcing, A est $\Delta_2^0(G)$. ■

Corollaire 4.10

Soit $(R_n)_{n \in \mathbb{N}}$ une suite calculable et $\mathcal{B} \subseteq \mathbb{N}^{\mathbb{N}}$ un fermé non vide dans l'espace de Baire n'ayant pas de membre calculable. Il existe un ensemble C cohésif pour $(R_n)_{n \in \mathbb{N}}$ ne calculant pas de membre de $\mathcal{B}$.

PREUVE. Soit $A = \emptyset''$. Par la proposition 4.8, il existe un ensemble X tel que $X' \geq_T \emptyset''$ et X ne calcule pas de membre de $\mathcal{B}$. Par le corollaire 4.7, X calcule un ensemble cohésif pour $(R_n)_{n \in \mathbb{N}}$. ■

Exercice 4.11. (★) (Wang [232]). Montrer à l'aide du forcing de Mathias calculable que si C n'est pas Σ_1^0 , pour toute suite calculable $(R_n)_{n \in \mathbb{N}}$, il existe un ensemble cohésif D tel que C n'est pas $\Sigma_1^0(D)$. ◇

4.2. Faiblesse calculatoire de RT_2^2

Le principe de cohésion étant calculatoirement très faible, une grande partie des propriétés calculatoires des instances non calculables du principe infini des tiroirs se retrouve dans les instances calculables du théorème de Ramsey pour les paires.

Rappelons qu'un problème P préserve une propriété de faiblesse $\mathcal{W}$ si pour tout $Z \in \mathcal{W}$ et toute instance $X \leq_T Z$ de P , il existe une solution Y telle que $Z \oplus Y \in \mathcal{W}$ (voir la section 24-1.1). Nous pouvons définir une notion forte de préservation en retirant la contrainte d'effectivité sur X .

Définition 4.12. Soit $\mathcal{W}$ une propriété de faiblesse. Un problème Q préserve fortement $\mathcal{W}$ si pour tout $Z \in \mathcal{W}$, toute instance (arbitraire) X de Q admet une solution Y telle que $Z \oplus Y \in \mathcal{W}$. ◇

Nous avons la correspondance formelle suivante entre les instances calculables de RT_2^2 et les instances arbitraires de RT_2^1 .

Proposition 4.13. Soit $\mathcal{B} \subseteq \mathbb{N}^{\mathbb{N}}$ un fermé non vide dans l'espace de Baire, et soit $\mathcal{W} = \{Z : Z \text{ ne calcule pas de membre de } \mathcal{B}\}$. Alors, RT_2^2 préserve $\mathcal{W}$ si, et seulement si, RT_2^1 préserve fortement $\mathcal{W}$. ★

PREUVE. Supposons que RT_2^2 préserve $\mathcal{W}$. Soit Z un ensemble ne calculant pas de membre de $\mathcal{B}$ et soit $g : \mathbb{N} \rightarrow 2$ une instance de RT_2^1 . Par la proposition 4.8, il existe un ensemble X tel que g est $\Delta_2^0(X)$ et $X \oplus Z$ ne calcule pas de membre de $\mathcal{B}$. Par le lemme de limite de Shoenfield, il existe une fonction X -calculable $f : [\mathbb{N}]^2 \rightarrow 2$ telle que pour tout $x \in \mathbb{N}$, on a $\lim_y f(\{x, y\}) = g(x)$. Comme RT_2^2 préserve $\mathcal{W}$, il existe un ensemble infini H homogène pour f tel que $H \oplus X \oplus Z$ ne calcule pas de membre de $\mathcal{B}$. En particulier, H est homogène pour g et $H \oplus Z$ ne calcule pas de membre de $\mathcal{B}$, donc RT_2^1 préserve fortement $\mathcal{W}$.

Supposons que RT_2^1 préserve fortement $\mathcal{W}$. Soit Z un ensemble ne calculant pas de membre de $\mathcal{B}$, et soit $f \leq_T Z$ une instance de RT_2^2 . Soit $(R_n)_{n \in \mathbb{N}}$ la suite calculable définie par $R_x = \{y : f(\{x, y\}) = 1\}$. Par le corollaire 4.10, il existe un ensemble C cohésif pour $(R_n)_{n \in \mathbb{N}}$ tel que $C \oplus Z$ ne calcule pas de membre de $\mathcal{B}$. Comme C est cohésif pour $(R_n)_{n \in \mathbb{N}}$, pour tout $x \in C$, $\lim_{y \in C} f(\{x, y\})$ existe. Soit $g : C \rightarrow 2$ le coloriage défini par $g(x) = \lim_{y \in C} f(\{x, y\})$. Par la proposition 2.2 et sachant que RT_2^1 préserve fortement $\mathcal{W}$, il existe un ensemble infini $H \subseteq C$ homogène pour g tel que $H \oplus C \oplus Z$ ne calcule pas de membre de $\mathcal{B}$. Par le fait 4.1, $H \oplus C \oplus Z$ calcule un ensemble infini Y homogène pour f . En particulier, $Y \oplus Z$ ne calcule pas de membre de $\mathcal{B}$, donc RT_2^2 préserve $\mathcal{W}$. ■

Nous avons à présent les éléments nécessaires pour montrer le théorème de Seetapun, qui stipule que RT_2^2 , contrairement à RT_2^3 , n'implique pas ACA_0 dans RCA_0 .

Théorème 4.14 (Seetapun [195])

Pour tout ensemble C non calculable et pour tout coloriage $f : [\mathbb{N}]^2 \rightarrow 2$ calculable, il existe un ensemble infini H homogène pour f tel que $C \not\leq_T H$.

PREUVE. Soit C un ensemble non calculable, et soit $\mathcal{W} = \{Z : C \not\leq_T Z\}$. Par le théorème 3.8 relativisé, RT_2^1 préserve fortement $\mathcal{W}$, donc par la proposition 4.13, RT_2^2 préserve $\mathcal{W}$. Comme C n'est pas calculable, $\emptyset \in \mathcal{W}$, de sorte que par la définition de la préservation pour $Z = \emptyset$, nous obtenons l'énoncé du théorème 4.14. ■

La preuve du théorème de Seetapun se relativise bien, et permet donc de créer, pour tout ensemble non calculable C , un idéal Turing satisfaisant RT_2^2 mais ne contenant pas C .

Corollaire 4.15 (Seetapun [195])

On a : $\text{RCA}_0 + \text{RT}_2^2 \not\vdash \text{ACA}_0$.

PREUVE. Soit $\mathcal{W} = \{X : \emptyset' \not\leq_T X\}$. Par le théorème 4.14, RT_2^2 préserve $\mathcal{W}$; Comme ACA_0 ne préserve pas $\mathcal{W}$, le résultat escompté découle alors du corollaire 24-1.12. ■

Jockusch [99] a prouvé l'existence d'une instance calculable de RT_2^2 sans solution Δ_2^0 , montrant ainsi que WKL_0 n'implique pas RT_2^2 dans RCA_0 . Sachant que RT_2^2 est strictement plus faible que ACA_0 , la dernière question ouverte en lien avec les cinq grands systèmes des mathématiques à rebours consiste à savoir si RT_2^2 implique WKL_0 dans RCA_0 . Cette question, connue sous le nom d'*énigme de Seetapun*, est également restée ouverte plusieurs décennies avant que Liu [146] n'y réponde négativement, montrant ainsi l'existence d'un théorème naturel qui n'est pas linéairement ordonné avec le Club des cinq.

Théorème 4.16 (Liu [146])

Pour toute fonction calculable $f : [\mathbb{N}]^2 \rightarrow 2$, il existe un ensemble infini H homogène pour f qui n'est pas de degré PA.

PREUVE. Soit $\mathcal{W} = \{X : X \text{ n'est pas de degré PA}\}$. Par le théorème 3.24 relativisé, RT_2^1 préserve fortement $\mathcal{W}$, donc par la proposition 4.13, RT_2^2 préserve $\mathcal{W}$. Comme $\emptyset \in \mathcal{W}$, par la définition de la préservation pour $Z = \emptyset$, nous obtenons l'énoncé du théorème 4.16. ■

Corollaire 4.17

On a : $\text{RCA}_0 + \text{RT}_2^2 \not\vdash \text{WKL}_0$.

PREUVE. Soit $\mathcal{W} = \{X : X \text{ n'est pas de degré PA}\}$.

Par le théorème 4.16, RT_2^2 préserve $\mathcal{W}$; or WKL_0 ne préserve pas $\mathcal{W}$, donc par le corollaire 24-1.12, $\text{RCA}_0 + \text{RT}_2^2 \not\vdash \text{WKL}_0$. ■

Exercice 4.18. Montrer que pour tout ensemble C non Σ_1^0 et tout coloriage calculable $f : [\mathbb{N}]^2 \rightarrow 2$, il existe un ensemble infini H homogène pour f tel que C n'est pas $\Sigma_1^0(H)$. ◇

Nous avons vu avec la proposition 4.4 et la proposition 4.4 l'existence d'une instance calculable de COH dont toutes les solutions ont un saut Turing de degré PA relativement à $\emptyset'$. Par la proposition 4.3, $\text{COH} \leq_W \text{RT}_2^2$, donc il existe une instance calculable de RT_2^2 dont toutes les solutions ont un saut Turing de degré PA relativement à $\emptyset'$. Peut-on faire mieux ? Le théorème

suivant de Cholak, Jockusch et Slaman [33] affirme d'une certaine manière l'optimalité de cette borne en montrant qu'il est toujours possible d'obtenir une solution dont le saut Turing est borné par un degré PA relativement à $\emptyset'$.

Théorème 4.19 (Cholak, Jockusch et Slaman [33])

Pour tout coloriage calculable $f : [\mathbb{N}]^2 \rightarrow 2$ et pour tout ensemble P de degré PA relativement à $\emptyset'$, il existe un ensemble infini H homogène pour f tel que $H' \leq_T P$.

PREUVE. Par l'exercice 14-1.12 relativisé à $\emptyset'$, il existe un ensemble P_1 de degré PA relativement à $\emptyset'$, tel que P est de degré PA relativement à P_1 . Par le théorème d'inversion de saut de Friedberg (voir le corollaire 10-3.33), il existe un ensemble X tel que $X' \equiv_T P_1$. En particulier, P est de degré PA relativement à X' .

Soit $(R_n)_{n \in \mathbb{N}}$ la suite calculable définie par $R_x = \{y : f(\{x, y\}) = 1\}$. Par le corollaire 4.7, X calcule un ensemble C cohésif pour $(R_n)_{n \in \mathbb{N}}$. En particulier, pour tout $x \in C$, $\lim_{y \in C} f(\{x, y\})$ existe.

Soit $g : C \rightarrow 2$ le coloriage $\Delta_2^0(C)$ défini par $g(x) = \lim_{y \in C} f(\{x, y\})$. Par la proposition 2.2 et le théorème 3.40 relativisé à C , il existe un ensemble infini $H \subseteq C$ homogène pour g tel que $(H \oplus C)' \leq_T P \oplus C \leq_T P$. Par le fait 4.1, $H \oplus C$ calcule un ensemble infini Y homogène pour f . En particulier, $Y' \leq_T (H \oplus C)' \leq_T P$. ■

Corollaire 4.20 (Cholak, Jockusch et Slaman [33])

Pour tout coloriage calculable $f : [\mathbb{N}]^2 \rightarrow 2$, il existe un ensemble infini H homogène pour f de degré low_2 .

PREUVE. Par le théorème 8-4.3 relativisé à $\emptyset'$, il existe un ensemble P de degré PA relativement à $\emptyset'$ tel que $P' \leq_T \emptyset''$. Par le théorème 4.19, il existe un ensemble infini H homogène pour f tel que $H' \leq_T P$. En particulier, $H'' \leq_T P' \leq_T \emptyset''$, donc H est de degré low_2 . ■

Le théorème de Seetapun se généralise à toute la hiérarchie arithmétique.

Proposition 4.21 (Wang [232]). Pour tout $n \geq 1$, tout ensemble C non Δ_n^0 et tout coloriage calculable $f : [\mathbb{N}]^2 \rightarrow 2$, il existe un ensemble infini H homogène pour f tel que C n'est pas $\Delta_n^0(H)$. ★

PREUVE. Pour $n = 1$, il s'agit du théorème 4.14. Soit $n \geq 2$. Si C n'est pas Δ_n^0 , alors il n'est pas $\Delta_{n-1}^0(\emptyset')$. Par le théorème de base d'évitement de cône relativisé à $\emptyset'$ (voir le théorème 8-4.7), il existe un ensemble P

de degré PA relativement à $\emptyset'$ tel que C n'est pas $\Delta_{n-1}^0(P)$. Par le théorème 4.19, il existe un ensemble infini H homogène pour f tel que $H' \leq_T P$. En particulier, C n'est pas $\Delta_{n-1}^0(H')$, et donc pas non plus $\Delta_n^0(H)$. ■

4.3. Invariants combinatoires du théorème de Ramsey

Rappelons ici la notion de préservation forte de la définition 4.12. On dira qu'un problème P *préserve fortement un cône* si pour tout C non calculable, tout ensemble Z ne calculant pas C , toute instance arbitraire de P admet une solution X telle que $Z \oplus X \not\leq_T C$. La notion de préservation (non forte) de cône de la définition 24-1.13 est identique, mais pour des instances Z -calculables de P .

Nous avons une grande différence combinatoire du point de vue de la calculabilité entre RT_k^1 et RT_k^2 . Le premier préserve fortement un cône, mais pas le deuxième : nous avons vu en effet que l'on pouvait construire une instance Δ_2^0 de RT_2^2 dont toutes les solutions calculent $\emptyset'$. Wang a découvert qu'un léger affaiblissement du théorème de Ramsey pour les paires permettait de rétablir la préservation forte de cône.

Notation

On dénote par $\text{RT}_{k,\ell}^n$ l'énoncé « Pour tout coloriage $f : [\mathbb{N}]^n \rightarrow k$, il existe un ensemble infini $H \subseteq \mathbb{N}$ tel que $|f([H]^n)| \leq \ell$. »

En particulier, $\text{RT}_{k,1}^n$ est l'énoncé RT_k^n . Une grande partie de l'analyse calculatoire du théorème de Ramsey se généralise à $\text{RT}_{k,\ell}^n$. En particulier, pour tout $k \geq \ell + 1$, $\text{RT}_{k,\ell}^n$ implique $\text{RT}_{k+1,\ell}^n$ dans RCA_0 par un argument de daltonisme (voir la proposition 2.3). Lorsque $k \leq \ell$, $\text{RT}_{k,\ell}^n$ est trivial, et donc calculatoirement vrai. Ainsi, seul le cas $\text{RT}_{\ell+1,\ell}^n$ est vraiment intéressant du point de vue des mathématiques à rebours.

Wang [231] a prouvé que pour tout n , lorsque ℓ est suffisamment grand par rapport à n , l'énoncé $\forall k \text{RT}_{k,\ell}^n$ évite fortement un cône. Cholak et Patey [34] ont étudié la borne exacte ℓ en fonction de n pour laquelle $\forall k \text{RT}_{k,\ell}^n$ évite fortement un cône, et celle-ci coïncide de manière surprenante avec les nombres de Catalan, très connus en combinatoire.

Définition 4.22. La *suite de Catalan* est définie de manière inductive comme suit :

$$C_0 = 1 \quad \text{et} \quad C_{n+1} = \sum_{i=0}^n C_i C_{n-i}.$$

◇

Les premières valeurs de la suite de Catalan sont 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16 796, 58 786, ... La suite de Catalan admet de nombreuses caractérisations combinatoires.

Exemple 4.23. Un *parenthésage* est une chaîne à valeurs dans $\{ (,) \}$. Il est dit *valide* si le nombre de parenthèses ouvrantes est égal au nombre de parenthèses fermantes, et si pour tout préfixe, le nombre de parenthèses fermantes n'est jamais plus grand que celui de parenthèses ouvrantes. Ainsi, $((()())()$ est un parenthésage valide, tandis que $)()$ et $((()))()$ ne le sont pas. Notons qu'un parenthésage valide est forcément de longueur paire. Soit C_n le nombre de parenthésages valides de longueur $2n$. Par exemple, il n'existe qu'un parenthésage valide de longueur 0, à savoir la chaîne vide, donc $C_0 = 1$. Il existe également un unique parenthésage valide de longueur 2, qui est $()$, donc $C_1 = 1$. En revanche, il existe deux parenthésages valides de longueur 4, qui sont $()()$ et $((()))$.

C'est avec l'élégante caractérisation suivante que nous concluons notre analyse du théorème de Ramsey du point de vue de la calculabilité.

Théorème 4.24 (Wang [231], Cholak et Patey [34])

Pour tout $n \geq 1$, $\forall k \text{RT}_{k,\ell}^n$ préserve fortement 1 cône si, et seulement si, $\ell \geq C_n$. Ainsi, $\forall k \text{RT}_{k,\ell}^{n+1}$ préserve un cône si, et seulement si, $\ell \geq C_n$.

Quatrième partie

Hypercalculabilité

Chapitre 26

Introduction

Il n'est pas aussi aisé de définir l'hypercalculabilité *per se* que nous ne l'avons fait pour les trois premières parties de cet ouvrage, à savoir la calculabilité classique, l'aléatoire algorithmique ou les mathématiques à rebours. En effet, l'hypercalculabilité ne manipule pas des objets mathématiques à correspondance épistémologique immédiate, comme c'est le cas pour les fonctions calculables ou les suites aléatoires. Ce domaine de recherche est avant tout un pont, entre la calculabilité classique d'une part, et la théorie des ensembles et la théorie descriptive des ensembles d'autre part. Pour ces raisons, l'hypercalculabilité se définit plus facilement en référence à ces théories, en adoptant soit une approche *descendante*, partant du point de vue de la théorie descriptive des ensembles et en considérant l'hypercalculabilité comme une effectivisation de ses concepts, soit une approche *ascendante*, en considérant l'hypercalculabilité comme une généralisation de la calculabilité classique pour laquelle on s'autorise *une infinité* d'étapes de calcul, via la notion *d'ordinaux*.

Pour le théoricien des ensembles, l'hypercalculabilité peut d'abord être vue comme l'étude des premiers niveaux de la hiérarchie de l'*univers des constructibles*. La constructibilité est un champ d'étude initié par Gödel, qui à partir d'un modèle de ZF, a montré comment construire un sous-modèle de ZF satisfaisant à la fois l'axiome du choix et l'hypothèse du continu. Le modèle de Gödel est l'univers des constructibles : les ensembles que l'on peut construire récursivement le long des ordinaux (que nous définirons formellement bientôt). Il s'agit toutefois d'un point de vue que nous ne développerons pas davantage dans cet ouvrage, et le lecteur curieux d'en apprendre plus à leur sujet pourra se reporter à [35] ou encore [192].

L'hypercalculabilité peut également être vue comme une version effective de la théorie descriptive des ensembles. Cette branche des mathématiques classe les ensembles en fonction de leur complexité définitionnelle. Elle est née de la continuation des travaux de Borel, Baire et Lebesgue, par Nikolai Luzin et Mikhail Souslin [219] au début du XX^e siècle. Nous reparlerons des aspects historiques de manière détaillée dans la section 29-1. La théorie descriptive des ensembles n'est au départ pas liée à la calculabilité, et ce sera essentiellement Stephen Cole Kleene qui en développera les aspects effectifs [114] [115] [117] [116] entre les années 1938 et 1955. Ce sont surtout les travaux de Kleene et de ses collègues que nous présenterons dans cette partie.

Pour le théoricien de la calculabilité, l'hypercalculabilité est une généralisation des concepts de la calculabilité classique à des calculs et des constructions en temps infini, via le concept d'ordinal. Cela permet d'étendre la notion de calcul et de ses intuitions à d'autres notions de définissabilité. Cet ouvrage traitant de calculabilité, il est naturel d'aborder l'hypercalculabilité principalement sous ce prisme.

Plus concrètement, de nombreuses définitions de calculabilité le long des entiers peuvent être généralisées à des définitions le long des ordinaux. Par exemple, la hiérarchie arithmétique peut être étendue à des niveaux ordinaux, pour former la hiérarchie hyperarithmétique. Nous verrons toute une correspondance entre la calculabilité classique et l'hypercalculabilité, où les ensembles hyperarithmétiques correspondront aux ensembles calculables, les ensembles Π_1^0 aux ensembles calculatoirement énumérables, et le $\mathcal{O}$ de Kleene au saut Turing. Ces notions seront toutes définies dans cette partie.

1. Motivations

Nous commençons par présenter un exemple, qui part de notions de calculabilité classique vues jusqu'ici, afin d'en montrer les limites ainsi que la nécessité de l'hypercalculabilité pour traiter certaines questions.

Souvenons-nous de la proposition 8-3.6, qui stipule que si une classe Π_1^0 contient exactement un élément X , alors X est calculable. Nous appellerons de tels éléments des *singletons* Π_1^0 . Qu'en est-il des singletons Π_2^0 ? La preuve de la proposition 19-1.2 montre que tout ensemble $\emptyset'$ -calculable est un singleton Π_2^0 . Il serait tentant de vouloir généraliser la situation des singletons Π_1^0 en conjecturant que les singletons Π_2^0 sont exactement les ensembles $\emptyset'$ -calculables, mais la situation est dans ce cas bien plus complexe.

Afin de voir cela, nous commençons par présenter une construction alternative pour montrer que $\emptyset'$ est un singleton Π_2^0 . Nous utilisons pour cela (2)

de l'exemple 17-3.7 : la classe $\mathcal{S}$ des ensembles qui sont des sauts Turing d'autres ensembles est Π_2^0 . L'exemple utilisait l'existence d'une fonctionnelle Φ_e telle que $\Phi_e(X)$ est totale pour tout X et telle que $\Phi_e(X') = X$ pour tout X . En utilisant Φ_e , nous avons défini $\mathcal{S}$ comme suit :

$$\bigcap_n \{X : X(n) = 1 \wedge \Phi_n(\Phi_e(X), n) \downarrow\} \cup \{X : X(n) = 0 \wedge \Phi_n(\Phi_e(X), n) \uparrow\}.$$

La classe $\mathcal{S}$ est bien une classe Π_2^0 , qui peut être utilisée pour montrer sans peine que l'arrêt $\emptyset'$ est un singleton Π_2^0 : c'est l'unique élément de la classe $\mathcal{S} \cap \{X : \Phi_e(X) = \emptyset\}$. On vérifie aisément que $\{X : \Phi_e(X) = \emptyset\}$ est une classe Π_1^0 , qui est donc d'après le lemme 17-3.12 également Π_2^0 . L'intersection des deux classes est dès lors tout aussi bien une classe Π_2^0 . Par ailleurs, il est clair d'après les définitions respectives de ces classes que l'intersection contient uniquement l'élément $\emptyset'$.

On peut à présent continuer pour montrer que $\emptyset''$ est lui aussi un singleton Π_2^0 . Soit $\mathcal{S}_1$ la classe Π_2^0 contenant exactement $\emptyset'$. Alors, on peut définir $\mathcal{S}_2$ la classe Π_2^0 contenant exactement $\emptyset''$ par

$$\mathcal{S}_2 = \mathcal{S} \cap \{X : \Phi_e(X) \in \mathcal{S}_1\}.$$

On voit aisément comment continuer avec $\mathcal{S}_{n+1} = \mathcal{S} \cap \{X : \Phi_e(X) \in \mathcal{S}_n\}$, faisant de chaque ensemble $\emptyset^{(n)}$ un singleton Π_2^0 .

Les singletons Π_2^0 peuvent donc être de complexité arithmétique arbitraire. Remarquons avant de continuer que l'inverse n'est déjà plus vrai : il existe des ensembles $\emptyset''$ -calculables qui ne sont pas des singletons Π_2^0 . Il suffit par exemple de remarquer que tout singleton Π_2^0 est un $\emptyset'$ -test de Martin-Löf, et de considérer un ensemble 2-aléatoire $\emptyset''$ -calculable. Les ensembles arithmétiques ne sont donc pas tous des singletons Π_2^0 . De leur côté, les singletons Π_2^0 sont-ils au moins tous arithmétiques ? Là encore, la situation n'est pas si simple...

Considérons l'ensemble $\emptyset^{(\omega)} = \bigoplus_n \emptyset^{(n)} : \langle n, m \rangle \in \emptyset^{(\omega)} \text{ ssi } n \in \emptyset^{(m)}$. L'ensemble $\emptyset^{(\omega)}$ ne peut pas être arithmétique, c'est-à-dire Σ_n^0 pour un certain n . En effet, si $\emptyset^{(\omega)}$ est Σ_n^0 , alors $\emptyset^{(n)} \geq_T \emptyset^{(\omega)}$. Par ailleurs, $\emptyset^{(\omega)} \geq_T \emptyset^{(n+1)}$, et donc $\emptyset^{(n)} \geq_T \emptyset^{(n+1)}$, ce qui est une contradiction. Si $\emptyset^{(\omega)}$ n'est pas arithmétique, il n'en est pas moins singleton Π_2^0 via la classe suivante :

$$\mathcal{S}_\omega = \left\{ \bigoplus_n X_n : \forall n \, X_n \in \mathcal{S}_n \right\}.$$

On peut bien sûr continuer avec les ensembles

$$\mathcal{S}_{\omega+n+1} = \mathcal{S} \cap \{X : \Phi_e(X) \in \mathcal{S}_{\omega+n}\}.$$

Jusqu'où cela va-t-il ? Cette question nous amène à l'étude et à la définition des ensembles dit *hyperarithmétiques*, qui comme nous le verrons sont exactement les ensembles Turing calculables par un singleton Π_2^0 .

2. Panorama de l'hypercalculabilité

La hiérarchie arithmétique admet plusieurs caractérisations. Elle est originellement définie en termes d'alternance de quantificateurs du premier ordre. Par le fameux théorème de Post, les ensembles Δ_n^0 et Σ_n^0 sont respectivement les ensembles calculables et calculatoirement énumérables à l'aide de l'oracle $\emptyset^{(n-1)}$. La hiérarchie arithmétique peut être étendue via cette correspondance, à l'aide d'itérations transfinies du saut Turing.

Itérations du saut Turing. Le saut Turing peut être vu comme une opération sur les ensembles, définie par $X \mapsto X' = \{e : \Phi_e^X(e) \downarrow\}$. Les ensembles $\emptyset^{(n)}$ sont définis en itérant cette opération le long des entiers naturels, à savoir

$$\emptyset^{(0)} = \emptyset \quad \text{et} \quad \emptyset^{(n+1)} = (\emptyset^{(n)})'.$$

Il est cependant possible d'aller au-delà des itérations finies du saut Turing, en définissant son ω -itération $\emptyset^{(\omega)} = \bigoplus_n \emptyset^{(n)}$, puis son $(\omega + 1)$ -itération $\emptyset^{(\omega+1)} = (\emptyset^{(\omega)})'$, et ainsi de suite.

Ordinaux. L'itération du saut Turing au-delà des nombres finis demande d'étendre la notion d'entier naturel à des nombres transfinis, appelés *ordinaux*. Il s'agit d'un concept introduit au départ par Cantor, et qui peut être vu comme une extension de la notion d'entier naturel pour laquelle le principe suivant reste vrai : tout sous-ensemble non vide d'ordinaux admet un plus petit élément. Les opérations arithmétiques standard se généralisent également aux ordinaux. Nous noterons donc $0, 1, 2, \dots$ les premiers ordinaux et ω le plus petit ordinal infini. De manière générale, on notera les ordinaux avec des lettres grecques : $\alpha, \beta, \gamma, \dots$

Les ordinaux peuvent être arbitrairement complexes, et sont pour la plupart indénombrables. On se restreindra donc, pour les besoins de l'hypercalculabilité, à une sous-collection d'ordinaux que l'on peut représenter à l'aide d'entiers naturels et de la calculabilité. Kleene a défini un ensemble $\mathcal{O} \subseteq \mathbb{N}$ de codes d'ordinaux, en associant à chaque entier $n \in \mathcal{O}$ un ordinal $|n|$. Les ordinaux admettant un code dans $\mathcal{O}$ sont appelés *constructibles*, et forment un segment initial. On notera ω_1^{ck} le plus petit ordinal non constructible. Ces notions seront étudiées dans le chapitre 27.

Ensembles hyperarithmétiques. Les sauts Turing peuvent être itérés le long des ordinaux constructibles. On peut étendre la hiérarchie arithmétique à des niveaux ordinaux, en définissant pour tout ordinal constructible α la classe Σ_α^0 des ensembles calculatoirement énumérables en $\emptyset^{(\alpha)}$. On obtient alors la *hiérarchie hyperarithmétique*. La correspondance entre calculabilité et définissabilité peut être étendue aux ensembles hyperarithmétiques, à l'aide de formules infinitaires.

Les ensembles hyperarithmétiques forment une classe relativement naturelle, et admettent notamment de nombreuses caractérisations. Une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ est un *modulus* d'un ensemble X si toute fonction g dominant f calcule X . Les ensembles hyperarithmétiques sont exactement les ensembles admettant un modulus, ou encore les ensembles calculables par un singleton Π_2^0 dans $2^{\mathbb{N}}$. Ces notions seront étudiées dans le chapitre 28.

Hiérarchie borélienne effective. De la même manière que la hiérarchie hyperarithmétique classe les ensembles d'entiers naturels en fonction de leur complexité définitionnelle, les classes dans l'espace de Cantor peuvent être classifiées en hiérarchies effectives. Nous avons défini les classes Σ_n^0 et Π_n^0 de réels pour tout $n \in \mathbb{N}$. Cette hiérarchie peut être étendue le long des ordinaux constructibles. On obtient la *hiérarchie borélienne effective*, ou les *classes hyperarithmétiques* qui seront présentées dans le chapitre 28.

Ensembles et classes Σ_1^1 et Π_1^1 . Au lieu d'itérer les constructions le long des ordinaux constructibles, on peut étendre la hiérarchie arithmétique et la hiérarchie borélienne effective en considérant les formules du second ordre. Un ensemble ou une classe est Σ_1^1 s'il est définissable par formule de la forme $\exists XF(X)$, où F est une formule arithmétique. De la même manière, un ensemble ou une classe est Π_1^1 s'il est définissable par formule de la forme $\forall XF(X)$. Les ensembles et classes Δ_1^1 sont ceux à la fois Σ_1^1 et Π_1^1 . L'usage du second ordre donne une puissance descriptive bien supérieure, et tous les ensembles et classes hyperarithmétiques sont en particulier Δ_1^1 . Kleene a montré que les ensembles et classes Δ_1^1 sont exactement les hyperarithmétiques, donnant ainsi une nouvelle définition purement définitionnelle des hyperarithmétiques sans avoir recours aux ordinaux. Ces notions seront abordées dans la section 29-1.

3. Correspondance avec la calculabilité classique

Les constructions présentées dans la section précédente sont principalement définitionnelles, et ne donnent pas un éclairage sur la nature calculatoire des ensembles au-delà des arithmétiques. Il existe cependant une correspondance informelle entre certains concepts de calculabilité classique, et ceux d'hypercalculabilité. Tout comme la notion d'ensemble calculable induit la réduction Turing par relativisation, on définit la *réduction hyperarithmétique* par $X \leq_h Y$ si X est hyperarithmétique relativement à Y . La correspondance entre calculabilité classique et hypercalculabilité est esquissée avec la figure 3.1.

Cette correspondance peut paraître surprenante sous certains aspects, notamment le fait que les ensembles hypercalculatoirement énumérables soient les ensembles Π_1^1 .

Calculabilité classique	Hypercalculabilité
Calculable	Hyperarithmétique
Ensemble c. e.	Ensemble Π_1^1
Problème de l'arrêt	$\mathcal{O}$ de Kleene
Classe Π_1^0	Classe Σ_1^1
Réduction Turing	Réduction hyperarithmétique

FIGURE 3.1 – *Correspondance calculabilité classique/hypercalculabilité*

Voici quelques éléments permettant de mieux comprendre la figure 3.1.

$\mathcal{O}$ de Kleene. Une propriété caractéristique du problème de l'arrêt est sa Σ_1^0 -complétude pour la réduction many-one. Autrement dit, le problème de l'arrêt est une forme d'ensemble calculatoirement énumérable universel. De la même manière, le $\mathcal{O}$ de Kleene est Π_1^1 -complet pour la réduction many-one.

Tout comme le problème de l'arrêt $\{e : \Phi_e(e) \downarrow\}$ induit par sa relativisation un opérateur $X \mapsto \{e : \Phi_e^X(e) \downarrow\}$ appelé saut Turing, on peut définir la notion d'ordinal X -constructible, ce qui induit un opérateur

$$X \mapsto \mathcal{O}^X = \{e : e \text{ code pour un ordinal } X\text{-constructible}\}.$$

Ensembles Π_1^1 . Si l'on dénote par $\mathcal{O}_{<\alpha}$ la restriction du $\mathcal{O}$ de Kleene aux codes d'ordinaux inférieurs à α , pour un ordinal constructible $\alpha < \omega_1^{ck}$, l'ensemble $\mathcal{O}_{<\alpha}$ est un ensemble $\Sigma_{\alpha+1}^0$, dès lors hyperarithmétique. On peut donc approximer le $\mathcal{O}$ de Kleene par des ensembles hyperarithmétiques croissants $\mathcal{O}_{<0}, \mathcal{O}_{<1}, \mathcal{O}_{<2}, \dots$ le long des ordinaux constructibles, avec

$$\mathcal{O} = \bigcup_{\alpha < \omega_1^{ck}} \mathcal{O}_{<\alpha}.$$

Le $\mathcal{O}$ de Kleene étant Π_1^1 complet pour la réduction many-one, ses approximations induisent des approximations de tout ensemble Π_1^1 à l'aide d'ensembles hyperarithmétiques le long des ordinaux constructibles. Les ensembles Π_1^1 sont donc les ensembles que l'on peut hypercalculatoirement énumérer.

Classes Σ_1^1 . De même qu'une classe Π_1^0 peut être vue comme l'ensemble des chemins d'un arbre calculable dans $2^{<\mathbb{N}}$, une classe Σ_1^1 $\mathcal{B} \subseteq 2^{\mathbb{N}}$ peut être codée par un arbre $T \subseteq \mathbb{N}^{<\mathbb{N}}$, de telle sorte que

$$\mathcal{B} = \{X : \exists f \in \mathbb{N}^{\mathbb{N}} \ X \oplus f \in [T]\}.$$

Nous prouverons plusieurs théorèmes de base pour les classes Σ_1^1 qui ad-

mettent une correspondance directe avec les théorèmes de base pour les classes Π_1^0 . Par exemple, toute classe Σ_1^1 non vide admet un membre *hyperlow*, c'est-à-dire un ensemble X tel que $\mathcal{O}^X \equiv_T \mathcal{O}$. De plus, si Y est un ensemble non hyperarithmétique, alors toute classe Σ_1^1 non vide contient un élément X tel que $Y \not\leq_h X$. Ces deux théorèmes de base sont des analogues des théorèmes de base low et d'évitement de cône pour les classes Π_1^0 .

Chapitre 27

Nombres transfinis

Nous allons aborder dans ce chapitre un concept fondamental en mathématiques permettant de réaliser des constructions itératives allant au-delà des entiers naturels, au sens où les itérations ne se contenteront pas d'atteindre toute étape finie : elles atteindront leur limite, et continueront au-delà. Afin d'illustrer ce mécanisme, nous allons montrer dans quelle mesure le saut Turing peut être itéré à l'infini et au-delà. Nous nous trouverons alors confrontés à des problématiques de gestion des étapes infinies, qui trouveront leur résolution à travers la notion d'ordinal, véritable révolution conceptuelle initiée par Cantor en 1883.

1. Motivation : itérations calculables du saut

Revenons sur la construction des itérations du saut Turing de $\emptyset$. On dispose sur les ensembles d'une opération $X \mapsto X' = \{e : \Phi_e^X(e) \downarrow\}$, que l'on va itérer à partir de l'ensemble vide $\emptyset$. Initialement, $\emptyset^{(0)} = \emptyset$. Puis, on définit à l'étape 1 le premier saut Turing $\emptyset^{(1)} = (\emptyset^{(0)})'$, puis à l'étape 2 le double saut $\emptyset^{(2)} = (\emptyset^{(1)})'$, et ainsi de suite, en définissant à l'étape $n + 1$ l'ensemble $\emptyset^{(n+1)} = (\emptyset^{(n)})'$. Nous avons également vu dans la section 26-1 que la construction ne s'arrêtait pas aux itérations finies. Il est possible de définir l' ω -saut à l'étape dite « ω » par $\emptyset^{(\omega)} = \bigoplus_n \emptyset^{(n)}$. Il est clair que $\emptyset^{(\omega)} >_m \emptyset^{(n)}$ pour tout n , où $\leq_m$ désigne la réduction many-one (voir la section 5-4). L'ensemble $\emptyset^{(\omega)}$ contient la réponse à toutes les questions arithmétiques, et comme nous l'avons vu, il n'est lui-même pas arithmétique. On peut à partir de là continuer à itérer le saut : pour tout n on définit

$$\emptyset^{(\omega+n+1)} = (\emptyset^{(\omega+n)})', \quad \text{puis} \quad \emptyset^{(\omega+\omega)} = \bigoplus_n \emptyset^{(\omega+n)}.$$

Là encore, on aura $\emptyset^{(\omega+\omega)} >_m \emptyset^{(\omega+n)}$ pour tout n .

Arrêtons-nous afin d'identifier, pour le moment de manière informelle, les étapes fondamentales de ce processus.

- (1) *Étape initiale.* Il s'agit du cas de base : le processus n'a pas encore été itéré. Dans le cas du saut Turing, l'itération zéro $\emptyset^{(0)}$ est simplement l'ensemble $\emptyset$ lui-même.
- (2) *Étape successeur.* Supposons que l'on ait itéré le processus un nombre α d'étapes, pour obtenir dans le cas du saut Turing l'ensemble $\emptyset^{(\alpha)}$, il est toujours possible de l'itérer une fois de plus. On passe donc à l'étape successeur $\alpha + 1$. Dans le cas du saut Turing, on définit $\emptyset^{(\alpha+1)} = (\emptyset^{(\alpha)})'$.
- (3) *Étape limite.* Étant donné des itérations $\emptyset^{(\alpha_n)}$ du saut pour $n \in \mathbb{N}$ et avec $\emptyset^{(\alpha_n)} <_m \emptyset^{(\alpha_{n+1})}$, on peut définir son itération *limite*

$$\emptyset^{(\gamma)} = \bigoplus_n \emptyset^{(\alpha_n)}.$$

On appelle ce genre de processus une itération *transfinie*, caractérisée par des étapes successeurs et des étapes limites. Il s'agit d'une extension des constructions par récurrence sur les entiers naturels, auxquelles l'ajout d'étapes limites permet d'aller au-delà du cas fini. Ce nouveau type d'étape mérite notre attention. Le premier cas limite $\emptyset^{(\omega)} = \bigoplus_n \emptyset^{(n)}$ est simple, mais il est important de remarquer qu'à degré Turing près — ou même à degré many-one près — il y aurait bien d'autres manières de définir $\emptyset^{(\omega)}$, par exemple en prenant $\bigoplus_n \emptyset^{(2^n)}$. À mesure que l'on avance dans les itérations transfinies, il n'y a rapidement plus nécessairement de choix canonique qui s'impose pour sélectionner une quantité dénombrable d'éléments permettant d'itérer le saut à ces étapes limites. On utilise alors pour cela un système de codage imaginé par Kleene et qui va nous permettre de procéder proprement.

1.1. $\mathcal{O}$ de Kleene

Afin de formaliser la construction transfinie sur les sauts Turing, nous allons commencer par nommer (ou coder) les étapes, en leur attribuant à chacune un entier naturel. Si l'on procède naïvement et associe l'entier 0 pour l'étape initiale, et l'entier $n + 1$ pour l'étape successeur de l'étape n , tous les entiers naturels seront utilisés pour nommer les premières étapes successeurs, et il n'en restera plus pour nommer les étapes limites. Nous allons donc avoir recours à un codage plus élaboré basé sur le théorème fondamental de l'arithmétique, qui affirme l'unicité de la décomposition en facteurs premiers.

Définition 1.1 (Kleene [114]). On définit inductivement un ensemble de notations $\mathcal{O} \subseteq \mathbb{N}$ avec un ordre partiel $<_o$ sur ses éléments :

- (1) $1 \in \mathcal{O}$;
- (2) si $a \in \mathcal{O}$ alors $2^a \in \mathcal{O}$. On a alors $a <_o 2^a$, et $b <_o a$ implique $b <_o 2^a$ pour tout b ;
- (3) si $\Phi_e : \mathbb{N} \rightarrow \mathbb{N}$ est une fonction totale calculable telle que

$$\forall n \ \Phi_e(n) \in \mathcal{O}, \quad \text{avec} \ \Phi_e(n) <_o \Phi_e(n+1) \text{ pour tout } n,$$

alors $3 \times 5^e \in \mathcal{O}$. De plus, pour chaque $b \in \mathcal{O}$ tel que $b <_o \Phi_e(n)$ pour un certain n , on a $b <_o 3 \times 5^e$. ◇

Formellement, l'ordre partiel $<_o \subseteq \mathbb{N} \times \mathbb{N}$ peut être défini comme le plus petit ensemble contenant $1 <_o 2$ et clos par les opérations (2) et (3). L'ensemble $\mathcal{O}$ est alors le domaine de cet ordre partiel.

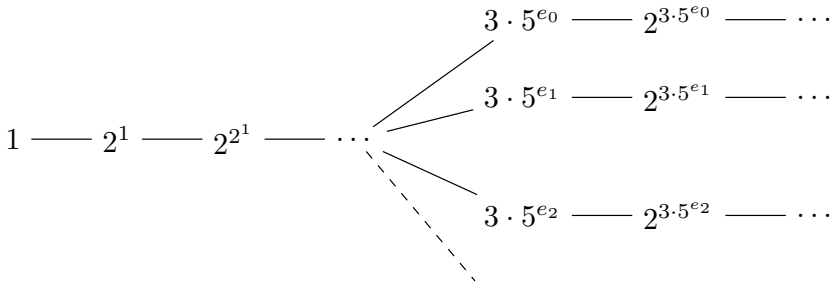
Intuitivement, le cas (1) définit le code pour l'étape 0, tandis que les cas (2) et (3) correspondent respectivement au cas successeur et au cas limite. Les détails de la définition sont principalement conventionnels, l'essentiel étant d'obtenir un système de codage pour les étapes successeurs et limites qui ne se « morde pas la queue » : les entiers 2, 3 et 5 utilisés sont tous des nombres premiers, et le théorème fondamental de l'arithmétique — l'unicité de la décomposition des nombres en facteurs premiers — nous garantit que la construction d'un élément de $\mathcal{O}$ à une étape limite ou successeur ne sera jamais égale à un code précédemment construit. En particulier, dans le cas (3), l'utilisation de 3×5^e plutôt que 3^e est simplement là pour éviter de retomber sur 1 dans le cas $e = 0$. On aurait aussi bien pu noter cette étape 3^{e+1} .

On notera que dans le cas limite — le cas (3) — seules les limites de certaines suites croissantes d'étapes admettent un codage par un entier, à savoir les limites de suites calculables d'étapes. Cette restriction est cependant nécessaire, dans la mesure où il n'y a qu'une quantité dénombrable d'entiers, tandis qu'il existe une quantité indénombrable de suites. La figure 1.2 illustre l'ordre $<_o$ sur $\mathcal{O}$.

Notation

On écrira parfois $a \leq_o b$ pour des éléments $a, b \in \mathcal{O}$ pour signifier $a <_o b$ ou $a = b$. On parlera parfois du « $\mathcal{O}$ de Kleene » pour parler de l'ensemble défini ci-dessus.

Le $\mathcal{O}$ de Kleene peut être à présent utilisé pour mener proprement les itérations transfinies du saut, effectuées traditionnellement avec la lettre H .

FIGURE 1.2 – *Ordre partiel $<_o$ sur $\mathcal{O}$*

Définition 1.3. On définit les itérations transfinies H_a du saut pour les éléments $a \in \mathcal{O}$:

- (1) $H_1 = \emptyset$;
- (2) $H_{2^a} = H'_a$;
- (3) $H_{3 \times 5^e} = \bigoplus_n H_{\Phi_e(n)}$.



Notons l'avantage de l'utilisation des codes : étant donné $a \in \mathcal{O}$, on peut énumérer tous les éléments $b \in \mathcal{O}$ tels que $b <_o a$. En effet, si a est de la forme 2^b , on énumère b , et l'on relance récursivement l'énumération sur b ; et, si a est de la forme 3×5^e , on énumère $\Phi_e(n)$ pour tout n , et l'on relance petit à petit récursivement l'énumération sur chaque $\Phi_e(n)$.

De ce fait, étant donné un ensemble H_a , sous réserve de la connaissance du code a , on peut alors énumérer tous les couples (H_b, b) pour $b <_o a$, en ajoutant à notre énumération l'ensemble H_b correspondant, que l'on peut retrouver de manière calculable à partir de H_a et de a . On a en particulier $H_b <_T H_a$ pour $b <_o a$. On pourrait en fait montrer avec un peu plus de travail que l'on a $H_b <_m H_a$ pour $b <_o a$. Cela sera une conséquence du corollaire 28-1.14.

Remarque

Toute itération finie du saut Turing est représentée de manière unique à l'aide de la définition 1.3. En effet, pour tout n , $\emptyset^{(n)} = H_a$ avec

$$a = \underbrace{2^{2^{\cdot^{2^1}}}}_{n \text{ copies de } 2}.$$

En revanche, il existe plusieurs codes de $\emptyset^{(\omega)}$, car une même fonction calculable possède une infinité de codes Turing. Ainsi, si $\Phi_e = \Phi_i$ et que $3 \times 5^e \in \mathcal{O}$, alors $3 \times 5^i \in \mathcal{O}$, et $H_{3 \times 5^e} = H_{3 \times 5^i}$. Comme expliqué précédemment, si l'on considère les itérations de la définition 1.3 du

point de vue des degrés Turing, il existe également des fonctions Φ_e et Φ_i qui ne sont pas équivalentes, mais telles que $H_{3 \times 5^e} \equiv_T H_{3 \times 5^i}$. Par exemple, si $3 \times 5^e \in \mathcal{O}$ et $\Phi_i(n) = \Phi_e(n+1)$ pour tout n , alors $3 \times 5^i \in \mathcal{O}$, et $H_{3 \times 5^e} \equiv_T H_{3 \times 5^i}$.

2. Ordinaux

Nous allons voir que les éléments de $\mathcal{O}$ sont utilisés en quelque sorte comme une extension des entiers, que l'on appelle *ordinaux* ou *nombre transfinis*. Les premiers codes de $\mathcal{O}$ sont de fait linéairement ordonnés comme les entiers : $1 < 2 < 2^2 < 2^{2^2} < \dots$.

Pour cette raison, on dira aussi que $1 \in \mathcal{O}$ est un code de 0, que $2 \in \mathcal{O}$ est un code de 1, etc., et l'on écrira $|1| = 0$, $|2| = 1$, et de manière générale si $|a| = n$, alors $|2^a| = n + 1$. Les notations $0, 1, 2, \dots$ et n désignent les *ordinaux finis* et sont utilisées pour éviter toute confusion avec leur codes correspondants.

Considérons à présent un code de fonction calculable e tel que $\Phi_e(n)$ renvoie $a_n \in \mathcal{O}$ pour tout n et où $|a_n| = n$. Alors, 3×5^e code en quelque sorte pour un *nombre transfini*, le plus petit « nombre » plus grand que chaque entier. Appelons ce nombre ω . Les éléments de $\mathcal{O}$ codant pour ω ne sont pas uniques. Considérons e_1, e_2 deux codes de fonctions telles que $|\Phi_{e_1}(n)| = 2n$ et $|\Phi_{e_2}(n)| = 2n + 1$. Les éléments 3×5^{e_1} et 3×5^{e_2} codent moralement pour la même chose, en l'occurrence ω , le plus petit nombre transfini plus grand que chaque entier. Comme expliqué dans la remarque 1.1, on a $H_{3 \times 5^{e_1}} \equiv_T H_{3 \times 5^{e_2}}$.

On aimerait une manière de pouvoir rendre compte formellement de cette équivalence. C'est précisément ce que permettent de faire les *ordinaux*. Si le concept fut introduit par Cantor en 1883, ce n'est toutefois pas sa définition que nous présenterons tout de suite, mais celle de von Neumann imaginée en 1923, qui présente l'avantage d'être plus concrète.

2.1. Approche de von Neumann : les ordinaux ensemblistes

John von Neumann définit les ordinaux de manière ensembliste. En substance, on a le principe suivant.

Principe des ordinaux de von Neumann

Un ordinal est donné par l'ensemble des ordinaux qui le précèdent, l'ordre sur les ordinaux étant alors établi par leur relation d'appartenance.

Une implémentation de ce principe peut se faire de manière informelle via la construction suivante.

- (1) L'ensemble vide $\emptyset$ est un ordinal.
- (2) Si α est un ordinal, alors $\alpha \cup \{\alpha\} = \{\beta : \beta = \alpha \text{ ou } \beta \in \alpha\}$ est l'*ordinal successeur* de α .
- (3) Soit $(\alpha_i)_{i \in I}$ une collection d'ordinaux n'ayant pas de plus grand élément pour la relation d'appartenance. Alors, $\bigcup_{i \in I} \alpha_i = \{\beta : \exists i \in I \beta \in \alpha_i\}$ est un *ordinal limite*.

On vérifie effectivement que si α est l'ensemble des ordinaux qui le précède, alors $\alpha \cup \{\alpha\}$ est un ensemble qui contient exactement α et tous les ordinaux qui précèdent α : il s'agit de l'ordinal successeur de α . De la même manière, si chaque α_i pour $i \in I$ est l'ensemble des ordinaux qui le précède, alors $\bigcup_{i \in I} \alpha_i$ est bien un ensemble d'ordinaux clos par le bas : il contient tous les ordinaux qui précèdent $\bigcup_{i \in I} \alpha_i$ au sens de la relation d'appartenance.

Moralement « l'ensemble » des ordinaux est le plus petit ensemble contenant $\emptyset$ et qui est clos par (2) et (3) de notre construction informelle. Nous verrons toutefois rapidement que « la collection » des ordinaux n'est pas un ensemble, et nous verrons bientôt une manière plus rigoureuse de les définir. Néanmoins, la construction précédente contient tout ce qu'il faut pour comprendre ce que sont les ordinaux.

Exemple 2.1. Les premiers ordinaux sont en correspondance directe avec les entiers naturels. Pour ces raisons, on les notera de la manière suivante.

- ▷ $0 = \emptyset$
- ▷ $1 = \{0\} = \{\emptyset\}$
- ▷ $2 = \{0, 1\} = \{\emptyset, \{\emptyset\}\}$
- ▷ $3 = \{0, 1, 2\} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$
- ▷ ...

Il est une pratique commune d'identifier les entiers naturels et leurs ordinaux correspondants : il s'agit d'un « codage » du même concept.

Le premier ordinal limite est donné par $\omega = \{0, 1, 2, \dots\}$.

Nous voyons à présent la définition formelle des ordinaux, qui nécessitera un peu de travail pour se ramener à l'intuition que nous en avons donnée et que nous rappelons ici : un ordinal est l'ensemble des ordinaux qui le précèdent.

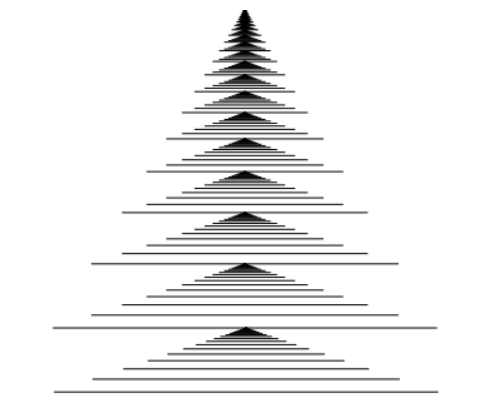


FIGURE 2.3 – Représentation des premiers ordinaux

Définition 2.2 (von Neumann). Un ensemble α est un ordinal si

- (1) α est *transitif* : $\forall \beta \in \alpha, \beta \subseteq \alpha$;
- (2) la relation d'appartenance sur α est un ordre total strict ;
- (3) toute partie non vide de α possède un plus petit élément pour l'appartenance.

On dira qu'un ordinal α est plus petit qu'un ordinal β , et l'on écrira $\alpha < \beta$ si $\alpha \in \beta$. ◇

Notons que la relation d'appartenance coïncidant avec la relation d'ordre, la condition (1) est équivalente à dire que l'ensemble α est clos par le bas : pour tout $\beta \in \alpha$, pour tout $\gamma < \beta$, $\gamma \in \alpha$. Voyons à présent quelques propositions permettant de clarifier la structure des ordinaux. La première proposition peut sembler curieuse, et n'a de fait d'intérêt qu'en l'absence de l'axiome de fondation (nous renvoyons le lecteur à la section 9-4 pour plus de détails sur ce sujet).

Proposition 2.4. Pour tout ordinal α , on a $\alpha \notin \alpha$. ★

PREUVE. Supposons $\alpha \in \alpha$. Alors, α contient un élément (lui-même, α) qui s'appartient à lui-même. La relation d'appartenance n'est donc pas un ordre strict sur α puisque que α contient un élément x tel que $x \in x$ (l'ordre n'est donc pas strict). Donc, α n'est pas un ordinal. ■

Nous voyons à présent une première direction permettant de revenir à notre intuition.

Proposition 2.5. Un ordinal α est toujours l'ensemble de tous les ordinaux plus petits que lui. ★

PREUVE. Par définition, si $\beta < \alpha$, alors $\beta \in \alpha$. Il reste à montrer que α ne contient que des ordinaux. Soit $x \in \alpha$, et soient $z \in y \in x$. Comme α est transitif, alors $z, y \in \alpha$. Comme la relation d'appartenance est une relation d'ordre sur α , alors $z \in x$. Donc, x est transitif. Puisque $x \subseteq \alpha$ et que la relation d'appartenance sur α est un ordre total strict, alors elle l'est également restreinte à x . Enfin, par transitivité, toute partie non vide de x est aussi une partie non vide de α , et contient donc un plus petit élément. Donc, x est un ordinal. ■

Nous montrons à présent que les ordinaux eux-mêmes sont totalement ordonnés entre eux.

Théorème 2.6

Soient α_1, α_2 deux ordinaux distincts. Alors, $\alpha_1 < \alpha_2$ ou $\alpha_2 < \alpha_1$.

PREUVE. Montrons d'abord la chose suivante. Pour tous ordinaux distincts α, β , on a $\alpha \subseteq \beta$ implique $\alpha < \beta$. Comme $\alpha \neq \beta$, alors $\beta \setminus \alpha$ est non vide, et possède donc un plus petit élément γ . Par minimalité de γ , tous ses éléments sont des éléments de α . On a donc $\gamma \subseteq \alpha$. Soit à présent $x \in \alpha \subseteq \beta$. Comme la relation d'appartenance est totale sur β , on a alors $x \in \gamma$ ou $\gamma \in x$ ou $\gamma = x$. Si $\gamma = x$, alors $\gamma \in \alpha$, ce qui contredit le choix de γ . De la même manière, si $\gamma \in x \in \alpha$, alors par transitivité de α on aurait $\gamma \in \alpha$, ce qui est aussi une contradiction. Donc, $x \in \gamma$, si bien que $\alpha \subseteq \gamma$, et donc $\alpha = \gamma$, égalité qui implique $\alpha < \beta$.

Passons à la preuve. Soit $A = \alpha_1 \setminus \alpha_2$. Si A est vide, alors $\alpha_1 \subseteq \alpha_2$, et donc $\alpha_1 < \alpha_2$. Sinon, A a un plus petit élément β tel que $\beta \notin \alpha_2$. En particulier, tous les éléments de β appartiennent à α_2 , et donc $\beta \subseteq \alpha_2$. On ne peut avoir par définition $\beta < \alpha_2$. Donc, $\beta = \alpha_2$ et $\alpha_2 < \alpha_1$. ■

Le théorème suivant permet de complètement se ramener à notre intuition de départ.

Théorème 2.7

Un ensemble α est un ordinal si, et seulement si, c'est un ensemble d'ordinaux clos par le bas.

PREUVE. Nous avons vu avec la proposition 2.5 qu'un ordinal α est toujours l'ensemble de tous les ordinaux plus petits que lui. Soient $\gamma \in \beta \in \alpha$. Par transitivité, $\gamma \in \alpha$, d'où il résulte que α est clos par le bas.

Supposons à présent que α soit un ensemble d'ordinaux clos par le bas. Soient $\gamma \in \beta \in \alpha$. Comme α est clos par le bas, alors $\gamma \in \alpha$. Ainsi, α est transitif. D'après la proposition 2.4, la relation d'appartenance sur α est anti-réflexive. D'après le théorème 2.6, elle est totale. Étant donné que α ne contient que des ordinaux qui sont des ensembles transitifs, la relation d'appartenance est aussi transitive. Elle est donc nécessairement aussi anti-symétrique, car si $x \in y$ et $y \in x$, alors par transitivité $x \in x$, ce qui contredit l'anti-réflexivité. La relation d'appartenance sur α est donc un ordre total strict. Enfin, soit $A \subseteq \alpha$ un ensemble non vide. Prenons $\beta \in A$. Ou bien β est le plus petit élément de A , auquel cas il n'y a rien à vérifier ; ou bien non, auquel cas $A \cap \beta \subseteq \beta$ est non vide, et comme β est un ordinal, alors $A \cap \beta$ possède un plus petit élément, qui est aussi le plus petit élément de l'ensemble A . ■

Corollaire 2.8

- (1) Soit α un ordinal. Alors, $\alpha \cup \{\alpha\}$ est un ordinal.
 (2) Soit $\{\alpha_i\}_{i \in I}$ une collection d'ordinaux. Alors, $\bigcup_{i \in I} \alpha_i$ est un ordinal.

PREUVE. Si α est un ordinal, alors $\alpha \cup \{\alpha\}$ est un ensemble d'ordinaux clos par le bas. De la même manière, si $\{\alpha_i\}_{i \in I}$ une collection d'ordinaux, alors $\bigcup_{i \in I} \alpha_i$ est un ensemble d'ordinaux clos par le bas. ■

Notation

Pour un ordinal α , on notera $\text{succ}(\alpha)$ pour $\alpha \cup \{\alpha\}$, l'ordinal successeur de α . Pour une collection d'ordinaux $(\alpha_i)_{i \in I}$, on notera $\sup_{i \in I} \alpha_i$ pour l'ordinal $\bigcup_{i \in I} \alpha_i$ (qui est limite si $(\alpha_i)_{i \in I}$ n'a pas de plus grand élément).

Notons que tout ordinal α est soit 0 , soit un ordinal successeur, soit un ordinal limite. En effet, si $\alpha \neq 0$, alors α admet au moins un élément. Si α admet un élément maximal β , alors $\alpha = \text{succ}(\beta)$. Sinon, $\alpha = \sup_{\beta \in \alpha} \beta$. Nous pouvons donc raisonner sur les ordinaux par analyse de cas sur cette trichotomie.

Notation

On note Ord « l'ensemble » de tous les ordinaux.

Le terme « ensemble » de la notation précédente a été mis entre guillemet. La raison est la suivante.

Proposition 2.9. La classe Ord n'est pas un ensemble. ★

PREUVE. Il est aisé de voir que la classe Ord des ordinaux, munie de la relation d'appartenance $\in$, satisfait les propriétés (1) à (3) de la définition 2.2.

On en déduit que Ord est un ordinal, et donc que $\text{Ord} \in \text{Ord}$, ce qui contredit la proposition 2.4. Cela est réglé de la manière suivante : Ord n'est pas un ensemble, et n'est donc pas sujet à la définition 2.2. ■

Si Ord n'est pas un ensemble, alors quelle est sa nature ? Il est aisé de construire une formule $F(x)$ du premier ordre de la théorie des ensembles telle que $F(x)$ est vraie ssi x est un ordinal. Pour autant, on ne peut pas en déduire que l'ensemble des éléments qui satisfont cette formule existe : afin d'éviter le paradoxe de Russell, l'axiome de compréhension nécessite un ensemble pré-existant : pour tout ensemble a et toute formule $G(x)$, l'ensemble $\{x \in a : G(x)\}$ existe, mais rien ne garantit l'existence de l'ensemble $\{x : G(x)\}$, et dans le cas de la formule $F(x)$ définissant les ordinaux, la proposition précédente montre que l'ensemble $\{x : F(x)\}$ n'existe pas, sous peine d'arriver à une contradiction. En quelque sorte, Ord est « trop gros » pour être un ensemble. On parlera alors de classes (à ne pas confondre avec notre utilisation du terme classe qui désigne tout au long de ce livre les sous-ensembles de $2^{\mathbb{N}}$) :

Définition 2.10. En théorie des ensembles une *classe* est une collection d'éléments qui satisfont une formule de la théorie des ensembles. ◇

Nous voyons tout de suite un exemple de l'utilisation de la notion de classe, qui sera réutilisée de temps à autre dans le cadre de notre manipulation des ordinaux.

Proposition 2.11. Toute classe non vide d'ordinaux possède un plus petit élément. ★

PREUVE. Soit $F(x)$ une formule de la théorie des ensembles. Supposons que pour au moins un ordinal α on ait $F(\alpha)$. Alors, l'ensemble A des ordinaux $\beta \in \text{succ}(\alpha)$ tels que $F(\beta)$ est vrai, est non vide. Comme $\text{succ}(\alpha)$ est un ordinal, alors A possède un plus petit élément, qui est aussi le plus petit élément β tel que $F(\beta)$. ■

2.2. Approche de Cantor : les bons ordres

La définition des ordinaux de von Neumann s'est imposée jusqu'à aujourd'hui comme standard. Le concept n'a toutefois pas été inventé par von Neumann, mais par Cantor [27], qui l'introduisit en 1883 dans la continuation de son travail sur l'infini et la cardinalité. Les nombres transfinis de von Neumann sont pratiques en ce sens qu'il s'agit d'objets « concrets ». Nous allons à présent dépouiller ces derniers de cette représentation concrète afin d'en dévoiler l'essence. Cette montée en abstraction fut le point de départ de la définition de Cantor : ce qui caractérise un ordinal, c'est son *type d'ordre*, c'est-à-dire l'ordre qu'il représente, à isomorphisme près.

Définition 2.12. Un ordre $<_R \subseteq A \times A$ strict sur un ensemble A est *bien fondé* s'il n'existe pas de suite infinie $(a_n)_{n \in \mathbb{N}}$ telle que $a_{n+1} <_R a_n$. Si de plus $<_R$ est total, on dira que $<_R$ est un *bon ordre*. $\diamond$

Voyons à présent un petit lemme qui permet de rapprocher la définition de bon ordre présentée ci-dessus, avec la définition des ordinaux de von Neumann.

Lemme 2.13. Soit $<_R \subseteq A \times A$ un ordre. Les deux énoncés suivants sont équivalents :

- (1) il n'existe pas de suite infinie $(x_n)_{n \in \mathbb{N}}$ de A telle que $x_{n+1} <_R x_n$ pour tout n ;
- (2) tout sous-ensemble $B \subseteq A$ possède un élément minimal. $\star$

PREUVE. Montrons (1) $\rightarrow$ (2), par contraposition. Supposons qu'il existe un sous-ensemble infini $B \subseteq A$ ne possédant pas d'élément minimal, c'est-à-dire tel que, pour tout $a \in B$, il existe $b \in B$ tel que $b <_R a$. On peut alors aisément construire une suite infinie $(x_n)_{n \in \mathbb{N}}$ de $B \subseteq A$ telle que $x_{n+1} <_R x_n$.

Montrons (2) $\rightarrow$ (1) de la même manière. Supposons qu'il existe une suite infinie $(x_n)_{n \in \mathbb{N}}$ de A telle que $x_{n+1} <_R x_n$. Alors, cette suite constitue un sous-ensemble $B \subseteq A$ ne possédant pas d'élément minimal. $\blacksquare$

Via notre nouveau vocabulaire, nous pouvons définir les ordinaux de von Neumann comme suit.

Ordinaux de von Neumann

Un ordinal de von Neumann est un ensemble transitif sur lequel la relation d'appartenance est un bon ordre.

Cantor définit les ordinaux comme étant les classes d'équivalence des bons ordres, via la relation d'équivalence d'isomorphisme entre les ordres.

Définition 2.14. Deux ordres $<_1 \subseteq A_1 \times A_1$ et $<_2 \subseteq A_2 \times A_2$ sur A_1 et A_2 respectivement sont *isomorphes* s'il existe une bijection $f : A_1 \rightarrow A_2$ telle que $x <_1 y$ ssi $f(x) <_2 f(y)$. $\diamond$

Nous allons maintenant montrer que la définition de von Neumann constitue en quelque sorte un représentant canonique de ces classes d'équivalence. Si un ordinal selon von Neumann n'est pas à proprement parler une relation d'ordre, il en induit une via la relation d'appartenance sur les éléments qu'il contiennent l'ordinal, et cette relation d'ordre est un bon ordre.

Théorème 2.15

Soit $<_R \subseteq A \times A$ un bon ordre sur un ensemble A . Alors, il existe un ordinal de von Neumann α et un isomorphisme $f : A \rightarrow \alpha$, pour lequel on a donc $x <_R y$ ssi $f(x) < f(y)$.

PREUVE. On définit la fonction f sur A par $f(y) = \{f(x) : x <_R y\}$. Une telle fonction f est définie par récurrence transfinie sur le bon ordre $<_R$. Nous verrons la validité d'une telle définition via le corollaire 3.10 de la section suivante. Supposons par l'absurde qu'il existe $x \in A$ tel que $f(x)$ n'est pas un ordinal. Comme $<_R$ est un bon ordre, il existe un tel x qui est minimal. En particulier, pour tout $y <_R x$, l'ensemble $f(y)$ est un ordinal. Donc, $f(x)$ est un ensemble d'ordinaux. Montrons que $f(x)$ est clos par le bas. Soit $\alpha \in f(y) \in f(x)$. Par définition de f , $\alpha = f(z)$ pour $z <_R y$. Par transitivité de $<_R$, on a $z <_R x$, et donc $f(z) \in f(x)$. Ainsi, $f(x)$ est bien clos par le bas, et est donc un ordinal, contradiction. Donc, pour tout $x \in A$, l'ensemble $f(x)$ est un ordinal. On montre de même que l'ensemble $\{f(x) : x \in A\}$ est clos par le bas, et donc qu'il s'agit d'un ordinal que nous appellerons α .

Il est clair par définition que $x <_R y$ implique $f(x) \in f(y)$. Réciproquement, si l'on n'a pas $x <_R y$, alors $x = y$ ou $y <_R x$ (car $<_R$ les bons ordres sont totaux), d'où $f(x) = f(y)$ ou $f(y) \in f(x)$. On n'a donc pas $f(x) \in f(y)$. Étant donné que $x <_R y$ ssi $f(x) < f(y)$, et que $x \neq y$ implique $x <_R y$ ou $y <_R x$, on obtient que f est injectif. Donc, f est un isomorphisme de A vers son image. ■

Notation

Étant donné un bon ordre $< \subseteq A \times A$ sur A , on écrit $|\cdot|$ pour dénoter l'ordinal α dont l'ordre des éléments est isomorphe à $<$.

Voici des exemples sur la représentation des ordinaux par des ordres sur $\mathbb{N}$.

Exemple 2.16.

- ▷ L'ordinal $n \in \omega$ correspond au bon ordre $0 < 1 < \dots < n - 1$.
- ▷ L'ordinal ω correspond à l'ordre de tous les entiers. Pour continuer il nous faut alors changer l'ordre naturel des entiers.
- ▷ L'ordinal $\text{succ}(\omega)$ correspond au bon ordre $1 < 2 < 3 < \dots < 0$.
- ▷ L'ordinal $\sup\{\text{succ}(\omega), \text{succ}(\text{succ}(\omega)), \text{succ}(\text{succ}(\text{succ}(\omega))), \dots\}$ correspond au bon ordre $0 < 2 < 4 < 6 < \dots < 1 < 3 < 5 < 7 < \dots$.

Tout comme il n'y a pas qu'un seul code de $\mathcal{O}$ pour un même ordinal, il n'y a pas non plus qu'un seul bon ordre pour un même ordinal. Par exemple, l'ordinal ω correspond également au bon ordre $1 < 0 < 2 < 3 < 4 < \dots$.

3. Induction et récurrence transfinie

Les ordinaux ont plusieurs intérêts. Nous verrons par exemple dans la section 4.1 qu'ils peuvent, à l'aide de l'axiome du choix, servir de socle à une approche unifiée de la cardinalité des ensembles. Ils permettent également de formaliser les itérations transfinies. C'est ce dernier aspect que nous développons dans cette section.

3.1. Exemple avec la hiérarchie borélienne

Souvenons-nous par exemple de la définition 17-3.1 sur les classes Σ_n^0 et Π_n^0 de la hiérarchie borélienne.

Cette définition est en fait incomplète, car elle ne capture pas toutes les classes boréliennes, dont nous donnons ci-après la définition complète.

Définition 3.1 (Classes boréliennes)

La collection des classes Boréliennes de $2^{\mathbb{N}}$ est la plus petite collection contenant les ouverts et les fermés, et qui est close par complémentaire et réunion dénombrable. $\diamond$

Selon cette définition, une réunion dénombrable $\bigcup_n \mathcal{B}_n$, où chaque $\mathcal{B}_n$ est une classe Π_n^0 , est bien une classe borélienne. Toutefois, si chaque $\mathcal{B}_n$ est un Π_n^0 propre, c'est-à-dire non Σ_n^0 et en particulier non Π_{n-1}^0 , alors $\bigcup_n \mathcal{B}_n$ n'est elle-même Σ_m^0 pour aucun $m \in \mathbb{N}$: elle est en fait Σ_ω^0 , et il est nécessaire d'utiliser les ordinaux pour capturer les complexités possibles de classes boréliennes.

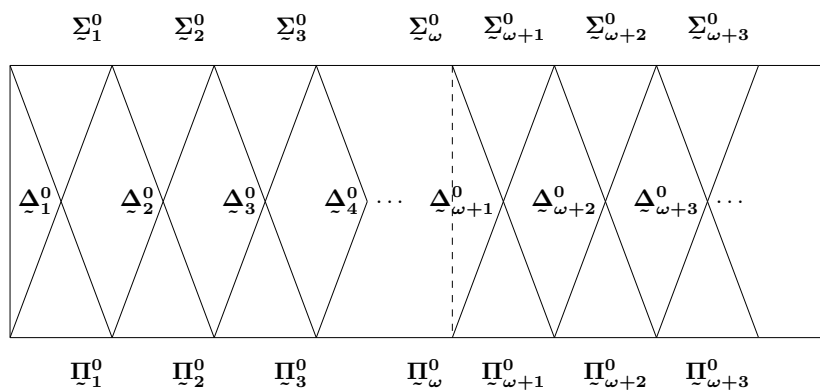


FIGURE 3.2 – Hiérarchie borélienne transfinie. Ici, $\omega + n$ désigne le n -ième successeur de ω (voir la définition 3.5 à venir).

Définition 3.3 (Hiérarchie borélienne)

La hiérarchie borélienne est définie par induction sur les ordinaux.

- ▷ Une classe Σ_1^0 est un ouvert de $2^{\mathbb{N}}$.
- ▷ Pour α ordinal, une classe Π_α^0 est le complémentaire d'une classe Σ_α^0 .
- ▷ Soit $\alpha = \text{succ}(\beta)$ un ordinal successeur. Une classe Σ_α^0 est une réunion dénombrable de classes Π_β^0 .
- ▷ Soit α un ordinal limite. Une classe Σ_α^0 est une réunion $\bigcup_n \mathcal{B}_n$ dénombrable, où chaque $\mathcal{B}_n$ est une classe $\Pi_{\beta_n}^0$ telle que $\sup_n \beta_n = \alpha$. $\diamond$

Notons qu'en l'absence d'ambiguïté possible et afin de rester cohérent avec les notations précédentes, nous écrivons Σ_1^0 et non Σ_1^0 . Il est possible de raccourcir un peu la définition précédente en rassemblant les cas limites et successeurs en un seul cas, ce qui sera parfois fait dans les différentes hiérarchies à venir.

Définition 3.4 (Hiérarchie borélienne (déf. alternative))

La hiérarchie borélienne est définie par induction sur les ordinaux.

- ▷ Une classe Σ_1^0 est un ouvert de $2^{\mathbb{N}}$.
- ▷ Pour α ordinal, une classe Π_α^0 est le complémentaire d'une classe Σ_α^0 .
- ▷ Soit α un ordinal. Une classe Σ_α^0 est une réunion dénombrable $\bigcup_n \mathcal{B}_n$, où chaque $\mathcal{B}_n$ est une classe $\Pi_{\beta_n}^0$ telle que $\sup_n (\text{succ}(\beta_n)) = \alpha$. $\diamond$

Notons que la hiérarchie borélienne est définie ci-dessus sur tous les ordinaux. En pratique, on suppose en général *l'axiome du choix dénombrable* qui implique que seuls les ordinaux dénombrables servent à la création de classes boréliennes. Nous verrons cela un peu plus en détail dans la section 4.

3.2. Exemple avec l'arithmétique des ordinaux

Nous voyons à présent un autre exemple de définition par récurrence sur les ordinaux, avec les extensions des opérations arithmétiques usuelles. L'addition, la multiplication et l'exponentiation se définissent de manière standard par récurrence sur les entiers, en traitant le cas de base, et le cas successeur. Pour leur extension aux ordinaux, on y rajoute une règle pour le cas limite. Comme nous l'avons vu dans l'exemple 2.1, les premiers ordinaux peuvent s'identifier aux entiers naturels, munis de leur ordre standard. Les définitions suivantes montrent que les ordinaux peuvent être vus comme une extension des entiers naturels. Les cas 1 et 2 des définitions suivantes correspondent aux définitions par récurrence de l'addition, la

multiplication et l'exponentiation sur les entiers. Par exemple, $n + 0 = 0$ et $n + (m + 1) = (n + m) + 1$ définissent complètement l'addition sur $\mathbb{N}$. Le cas 3, à savoir le cas limite, permet de « dépasser » la barrière de ω , et continuant l'itération sur des ordinaux arbitraires.

Définition 3.5

1. $\alpha + 0 = \alpha$,	1. $\alpha \times 0 = 0$,
2. $\alpha + \text{succ}(\beta) = \text{succ}(\alpha + \beta)$,	2. $\alpha \times \text{succ}(\beta) = (\alpha \times \beta) + \alpha$,
3. $\alpha + \beta = \sup_{\gamma \in \beta} (\alpha + \gamma)$ si β limite.	3. $\alpha \times \beta = \sup_{\gamma \in \beta} (\alpha \times \gamma)$ si β limite.
-
- | |
|--|
| 1. $\alpha^0 = 1$, |
| 2. $\alpha^{\text{succ}(\beta)} = (\alpha^\beta) \times \alpha$, |
| 3. $\alpha^\beta = \sup_{\gamma \in \beta} (\alpha^\gamma)$ si β limite. |

◇

Les définitions précédentes ne sont pas forcément évidentes à appréhender et le lecteur peut consulter la figure 3.6 pour une représentation graphique des premiers ordinaux ainsi que de l'utilisation de la multiplication et de la puissance ordinale.

Notons que l'addition et la multiplication ne sont plus commutatives dans les ordinaux : $\omega + 2 = \text{succ}(\text{succ}(\omega))$ et $2 + \omega = \sup_{n \in \omega} (2 + n) = \omega$.

La raison clef qui fait que les définitions précédentes sont valides est le fait que les ordinaux sont bien ordonnés, et la difficulté supplémentaire par rapport à l'induction sur les entiers est la compréhension des étapes limites. Si l'on veut effectuer l'opération $\alpha + \text{succ}(\beta)$, où β est limite, nous avons besoin d'après la définition précédente de connaître $\alpha + \beta$, et pour connaître $\alpha + \beta$ nous avons alors besoin de connaître $\alpha + \gamma$ pour tout $\gamma < \beta$, et ainsi de suite pour chacune de ces additions $\alpha + \gamma$. Contrairement à l'induction sur les entiers, la définition de $\alpha + \text{succ}(\beta)$ dépend ici d'une *infinité* « d'étapes précédentes ». L'addition est néanmoins bien définie, car l'ordre sur les ordinaux est bien fondé : si la valeur de $\alpha + \beta_1$ a besoin de la valeur de $\alpha + \beta_2$, laquelle a besoin de la valeur de $\alpha + \beta_3$, et ainsi de suite pour $\beta_1 > \beta_2 > \beta_3 > \dots$, on arrivera nécessairement à $\beta_n = 0$ pour un certain entier n .

Nous allons voir dans la prochaine section une justification formelle aux définitions par récurrence sur les ordinaux. Avant de nous y atteler, voici un petit exercice pour apprendre à manipuler un peu le transfini.

Exercice 3.7. (★★) Des Blorks — créatures bêtes et méchantes — se trouvent devant vous dans une file indienne transfinie indexée par tous les ordinaux successeurs. Il n'y a que ω Blorks pour le moment. Vous avez à votre disposition un marteau de guerre avec lequel vous devez écraser chaque Bork l'un après l'autre. Le problème est que dès que vous écrasez un

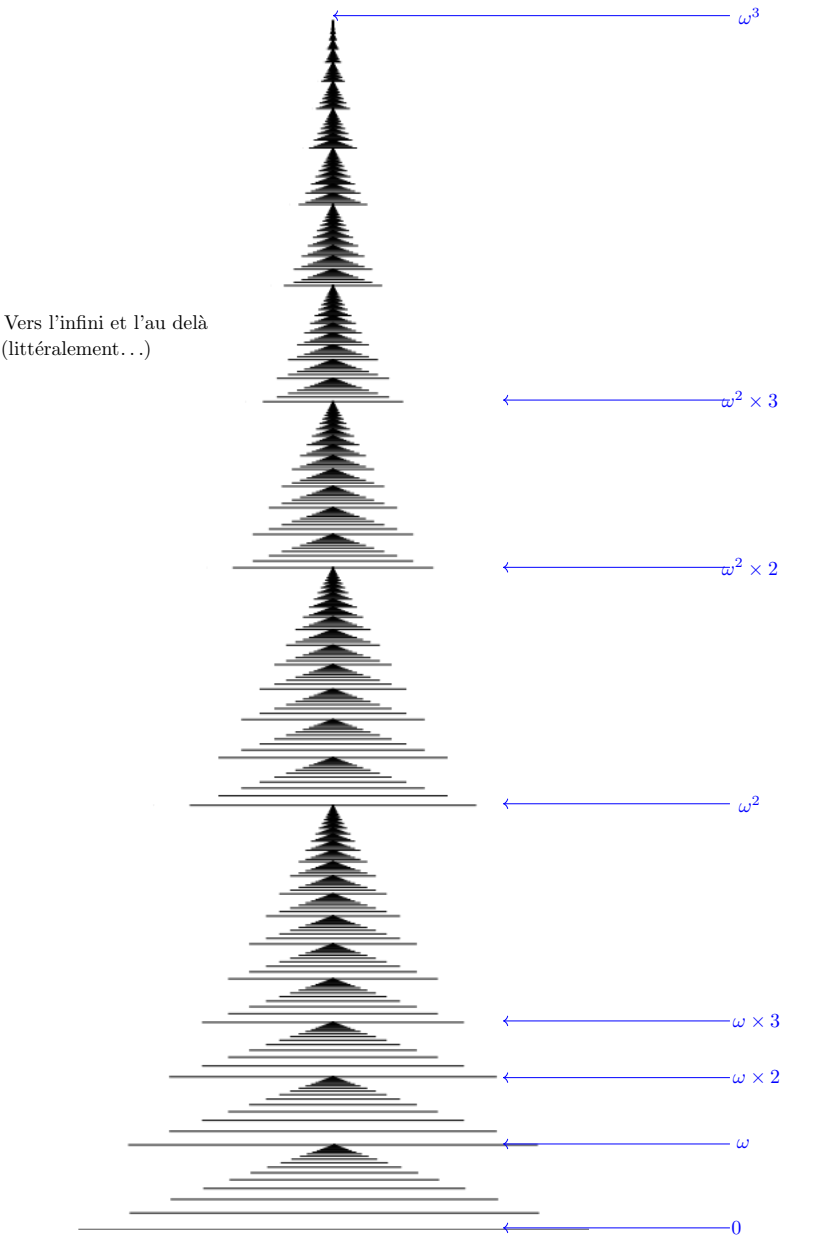


FIGURE 3.6 – *Représentation des premiers ordinaux. Chaque barre représente un ordinal.*

Blorck, un autre apparaît au fond de la file indienne. Ainsi, après écrasement du Blorck numéro 0, un autre apparaît en position $\omega + 1$. Après écrasement du Blorck numéro 1, un autre apparaît en position $\omega + 2$, etc.

- (1) À quelle étape transfinie aurez-vous enfin anéanti tous les Blorcks ?
- (2) Même question, mais cette fois-ci dès que vous écrasez un Blorck, ce sont ω Blorcks qui apparaissent au fond de la file indienne !
- (3) Laissons de côté votre marteau, et supposons à présent qu'à chaque étape un éclair foudroie un Blorck *au hasard*. À ce moment, si un Blorck est bien foudroyé (ce qui est le cas s'il reste au moins un Blorck) ω Blorcks apparaissent au fond de la file indienne. Peut-on être sûr qu'à une certaine étape transfinie, il n'y aura plus de Blorck ? Ou bien cela dépend-il du sort ? (Cette dernière question est difficile.) $\diamond$

3.3. Justification formelle de la récurrence ordinale

Commençons par une généralisation de l'induction arithmétique aux ordinaux : si une propriété P est vraie sur 0 et si le fait qu'elle soit vraie sur un entier n implique qu'elle le soit sur l'entier $n + 1$, alors elle est vraie sur tout $n \in \mathbb{N}$. La proposition suivante étend ce principe aux ordinaux, et vient simplement du fait que l'ordre sur les ordinaux est bien fondé.

Proposition 3.8 (Induction transfinie sur les ordinaux). Soit $F(x)$ une formule du premier ordre, telle que pour tout ordinal α , si $F(\beta)$ est vraie pour tout $\beta < \alpha$, alors $F(\alpha)$ est vraie. Alors, $F(\alpha)$ est vraie pour tout ordinal α . $\star$

PREUVE. Supposons par l'absurde que F ne soit pas vraie pour tous les ordinaux α . D'après la proposition 2.11, il existe un plus petit ordinal α tel que $F(\alpha)$ est faux. En particulier, par minimalité de α , on a $F(\beta)$ pour tout $\beta < \alpha$. Donc, $F(\alpha)$ est vraie, ce qui est une contradiction. $\blacksquare$

Nous avons vu avec la définition 3.5 des fonctions définies par récurrence sur les ordinaux. Tout comme dans le cas de l'induction, il s'agit formellement d'une extension de la récurrence sur les entiers : il est possible de définir une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ en réutilisant pour la définition de $f(n)$ des valeurs $f(0), \dots, f(n-1)$. Souvent, seules quelques valeurs précédentes de f sont utilisées pour définir $f(n+1)$, comme dans la définition de la suite de Fibonacci : $f(0) = 0$, $f(1) = 1$ et $f(n+2) = f(n) + f(n+1)$, mais ce n'est pas toujours le cas. Par exemple, la définition 25-4.22 récursive des nombres de Catalan, $C_{n+1} = \sum_{i=0}^n C_i C_{n-i}$, utilise toutes les valeurs précédentes. Ce genre de définition est tout à fait valide, et le théorème général qui pourrait être formulé est que, pour toute fonction $g : \mathbb{N}^{<\mathbb{N}} \rightarrow \mathbb{N}$, il existe une unique fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $f(n) = g(f \upharpoonright_n)$ pour tout n . Par exemple, une

telle fonction $g : \mathbb{N}^{<\mathbb{N}} \rightarrow \mathbb{N}$ correspondant à la suite de Fibonacci est donnée par $g(\sigma) = 0$ si $|\sigma| = 0$, $g(\sigma) = 1$ si $|\sigma| = 1$, et $g(\sigma) = \sigma(|\sigma| - 2) + \sigma(|\sigma| - 1)$ sinon. Nous voyons une extension de ce théorème, qui demande un peu de travail pour être démontré en toute généralité sur la classe des ordinaux. Pour commencer, nous avons besoin de manipuler des objets un peu plus généraux que les fonctions afin de traiter uniformément de l'induction sur tous les ordinaux, et nous introduisons pour cela le concept de relation fonctionnelle suivant.

Notation

Une relation fonctionnelle $G : \mathcal{V} \rightarrow \mathcal{V}$, où $\mathcal{V}$ est la classe de tous les ensembles, est simplement une formule du premier ordre telle que pour tout x il existe un unique ensemble r pour lequel $G(x, r)$ est vrai. On écrira alors $G(x) = r$.

Via cette nouvelle notation, le sens formel par exemple de l'addition de la définition 3.5 en théorie des ensembles est le suivant : pour tout α , nous avons une relation fonctionnelle G_α telle que si f est une fonction définie sur un ordinal γ quelconque, alors

$$G_\alpha(f) = \alpha \quad \text{dans le cas où } \gamma = 0,$$

$$G_\alpha(f) = \text{succ}(f(\gamma')) \quad \text{dans le cas où } \gamma = \text{succ}(\gamma')$$

et

$$G_\alpha(f) = \sup_{\gamma' < \gamma} f(\gamma') \quad \text{dans le cas où } \gamma \text{ est limite.}$$

Bien entendu, l'objectif est d'utiliser G_α non pas sur n'importe quelle fonction f , mais bien sur la fonction f qui consiste déjà en l'addition de α à γ' pour $\gamma' < \gamma$. Le théorème suivant nous garantit l'existence et l'unicité de cette fonction f à chaque étape, ce qui donne la validité des définitions par induction sur la classe des ordinaux.

Théorème 3.9 (Récursion transfinie sur les ordinaux)

Soit $\mathcal{V}$ la classe de tous les ensembles. Soit $G : \mathcal{V} \rightarrow \mathcal{V}$ une relation fonctionnelle. Alors, il existe une unique relation fonctionnelle

$$F : \text{Ord} \rightarrow \mathcal{V} \quad \text{telle que } F(\alpha) = G(F \upharpoonright_\alpha).$$

PREUVE

On définit le prédicat $H(\alpha, f)$ comme étant : « α est un ordinal, f est une fonction de domaine α telle que, pour tout $\beta \in \alpha$, on a $G(f \upharpoonright_\beta) = f(\beta)$ ».

La relation $F(\alpha) = r$ est alors définie comme étant : « il existe une fonction f telle que $H(\alpha, f)$ et $G(f) = r$. »

Montrons d'abord que pour tout ordinal α , pour toutes fonctions f_1, f_2 définies sur α , si $H(\alpha, f_1)$ et $H(\alpha, f_2)$, alors $f_1 = f_2$. D'après la proposition 3.8, il suffit de montrer que si c'est le cas pour tout $\beta < \alpha$, alors c'est le cas pour α . Supposons effectivement que ce soit le cas pour tout $\beta < \alpha$. Soient f_1, f_2 telles que $H(\alpha, f_1)$ et $H(\alpha, f_2)$. Par définition de H , pour tout $\beta < \alpha$, on a $H(\beta, f_1 \upharpoonright_\beta)$ et $H(\beta, f_2 \upharpoonright_\beta)$ (où $f \upharpoonright_\beta$ désigne la restriction de f aux éléments de β). Par hypothèse, on a donc $f_1 \upharpoonright_\beta = f_2 \upharpoonright_\beta$ pour tout $\beta < \alpha$. Si α est limite ou égal à 0 , on a donc $f_1 = f_2$. Si $\alpha = \text{succ}(\beta)$, alors $f_1(\beta) = G(f_1 \upharpoonright_\beta) = G(f_2 \upharpoonright_\beta) = f_2(\beta)$, et donc $f_1 = f_2$. Ainsi, pour tout ordinal α , on a $H(\alpha, f_1)$ et $H(\alpha, f_2)$ implique $f_1 = f_2$.

Montrons que, pour tout ordinal α , il existe un unique ensemble r (égal à $G(F \upharpoonright_\alpha)$) tel que $F(\alpha) = r$. D'après la proposition 3.8, il suffit de montrer que si c'est le cas pour tout $\beta < \alpha$, alors c'est le cas pour α . Supposons effectivement que ce soit le cas pour tout $\beta < \alpha$. Si $\alpha = \text{succ}(\beta)$, alors il existe une fonction f_β telle que $H(\beta, f_\beta)$. Soit $r = G(f_\beta)$. La fonction f définie sur α par $f(\beta) = r$ et par $f(\gamma) = f_\beta(\gamma)$ pour $\gamma < \beta$ existe et vérifie $H(\alpha, f)$. On a en particulier $F(\alpha) = r$, et cette valeur r est unique. À présent, si α est limite ou égale à 0 , alors pour tout $\beta < \alpha$ il existe une unique fonction f_β telle que $H(\beta, f_\beta)$. En utilisant l'axiome de remplacement, l'ensemble $\{f : \exists \beta < \alpha \ H(\beta, f)\}$ existe et, d'après le paragraphe précédent, la fonction $f = \bigcup_{\beta < \alpha} f_\beta$ est bien définie, est unique et vérifie $H(\alpha, f)$. On a alors $F(\alpha) = r$ pour $r = G(f)$. ■

D'après la correspondance entre ordinaux et bons ordres, on peut donc faire des définitions par récurrence transfinie sur n'importe quel bon ordre. Dans le corollaire suivant la notation $Y^{<A}$ désigne l'ensemble des segments initiaux de A via la relation de bon ordre sur A .

Corollaire 3.10 (Récursion transfinie)

Soit $<_R$ un bon ordre sur un ensemble A et soit Y un ensemble.

Considérons $g : Y^{<A} \rightarrow Y$ une fonction. Il existe alors une unique fonction $f : A \rightarrow Y$ telle que pour tout $a \in A$, $f(a) = g(f \upharpoonright_a)$.

4. Ordinaux dénombrables et indénombrables

Par définition, un ordinal α est dénombrable s'il existe $f : \alpha \rightarrow \mathbb{N}$ bijective. Notons qu'il s'agit d'une bijection au sens ensembliste, c'est-à-dire que l'on n'impose pas que l'ordre soit respecté. De manière équivalente, un ordinal est dénombrable si l'on peut le représenter par un bon ordre sur $\mathbb{N}$.

Proposition 4.1. Soit α un ordinal. Alors, α est dénombrable si, et seulement si, il existe $X \in 2^{\mathbb{N}}$ tel que la relation $<_X \subseteq \mathbb{N} \times \mathbb{N}$ définie par $a <_X b$ ssi $\langle a, b \rangle \in X$ est un bon ordre sur $\mathbb{N}$ pour lequel $|\langle <_X \rangle| = \alpha$. ★

PREUVE. Supposons α dénombrable. Soit $f : \mathbb{N} \rightarrow \alpha$ une bijection. On définit X par $\langle a, b \rangle \in X$ ssi $f(a) \in f(b)$. L'ordre $<_X$ est par définition isomorphe à celui des éléments de α pour la relation d'appartenance. On a donc $|\langle <_X \rangle| = \alpha$.

Réciproquement, soit $X \in 2^{\mathbb{N}}$ tel que la relation $<_X$ soit un bon ordre sur $\mathbb{N}$, pour lequel $\alpha = |\langle <_X \rangle|$. D'après le théorème 2.15, on a $f : \mathbb{N} \rightarrow \beta$ qui est un isomorphisme entre $<_X$ et un certain ordinal β , lequel est donc tel que $|\langle <_X \rangle| = \beta$. Par hypothèse, $|\langle <_X \rangle| = \alpha$, et donc $\alpha = \beta$. Ainsi, on a bien une bijection entre $\mathbb{N}$ et α . ■

On introduit à présent le plus petit ordinal non dénombrable.

Notation

Soit $\omega_1 = \{\alpha : \alpha \text{ est un ordinal dénombrable}\}$.

Par le théorème 2.7, ω_1 est un ordinal. Il ne peut pas en revanche être dénombrable, car on aurait alors $\omega_1 \in \omega_1$. On en conclut que l'ensemble des ordinaux dénombrables ne l'est pas. De plus, si α est un autre ordinal non dénombrable, on a nécessairement $\omega_1 < \alpha$ puisque par définition de ω_1 on n'a pas $\alpha \in \omega_1$. Ainsi, ω_1 est le plus petit ordinal non dénombrable. En pratique, nous travaillerons ici uniquement avec des ordinaux dénombrables, et donc représentables par un bon ordre sur $\mathbb{N}$.

4.1. Les ordinaux et l'axiome du choix

La présente section utilise les notions de théorie des ensembles développées dans la section 9-4. L'axiome du choix permet de montrer que tout ensemble peut être mis en bijection avec un ordinal, ce qui a pour conséquence les énoncés qui suivent.

- ▷ Tout ensemble est bien ordonné : étant donné un ensemble A quelconque, on peut construire un bon ordre $< \subseteq A \times A$ sur ses éléments.
- ▷ Le cardinalité de tout ensemble est comparable : cela vient directement du fait que pour tous ordinaux α, β , soit $\alpha \subseteq \beta$ soit $\beta \subseteq \alpha$.

Au fond, l'axiome du choix est équivalent au fait que tout ensemble soit bien ordonné. Ce n'est pas une équivalence dont le père des ordinaux et de la cardinalité des ensembles avait pleinement conscience. Cantor écrivit par exemple (voir [12, p. 143]) ce qui suit.

« *Qu'il soit toujours possible de mettre tout ensemble bien défini sous la forme d'un ensemble bien ordonné est, me semble-t-il, une loi de la pensée, fondamentale, riche de conséquences et particulièrement remarquable par son universalité, loi sur laquelle je me promets de revenir dans un travail ultérieur.* »

Cet extrait des écrits de Cantor illustre bien la part importante d'intuition qu'il y avait dans son travail. C'est finalement Zermelo, et non Cantor, qui reviendra sur « cette loi fondamentale de la pensée » dans un travail ultérieur, et qui comprendra que la possibilité de pouvoir bien ordonner tout ensemble nécessite un axiome, équivalent à l'axiome du choix.

Théorème 4.2 (Zermelo)

L'axiome du choix est équivalent au fait que tout ensemble est bien ordonnable.

PREUVE. Rappelons pour cette preuve que $\mathcal{P}(A)$ dénote, pour un ensemble A , l'ensemble de ses parties. Supposons que l'on dispose de l'axiome du choix. Soit A un ensemble que l'on souhaite bien ordonner. Il existe en particulier une fonction de choix f définie sur $\mathcal{P}(A) \setminus \{\emptyset\}$ et qui à chaque partie non vide $B \subseteq A$ associe un élément de B . On définit par induction sur les ordinaux $g(0) = f(A)$, puis pour tout ordinal α tel que $A \setminus \bigcup_{\beta < \alpha} \{g(\beta)\}$ est non vide, on définit $g(\alpha) = f(A \setminus \bigcup_{\beta < \alpha} \{g(\beta)\})$. Par l'axiome de compréhension, soit $B \subseteq A$ le sous-ensemble des éléments de A égaux à $g(\alpha)$ pour un certain ordinal α . Par l'axiome de remplacement, soit α l'ordinal supremum des $g^{(-1)}(a)$ pour $a \in B$. Alors, on a forcément $A \setminus B$ vide, sinon $g(\text{succ}(\alpha))$ serait envoyé vers un élément de $A \setminus B$, ce qui contredit la définition de B . Donc, g est une bijection d'un ordinal α vers A , via laquelle A est bien ordonné.

Réciproquement, supposons tout ensemble bien ordonnable. Soit $(A_i)_{i \in I}$ une collection d'ensembles non vides ; on met alors un bon ordre sur $\bigcup_{i \in I} A_i$, et l'on définit la fonction de choix f en assignant à un élément $i \in I$ le plus petit élément de $x \in \bigcup_{i \in I} A_i$ tel que $x \in A_i$. ■

4.2. Axiome du choix dénombrable et boréliens

Zermelo a mis en évidence que l'axiome du choix était utilisé de manière intuitive et instinctive par plusieurs mathématiciens. Ainsi, par exemple le fait que tout espace vectoriel de dimension infinie ait une base, nécessite l'axiome du choix. Pour ce qui est de l'étude de la cardinalité des ensembles, l'axiome du choix est nécessaire pour montrer que la cardinalité de tout ensemble est comparable, ou par exemple pour montrer qu'une réunion

dénombrable d'ensembles dénombrables est dénombrable. Cette dernière propriété nous intéresse particulièrement, et ne nécessite qu'une version faible de l'axiome du choix.

Il est fréquent aujourd'hui, de la part des théoriciens des ensembles, de travailler avec l'axiome du choix restreint aux collections dénombrables d'ensembles (qui eux ne sont pas nécessairement dénombrables), sans forcément admettre l'axiome du choix dans sa globalité : « Pour toute collection d'ensembles $(\mathcal{A}_n)_{n \in \mathbb{N}}$, il existe une fonction de choix de n vers $\mathcal{A}_n$ ». La version dénombrable de l'axiome du choix permet de montrer que la réunion dénombrable d'ensembles dénombrables est dénombrable. Elle a en particulier la conséquence suivante : si $(\alpha_n)_{n \in \mathbb{N}}$ est une suite dénombrable d'ordinaux dénombrables, alors $\sup_n \alpha_n$ est un ordinal dénombrable. Cela implique en particulier que toute classe borélienne est toujours Σ_α^0 pour un certain α dénombrable. Cela est dû au fait que les boréliens se construisent par réunion dénombrable de boréliens de rang inférieur. Aussi est-il impossible d'avoir une suite d'ordinaux dénombrables $(\beta_n)_{n \in \mathbb{N}}$ pour laquelle on a $\sup_n \beta_n = \omega_1$. Il n'y a donc aucun nouveau borélien à l'étape ω_1 , et donc aux étapes suivantes non plus. Indépendamment des axiomes utilisés, on considérera toujours dans ce livre que les boréliens sont construits le long des ordinaux dénombrables, et si par un hasard incongru une suite d'ordinaux dénombrables $(\alpha_n)_{n \in \mathbb{N}}$ avec $\sup_n \alpha_n = \omega_1$ venait à profiter de l'absence de cet axiome pour faire son apparition, alors une classe $\bigcup_n \mathcal{B}_n$, où chaque $\mathcal{B}_n$ serait strictement $\Sigma_{\alpha_n}^0$, ne serait pas considérée comme borélienne.

On sait aujourd'hui que la version dénombrable de l'axiome du choix n'entraîne pas les situations contre-intuitives qui surviennent avec sa version indénombrable : l'axiome du choix dénombrable est par exemple compatible avec le fait que tout ensemble soit mesurable (voir la section 17-4) ou ait la propriété de Baire (voir la définition 10-4.9). Le sujet dépasse le cadre de cet ouvrage, et le lecteur pourra consulter l'excellent ouvrage de théorie des ensembles de Patrick Dehornoy [44] pour plus de détails.

4.3. Ordinaux et hypothèse du continu

L'ordinal ω_1 représente en quelque sorte « le plus petit bon ordre » non dénombrable. Cet infini coïncide-t-il avec celui des nombres réels ? En d'autres termes, existe-t-il une bijection entre ω_1 et $2^{\mathbb{N}}$? Cette question peut être vue comme une formulation de l'hypothèse du continu. La question originelle de Cantor portait sur les sous-ensembles de réels (si $\mathcal{A} \subseteq \mathbb{R}$ est indénombrable, existe-t-il une injection de $2^{\mathbb{N}}$ vers $\mathcal{A}$?). Si l'on admet l'axiome du choix, la question de Cantor est alors équivalente à celle de savoir s'il existe une injection de $2^{\mathbb{N}}$ vers ω_1 . Notons que sans l'axiome du choix il n'est pas clair non plus qu'une injection existe de ω_1 vers $2^{\mathbb{N}}$:

Proposition 4.3. « Moralement », il existe une injection de ω_1 vers $2^{\mathbb{N}}$, c'est-à-dire qu'il existe une fonction f qui à $\alpha < \omega_1$ associe une classe non vide de réels A_α telle que $\beta_1 \neq \beta_2$ implique $A_{\beta_1} \cap A_{\beta_2} = \emptyset$. ★

PREUVE. Par la proposition 4.1, pour tout α dénombrable, il existe X tel que $|\langle_X| = \alpha$, où $\langle_X$ est défini par $a \langle_X b$ ssi $\langle a, b \rangle \in X$. On associe à chaque α dénombrable la classe des réels X tels que $|\langle_X| = \alpha$. ■

La proposition précédente est bien évidemment insatisfaisante. On a une injection qui associe $\alpha < \omega_1$ vers des ensembles non vides A_α de réels, deux à deux disjoints. Mais comment choisir un unique réel dans chaque ensemble A_α ? Il s'agit de quelque chose d'impossible à faire sans l'axiome du choix, avec lequel on construit en revanche aisément une injection de ω_1 dans $2^{\mathbb{N}}$.

Rappelons qu'un ensemble X est subpotent (resp. équipotent) à un ensemble Y s'il existe une injection (resp. bijection) de X vers Y (voir le chapitre 2). On peut itérer la définition de ω_1 de la manière suivante.

Définition 4.4. Supposons ω_n défini pour $n \in \mathbb{N}$. Soit

$$\omega_{n+1} = \{\beta : \beta \text{ est subpotent (strictement ou pas) à } \omega_n\}. \quad \diamond$$

On montre aisément par induction que ω_n n'est pas équipotent à ω_{n+1} : il s'agit d'une hiérarchie d'infinis de plus en plus grands.

Gödel a construit un modèle de ZFC — son fameux univers des constructibles — dans lequel ω_1 est équipotent à $2^{\mathbb{N}}$. Cohen a montré comment, pour tout n , construire un modèle de ZFC dans lequel $2^{\mathbb{N}}$ est équipotent à ω_n .

Indépendamment des axiomes utilisés, nous verrons bientôt que pour un grand nombre de classes d'ensembles, l'hypothèse du continu est vérifiée dans le sens de Cantor : soit la classe est dénombrable, soit la classe contient un fermé parfait et donc une injection continue de $2^{\mathbb{N}}$ vers elle-même. Ce sera en particulier le cas pour toute classe borélienne.

5. Ordinaux effectifs

Les ordinaux étant en quantité indénombrable, ils ne peuvent évidemment pas tous être décrits par un algorithme. Dans cette section, nous allons étudier deux définitions d'ordinaux effectifs, basées respectivement sur l'approche de von Neumann et celle de Cantor, et montrer leur équivalence. La classe des ordinaux effectifs servira de socle pour le développement de l'hypercalculabilité.

5.1. Ordinaux constructifs

La première approche, celle de Kleene, suit le principe des ordinaux de von Neumann, en définissant les ordinaux de manière inductive en termes de successeurs et d'ordinaux limites. Comme nous l'avons vu dans la section 1.1, Kleene a défini un ensemble $\mathcal{O}$ de codes d'ordinaux en se basant sur l'unicité de la décomposition d'un entier en le produit de ses facteurs premiers. Rappelons ici la définition de $\mathcal{O}$.

Définition 5.1 (Kleene [114]). On définit inductivement un ensemble de notations $\mathcal{O} \subseteq \mathbb{N}$, avec un ordre partiel $<_o$ sur ses éléments.

- (1) $1 \in \mathcal{O}$.
- (2) Si $a \in \mathcal{O}$, alors $2^a \in \mathcal{O}$. Par suite, $a <_o 2^a$, et $b <_o a$ implique $b <_o 2^a$ pour tout b .
- (3) Si $\Phi_e : \mathbb{N} \rightarrow \mathbb{N}$ est une fonction totale calculable telle que

$$\forall n \ \Phi_e(n) \in \mathcal{O}, \text{ avec } \Phi_e(n) <_o \Phi_e(n+1) \text{ pour tout } n,$$

alors $3 \times 5^e \in \mathcal{O}$. De plus, pour chaque $b \in \mathcal{O}$ tel que $b <_o \Phi_e(n)$ pour un certain n , on a $b <_o 3 \times 5^e$. $\diamond$

Définition 5.2. On assigne naturellement un ordinal à chaque code de $\mathcal{O}$ de la manière suivante.

1. $|1| = 0$,
2. $|2^a| = \text{succ}(|a|)$,
3. $|3 \times 5^e| = \sup_n |\Phi_e(n)|$.

Un ordinal α est *constructif* s'il existe un code $a \in \mathcal{O}$ tel que $\alpha = |a|$. $\diamond$

Notation

Soit α un ordinal. On écrira $\mathcal{O}_{<\alpha}$ (resp. $\mathcal{O}_{=\alpha}$) pour l'ensemble des éléments $a \in \mathcal{O}$ tels que $|a| < \alpha$ (resp. $|a| = \alpha$).

Les ordinaux ne sont pas, bien entendu, tous constructifs. Comme les codes de $\mathcal{O}$ sont en quantité dénombrable, et qu'il existe une quantité indénombrable d'ordinaux dénombrables, il existe alors un plus petit ordinal dénombrable qui n'est pas constructif.

Notation

On note ω_1^{ck} le plus petit ordinal non constructif.

L'exposant « ck » de l'ordinal ω_1^{ck} est l'acronyme de « Church Kleene » pour Alonzo Church et Stephen Cole Kleene, qui ont introduit [36] le concept. Notons qu'aucun ordinal plus grand que ω_1^{ck} n'est constructif.

Exercice 5.3. Montrer que les ordinaux constructifs sont clos par le bas.
 $\diamond$

Nous verrons petit à petit de nombreuses propriétés remarquables de l'ordinal ω_1^{ck} . Nous verrons notamment avec le théorème 29-6.1 qu'il s'agit du plus petit ordinal non définissable, pour de très puissantes notions de définissabilité.

Les ordinaux constructifs permettent de faire des définitions de fonctions calculables par induction sur les ordinaux. On peut en particulier calculer une fonction d'addition sur les codes d'ordinaux constructifs.

Exemple 5.4. On définit la fonction d'addition $+_o : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ totale calculable, telle que pour tout $a, b \in \mathcal{O}$ la fonction renvoie $(a +_o b) \in \mathcal{O}$ avec $|a| + |b| = |a +_o b|$ et $a \leq_o (a +_o b)$:

$$\begin{aligned} a +_o 1 &= a, \\ a +_o 2^b &= 2^{a+_o b}, \\ a +_o 3 \times 5^e &= 3 \times 5^{f(e,a)}, \quad \text{où } f(e,a) \text{ est le code de la fonction telle} \\ &\quad \text{que } \Phi_{f(e,a)}(n) = a +_o \Phi_e(n), \end{aligned}$$

$$a +_o b = 1 \quad \text{si } b \text{ n'est pas de la forme } 1, 2^c \text{ ou } 3 \times 5^e.$$

Notons que la définition de $+_o$ utilise le théorème du point fixe, afin de réutiliser le code de $+_o$ dans la fonction $f(e, a)$.

Exercice 5.5. (*) Montrer que pour tous $a, b \in \mathcal{O}$, $|a| + |b| = |a +_o b|$. $\diamond$

5.2. Les ordinaux calculables

La définition des ordinaux constructifs de Kleene est dans l'esprit de celle de von Neumann : un ordinal est un objet qui « contient » les ordinaux qui le précèdent. Cette définition étant équivalente à celle de Cantor dans le cas général, il est naturel de se demander si une définition à la Cantor des ordinaux effectifs n'aurait pas elle aussi son intérêt. On introduit donc la définition suivante.

Définition 5.6. Un ordinal α est *calculable* s'il est fini ou si $\alpha = |<|$ pour $< \subseteq \mathbb{N} \times \mathbb{N}$ un bon ordre calculable sur $\mathbb{N}$. $\diamond$

Un premier résultat facile est l'inclusion des ordinaux constructifs dans les ordinaux calculables.

Proposition 5.7. Les ordinaux constructifs sont calculables. On peut en fait uniformément transformer un code $a \in \mathcal{O}$ en un code e calculant un bon ordre $<_e$ tel que $|a| = |<_e|$. $\star$

La preuve tient dans les deux lemmes qui suivent.

Lemme 5.8. Soit $<_R \subseteq \mathbb{N} \times \mathbb{N}$ un bon ordre c.e. sur $A \subseteq \mathbb{N}$ avec $|A|$ infini. Alors, il existe un bon ordre c.e. $<_S \subseteq \mathbb{N} \times \mathbb{N}$ sur $\mathbb{N}$ tel que $|<_R| = |<_S|$. ★

PREUVE. Si $<_R$ est une relation calculatoirement énumérable sur A , alors A est bien sûr lui aussi calculatoirement énumérable : quand $\langle a, b \rangle$ est énuméré dans $<_R$, on énumère a et b dans A . Notons qu'une énumération de A avec $|A|$ infini induit naturellement une bijection calculable $f : \mathbb{N} \rightarrow A$ en définissant $f(n)$ comme étant le n -ième élément énuméré dans A (en ne prenant pas en compte les redondances). On définit alors $<_S$ comme étant $\langle f^{-1}(a), f^{-1}(b) \rangle$ pour chaque $\langle a, b \rangle$ énuméré dans $<_R$. ■

Lemme 5.9. Soit $<_R \subseteq \mathbb{N} \times \mathbb{N}$ un bon ordre c.e. sur $\mathbb{N}$. Alors, $<_R$ est calculable. ★

PREUVE

Si $<_R$ est un ordre total sur $\mathbb{N}$, on a pour tout entier a, b avec $a \neq b$, soit $a <_R b$ soit $b <_R a$. Ainsi, $\langle b, a \rangle \notin <_R$ ssi $a = b$ ou $\langle a, b \rangle \in <_R$. Le complémentaire de $<_R$ est donc lui aussi calculatoirement énumérable. On conclut que $<_R$ est calculable. ■

PREUVE DE LA PROPOSITION 5.7. Supposons que $a \in \mathcal{O}$ code pour un ordinal infini. On peut énumérer la relation $<_a \subseteq \mathbb{N} \times \mathbb{N}$ définie comme étant la relation $<_o$ restreinte aux éléments $b, c <_o a$, ce qui nous donne un bon ordre c.e. sur une partie $A \subseteq \mathbb{N}$ avec A infini et tel que $|<_a| = |A|$. On applique alors le lemme 5.8 et le lemme 5.9. ■

Les avantages des ordinaux constructifs par rapport aux ordinaux calculables sont clairs : étant donné un code $e \in \mathcal{O}$, on voit tout de suite si e code pour un ordinal successeur (auquel cas on peut obtenir un code pour l'ordinal prédécesseur) ou si e code pour un ordinal limite (auquel cas on peut obtenir une suite calculable de codes pour les ordinaux constituants la limite). Cela permet de faire des définitions de fonctions calculables par induction sur les ordinaux constructifs comme avec l'exemple 5.4. Il n'est pas possible de le faire avec les ordinaux calculables. Le double saut est par exemple nécessaire pour savoir si un ordinal calculable est limite ou successeur. Pourtant les deux notions coïncident.

Théorème 5.10 (Kleene, Markwald)

Les ordinaux calculables sont constructifs.

On peut uniformément transformer le code e d'un ordre total calculable bien fondé $<_e$ en un code $a \in \mathcal{O}$ tel que $|<_e| \leq |a|$.

La preuve du théorème utilise le lemme suivant.

Lemme 5.11. Il existe une fonction totale calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que si $W_e \subseteq \mathcal{O}$, alors $f(e) \in \mathcal{O}$, avec $\sup_{a \in W_e} |a| \leq |f(e)|$. ★

PREUVE. On peut considérer sans perte de généralité que W_e est infini. Afin de s'en assurer, on pourra par exemple y énumérer en plus de ce qui s'y trouve déjà tous les codes constructifs d'ordinaux finis. On définit $f(e)$ comme étant 3×5^a , où a est tel que $\Phi_a(n)$ renvoie la somme finie des n premiers codes distincts et différents de 1 (le code de 0), d'ordinaux énumérés dans W_e . La somme finie est faite via la fonction $+_o$ de l'exemple 5.4. Notons que l'on a bien $\Phi_a(n) <_o \Phi_a(n+1)$, et l'on a donc bien $3 \times 5^a \in \mathcal{O}$. ■

PREUVE DU THÉORÈME 5.10. Soit $<_R \subseteq \mathbb{N} \times \mathbb{N}$ un bon ordre calculable sur $\mathbb{N}$. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ la fonction du lemme précédent. On définit une fonction $g : \mathbb{N} \rightarrow \mathbb{N}$ qui sur a calcule le code e_a de l'ensemble

$$W_{e_a} = \{g(b) : b <_R a\}$$

et renvoie $f(e_a)$. Notons que l'existence de la fonction g fait appel au théorème du point fixe.

Voyons ce que donne la fonction g sur le premier élément de $<_R$. Soit a le plus petit élément de $<_R$. Alors, $W_{e_a} = \emptyset$, et l'on a bien $f(e_a) \in \mathcal{O}$ d'après le lemme précédent, de sorte que $g(a) \in \mathcal{O}$. En partant de ce cas initial, on montre facilement par une induction que, pour tout $a \in \mathbb{N}$, on a $g(a) \in \mathcal{O}$, avec $|<_R \upharpoonright_{X_a}| \leq |g(a)|$ où X_a est l'ensemble des éléments plus petits que l'entier a via $<_R$.

On calcule ensuite aisément le code e de la fonction qui énumère $g(a)$ pour tout $a \in \mathbb{N}$ et l'on renvoie $f(e)$. On a alors $|<_R| \leq |f(e)|$. Comme les ordinaux constructifs sont clos par le bas, l'ordinal $|<_R|$ est donc constructif. ■

Notons que l'on n'a pas de conversion effective du code d'un ordinal calculable $<_R$ en un code a d'ordinal constructif tel que $|<_R| = |a|$. On a en effet à la place seulement $|<_R| \leq |a|$, et il est possible de montrer que l'égalité ne peut être obtenue de manière uniforme.

Exercice 5.12. (★) Montrer l'existence d'une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ totale calculable telle que $f(n)$ code toujours pour un ordinal calculable via une relation $<_{f(n)}$

telle que $n \in \emptyset' \rightarrow |<_{f(n)}| = \omega + 1$ et telle que $n \notin \emptyset' \rightarrow |<_{f(n)}| = \omega$.

En déduire l'impossibilité de transformer uniformément un code d'ordinal calculable en un code constructif pour le même ordinal. ◇

5.3. Représentation des ordinaux par des arbres

Il existe une autre représentation usuelle des ordinaux dénombrables, facile à manipuler et bien utile dans de nombreux cas : via des arbres de l'espace de Baire $\mathbb{N}^{\mathbb{N}}$, comme introduits dans la section 8-7.

On rappelle que la manipulation des arbres et des chaînes de l'espace de Baire est analogue à celle de l'espace de Cantor, pour lequel on reprend les mêmes notations. On en introduit deux nouvelles, qui seront utilisées de temps à autre dans ce qui suit.

Notation

Soit T un arbre avec $\sigma \in T$.

- (1) la notation $T \upharpoonright_{\sigma}$ désigne l'arbre des nœuds de T comparables avec σ .
- (2) La notation $T \downarrow_{\sigma}$ désigne l'arbre $\{\tau \in \mathbb{N}^{<\mathbb{N}} : \sigma\tau \in T\}$, c'est-à-dire l'arbre $T \upharpoonright_{\sigma}$ mais en « prenant σ comme racine ».

Définition 5.13. Un arbre $T \subseteq \mathbb{N}^{<\mathbb{N}}$ est *bien fondé* s'il ne contient pas de chemins infinis, c'est-à-dire si $[T]$ est vide. Pour T bien fondé et $\sigma \in T$, on définit par induction $|\sigma| = \sup\{|\sigma n| + 1 : n \in \mathbb{N}, \sigma n \in T\}$. On définit enfin $|T| = |\epsilon|$, où ϵ est la racine de T . On dira parfois que $|T|$ est la *hauteur* de T . $\diamond$

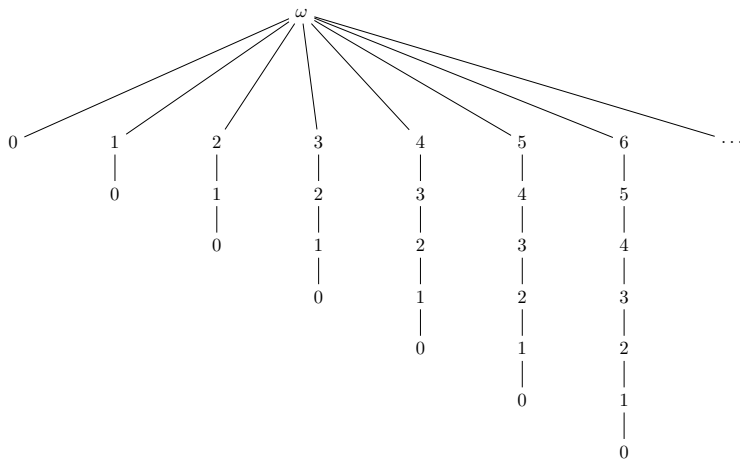


FIGURE 5.14 – Représentation de l'ordinal ω par un arbre T . On a noté pour chaque nœud σ l'ordinal correspondant à σ , c'est-à-dire l'ordinal $|T \upharpoonright_{\sigma}|$.

Notons que la valeur $|\sigma|$ du nœud σ d'un arbre bien fondé T dépend de T . Si jamais il y a une possible ambiguïté, nous utiliserons plutôt $|T \upharpoonright_{\sigma}|$, qui

est égal à $|\sigma|$ et se place explicitement dans le contexte de T .

Pour T un arbre mal fondé, la valeur $|T|$ n'est pas définie, mais on l'utilisera parfois par abus de notation en la considérant plus grande que tous les ordinaux.

Notation

On note $\mathcal{T}$ l'ensemble des codes c.e. d'arbres bien fondés. On note $\mathcal{T}_{<\alpha}$ (resp. $\mathcal{T}_{=\alpha}$) l'ensemble des codes c.e. d'arbres $T \in \mathcal{T}$ tels que $|T| < \alpha$ (resp. tels que $|T| = \alpha$).

Il est à relever que les nœuds de T forment un ordre bien fondé via leur relation de suffixe : $\sigma < \tau$ ssi $\tau \prec \sigma$. Via cet ordre $<$ la définition 5.13 est équivalente à

$$|\sigma| = \sup\{|\tau| + 1 : \tau < \sigma, \tau \in T\}.$$

Si cet ordre n'est pas total, il permet néanmoins de représenter des ordinaux. On aura parfois besoin de « rendre total » la relation d'ordre sur les nœuds d'un arbre bien fondé T . Cela peut se faire via l'ordre de Kleene-Brouwer. On définit $\sigma < \tau$ pour $\sigma, \tau \in T$ ssi $\tau \prec \sigma$ ou alors si σ, τ ne sont pas préfixes l'un de l'autre et que σ est lexicographiquement plus petit que τ . On vérifie facilement que l'ordre ainsi donné est total sur les nœuds de T , et qu'il s'agit d'un bon ordre quand T est bien fondé.

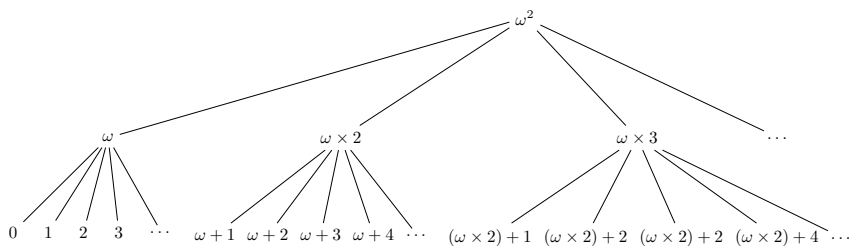


FIGURE 5.15 – Représentation de l'ordinal ω^2 avec l'ordre de Kleene-Brouwer sur les arbres

Voyons formellement l'ordinal associé aux nœuds d'un arbre via l'ordre de Kleene-Brouwer.

Définition 5.16. Pour un arbre T bien fondé, l'ordre $<_{KB}$ de Kleene-Brouwer sur les éléments de T est défini par $\sigma <_{KB} \tau$ si $\tau \prec \sigma$ ou alors si σ, τ ne sont pas préfixes l'un de l'autre et si σ est lexicographiquement plus petit que τ . Pour un nœud $\sigma \in T$, on définit par induction :

$$|\sigma|_{KB} = \sup\{(|\tau|_{KB} + 1) : \tau <_{KB} \sigma, \tau \in T\}.$$

On définit enfin $|T|_{\text{KB}} = |\epsilon|_{\text{KB}}$, où ϵ est la racine de T . ◇

Exercice 5.17. (★) Montrer par induction que l'on a toujours $|T| \leq |T|_{\text{KB}}$, pour un arbre T bien fondé. ◇

Montrons à présent que les arbres c. e. sont des représentations d'ordinaux calculables.

Proposition 5.18. Un ordinal est calculable si, et seulement si, il est égal à $|T|$ pour un arbre c. e. $T \subseteq \mathbb{N}^{<\mathbb{N}}$. On peut uniformément transformer le code d'un bon ordre $<_R$ en un code d'un arbre c. e. T tel que $|<_R| \leq |T|$, et réciproquement, on peut uniformément transformer le code d'un arbre c. e. T en un code de bon ordre $<_R$ tel que $|T| \leq |<_R|$. ★

PREUVE. Soit $<_R$ un bon ordre calculable sur $\mathbb{N}$. On énumère dans T tous les nœuds n , pour $n \in \mathbb{N}$. Puis, inductivement, pour chaque nœud de la forme σa énuméré dans T , on énumère comme fils de σa tous les nœuds σab pour b tel que $b <_R a$. L'arbre T ainsi défini est bien fondé, car un chemin infini de T induirait une suite infinie $\dots <_R a_2 <_R a_1 <_R a_0$, ce qui ne peut arriver car $<_R$ est un bon ordre. On vérifie facilement par induction que $|<_R| \leq |T|$.

Pour la réciproque, il suffit de considérer l'ordre de Kleene-Brouwer sur les nœuds de T . Par l'exercice 5.17, on a $|T| \leq |T|_{\text{KB}}$ et, via un codage des éléments de $\mathbb{N}^{<\mathbb{N}}$, on énumère un bon ordre $<_R$ sur une partie $A \subseteq \mathbb{N}$ telle que $|T|_{\text{KB}} = |<_R|$. Le lemme 5.8 et le lemme 5.9 permettent de conclure. ■

Notons qu'en suivant ce même principe, un ordinal est dénombrable si, et seulement si, il est égal à $|T|$ pour un arbre bien fondé T — mais qui n'est pas nécessairement c. e.

6. Relativisation

Les notions d'ordinaux constructibles et calculables se relativisent à un oracle.

Définition 6.1. Soit $X \in 2^{\mathbb{N}}$. On définit l'ensemble $\mathcal{O}^X$ de la manière suivante.

1. $1 \in \mathcal{O}^X$, avec $|1| = \emptyset$.
2. Si $a \in \mathcal{O}^X$, alors $2^a \in \mathcal{O}^X$ avec $|2^a| = \text{succ}(|a|)$. De plus $a <_o 2^a$, et $b <_o a$ implique $b <_o 2^a$ pour tout b .
3. Si Φ_e est une fonctionnelle totale sur X avec

$$\Phi_e(X, n) \in \mathcal{O}^X \text{ et } \Phi_e(X, n) <_o \Phi_e(X, n+1), \text{ pour tout } n,$$

alors $3 \times 5^e \in \mathcal{O}^X$, avec $|3 \times 5^e| = \sup_n |\Phi_e(X, n)|$. De plus, pour tout a , si $a <_o \Phi_e(X, n)$ pour un certain n , alors $a <_o 3 \times 5^e$.

Un ordinal α est *X-constructif* s'il existe $a \in \mathcal{O}^X$ tel que $\alpha = |a|$. ◇

Étant donné X fixé, l'ordre $<_o$ sur les éléments de $\mathcal{O}^X$ n'est bien entendu pas le même que l'ordre $<_o$ sur les éléments de $\mathcal{O}$. Aussi écrira-t-on parfois $<_o^X$ pour lever toute ambiguïté quand il y a lieu, tout comme il nous arrivera d'écrire $|a|^X$ à la place de $|a|$ pour un ordinal $a \in \mathcal{O}^X$.

Définition 6.2

Un ordinal α est *X-calculable* s'il est fini ou s'il existe $<_R \subseteq \mathbb{N} \times \mathbb{N}$ un bon ordre *X-calculable* sur $\mathbb{N}$ tel que $|<_R| = \alpha$. ◇

Notation

On note $\mathcal{T}^X$ l'ensemble des codes d'arbres *X-c.e.* bien fondés de l'espace de Baire.

Les différentes équivalences vues jusqu'ici se relativisent sans problème à un oracle X .

Théorème 6.3

Un ordinal est X-constructif si, et seulement si, il est X-calculable, ou ce qui revient au même s'il est égal à $|T|$ pour un arbre X-c.e. $T \in \mathcal{T}^X$.

Souvenons-nous à présent du symbole ω_1^{ck} désignant le plus petit ordinal non constructible, ou de manière équivalente non calculable. Là encore, la notion se relativise.

Notation

On note ω_1^X le plus petit ordinal non *X-calculable*.

Notons que si ω_1^X peut être supérieur à ω_1^{ck} , il n'en reste pas moins dénombrable. Réciproquement, tout ordinal dénombrable est calculable pour un certain X , en considérant par exemple un oracle qui code pour une relation d'ordre représentant l'ordinal. Cela nous donne l'énoncé qui suit.

Proposition 6.4. L'ordinal ω_1 est le supremum des ordinaux *X-calculables* pour un certain oracle X . ★

Chapitre 28

Ensembles hyperarithmétiques

De la même manière que les classes Σ_n^0 et Π_n^0 de la hiérarchie borélienne peuvent être généralisées en des classes Σ_α^0 et Π_α^0 pour tout ordinal α dénombrable afin de définir l'intégralité des boréliens, la hiérarchie arithmétique peut également être itérée de long des ordinaux calculables, produisant une nouvelle classe d'ensembles appelés *ensembles hyperarithmétiques*. Nous verrons à travers ce chapitre et le chapitre 29, qui suit, que cette classe est extrêmement robuste, dans le sens où elle admet de nombreuses caractérisations, en termes d'itérations du saut Turing, de singletons Π_2^0 , ou de modulus.

Ce chapitre va nécessiter le développement d'un arsenal de codages de boréliens et d'ensembles d'entiers à l'aide d'entiers naturels. Les preuves feront souvent appel à des manipulations un peu laborieuses de ces codages, mais le lecteur sera récompensé par l'ajout des ensembles hyperarithmétiques à sa boîte à outils. Ce développement culminera avec deux résultats : d'une part, le corollaire 1.14, qui affirme que les ensembles H_a pour $a \in \mathcal{O}$ — autrement dit les α -itérations du saut Turing de $\emptyset$ — sont complets aux différents niveaux de la hiérarchie de Kleene, et d'autre part le théorème 4.4, qui caractérise les classes Σ_α^0 de la hiérarchie borélienne effective en termes des α -itérations du saut Turing.

1. Hiérarchie de Kleene

On généralise la définition 5-1.1 des ensembles arithmétiques de la manière suivante.

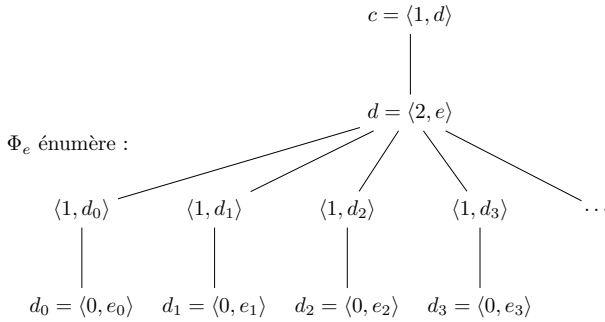


FIGURE 1.2 – Le dépliage le long d'un arbre d'un Π_2^0 -code c , qui code pour le complémentaire du Σ_2^0 codé par $\langle 2, e \rangle$, où Φ_e énumère des Π_1^0 -codes qui correspondent eux-mêmes à des complémentaires d'ensembles Σ_1^0 codés par chaque e_i .

Définition 1.1. La hiérarchie hyperarithmétique de Kleene est définie par induction sur les ordinaux.

- ▷ Un Σ_1^0 -code est donné par une paire $\langle 0, e \rangle$. L'ensemble $A \subseteq \mathbb{N}$ correspondant est donné par $A = W_e$.
- ▷ Un Π_α^0 -code est donné par une paire $\langle 1, e \rangle$, où e est un Σ_α^0 -code. L'ensemble $A \subseteq \mathbb{N}$ correspondant est donné par $A = \mathbb{N} \setminus B$, où B est l'ensemble correspondant au code e .
- ▷ Un Σ_α^0 -code est donné par une paire $\langle 2, e \rangle$, où W_e est non vide et énumère des $\Pi_{\beta_n}^0$ -codes pour $\beta_n < \alpha$, avec $\sup_n(\beta_n + 1) = \alpha$. L'ensemble $A \subseteq \mathbb{N}$ correspondant est donné par $\bigcup_n A_n$, où A_n est l'ensemble correspondant au n -ième code énuméré par W_e .

On dit qu'un ensemble A est Σ_α^0 (resp. Π_α^0) s'il correspond à un Σ_α^0 -code e (resp. Π_α^0 -code e). On dit que A est Δ_α^0 s'il est à la fois Σ_α^0 et Π_α^0 . On dira enfin qu'un ensemble A est *hyperarithmétique* s'il est Σ_α^0 pour un certain ordinal α . ◇

Souvenons-nous des ensembles H_a de la définition 27-1.3, correspondants aux versions itérées du saut pour $a \in \mathcal{O}$. Pour tout $a \in \mathcal{O}$ tel que $|a| = n$, l'ensemble $H_a = \emptyset^{(n)}$ est Σ_n^0 . Pour $a = 3 \times 5^e$ tel que $|a| = \omega$, l'ensemble H_a est quant à lui Δ_ω^0 . En effet, on a $\langle n, m \rangle \in H_a$ ssi $n \in H_{\Phi_e(m)}$, et donc aussi $\langle n, m \rangle \notin H_a$ ssi $n \notin H_{\Phi_e(m)}$. Les ensembles

$$\{\langle n, m \rangle : n \in H_{\Phi_e(m)}\} \quad \text{et} \quad \{\langle n, m \rangle : n \notin H_{\Phi_e(m)}\}$$

sont $\Delta_1^0(H_{\Phi_e(m)})$ uniformément en m , et donc $\Pi_{|\Phi_e(m)|+1}^0$ uniformément en m , d'après le théorème 5-5.5. Cela nous permet de faire une définition Σ_ω^0

de H_a ainsi que de son complémentaire. L'ensemble H_{2^a} est quant à lui Σ_ω^0 , et cela s'itère sans problème sur les éléments de $\mathcal{O}$. Pour être tout à fait rigoureux, nous aurons toutefois besoin de quelques lemmes de manipulation des codes Σ_α^0 et Π_α^0 , à commencer par une généralisation des deux lemmes 5-1.7 et 5-1.6.

Lemme 1.3. Les ensembles Σ_α^0 et Π_α^0 sont stables par réunions et intersections finies. ★

PREUVE. On peut récursivement utiliser les relations suivantes pour des ensembles quelconques A , B et A_n , B_n , avec $n \in \mathbb{N}$:

$$\begin{aligned} (\bigcup_{n \in \mathbb{N}} A_n) \cap (\bigcup_{n \in \mathbb{N}} B_n) &= \bigcup_{n_1, n_2 \in \mathbb{N}} (A_{n_1} \cap A_{n_2}); \\ (\bigcup_{n \in \mathbb{N}} A_n) \cup (\bigcup_{n \in \mathbb{N}} B_n) &= \bigcup_{n \in \mathbb{N}} C_n, \text{ avec } C_{2n} = A_n \text{ et } C_{2n+1} = B_n; \\ A \cap (\bigcup_{n \in \mathbb{N}} B_n) &= \bigcup_{n \in \mathbb{N}} A \cap B_n; \\ A \cup (\bigcup_{n \in \mathbb{N}} B_n) &= \bigcup_{n \in \mathbb{N}} C_n, \text{ avec } C_0 = A \text{ et } C_{n+1} = B_n; \\ (\mathbb{N} \setminus A) \cap (\mathbb{N} \setminus B) &= \mathbb{N} \setminus (A \cup B); \\ (\mathbb{N} \setminus A) \cup (\mathbb{N} \setminus B) &= \mathbb{N} \setminus (A \cap B). \end{aligned}$$

En utilisant le théorème du point fixe, on crée les fonctions calculables de réunion et d'intersection de deux codes, qui propagent leur application en utilisant les relations ci-dessus. ■

Lemme 1.4. Les ensembles Σ_α^0 sont stables par réunions dénombrables uniformes. ★

PREUVE. Étant donné des Σ_α^0 -codes $(\langle 2, e_n \rangle)_{n \in \mathbb{N}}$, il suffit bien entendu de créer le code $\langle 2, e \rangle$, où e énumère $\bigcup_n W_{e_n}$. ■

Nous terminons par deux lemmes de manipulation des codes qui nous seront utiles de temps à autre.

Lemme 1.5. Soit $x \in \mathbb{N}$. Il existe une fonction calculable qui sur le code Σ_α^0 (resp. Π_α^0) d'un ensemble A renvoie le code Σ_α^0 (resp. Π_α^0) d'un ensemble B tel que $B = \mathbb{N}$ si $x \in A$, et $B = \emptyset$ sinon. ★

PREUVE. Il suffit de « déplier » le code et de remplacer chaque feuille correspondant à un ensemble W_k par un ensemble c. e. égal à $\mathbb{N}$ si $x \in W_k$ et égal à $\emptyset$ sinon.

La procédure fonctionne trivialement pour les codes Σ_1^0 . Supposons que la procédure fonctionne pour les codes Σ_α^0 . Soit $a = \langle 1, b \rangle$ un code Π_α^0 d'un ensemble $A = \mathbb{N} \setminus B$, où b est un code de B . Alors, la procédure appliquée à b renvoie $\mathbb{N}$ si $x \in B$ et $\emptyset$ sinon, ce qui appliqué à a passe au complémentaire et donne bien $\mathbb{N}$ si $x \in A$, et $\emptyset$ sinon. On vérifie de même par induction que l'on a le résultat attendu avec les codes de la forme $\langle 2, b \rangle$ correspondant à des réunions. ■

Lemme 1.6. Soit A un ensemble Σ_α^0 et soit f une fonction calculable. Alors, $f(A)$ est un ensemble Σ_α^0 . ★

PREUVE. D'après le lemme 1.5, on peut créer pour tout x le code Σ_α^0 d'un ensemble B_x égal à $\mathbb{N}$ si $x \in A$, et égal à $\emptyset$ sinon.

On a $f(A) = \bigcup_{x \in \mathbb{N}} \{f(x)\} \cap B_x$. D'après le lemme 1.3, l'ensemble $\{f(x)\} \cap B_x$ est Σ_α^0 uniformément en x . D'après le lemme 1.4, l'ensemble $f(A)$ est donc lui aussi Σ_α^0 . ■

Nous avons à présent tous les outils nécessaires pour attaquer la complexité des ensembles issus de l'itération transfinie du saut Turing.

Proposition 1.7. Pour tout ordinal $\alpha \geq \omega$ et tout $a \in \mathcal{O}_{=\alpha}$, l'ensemble H_{2^a} est Σ_α^0 uniformément en a . ★

PREUVE. On montre la proposition par induction sur les éléments de $\mathcal{O}$. L'induction commence avec les ensembles H_a pour $|a|$ fini, qui sont tous $\Sigma_{|a|}^0$ d'après la proposition 5-5.3. Nous verrons dans le reste de la preuve (comme cela a été esquissé dans le paragraphe suivant la définition 1.1) que cela amène à avoir que H_{2^a} est $\Sigma_{|a|}^0$ pour $a \in \mathcal{O}_{=\omega}$. Supposons la proposition vraie pour tout $b <_o a$ et montrons que H_a est $\Delta_{|a|}^0$.

Si $a = 2^b$ code pour un ordinal successeur, alors par hypothèse d'induction l'ensemble H_a est $\Sigma_{|b|}^0$. Donc, lui et son complémentaire sont tous deux $\Delta_{|a|}^0$. Si $a = 3 \times 5^e$ code pour un ordinal limite, on a

$$\langle n, m \rangle \in H_a \Leftrightarrow n \in H_{\Phi_e(m)}, \quad \text{ainsi que} \quad \langle n, m \rangle \notin H_a \Leftrightarrow n \notin H_{\Phi_e(m)}.$$

Vu que l'ensemble $H_{\Phi_e(m)}$ est par induction uniformément $\Sigma_{|\Phi_e(m)|}^0$, on a d'après le lemme 1.6 appliqué à chaque $H_{\Phi_e(m)}$, un $\Sigma_{|a|}^0$ -code de H_a ainsi que de son complémentaire. Donc, H_a est un ensemble $\Delta_{|a|}^0$.

Dans les deux cas, H_a est un ensemble $\Delta_{|a|}^0$. Par ailleurs, on a $n \in H_{2^a}$ si, et seulement si, $\Phi_n(H_a, n) \downarrow$, ce qui est une condition $\Sigma_1^0(H_a)$. Comme l'ensemble H_a et son complémentaire sont $\Delta_{|a|}^0$, alors l'ensemble H_{2^a} est $\Sigma_{|a|}^0$ via le prédicat $n \in H_{2^a}$ si, et seulement si,

$$\exists \sigma \exists t \Phi_n(\sigma, n)[t] \downarrow \wedge \forall i < |\sigma| \left((\sigma(i) = 0 \wedge i \notin H_a) \vee (\sigma(i) = 1 \wedge i \in H_a) \right).$$

La manière d'obtenir réellement un $\Sigma_{|a|}^0$ -code pour H_{2^a} à partir de la formule ci-dessus n'est pas forcément claire, et nous détaillons donc comment procéder. Étant donné une chaîne σ et un entier $i < |\sigma|$, on peut calculer uniformément le code $e_{\sigma, i}$ d'un ensemble $\Sigma_{|a|}^0$ égal à $\mathbb{N}$ si $\sigma(i) = 0$ et $i \notin H_a$, ou si $\sigma(i) = 1$ et $i \in H_a$, et égal à $\emptyset$ sinon : il suffit d'appliquer le lemme 1.5

à i et $\mathbb{N} - H_a$ si $\sigma(i) = 0$, et à i et H_a si $\sigma(i) = 1$. Soit $B_{e_{\sigma,i}}$ l'ensemble $\Sigma_{|a|}^0$ ainsi décrit. Alors,

$$H_{2^a} = \bigcup_{\sigma} \left(\{n : \Phi_n(\sigma, n) \downarrow\} \cap \bigcap_{i < |\sigma|} B_{e_{\sigma,i}} \right).$$

D'après le lemme 1.3 et le lemme 1.4, il s'agit bien d'un ensemble $\Sigma_{|a|}^0$. ■

Alternance de quantificateurs

Notons que dans la preuve précédente, à partir du fait que H_{2^a} soit $\Sigma_1^0(X)$ avec X un ensemble $\Delta_{|a|}^0$, on obtient une description $\Sigma_{|a|+1}^0$ de H_{2^a} . Ainsi, l'équivalence entre la quantification $\exists$ et la réunion, tout comme celle entre la quantification $\forall$ et l'intersection, se poursuivent « moralement » dans le transfini, mais le langage du premier ordre ne nous permet simplement plus d'en rendre compte formellement (il nous faudrait des phrases de taille infinie).

Complexité des itérations du saut

Notons que pour $a \in \mathcal{O}_{<\omega}$ l'ensemble H_a est $\Sigma_{|a|}^0$ (c'est-à-dire que $\emptyset^{(n)}$ est Σ_n^0), alors que pour $a \in \mathcal{O}_{\geq\omega}$ c'est l'ensemble H_{2^a} qui est $\Sigma_{|a|}^0$. Ce phénomène est dû aux étapes limites.

Nous avons vu antérieurement, avec la proposition 5-5.3, que pour $a \in \mathcal{O}_{<\omega}$, l'ensemble H_a est $\Sigma_{|a|}^0$ -complet. Nous allons à présent démontrer que chaque ensemble H_{2^a} pour $|a| \geq \omega$, est $\Sigma_{|a|}^0$ -complet, c'est-à-dire que chaque ensemble $\Sigma_{|a|}^0$ est many-one réductible à H_{2^a} . Cela demande toutefois un peu de travail, et nous commençons par une transformation effective des Σ_α^0 -codes en arbres bien fondés de hauteur α .

Proposition 1.8. Il existe une fonction totale calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que e est un Σ_α^0 -code de la hiérarchie effective de Kleene si, et seulement si, $f(e) \in \mathcal{T}_{=\alpha}$. ★

La preuve de la proposition précédente n'est pas difficile et nous en donnons simplement ici l'idée générale, en laissant les détails au lecteur. La fonction f renvoie le code de l'arbre correspondant au dépliage du Σ_α^0 -code en omettant simplement le passage au complémentaire de Π_α^0 vers Σ_α^0 . Si à un certain moment on s'aperçoit que e n'est pas un code valide, la fonction renvoie alors le code d'un arbre mal fondé. Notons que cela implique d'avoir ω_1^{ck} comme borne sur les ensembles Σ_α^0 .

Corollaire 1.9

Si un sous-ensemble de $\mathbb{N}$ est Σ_α^0 , alors $\alpha < \omega_1^{ck}$.

Voyons à présent le théorème clef dans la preuve que chaque ensemble H_{2^a} est $\Sigma^0_{|a|}$ -complet.

Théorème 1.10 (Spector [214])

Pour tout ordinal $\alpha < \omega_1^{ck}$ et tout $a \in \mathcal{O}_{=\alpha}$, on a

1. $\mathcal{O}_{<\alpha} \leq_T H_{2^a}$, uniformément en a ;
2. $\mathcal{T}_{<\alpha} \leq_T H_{2^a}$, uniformément en a .

PREUVE. On veut deux fonctionnelles Ψ_1, Ψ_2 telles que pour tout $a \in \mathcal{O}_{=\alpha}$ et $x \in \mathbb{N}$ on a :

- ▷ $x \in \mathcal{O}_{<\alpha} \rightarrow \Psi_1(H_{2^a}, a, x) = 1$ et $x \notin \mathcal{O}_{<\alpha} \rightarrow \Psi_1(H_{2^a}, a, x) = 0$;
- ▷ $x \in \mathcal{T}_{<\alpha} \rightarrow \Psi_2(H_{2^a}, a, x) = 1$ et $x \notin \mathcal{T}_{<\alpha} \rightarrow \Psi_2(H_{2^a}, a, x) = 0$.

La fonction $\Psi_1(H_{2^a}, a, x)$ fait les calculs suivants.

1. Si $a = 1$, alors le calcul renvoie 0, et effectivement $\mathcal{O}_{<|1|}$ est vide.
2. Si $a = 2^b$, alors le calcul renvoie 1 si la condition suivante est vraie (et renvoie 0 sinon) :
 - (a) $\Psi_1(H_{2^b}, b, x) = 1$, ce qui correspond au fait que $x \in \mathcal{O}_{<|b|}$. Il reste à couvrir les cas où $|x| = |b|$;
 - (b) ou $x = 2^y$, $|b|$ est successeur et $\Psi_1(H_{2^b}, b, y) = 1$, ce qui correspond au fait que $x \in \mathcal{O}_{=|b|}$ avec b successeur ;
 - (c) ou $x = 3 \times 5^e$, b est limite et l'énoncé $\Pi_2^0(H_b)$ suivant est vrai : pour tout $n \in \mathbb{N}$, on a $\Phi_e(n) \downarrow_{<_o} \Phi_e(n+1) \downarrow$ et pour tout n il existe $c <_o b$ tel que $\Psi_1(H_{2^c}, c, \Phi_e(n)) = 1$. Ce cas correspond au fait que $x \in \mathcal{O}_{=|b|}$ avec b limite. Notons que l'énoncé $\Pi_2^0(H_b)$ est aussi $\Pi_1^0(H'_b)$. Comme $H_{2^a} = H''_b$, c'est une question à laquelle on peut répondre à l'aide de notre oracle.
3. Si $a = 3 \times 5^e$, alors le calcul renvoie 1 si la condition $\Sigma_1^0(H_a)$ suivante est vraie (et renvoie 0 sinon) : il existe $b <_o a$ tel que $\Psi_1(H_{2^b}, b, x) = 1$. Ce cas correspond au fait que $x \in \mathcal{O}_{<|a|}$, avec a limite. Comme $H_{2^a} = H'_a$, c'est une question à laquelle on peut répondre à l'aide de notre oracle.

On vérifie aisément par induction que le calcul fait bien ce qui est attendu. La fonction $\Psi_2(H_{2^a}, a, x)$ fait quant à elle les calculs suivants.

1. Si $a = 1$, alors $\Psi_2(H_{2^a}, a, x) = 0$, et effectivement $\mathcal{T}_{<|1|}$ est vide.
2. Si $a = 2^b$, soit T l'arbre codé par x . Alors, $\Psi_2(H_{2^a}, a, x) = 1$ ssi l'énoncé $\Pi_1^0(H_{2^b})$ suivant est vrai : pour tout $n \in T$ — autrement dit, pour tout nœud de profondeur 1 dans T — et tout x_n où x_n est le code de $T \upharpoonright_n$, on a $\Psi_2(H_{2^b}, b, x_n) = 1$. Cela correspond au fait que pour chaque sous-arbre $T \upharpoonright_n$ on ait $|T \upharpoonright_n| < |b|$, et donc au fait que $|T| < |a|$.

3. Si $a = 3 \times 5^e$, alors $\Psi_2(H_{2^a}, a, x) = 1$ ssi l'énoncé $\Sigma_1^0(H_a)$ suivant est vrai : il existe $b <_o a$ tel que $\Psi_2(H_{2^b}, b, x) = 1$. Ce cas correspond au fait que $x \in \mathcal{T}_{<|a|}$ avec a limite. ■

Le théorème précédent ne peut pas être amélioré. Pour beaucoup d'ordinaux α , l'ensemble $\mathcal{O}_{<\alpha}$ sera en fait calculable par des ensembles H_a pour $a \in \mathcal{O}_{<\alpha}$, mais pour les ordinaux α qui sont limites de limites, il est possible de montrer que l'ensemble $\mathcal{T}_{<\alpha}$ est Σ_α^0 mais pas Π_α^0 , et que l'ensemble $\mathcal{T}_{<\alpha+1}$ est $\Pi_{\alpha+1}^0$ mais pas $\Sigma_{\alpha+1}^0$. Les bornes précises sont données dans l'exercice suivant.

Exercice 1.11. (**) Montrez que pour tout $\alpha = 0$ ou limite, et pour tout $\mathbb{k}, \mathbb{p} \in \omega$:

1. l'ensemble $\mathcal{T}_{<\omega(\alpha+\mathbb{k})}$ est $\Sigma_{\alpha+2\mathbb{k}}^0$ -complet ;
2. l'ensemble $\mathcal{T}_{\leq \omega(\alpha+\mathbb{k})+\mathbb{p}}$ est $\Pi_{\alpha+2\mathbb{k}+1}^0$ -complet. ◇

Montrons à présent que les ensembles de la forme H_a sont complets pour leurs classes de complexité.

Théorème 1.12

Soit $\alpha < \omega_1^{ck}$, et soient $a \in \mathcal{O}_{=\alpha}$ et A un ensemble Σ_α^0 . Alors, A est many-one réductible à H_{2^a} , uniformément en a et en un Σ_α^0 -code de A .

Nous aurons besoin du lemme suivant.

Lemme 1.13. Soit e un Σ_α^0 -code. Soit $b \in \mathcal{O}$ tel que $\alpha + 2 \leq |b|$. Alors, H_b peut retrouver uniformément en b et en e un code $a <_o b$ tel que $|a| = \alpha$.★

PREUVE. On commence par transformer e en un code $f(e) \in \mathcal{T}_{=\alpha}$ via la proposition 1.8. Étant donné b , on énumère tous les codes a tels que $2^a <_o b$ tout en cherchant le plus petit tel que $f(e) \in \mathcal{T}_{=|a|}$, c'est-à-dire l'unique a tel que :

- (1) $f(e) \in \mathcal{T}_{<\text{succ}(|a|)}$,
- (2) $\forall c <_o a$ $f(e) \notin \mathcal{T}_{<\text{succ}(|c|)}$.

Pour (1), d'après le théorème 1.10, on a $\mathcal{T}_{<\text{succ}(|a|)} \leq_T H_{2^{2^a}}$, qui est au pire égal à H_b . Pour (2), la question requiert — toujours en utilisant le théorème 1.10 — H'_{2^a} , qui dans le pire des cas est égal à H_b . L'oracle H_b suffit donc pour identifier a . ■

PREUVE DU THÉORÈME 1.12. On cherche à définir une fonction totale calculable $f : \mathbb{N}^3 \rightarrow \mathbb{N}$ telle que pour tout $\alpha < \omega_1^{ck}$, pour tout Σ_α^0 -code e d'un ensemble A on a $x \in A$ ssi $f(a, e, x) \in H_{2^a}$.

La fonction $f(a, e, x)$ renvoie le code de la fonctionnelle dont on suppose qu'elle dispose de l'oracle H_a et qui traite les cas suivants.

1. Si $\alpha = 1$ — le cas initial correspondant aux ensembles Σ_1^0 —, alors la fonctionnelle s'arrête sur sa propre entrée ssi x appartient à l'ensemble Σ_1^0 de code e .
2. Sinon, si α est limite, la fonctionnelle énumère les $\Sigma_{\beta_n}^0$ -codes e_n d'ensembles A_n tels que $A = \bigcup_n (\mathbb{N} \setminus A_n)$ et avec $\sup_n (\beta_n + 1) = \alpha$. Notons que l'on a $\beta_n + 2 \leq \alpha$ pour chaque n . On peut donc appliquer le lemme 1.13 avec l'oracle H_a pour trouver pour chaque e_n un code $b_n <_o a$ tel que $|b_n| = \beta_n$. La fonctionnelle s'arrête sur sa propre entrée ssi la condition $\Sigma_1^0(H_a)$ suivante est vraie : $\exists n f(b_n, e_n, x) \notin H_{b_n}$.
3. Sinon, si $\alpha = \text{succ}(\beta)$ avec $a = 2^b$, la fonctionnelle énumère les $\Sigma_{\beta_{n,m}}^0$ -codes $e_{n,m}$ d'ensembles $A_{n,m}$, avec

$$\sup_m (\beta_{n,m} + 1) = \alpha_n \leq \beta, \quad \sup_n (\alpha_n + 1) = \alpha \quad \text{et} \quad A = \bigcup_n \bigcap_m A_{n,m}.$$

Notons que l'on a $\beta_{n,m} + 2 \leq \alpha$ pour chaque n, m . On peut donc appliquer le lemme 1.13 avec l'oracle H_a pour trouver pour chaque $e_{n,m}$ un code $b_{n,m} <_o a$ tel que $|b_{n,m}| = \beta_{n,m}$. La fonctionnelle s'arrête sur sa propre entrée si, et seulement si, la condition $\Sigma_2^0(H_b)$ suivante est vraie : $\exists n \forall m f(b_{n,m}, e_{n,m}, x) \in H_{b_{n,m}}$. ■

Nos efforts sont à présent récompensés par trois corollaires qui viennent structurer un peu nos connaissances des ensembles hyperarithmétiques. Le premier d'entre eux est simplement la juxtaposition du théorème 1.12 et de la proposition 1.7.

Corollaire 1.14

Pour tout $\alpha < \omega_1^{ck}$ tel que $\alpha \geq \omega$, et tout $a \in \mathcal{O}_{=\alpha}$, l'ensemble H_{2^a} est Σ_α^0 -complet uniformément en a .

Le deuxième corollaire nous indique qu'aucun niveau n'est superflu dans la hiérarchie de Kleene : pour tout $\alpha < \omega_1^{ck}$ il existe un ensemble Σ_α^0 qui n'est pas Π_α^0 .

Corollaire 1.15

La hiérarchie de Kleene est stricte.

PREUVE. Pour un ordinal calculable $\alpha \geq \omega$, pour $a \in \mathcal{O}_{=\alpha}$, chaque ensemble H_{2^a} est Σ_α^0 . Supposons par l'absurde que l'un d'eux soit aussi Π_α^0 . Alors, son complémentaire est Σ_α^0 . On a donc d'après le théorème 1.12 une fonction totale calculable f telle que $e \notin H_{2^a}$ ssi $f(e) \in H_{2^a}$, c'est-à-dire $\Phi_e(H_a, e) \uparrow$ ssi $\Phi_{f(e)}(H_a, f(e)) \downarrow$.

Notons que l'on a aussi une fonction totale calculable g telle que $\Phi_e(H_a, e) \downarrow$ implique $\forall n \ \Phi_{g(e)}(H_a, n) \downarrow$, et $\Phi_e(H_a, e) \uparrow$ implique $\forall n \ \Phi_{g(e)}(H_a, n) \uparrow$. Donc, en particulier, $\Phi_e(H_a, e) \uparrow$ ssi $\Phi_{g(f(e))}(H_a, e) \downarrow$. D'après le théorème du point fixe, il existe e tel que $\Phi_e(H_a, e) = \Phi_{g(f(e))}(H_a, e)$, ce qui est une contradiction. ■

Le dernier corollaire va finalement nous permettre de nous abstraire un peu de notre système de notation pour les ordinaux.

Corollaire 1.16

Soit $\alpha < \omega_1^{ck}$, et soient $a, b \in \mathcal{O}_{=\alpha}$. Alors, $H_{2^a} \equiv_m H_{2^b}$.

PREUVE. Le corollaire est évident quand α est un ordinal fini. Pour $\alpha \geq \omega$, il suffit de constater que chaque H_{2^a} est $\Sigma_{|a|}^0$ -complet. ■

Remarquons que les équivalences many-one ne fonctionnent que pour les cas successeurs. Qu'en est-il des ensembles H_a pour a un ordinal limite ? Ces ensembles sont $\Delta_{|a|}^0$ et si l'on a toujours $H_a \equiv_T H_b$ pour $a, b \in \mathcal{O}_{=\alpha}$ avec α limite, on n'aura pas nécessairement $H_a \equiv_m H_b$. Moschovakis [162] a montré que l'équivalence tient malgré tout quand α est de la forme $\beta + \omega$, mais que la structure des degrés many-one au sein du degré Turing d'un H_a est très chaotique dans le cas où $|a|$ est une limite de limite.

On utilisera le corollaire précédent pour considérer les itérations α du saut à degré many-one près (ou à degré Turing près dans le cas limite).

Notation

Pour $\alpha < \omega_1^{ck}$, on écrira parfois $\emptyset^{(\alpha)}$ pour signifier l'ensemble H_a pour un certain $a \in \mathcal{O}_{=\alpha}$ sans préciser davantage.

En particulier, chaque ensemble $\emptyset^{(\alpha+1)}$ est à degré many-one près l'ensemble Σ_α^0 -complet, et pour α limite l'ensemble $\emptyset^{(\alpha)}$ est bien défini à degré Turing près.

2. Les singletons Π_2^0

Nous avons vu dans la section 26-1 que les singletons Π_2^0 dans l'espace de Cantor dépassaient la hiérarchie arithmétique, et notamment que l' ω -saut Turing de $\emptyset$ était un singleton Π_2^0 . Nous allons maintenant voir que l'on peut trouver des singletons Π_2^0 à tous les niveaux de la hiérarchie de Kleene, et en particulier que tout ensemble hyperarithmétique est calculable par un

singleton Π_2^0 . Nous verrons plus tard dans la section 29-5 que la réciproque est vraie.

Théorème 2.1

Il existe $\mathcal{P} \subseteq \mathbb{N} \times 2^{\mathbb{N}}$ une classe Π_2^0 telle que pour tout $a \in \mathcal{O}$ la classe

$$\{X : (a, X) \in \mathcal{P}\}$$

est le singleton H_a .

PREUVE. Souvenons-nous de $\mathcal{S}$ de l'exemple 17-3.7 : la classe Π_2^0 contenant exactement les ensembles qui sont des sauts Turing d'autres ensembles. Notons également Ψ la fonctionnelle telle que $\Psi(X') = X$ pour tout X .

En utilisant le théorème du point fixe, on définit alors la classe $\mathcal{P}$ comme étant :

$$\mathcal{P} = \left\{ (a, X) : \begin{array}{l} a=1 \text{ et } X=\emptyset \text{ ou} \\ a=2^b \text{ et } X \in \mathcal{S} \text{ et } (b, \Psi(X)) \in \mathcal{P} \text{ ou} \\ a=3 \times 5^b \text{ et } X = \bigoplus_n X_n \text{ avec } \forall n (\Phi_b(n), X_n) \in \mathcal{P} \end{array} \right\}.$$

Voyons un peu plus formellement comment est définie la classe $\mathcal{P}$. Pour tout $n \in \mathbb{N}$, soit $S_n \subseteq 2^{<\mathbb{N}}$ l'ensemble Σ_1^0 tel que $\mathcal{S} = \bigcap_n [S_n]$. On peut supposer sans perte de généralité que chaque S_n est clos par suffixe : si $\sigma \in S_n$, alors toute extension de σ appartient à S_n . Afin d'utiliser le théorème du point fixe, on définit la fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ qui sur un code e renvoie le code $f(e)$ tel que :

$$\begin{aligned} \Phi_{f(e)}(a, n, \sigma) \downarrow &\leftrightarrow \vee \begin{array}{l} (a = 1 \wedge \sigma \prec 0^\infty) \\ (a = 2^b \wedge \sigma \in S_n \wedge \Phi_e(b, n, \Psi(\sigma)) \downarrow) \\ (a = 3 \times 5^b \wedge \Phi_e(\Phi_b(n), n, \sigma_n), \text{ où } \sigma = \bigoplus_m \sigma_m) \end{array} \end{aligned}$$

D'après le théorème du point fixe, soit e tel que $\Phi_{f(e)} = \Phi_e$. On montre alors par induction que pour tout $a \in \mathcal{O}$, la classe Π_2^0 donnée par

$$\bigcap_n [\{\sigma : \Phi_e(a, n, \sigma) \downarrow\}]$$

contient l'ensemble H_a , et uniquement lui. ■

Corollaire 2.2

Tout ensemble hyperarithmétique est calculable par un singleton Π_2^0 .

PREUVE. D'après le théorème 1.12 et le théorème 2.1. ■

À quoi ressemble $\{X : (a, X) \in \mathcal{P}\}$ pour un élément $a \notin \mathcal{O}$? Pour la plupart de ces éléments, la classe en question sera vide, mais nous verrons avec le théorème 31-3.3 que ce n'est pas toujours le cas : il existe des éléments $a \notin \mathcal{O}$, mais qui ressemblent à des éléments de $\mathcal{O}$. Plus précisément, il existe des éléments $a \notin \mathcal{O}$ qui ressemblent à des éléments de $\mathcal{O}$: ils sont de la forme 2^b ou 3×5^e , et récursivement sur les éléments « plus petits qu'eux ».

Simplement, cette énumération contiendra une suite infinie d'éléments

$$\dots < b_4 < b_3 < b_2 < b_1 < a.$$

Ce phénomène-là, aux conséquences fascinantes, sera étudié ultérieurement dans les sections 31-3 et 31-4.

Voyons à présent une notion introduite par Groszek et Slaman, et qui comme nous le verrons avec le théorème 29-5.4 caractérise exactement les ensembles hyperarithmétiques. La définition suivante peut être vue comme une extension de la définition de modulus des ensembles Δ_2^0 (voir la définition 4-7.7).

Définition 2.3 (Groszek et Slaman [79]). Un ensemble X admet un *modulus* s'il existe une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que, pour toute fonction $g \geq f$, on a $g \geq_T X$. ◇

Théorème 2.4

Tout ensemble hyperarithmétique admet un modulus.

PREUVE. Soit $\bigcap_n \mathcal{U}_n$ une classe Π_2^0 contenant un unique élément X . Soit une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $X \in \mathcal{U}_n[f(n)]$ pour tout n . Pour tout $g \geq f$, la classe $\bigcap_n \mathcal{U}_n[g(n)]$ est $\Pi_1^0(g)$ et ne contient que le point X . D'après la proposition 8-3.6, pour tout $g \geq f$ l'ensemble X est g -calculable. Tout singleton Π_2^0 admet donc un modulus. Le corollaire 2.2 permet de conclure. ■

Nous verrons la réciproque du précédent résultat avec le théorème 29-5.4, qui nous fournira une élégante caractérisation des ensembles hyperarithmétiques.

3. Relativisation

L'itération du saut se relativise à n'importe quel oracle X .

Définition 3.1. Soit $X \in 2^{\mathbb{N}}$. On définit les itérations transfinies H_a^X du saut pour les éléments $b \in \mathcal{O}^X$:

1. $H_1^X = X$:
2. $H_{2^a}^X = (H_a^X)'$:
3. $H_{3 \cdot 5^e}^X = \bigoplus_n H_{\Phi_e(X, n)}^X$. ◇

La hiérarchie de Kleene se relativise elle aussi sans problème à un oracle X , le long des ordinaux $\alpha < \omega_1^{ck}$, de même que les différents théorèmes vus jusqu'ici. En voici, notamment, trois exemples.

Théorème 3.2

Pour tout $X \in 2^{\mathbb{N}}$, tout ordinal $\alpha < \omega_1^X$ et tout $a \in \mathcal{O}_{=\alpha}^X$, on a

1. $\mathcal{O}_{<\alpha}^X \leq_T H_{2^a}^X$, uniformément en a ;
2. $\mathcal{T}_{<\alpha}^X \leq_T H_{2^a}^X$, uniformément en a .

Théorème 3.3

Soit $X \in 2^{\mathbb{N}}$, et soit $\alpha < \omega_1^X$. Soit $a \in \mathcal{O}_{=\alpha}^X$. L'ensemble $H_{2^a}^X$ est $\Sigma_\alpha^0(X)$ -complet, uniformément en X et a .

Théorème 3.4

Il existe $\mathcal{P} \subseteq \mathbb{N} \times 2^{\mathbb{N}} \times 2^{\mathbb{N}}$ une classe Π_2^0 telle que pour tout X , pour tout $a \in \mathcal{O}^X$, la classe $\{Y : (a, X, Y) \in \mathcal{P}\}$ est le singleton H_a^X .

Notons dans les théorèmes précédents que l'on doit également relativiser le plus petit ordinal non calculable ω_1^{ck} , et utiliser à la place ω_1^X , le plus petit ordinal non X -calculable.

4. Hiérarchie borélienne effective

En suivant l'exemple de la hiérarchie de Kleene pour les ensembles d'entiers, il est possible de donner une version effective de la hiérarchie borélienne dont la version complète fut introduite avec la définition 27-3.3. Nous en donnons ici directement la version relativisée à n'importe quel oracle X .

Définition 4.1. La hiérarchie borélienne effective est définie par induction sur les ordinaux, relativement à un oracle $X \in 2^{\mathbb{N}}$.

- ▷ Un $\Sigma_1^0(X)$ -code est donné par une paire $\langle 0, e \rangle$. La classe $\mathcal{U}$ correspondante est donnée par $\mathcal{U} = \bigcup_{\sigma \in W_e^X} [\sigma]$.
- ▷ Un $\Pi_\alpha^0(X)$ -code est donné par une paire $\langle 1, e \rangle$, où e est un $\Sigma_\alpha^0(X)$ -code. La classe $\mathcal{B}$ correspondante est donnée par $\mathcal{B} = 2^{\mathbb{N}} \setminus \mathcal{A}$ où $\mathcal{A}$ est l'ensemble correspondant au code e .
- ▷ Un $\Sigma_\alpha^0(X)$ -code est donné par une paire $\langle 2, e \rangle$, où $W_e^X \neq \emptyset$ énumère avec l'oracle X des $\Pi_{\beta_n}^0(X)$ -codes pour $\beta_n < \alpha$, avec $\sup_n(\beta_n + 1) = \alpha$. La classe $\mathcal{B}$ correspondante est donnée par $\bigcup_n \mathcal{A}_n$ où $\mathcal{A}_n$ est la classe correspondant au n -ième code énuméré par W_e^X . ♦

On dit qu'une classe $\mathcal{B} \subseteq 2^{\mathbb{N}}$ est $\Sigma_\alpha^0(X)$ (resp. $\Pi_\alpha^0(X)$) si elle correspond à un $\Sigma_\alpha^0(X)$ -code (resp. un $\Pi_\alpha^0(X)$ -code). On dit que la classe $\mathcal{B}$ est $\Delta_\alpha^0(X)$ si elle est à la fois $\Sigma_\alpha^0(X)$ et $\Pi_\alpha^0(X)$.

Souvenons-nous de la définition des boréliens non effectifs à travers les ordinaux. Nous avons insisté sur l'utilisation de l'axiome du choix dénombrable

afin de garantir que tout borélien est bien Σ_α^0 pour un certain ordinal α dénombrable. Le principal intérêt est alors d'avoir tout borélien comme étant effectif relativement à un certain oracle X : comme α est dénombrable il peut être encodé par un oracle, qui peut ensuite encoder l'arbre bien fondé correspondant au dépliage du code du borélien.

On obtient ainsi que tout borélien est $\Sigma_\alpha^0(X)$ pour un certain oracle X .

Proposition 4.2. Si une classe est $\Sigma_\alpha^0(X)$ alors $\alpha < \omega_1^X$. ★

PREUVE. Il suffit de répéter la preuve de la proposition 1.8 pour transformer un $\Sigma_\alpha^0(X)$ -code en élément de $\mathcal{T}^X$. On en déduit que $\alpha < \omega_1^X$. ■

Tout comme la hiérarchie de Kleene est stricte, il est possible de montrer que la hiérarchie borélienne est elle aussi stricte, dans un sens fort.

Théorème 4.3

Soit $X \in 2^\mathbb{N}$. Pour tout $\alpha < \omega_1^X$, il existe une classe $\Sigma_\alpha^0(X)$ qui n'est pas Π_α^0 .

PREUVE. La preuve détaillée est laissée en exercice. On suppose pour simplifier la présentation $X = \emptyset$, la relativisation ne posant pas de problème particulier. L'idée générale est la suivante : on définit une fonctionnelle Ψ telle que, pour tout $a \in \mathcal{O}$, on ait :

- (1) pour tout $Y \in 2^\mathbb{N}$, la valeur $\Psi(Y, a)$ est un $\Sigma_{|a|}^0(Y)$ -code ;
- (2) pour toute classe $\Sigma_{|a|}^0 \mathcal{A}$, il existe un oracle Y tel que $\Psi(Y, a)$ est un $\Sigma_{|a|}^0(Y)$ -code de $\mathcal{A}$;
- (3) pour tout $Y \in 2^\mathbb{N}$, la classe

$$\{Y \in 2^\mathbb{N} : Y \text{ appartient à la classe de code } \Psi(Y, a)\}$$
 est Σ_α^0 .

La fonctionnelle Ψ procède simplement à un système de codage des boréliens via son oracle, en utilisant $a \in \mathcal{O}$ pour contrôler la hauteur du code produit. On montre alors que la classe

$$\mathcal{C} = \{Y \in 2^\mathbb{N} : Y \text{ n'appartient pas à la classe de code } \Psi(Y, a)\}$$

ne peut pas être $\Sigma_{|a|}^0$. En effet, si tel était le cas, il y aurait un oracle Y tel que $\Psi(Y, a)$ renvoie un $\Sigma_{|a|}^0(Y)$ -code pour ce borélien. On aurait alors $Y \notin \mathcal{C}$ ssi $Y \in \mathcal{C}$, ce qui est une contradiction. ■

Nous faisons à présent le lien entre la hiérarchie de Kleene et la hiérarchie borélienne effective.

Théorème 4.4

Une classe $\mathcal{A} \subseteq 2^{\mathbb{N}}$ est Σ_{α}^0 si, et seulement si, il existe pour $a \in \mathcal{O}_{=\alpha}$ un entier n tel que

$$\mathcal{A} = \{X : n \in H_{2a}^X\}.$$

PREUVE. Montrons que pour $n \in \mathbb{N}$ et $a \in \mathcal{O}_{=\alpha}$ la classe $\{X : n \in H_{2a}^X\}$ est Σ_{α}^0 , uniformément en n et a . D'après le théorème 3.3, l'ensemble H_{2a}^X est $\Sigma_{\alpha}^0(X)$, uniformément en X et a .

On a donc une fonction calculable $g : \mathbb{N} \rightarrow \mathbb{N}$ telle que $g(a)$ est un $\Sigma_{\alpha}^0(X)$ -code de l'ensemble H_{2a}^X pour tout X .

Définissons à présent une fonction calculable $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ telle que :

- (1) l'entier $f(a, n)$ est le Σ_{α}^0 -code d'une classe borélienne pour tout $a \in \mathcal{O}_{=\alpha}$ et pour tout n ;
- (2) l'entier $n \in H_{2a}^X$ si, et seulement si, X appartient à la classe décrite par le Σ_{α}^0 -code $f(a, n)$.

La fonction $f(a, n)$ calcule le code $g(a)$, qui est pour tout X un $\Sigma_{\alpha}^0(X)$ -code de l'ensemble H_{2a}^X . On utilise ensuite une fonction $h_n : \mathbb{N} \rightarrow \mathbb{N}$ définie inductivement sur les nœuds de l'arbre décrit par le code $g(a)$ (dépendamment d'un oracle X). Sur un code $\langle 2, e \rangle$, la fonction h_n renvoie un code correspondant à la réunion des $[\sigma] \cap \mathcal{B}_{\sigma}$ pour chaque σ tel que W_e^{σ} énumère un code e_{σ} , et où $\mathcal{B}_{\sigma}$ est la classe codée par $h_n(e_{\sigma})$. Sur un code $\langle 1, e \rangle$, la fonction h_n renvoie $\langle 1, h_n(e) \rangle$. Sur un code $\langle 0, e \rangle$, la fonction h_n renvoie le code $\langle 0, d \rangle$, où d est le code de la classe Σ_1^0 qui énumère les chaînes σ telles que $n \in W_e^{\sigma}$. La fonction $f(a, n)$ renvoie finalement $h_n(g(a))$. Il suffit de montrer par induction que f satisfait les points (1) et (2) ci-dessus.

Montrons à présent que pour $\mathcal{A}$ une classe Σ_{α}^0 , on peut uniformément trouver un entier e tel que $\mathcal{A} = \{X : e \in H_{2a}^X\}$. Pour ce faire, on montre d'abord comment trouver un entier d qui est pour tout X le $\Sigma_{\alpha}^0(X)$ -code d'un ensemble contenant 0 ssi $X \in \mathcal{A}$. On transforme pour cela uniformément en X l'arbre correspondant au Σ_{α}^0 -code de $\mathcal{A}$ en $\Sigma_{\alpha}^0(X)$ -code du même arbre, excepté que chaque feuille correspondant à $\mathcal{U} \subseteq 2^{\mathbb{N}}$, une classe Σ_1^0 , est remplacée par une feuille correspondant à $U \subseteq \mathbb{N}$ un ensemble $\Sigma_1^0(X)$ tel que $0 \in U$ ssi un préfixe σ de X est tel que $[\sigma] \subseteq \mathcal{U}$. On montre facilement par induction que 0 appartient au $\Sigma_{\alpha}^0(X)$ -code d ssi $X \in \mathcal{A}$. On utilise à présent le théorème 3.3 pour trouver une fonction calculable h telle que 0 appartient à notre ensemble $\Sigma_{\alpha}^0(X)$ ssi $h(0) \in H_{2a}^X$. On a alors $\mathcal{A} = \{X : h(0) \in H_{2a}^X\}$. ■

Nous voyons pour finir une autre version du théorème précédent, mais pour les classes $\{Y : n \in \mathcal{O}_{<\alpha}^Y\}$.

Théorème 4.5

Soit $Y \in 2^{\mathbb{N}}$. Pour tout $n \in \mathbb{N}$ et $\alpha < \omega_1^Y$, la classe $\{X : n \in \mathcal{O}_{<\alpha}^X\}$ est une classe $\Sigma_{\alpha+1}^0(Y)$, uniformément en Y et en n .

PREUVE. On peut faire la même construction que dans le théorème précédent, d'une fonction calculable $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ telle que :

- (1) l'entier $f(a, n)$ est le $\Sigma_{\alpha+1}^0(Y)$ -code d'une classe borélienne pour tout a dans $\mathcal{O}_{=\alpha}^Y$ et pour tout n ;
- (2) l'entier $n \in \mathcal{O}_{<\alpha}^X$ si, et seulement si, X appartient à la classe décrite par le $\Sigma_{\alpha+1}^0(Y)$ -code $f(a, n)$.

D'après le théorème 3.2, pour tous X, Y , et pour tout $\alpha < \omega_1^Y$, on a

$$\mathcal{O}_{<\alpha}^{X \oplus Y} \leq_T H_{2^a}^{X \oplus Y},$$

uniformément en X, Y , et en $a \in \mathcal{O}_{=|\alpha|}^Y$. On a clairement $\mathcal{O}_{<\alpha}^X \leq_m \mathcal{O}_{<\alpha}^{X \oplus Y}$, car pour savoir si $e \in \mathcal{O}_{<\alpha}^X$ il suffit de regarder si $e' \in \mathcal{O}_{<\alpha}^{X \oplus Y}$, où e' fait la même chose que e , mais en utilisant « la moitié » de son oracle correspondant à X . D'après le théorème 3.3, l'ensemble $H_{2^a}^{X \oplus Y}$ est $\Sigma_{\alpha}^0(X \oplus Y)$, uniformément en X, Y et a . L'ensemble $\mathcal{O}_{<\alpha}^X$ est donc $\Delta_{\alpha+1}^0(X \oplus Y)$, uniformément en X, Y et en $\alpha < \omega_1^Y$, et il est donc en particulier $\Sigma_{\alpha+1}^0(X \oplus Y)$.

On peut alors procéder de manière similaire à la preuve précédente pour créer f . Les détails sont laissés au lecteur. ■

Exercice 4.6. (★) Montrer que l'on peut considérer sans perte de généralité que les classes Σ_{α}^0 (de la forme $\bigcup_n \mathcal{B}_n$) sont croissantes (c'est-à-dire qu'elles vérifient $\mathcal{B}_n \subseteq \mathcal{B}_{n+1}$). ◇

Exercice 4.7. (★) Montrer que la proposition 17-4.3 se poursuit dans le transfini : pour $\mathcal{A}$ une classe Σ_{α}^0 , l'ensemble $\{q \in \mathbb{Q} : \lambda(\mathcal{A}) > q\}$ est Σ_{α}^0 , uniformément en un code de $\mathcal{A}$. ◇

Chapitre 29

Au delà des hyperarithmétiques

Qu'y a-t-il au-delà des ensembles hyperarithmétiques ? D'après une variation de l'exercice 28-1, les fragments du $\mathcal{O}$ de Kleene sont arbitrairement complexes dans la hiérarchie hyperarithmétique

Dans ce chapitre, nous allons introduire les classes et ensembles Σ_1^1 et Π_1^1 définis à l'aide de quantifications du second ordre. Ces notions vont nous permettre de donner une caractérisation des ensembles hyperarithmétiques purement en termes de définissabilité par des formules du second ordre, et d'en déduire de nouvelles caractérisations en termes de modulus ou de singletons Π_2^0 .

Les ensembles Π_1^1 jouent un rôle central, notamment dans la correspondance entre calculabilité classique et hypercalculabilité. Nous verrons dans quelle mesure les ensembles Π_1^1 peuvent être perçus comme des ensembles « hypercalculatoirement énumérables ». Nous verrons en particulier que le $\mathcal{O}$ de Kleene joue pour les hyperarithmétiques un rôle analogue à celui joué par $\emptyset'$ pour les calculables. Cette correspondance entre calculabilité et hypercalculabilité sera poussée plus loin dans le chapitre 30, où nous verrons que les classes Σ_1^1 peuvent être vues comme le pendant hypercalculable des classes Π_1^0 .

1. Un peu d'histoire : l'école de Moscou

Au XIX^e siècle, les écrivains Nicolas Gogol et Fiodor Dostoïevski théorisent « l'âme russe », concept flou encore aujourd'hui, souvent perçu comme un mélange de mysticisme, d'irrationalité, de démesure et d'abattement.

On ne peut s'empêcher de voir la manifestation de cette fameuse âme russe — qu'elle soit fantasmée ou réelle — quand on se plonge dans l'histoire de la théorie des ensembles moscovite du début du XX^e siècle. Les mathématiciens Jean-Michel Kantor et Loren Graham ont mené une enquête historique approfondie dans laquelle ils étudient l'impact des différences culturelles entre la Russie et l'Europe occidentale, sur l'accueil et la perception des nouvelles mathématiques que constitue alors la théorie des ensembles. Ils écrivent notamment [107] :

« Les mathématiciens russes de la fin du XIX^e siècle et du début du XX^e siècle pensaient que leurs travaux étaient étroitement liés à la philosophie, à la religion et à l'idéologie en général. De ce point de vue, ils se démarquaient de la plupart de leurs collègues français. »

Egorov et Luzin

Nous avons parlé dans la section 17-1 du célèbre trio Français : Borel, Baire et Lebesgue, dont le travail avait éclos au début du XX^e siècle sur un ensemble de résultats d'une grande richesse faisant suite aux travaux de Cantor sur l'infini. Les concepts d'ensembles boréliens font alors peu à peu leur chemin dans la communauté mathématique, qui en perçoit l'esthétique et l'intérêt. C'est dans la lignée de leurs travaux que va naître sous l'impulsion de Dimitri Egorov et de Nicolai Luzin l'*École mathématique de Moscou*, une des plus impressionnantes qui ait jamais existé. Egorov et Luzin ont étudié les mathématiques auprès de Nikolai Vasilievich Bugaev (1837-1903), qui s'intéressa beaucoup à la théorie des fonctions discontinues, lesquelles avaient pour lui un intérêt philosophique qui allait bien au-delà des considérations logiques des mathématiciens français et allemands [107, p.89] :

« La discontinuité est une affirmation de l'individualité indépendante et autonome. La discontinuité intervient aussi là où surgissent les questions des causes finales et les problèmes esthétiques et éthiques. »

On voit le contraste saisissant avec les déclarations de Poincaré (voir la section 17-1.1) pour qui la discontinuité est une aberration logique sans lien avec le monde réel, là où Bugaev voit dans les objets mathématiques discontinus un intérêt abstrait, presque mystique. Bugaev fut le professeur de mathématiques d'Egorov, qui au cours de plusieurs séjours en France et en Allemagne, se familiarisa avec les tout récents développements de la théorie des ensembles. Cette théorie le passionne, et Egorov décide de l'enseigner une fois de retour à Moscou. Luzin figure parmi ses étudiants préférés, et ils monteront ensemble ce qui sera d'abord un séminaire de mathématiques baptisé *Lusitanie* — on ne sait pas vraiment aujourd'hui si ce nom fut choisi en l'honneur de Luzin ou non — qui constituera les

prémises de ce qui deviendra plus tard l'École de mathématique de Moscou. L'histoire de la Lusitanie s'inscrit elle-même dans la grande Histoire et les profonds bouleversements qui agiteront la Russie au début du XX^e siècle. Mentionnons à ce sujet une anecdote sur la jeunesse de Luzin. Ce dernier est profondément marqué par la révolution manquée de 1905. Il perd son intérêt pour les mathématiques et traverse trois années durant une crise profonde, comme en témoignent les lettres qu'il écrit à Egorov [107, p.106] :

« Quand vous m'avez rencontré à l'Université, je n'étais qu'un enfant ignorant. Je ne sais pas ce qui s'est passé, mais je ne peux me satisfaire aujourd'hui des fonctions analytiques et des séries de Taylor... La misère du peuple, les tourments de la vie... sont des visions insupportables... je ne peux plus vivre uniquement de science... je n'ai rien, pas de vision du monde et pas d'éducation. »

Luzin sortira de sa dépression avec l'aide de son ami et guide spirituel Pavel Florensky — prêtre et mathématicien —, notamment suite à la lecture de la thèse de Florensky *« De la vérité religieuse »*. Luzin reprend alors ses travaux mathématiques. Toute sa vie, il s'intéressera essentiellement à la théorie des ensembles, qu'il étudiera avec toute la rigueur scientifique d'un mathématicien, mais aussi avec une conviction qui relève d'une certaine forme de foi religieuse. On trouvera par exemple dans ses notes :

« Nous pensons que les nombres entiers existent objectivement. Nous pensons que la totalité des nombres transfinis de seconde classe existe objectivement. Nous désirons la chose suivante : ayant supposé leur évidence, nous associons à chacun des nombres transfinis une définition, un nom, et cela pour tous les nombres transfinis que nous envisageons. »

La Lusitanie

Egorov et Luzin lancent peu de temps avant la Seconde Guerre mondiale un séminaire pour un petit groupe d'étudiants motivés : la Lusitanie, qui connaîtra un succès et une pérennité inattendue.

Dans la première décennie du XX^e siècle, la Russie ne comptait quasiment aucun mathématicien de renommée mondiale. Cela changera avec la création de la Lusitanie. En dix ans à peine, de nombreux très jeunes mathématiciens issus de cette école vont se faire une place sur la scène internationale, parmi lesquels on peut citer Pavel Alexandrov, qui développera une part importante de la topologie moderne, Pavel Urysohn, autre topologue de renom, ou encore Nina Bari, connue pour ses travaux sur les séries trigonométriques. Andreï Kolmogorov, certainement le plus célèbre mathématicien russe, passera aussi par la Lusitanie, mais un peu plus tard. En 1930, Moscou sera devenu une des principales capitales mathématiques mondiales.

Le séminaire d'Egorov et de Lusin prend rapidement : entre la pédagogie et l'investissement passionné des deux professeurs, la motivation et la qualité scientifique exceptionnelle des étudiants, l'alchimie se fait. Rapidement, un groupe très soudé, joyeux et travailleur se forme, et découvre avec admiration les derniers développements de ces « nouvelles mathématiques » que constitue la théorie des ensembles, à travers lesquelles se forme le sentiment de prendre part à quelque chose d'important, à une aventure intellectuelle plus grande que soi.

L'histoire de la Lusitanie est d'autant plus remarquable que les nombreux développements mathématiques de premier plan qui y eurent lieu se firent dans des conditions matérielles catastrophiques : la Première Guerre mondiale participe à la famine et aux pénuries généralisées qui sévissent dans le pays. La révolution d'Octobre (1917) plonge le pays dans une violente guerre civile. Les étudiants de Lusin et d'Egorov souffrent du froid et de la faim. La température des salles de classe descend parfois sous les 0-degrés [107, p.137]. Qu'importe, les étudiants viennent tout de même.

La théorie descriptive des ensembles

Le premier accomplissement d'importance de la Lusitanie est celui qui concerne l'objet du chapitre à venir : la naissance de la théorie descriptive des ensembles. Une des questions centrales était à l'époque celle de l'hypothèse du continu de Cantor. Alexandrov, alors à peine âgé de dix-huit ans montre que l'hypothèse du continu est vraie pour toutes les classes boréliennes. Nous verrons une forme moderne et effective de ce théorème avec le corollaire 30-3.3. Nous le savons aujourd'hui, l'hypothèse du continu demandera des efforts et développements bien plus conséquents, mais pour l'époque, il s'agissait tout de même d'une étape importante vers sa résolution. Pouvait-on étendre la preuve d'Alexandrov à toutes les classes ? À l'époque, l'existence de classes non boréliennes n'était pas encore complètement claire, et des exemples précis de ces classes l'étaient encore moins. Un an plus tard, un autre jeune étudiant de la Lusitanie, Mikhaïl Souslin, repère une erreur dans une preuve de Lebesgue datant de 1905, qui restera fameuse comme « l'erreur de Lebesgue », marquant le point de départ d'un champ de recherche nouveau : *la théorie descriptive des ensembles*. La démonstration de Lebesgue porte sur l'énoncé suivant : l'image d'une classe borélienne par une fonction continue est aussi une classe borélienne. Souslin montrera que ce n'est pas nécessairement vrai : l'image de certains boréliens par certaines fonctions continues sont des ensembles *strictement plus complexes* que les boréliens. C'est la découverte par Souslin des classes dites *analytiques* ou comme nous les appelons dans cet ouvrage, les classes Σ_1^1 . Souslin [219] montrera par la suite que les boréliens sont exactement les classes Σ_1^1 dont le complémentaire est également Σ_1^1 . Plus tard,

Kleene [116] donnera — en s'appuyant sur des résultats de Spector [214] — une version effective du théorème de Souslin, et qui s'applique non seulement aux classes, mais aussi aux ensembles d'entiers. C'est ce résultat que nous présenterons dans la section 5.

La fin

Si l'héritage de la Lusitanie perdure jusqu'à aujourd'hui, les troubles politiques du pays auront raison de la Lusitanie elle-même, qui s'achèvera vers les années 1920. La nouvelle génération de mathématiciens formée par Luzin et Egorov s'adapte au nouvel ordre des choses dicté par le pouvoir en place, là où l'ancienne génération ne s'y fait pas. Egorov ne cache ni ses convictions religieuses, ni son animosité envers le nouveau régime. Il est arrêté en 1930, accusé de mélanger mathématiques et religion. En prison, Egorov entame une grève de la faim, suite à laquelle il doit être transféré à l'hôpital où il meurt peu de temps après.

Luzin quant à lui fait profil bas. L'arrestation de son collègue et ami le terrorise, aussi se fait-il aussi discret que possible. Il cache ses convictions religieuses, et essaye de montrer sa loyauté envers l'état soviétique. Mais son passé le rattrape. Il est la cible d'attaques répétées jusqu'en 1936, au moment où il est accusé lors d'un procès d'être, en substance, un ennemi du parti soviétique. Plusieurs de ses anciens étudiants témoignent contre lui, l'accusant de plagiat et de népotisme. C'est un coup terrible pour Luzin, qui pour une raison encore floue aujourd'hui, et malgré une condamnation officielle, ne sera finalement ni arrêté ni expulsé de l'Académie des sciences.

En dépit de cet épisode douloureux de l'histoire des mathématiques, L'école fondée par Egorov et Luzin a perduré, et à travers elle, la Russie a posé une marque profonde et durable sur les mathématiques, apportant peut-être un peu de cette mystérieuse âme russe, à travers un style, et une intensité dans l'engagement mathématique. Nous terminons notre interlude historique par une citation du livre de Jean Michel Kantor et Loren Graham, qui montre bien la façon dont l'école russe a marqué l'imaginaire du mathématicien :

« On citait encore avec une admiration mêlée de crainte les légendaires séminaires de mathématiques de Moscou où les exposés, commencés dans l'après-midi, se prolongeaient souvent tard dans la soirée, organisateurs et assistants du séminaire soumettant les orateurs à une longue suite de questions dans le but de comprendre. Chaque personne assistant au séminaire pouvait d'ailleurs être envoyée au tableau à l'improviste, ce qui rendait impossible l'habitude occidentale d'assister à un séminaire en spectateur distant et distrait. »

Les auteurs de ce livre, qui ont eu la chance de travailler avec leurs collègues moscovites, peuvent attester que la légende comporte bien un fond de vérité...

2. Quantifications du second ordre

Nous avons vu avec la hiérarchie de Kleene que la hiérarchie arithmétique pouvait être étendue de manière naturelle le long des ordinaux calculables, pour donner les ensembles hyperarithmétiques. D'après la remarque suivant la proposition 28-1.7, cette extension peut-être pensée en termes d'alternance de quantificateurs sur des formules infinies.

Il existe une autre approche naturelle pour étendre la hiérarchie arithmétique, consistant à autoriser les quantificateurs du second ordre, autrement dit les quantificateurs sur les ensembles d'entiers. On obtient alors une hiérarchie dont le premier niveau contient déjà tous les ensembles hyperarithmétiques, comme nous le verrons par la suite.

Définition 2.1. Soit un ensemble $A \subseteq \mathbb{N}$.

1. A est Σ_1^1 s'il existe une classe arithmétique $\mathcal{B} \subseteq 2^{\mathbb{N}} \times \mathbb{N}$ telle que

$$A = \{n \in \mathbb{N} : \exists X \in 2^{\mathbb{N}} (X, n) \in \mathcal{B}\}.$$

2. A est Π_1^1 s'il existe une classe arithmétique $\mathcal{B} \subseteq 2^{\mathbb{N}} \times \mathbb{N}$ telle que

$$A = \{n \in \mathbb{N} : \forall X \in 2^{\mathbb{N}} (X, n) \in \mathcal{B}\}.$$

3. A est Δ_1^1 si A est Π_1^1 et Σ_1^1 .

Les ensembles Σ_1^1 et Π_1^1 sont aussi qualifiés respectivement d'*analytiques* et *co-analytiques* effectifs. $\diamond$

Notons que les Π_1^1 sont les complémentaires des Σ_1^1 , et vice-versa. Un ensemble Σ_1^1 (resp. Π_1^1) a le droit d'utiliser des quantifications existentielles (resp. universelles) non pas sur les entiers, mais sur les ensembles d'entiers. Cela donne beaucoup plus de pouvoir expressif, et nous verrons que les ensembles hyperarithmétiques sont tous Π_1^1 et Σ_1^1 . Nous verrons en fait avec le théorème 5.2 que les ensembles hyperarithmétiques sont exactement les ensembles Δ_1^1 .

On définit de la même manière les notions de Σ_1^1 et Π_1^1 pour les classes.

Définition 2.2. Soit une classe $\mathcal{A} \subseteq 2^{\mathbb{N}}$.

1. $\mathcal{A}$ est Σ_1^1 s'il existe une classe arithmétique $\mathcal{B} \subseteq 2^{\mathbb{N}} \times 2^{\mathbb{N}}$ telle que

$$\mathcal{A} = \{X \in 2^{\mathbb{N}} : \exists Y \in 2^{\mathbb{N}} (Y, X) \in \mathcal{B}\}.$$

2. $\mathcal{A}$ est Π_1^1 s'il existe une classe arithmétique $\mathcal{B} \subseteq 2^{\mathbb{N}} \times 2^{\mathbb{N}}$ telle que

$$\mathcal{A} = \{X \in 2^{\mathbb{N}} : \forall Y \in 2^{\mathbb{N}} (Y, X) \in \mathcal{B}\}.$$

3. $\mathcal{A}$ est Δ_1^1 si, seulement si, $\mathcal{A}$ est Π_1^1 et Σ_1^1 .

Les classes Σ_1^1 et Π_1^1 sont aussi qualifiées respectivement d'*analytiques* et *co-analytiques* effectives. $\diamond$

Notons que les définitions de Σ_1^1 et Π_1^1 s'étendent naturellement à l'espace de Baire ainsi qu'au produit de différents espaces sur lesquels la définition a cours.

Exemple 2.3. Une classe $\mathcal{A} \subseteq \mathbb{N}^{\mathbb{N}} \times 2^{\mathbb{N}} \times \mathbb{N}$ est Σ_1^1 s'il existe un ensemble arithmétique $\mathcal{B} \subseteq 2^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \times 2^{\mathbb{N}} \times \mathbb{N}$ tel que

$$A = \{(f, X, n) : \exists Y \in 2^{\mathbb{N}} (Y, f, X, n) \in \mathcal{B}\}.$$

2.1. Propriétés de clôture

Nous allons voir que, de manière équivalente, les quantifications des classes et ensembles Σ_1^1 et Π_1^1 peuvent se faire sur les fonctions plutôt que sur les ensembles d'entiers. De manière générale, ces classes seront plus simples à manipuler via des quantifications sur les fonctions. Nous introduisons pour cela quelques notations qui nous seront utiles.

Notation

- ▷ Soit X un ensemble infini. Alors, f_X désigne la fonction de $\mathbb{N}$ dans $\mathbb{N}$ encodée par X , c'est-à-dire la fonction telle que $f_X(n)$ renvoie le n -ième élément de X .
- ▷ La notation $\exists X \oplus Y \mathcal{B}(X, Y)$ est un raccourci pour signifier $\exists Z \mathcal{B}(Z^{[0]}, Z^{[1]})$, où $Z^{[i]} = \{2n + i : n \in Z\}$.
- ▷ Il en va de même pour la notation $\exists(\bigoplus_m X_m) \forall m \mathcal{B}(X_m, m)$.

La proposition suivante implique en particulier que les classes ou ensembles Σ_1^1 peuvent être définis avec un nombre arbitraire de quantificateurs existentiels sur les ensembles, essentiellement en codant deux quantifications en une seule à l'aide de la jointure effective.

Proposition 2.4.

- (1) Les ensembles ou classes Σ_1^1 sont clos par réunion effective indexée par les ensembles d'entiers ou indexée par les fonctions de $\mathbb{N}$ dans $\mathbb{N}$.
- (2) Les ensembles ou classes Π_1^1 sont clos par intersection effective indexée par les ensembles d'entiers ou par les fonctions de $\mathbb{N}$ dans $\mathbb{N}$. $\star$

PREUVE. Notons que (1) est équivalent à (2), par passage au complémentaire. Nous montrons donc uniquement (1), et nous le faisons à la fois pour les classes et les ensembles en considérant des classes de l'espace produit $2^{\mathbb{N}} \times \mathbb{N}$.

Soit $\mathcal{A} \subseteq 2^{\mathbb{N}} \times 2^{\mathbb{N}} \times \mathbb{N}$ la classe Σ_1^1 égale à $\{(Y, X, n) : \exists Z (Z, Y, X, n) \in \mathcal{B}\}$, pour $\mathcal{B}$ un ensemble arithmétique. Montrons que la classe

$$\{X : \exists Y (Y, X, n) \in \mathcal{A}\}$$

est aussi Σ_1^1 . On a

$$\exists Y (Y, X, n) \in \mathcal{A} \Leftrightarrow \exists Z \exists Y (Z, Y, X, n) \in \mathcal{B} \Leftrightarrow \exists Z \oplus Y (Z, Y, X, n) \in \mathcal{B}.$$

On obtient ainsi la clôture pour les réunions indexées par des ensembles d'entiers.

Soit $\mathcal{A} \subseteq \mathbb{N}^{\mathbb{N}} \times 2^{\mathbb{N}} \times \mathbb{N}$ la classe Σ_1^1 égale à $\{(f, X, n) : \exists Y (Y, f, X, n) \in \mathcal{B}\}$, pour $\mathcal{B}$ un ensemble arithmétique. Montrons que la classe

$$\{(X, n) : \exists f (f, X, n) \in \mathcal{A}\}$$

est aussi Σ_1^1 . On a

$$\exists f (f, X, n) \in \mathcal{A} \Leftrightarrow \exists Y \exists f (Y, f, X, n) \in \mathcal{B}$$

$$\Leftrightarrow \exists Y \exists Z \text{ tel que } (Z \text{ est infini et } (Y, f_Z, X, n) \in \mathcal{B}).$$

Être infini est bien un prédicat arithmétique. Par la propriété de clôture du paragraphe précédent, la classe est bien Σ_1^1 . ■

Quantification sur $\mathbb{N}^{\mathbb{N}}$ vs $2^{\mathbb{N}}$

La proposition précédente a une implication importante : les ensembles ou classes Σ_1^1 peuvent se décrire de manière équivalente via des quantifications du second ordre sur $\mathbb{N}^{\mathbb{N}}$ ou bien sur $2^{\mathbb{N}}$. C'est une propriété que nous utiliserons pour le théorème de forme normale à venir.

Pour la preuve de la proposition suivante, nous utiliserons l'axiome du choix dénombrable, afin de nous simplifier la vie. Cet axiome n'est toutefois pas absolument nécessaire : nous verrons plus tard avec la proposition 30-1.2 qu'il est automatiquement vérifié pour les classes Σ_1^1 (qui sont celles sur lesquelles on l'utilise dans la proposition).

Proposition 2.5. Les ensembles ou classes Σ_1^1/Π_1^1 sont clos par réunion ou intersection effective indexée par les entiers. ★

PREUVE. Là encore, on montre la proposition à la fois pour les classes et les ensembles. Soit $\mathcal{A} \subseteq \mathbb{N} \times (2^{\mathbb{N}} \times \mathbb{N})$ la classe Σ_1^1 égale à

$$\{(m, X, n) : \exists f (f, m, X, n) \in \mathcal{B}\},$$

pour $\mathcal{B}$ un ensemble arithmétique. Montrons que la classe

$$\{(X, n) : \exists m (m, X, n) \in \mathcal{A}\}$$

est aussi Σ_1^1 . On a

$$\exists m (m, X, n) \in \mathcal{A} \Leftrightarrow \exists m \exists f (f, m, X, n) \in \mathcal{B} \Leftrightarrow \exists f \exists m (f, m, X, n) \in \mathcal{B}.$$

Le prédicat $\exists m (f, m, X, n) \in \mathcal{B}$ étant arithmétique, la classe est bien Σ_1^1 .

Montrons que la classe $\{(X, n) : \forall m (m, X, n) \in \mathcal{A}\}$ est aussi Σ_1^1 . On a $\forall m (m, X, n) \in \mathcal{A}$ ssi $\forall m \exists f (f, m, X, n) \in \mathcal{B}$. En particulier, pour tout m , il existe une fonction f telle que $(f, m, X, n) \in \mathcal{B}$. En utilisant l'axiome du choix, il suffit de choisir pour chaque m une fonction f_m telle que $(f_m, m, X, n) \in \mathcal{B}$. En considérant la fonction $\bigoplus_m f_m$, on a alors

$$\forall m \exists f (f, m, X, n) \in \mathcal{B} \Leftrightarrow \exists \left(\bigoplus_m f_m \right) \forall m (f_m, m, X, n) \in \mathcal{B}.$$

Le prédicat $\forall m (f_m, m, X, n) \in \mathcal{B}$ étant arithmétique, la classe est bien Σ_1^1 .

Par passage au complémentaire, les ensembles ou classes Π_1^1 sont clos par réunion ou intersection effective indexée par les entiers. ■

Nous verrons avec la proposition 30-1.2 que l'on peut construire des fonctions de choix sur les suites dénombrables de classes Σ_1^1 — sans utiliser bien sûr l'axiome du choix. Dans la deuxième partie de la proposition précédente, les classes $\mathcal{B}_m = \{f \in \mathbb{N}^{\mathbb{N}} : (f, m, X, n) \in \mathcal{B}\}$ sont arithmétiques (avec l'oracle X), et donc en particulier $\Sigma_1^1(X)$. On peut dès lors choisir dans chaque $\mathcal{B}_m$ une fonction f_m sans utiliser l'axiome du choix. Il faut simplement vérifier que les développements menant à la construction de cette fonction de choix n'utilisent pas la proposition précédente. C'est bien le cas, les développements en question tiennent essentiellement au théorème de forme normal de Kleene de la section suivante, qui n'a pas besoin de la clôture des Σ_1^1 par réunion ou intersection effective.

2.2. Formes normales

L'avantage d'utiliser des quantifications sur les fonctions plutôt que sur les ensembles d'entiers, est la simplicité des formes normales suivantes, pour les prédicats Σ_1^1 et Π_1^1 .

Théorème 2.6 (Kleene [115])

Soit une classe $\mathcal{A} \subseteq 2^{\mathbb{N}} \times \mathbb{N}$.

- (1) La classe $\mathcal{A}$ est Σ_1^1 si, et seulement si, il existe une fonctionnelle Φ_e telle que $(X, n) \in \mathcal{A} \Leftrightarrow \exists f \Phi_e(f, X, n) \uparrow$.
- (2) La classe $\mathcal{A}$ est Π_1^1 si, et seulement si, il existe une fonctionnelle Φ_e telle que $(X, n) \in \mathcal{A} \Leftrightarrow \forall f \Phi_e(f, X, n) \downarrow$.

PREUVE

Notons que (1) est équivalent à (2) par passage au complémentaire. On montre donc tout simplement (1). Sans perte de généralité, une classe Σ_1^1 de $2^{\mathbb{N}} \times \mathbb{N}$ est de la forme $\{(X, n) : \exists f \mathcal{B}(f, X, n)\}$. Par ailleurs, $\mathcal{B}$ est de la forme $\exists x_1 \forall x_2 \dots \mathcal{R}(x_1, x_2, \dots, x_m, f, X, n)$, pour un certain m et un certain prédicat récursif $\mathcal{R}$. On montre par induction que l'on peut « avaler » les quantifications de $\mathcal{B}$ par la quantification existentielle sur les fonctions.

Soit $\mathcal{A} = \{(X, n) : \exists f \exists y (f, y, X, n) \in \mathcal{B}\}$ pour un prédicat arithmétique $\mathcal{B}$. Alors, on peut avaler y dans la quantification de f , car on a aussi $\mathcal{A} = \{(X, n) : \exists f (f \upharpoonright_{\mathbb{N}^*}, f(0), X, n) \in \mathcal{B}\}$, où $f \upharpoonright_{\mathbb{N}^*}$ est la fonction telle que $f \upharpoonright_{\mathbb{N}^*}(n) = f(n+1)$ pour tout n .

Soit $\mathcal{A} = \{(X, n) : \exists f \forall y \exists z (f, y, z, X, n) \in \mathcal{B}\}$, pour un prédicat arithmétique $\mathcal{B}$. Alors, on a aussi $\mathcal{A} = \{(X, n) : \exists f \oplus g \forall y (f, y, g(y), X, n) \in \mathcal{B}\}$.

On peut ainsi « avaler » les quantificateurs arithmétiques jusqu'à arriver à un prédicat du type $\mathcal{A} = \{(X, n) : \exists f \forall y (f, y, X, n) \in \mathcal{R}\}$, où $\mathcal{R}$ est un prédicat calculable, que l'on transforme facilement en fonctionnelle Φ_e telle que $\mathcal{A} = \{(X, n) : \exists f \Phi_e(f, X, n) \uparrow\}$. ■

2.3. Relativisation

Tout comme pour les hiérarchies hyperarithmétiques, les notions d'ensembles/classes Σ_1^1/Π_1^1 se relativisent à un oracle. Le théorème de forme normale se relativise aussi sans problème.

Théorème 2.7 (Forme normale relativisée)

Soit une classe $\mathcal{A} \subseteq 2^{\mathbb{N}} \times \mathbb{N}$ et soit $Z \in 2^{\mathbb{N}}$.

1. La classe $\mathcal{A}$ est $\Sigma_1^1(Z)$ si, et seulement si, il existe une fonctionnelle Φ_e telle que $(X, n) \in \mathcal{A} \Leftrightarrow \exists f \Phi_e(f, Z, X, n) \uparrow$.
2. La classe $\mathcal{A}$ est $\Pi_1^1(Z)$ si, et seulement si, il existe une fonctionnelle Φ_e telle que $(X, n) \in \mathcal{A} \Leftrightarrow \forall f \Phi_e(f, Z, X, n) \downarrow$.

Tout comme pour les boréliens, les classes Σ_1^1/Π_1^1 ont aussi une version non effective. Ce sont ces versions non effectives qui furent d'abord découvertes et étudiées par Mikhail Souslin [219] en 1917.

Définition 2.8. Une classe $\mathcal{A} \subseteq 2^{\mathbb{N}} \times \mathbb{N}$ est Σ_1^1 si $\mathcal{A}$ est $\Sigma_1^1(Z)$ pour un certain Z . De la même manière, $\mathcal{A}$ est Π_1^1 si $\mathcal{A}$ est $\Pi_1^1(Z)$ pour un certain Z . ◇

La définition que nous donnons ici n'est pas la définition originelle : les classes Σ_1^1 , ou *analytiques*, étudiées par Souslin étaient les images de classes boréliennes par des fonctions continues. Nous verrons dans la section 30-1 que les classes $\Sigma_1^1(X)$ admettent effectivement une représentation canonique qui en font les images de $\mathbb{N}^{\mathbb{N}}$ par des fonctions calculables en $\mathcal{O}^X$. Les classes Π_1^1 sont quant à elles qualifiées aussi de *co-analytiques*.

Exercice 2.9. Montrer que si X est $\Sigma_1^1(Y)$ et Y est $\Delta_1^1(Z)$, alors X est $\Sigma_1^1(Z)$. ◇

2.4. Séparations

Nous montrons finalement que les ensembles/classes Σ_1^1/Π_1^1 sont distincts. Ces résultats de séparation sont fondamentaux, notamment à la lumière du théorème 5.2 et du corollaire 5.15 à venir, qui nous fourniront alors une séparation entre les classes boréliennes et les classes Σ_1^1/Π_1^1 .

Théorème 2.10

Il existe des ensembles Π_1^1 qui ne sont pas Σ_1^1 , et vice-versa.

PREUVE. Soit $A \subseteq \mathbb{N}$ la classe Π_1^1 suivante : $n \in A$ ssi $\forall f \Phi_n(f, n) \downarrow$. Supposons par l'absurde que le complémentaire de A soit Π_1^1 . Alors, d'après la forme normale de Kleene, il existe un code e tel que $n \in \mathbb{N} \setminus A$ ssi $\forall f \Phi_e(f, n) \downarrow$. On a donc $e \in \mathbb{N} \setminus A$ ssi $\forall f \Phi_e(f, e) \downarrow$ ssi $e \in A$, ce qui est une contradiction. Donc, A est Π_1^1 , mais pas Σ_1^1 . Son complémentaire est donc Σ_1^1 , mais pas Π_1^1 . ■

La preuve du théorème précédent est très similaire à celle qui montre que $\emptyset'$ est un ensemble Σ_1^0 qui n'est pas Π_1^0 . Cette vision des choses sera renforcée dans la section 4 avec l'analogie entre ensembles Π_1^1 et ensembles c. e.

Théorème 2.11

Il existe des classes Π_1^1 qui ne sont pas Σ_1^1 , et des classes Σ_1^1 qui ne sont pas Π_1^1 .

PREUVE. Soit $\mathcal{A}$ la classe Π_1^1 égale à $\{1^n 0X : \forall f \Phi_n(f, X, 1^n 0X) \downarrow\}$. Supposons par l'absurde que $\mathcal{A}$ soit Σ_1^1 , c'est-à-dire que $2^{\mathbb{N}} \setminus \mathcal{A}$ soit Π_1^1 . Alors, il existe Y tel que $2^{\mathbb{N}} \setminus \mathcal{A}$ est $\Pi_1^1(Y)$, et il existe donc e tel que

$$2^{\mathbb{N}} \setminus \mathcal{A} = \{X : \forall f \Phi_e(f, Y, X) \downarrow\}.$$

À présent, on a $1^e 0Y \in \mathcal{A}$ ssi $\forall f \Phi_e(f, Y, 1^e 0Y) \downarrow$ ssi $1^e 0Y \in 2^{\mathbb{N}} \setminus \mathcal{A}$, ce qui est une contradiction. Donc, $\mathcal{A}$ n'est pas Σ_1^1 et, par passage au complémentaire, $2^{\mathbb{N}} \setminus \mathcal{A}$ est une classe Σ_1^1 qui n'est pas Π_1^1 . ■

3. Les Π_1^1 et les bons ordres

Nous voyons dans cette section le lien étroit entre les ensembles Π_1^1 et la notion de bon ordre.

Théorème 3.1

L'ensemble $\mathcal{O}^X$ est $\Pi_1^1(X)$ uniformément en X .

PREUVE. On montre le théorème pour $\mathcal{O}$. La preuve se relativise sans problème uniformément à un oracle X . Nous étendons pour cette preuve l'ordre $<_o$ au maximum d'entiers possibles.

Pour un entier a quelconque, on écrira $a <_o 2^a$. Pour tout entier e , si $\Phi_e(n) \downarrow$, on écrira $\Phi_e(n) <_o 3 \times 5^e$. On clôt ensuite la relation $<_o$ transitivement (notons que des cycles peuvent apparaître et que $<_o$ n'est plus une relation d'ordre sur certains sous-ensembles d'entiers).

On définit une première condition Π_2^0 pour qu'un élément a appartienne à $\mathcal{O}$. Soit A l'ensemble contenant a ainsi que les éléments $b <_o a$; alors, la condition est la suivante.

- (1) La relation $<_o$ est un ordre total restreint aux éléments de A , et ces derniers sont égaux à 1, ou à 2^b pour un certain b , ou à 3×5^e pour un code e tel que Φ_e est total avec $\Phi_e(n) <_o \Phi_e(n+1)$ pour tout n .

Cette condition n'est pas suffisante : il se pourrait qu'un code a la satisfasse mais soit tel que l'énumération de A contienne une suite infinie d'éléments de plus en plus petits : $\dots <_o a_3 <_o a_2 <_o a_1 <_o a$. Il faut alors en plus satisfaire la condition Π_1^1 suivante.

- (2) Pour tout $B \subseteq A$, si B est non vide, alors B contient un plus petit élément pour $<_o$.

Il est clair que tout élément de $\mathcal{O}$ satisfait (1) et (2). Réciproquement, montrons que si a satisfait (1) et (2), alors $a \in \mathcal{O}$. Raisonnons par l'absurde et supposons que $a \notin \mathcal{O}$. Comme $a \in A$, $A \setminus \mathcal{O} \neq \emptyset$, donc par (2), $A \setminus \mathcal{O}$ possède un plus petit élément, que l'on note d . Par (1), soit $d = 1$, soit d est de la forme 2^b , soit de la forme 3×5^e avec $\Phi_e(n) <_o \Phi_e(n+1)$ pour tout n . Comme $1 \in \mathcal{O}$, d est donc d'une des deux formes suivantes.

- ▷ Si $d = 2^b$. Comme $b <_o 2^b$ par définition de $<_o$, alors $b \in A$ par clôture de A par le bas. Par minimalité de d , $b \in \mathcal{O}$. Mais alors $2^b \in \mathcal{O}$, ce qui contredit $d \notin \mathcal{O}$.
- ▷ Si $d = 3 \times 5^e$. Comme pour tout n , $\Phi_e(n) <_o 3 \times 5^e$ par définition de $<_o$, alors $\Phi_e(n) \in A$ par clôture de A par le bas. Par minimalité de d , $\Phi_e(n) \in \mathcal{O}$ pour tout n . Alors, $3 \times 5^e \in \mathcal{O}$, ce qui contredit $d \notin \mathcal{O}$.

Dans les deux cas, nous obtenons une contradiction. Ainsi, pour tout a qui satisfait (1) et (2), il vient $a \in \mathcal{O}$. ■

Théorème 3.2

Soit $\mathcal{A} \subseteq 2^{\mathbb{N}} \times \mathbb{N}$. Alors, $\mathcal{A}$ est Π_1^1 si, et seulement si, il existe $h : \mathbb{N} \rightarrow \mathbb{N}$ totale calculable telle que $(X, n) \in \mathcal{A} \Leftrightarrow h(n) \in \mathcal{O}^X$.

PREUVE. Supposons $(X, n) \in \mathcal{A}$ ssi $h(n) \in \mathcal{O}^X$. En utilisant le théorème 3.1, on donne facilement une description Π_1^1 de la classe $\mathcal{A}$.

Montrons à présent la réciproque. Soit $\mathcal{A}$ un ensemble Π_1^1 . D'après le théorème de forme normale, il existe un code e tel que

$$(X, n) \in \mathcal{A} \Leftrightarrow \forall f \Phi_e(f, X, n) \downarrow.$$

Uniformément en n et X , on définit un arbre calculable T_n^X tel que

$$\sigma \in T_n^X \Leftrightarrow \Phi_e(\sigma, X, n)[|\sigma|] \uparrow.$$

L'arbre T_n^X est bien fondé ssi $\forall f \ f \notin [T_n^X]$ ssi $\forall f \ \exists t \ \Phi_e(f \upharpoonright_t, X, n)[t] \downarrow$ ssi $(X, n) \in \mathcal{A}$.

La fonction h transforme simplement l'arbre T_n^X en un code d'ordinal calculable à l'aide de l'ordre de Kleene-Brouwer comme utilisé dans la preuve de la proposition 27-5.18, puis transforme le code de l'ordinal calculable en code d'un ordinal constructif en utilisant la transformation des ordinaux calculables en ordinaux constructifs comme effectué dans la preuve du théorème 27-5.10. Notons que si l'arbre T_n^X est mal fondé les transformations effectuées dans chacune de ces preuves restent valides, mais simplement aboutissent à un entier n'appartenant pas à $\mathcal{O}^X$. ■

Remarque

Le théorème précédent généralise les deux cas particuliers qui nous intéressent.

- ▷ Un ensemble $A \subseteq \mathbb{N}$ est Π_1^1 si, et seulement si, il existe $h : \mathbb{N} \rightarrow \mathbb{N}$ totale calculable telle que $n \in A \Leftrightarrow h(n) \in \mathcal{O}$.
- ▷ Une classe $\mathcal{A} \subseteq 2^{\mathbb{N}}$ est Π_1^1 si, et seulement si, il existe $e \in \mathbb{N}$ tel que $X \in \mathcal{A} \Leftrightarrow e \in \mathcal{O}^X$.

Corollaire 3.3

Pour tout X , l'ensemble $\mathcal{O}^X$ est many-one complet pour les ensembles $\Pi_1^1(X)$ (ou encore $\Pi_1^1(X)$ -complet).

PREUVE. Il s'agit d'une simple reformulation du théorème 3.2. ■

Corollaire 3.4

Pour tout X , l'ensemble $\mathcal{O}^X$ est $\Pi_1^1(X)$, mais n'est pas $\Sigma_1^1(X)$.

PREUVE. On montre le corollaire pour $\mathcal{O}$, la preuve se relativisant sans problème à $\mathcal{O}^X$ pour tout X . On montre aisément que tout ensemble many-one réductible à un ensemble Π_1^1 est aussi Π_1^1 , et que tout ensemble many-one réductible à un ensemble Σ_1^1 est aussi Σ_1^1 . Donc, si $\mathcal{O}$ est Σ_1^1 , alors tout ensemble Π_1^1 est aussi Σ_1^1 , ce qui contredit le théorème 2.10. ■

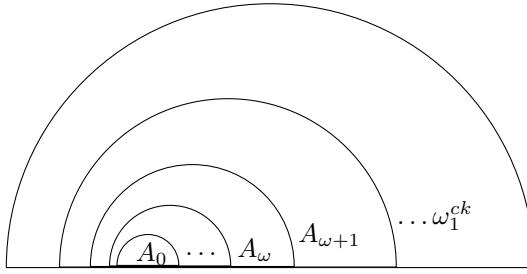


FIGURE 3.5 – Illustration du fait qu'un ensemble Π_1^1 est une réunion croissante le long de ω_1^{ck} d'ensembles $\Sigma_{\alpha+1}^0$

Nous voyons à présent deux autres corollaires importants : les ensembles/-classes Π_1^1 sont des réunions uniformes de boréliens indexés par les ordinaux dénombrables.

Corollaire 3.6

Soit $A \subseteq \mathbb{N}$ un ensemble Π_1^1 . Alors, pour tout $\alpha < \omega_1^{ck}$, uniformément en un code de α , on peut trouver le code de A_α , un ensemble $\Sigma_{\alpha+1}^0$, tel que $A = \bigcup_{\alpha < \omega_1^{ck}} A_\alpha$.

PREUVE. D'après le théorème 3.2, on a une fonction calculable $h : \mathbb{N} \rightarrow \mathbb{N}$ telle que $n \in A \Leftrightarrow h(n) \in \mathcal{O}$. En particulier, $A = \bigcup_{\alpha < \omega_1^{ck}} \{n : h(n) \in \mathcal{O}_{<\alpha}\}$. D'après le théorème 28-1.10, l'ensemble $\mathcal{O}_{<\alpha}$ est uniformément Turing réductible à un ensemble Σ_α^0 , et est donc $\Delta_{\alpha+1}^0$. L'ensemble $\{n : h(n) \in \mathcal{O}_{<\alpha}\}$ est en particulier $\Sigma_{\alpha+1}^0$. ■

Le corollaire précédent est remarquable : on a réussi à transformer une quantification universelle sur les fonctions de l'espace de Baire en quantification existentielle sur les ordinaux calculables. On passe en particulier d'une quantification ayant la puissance du continu à une quantification dénombrable. Le corollaire suivant suit la même idée, mais pour les classes.

Corollaire 3.7

Soit $\mathcal{A} \subseteq 2^{\mathbb{N}}$ une classe Π_1^1 . Alors, pour tout $\alpha < \omega_1$, uniformément en un oracle X codant pour α , on peut trouver le code de $\mathcal{A}_\alpha$, une classe $\Sigma_{\alpha+1}^0(X)$, telle que $\mathcal{A} = \bigcup_{\alpha < \omega_1} \mathcal{A}_\alpha$.

PREUVE. D'après le théorème 3.2, on a un code e tel que $Y \in \mathcal{A} \Leftrightarrow e \in \mathcal{O}^Y$. En particulier, $\mathcal{A} = \bigcup_{\alpha < \omega_1} \{Y : e \in \mathcal{O}_{<\alpha}^Y\}$. D'après le théorème 28-4.5, pour tout $\alpha < \omega_1$, pour tout X tel que $\alpha < \omega_1^X$ et pour tout $a \in \mathcal{O}_{=\alpha}^X$, l'ensemble $\{Y : e \in \mathcal{O}_{<\alpha}^Y\}$ est $\Sigma_{\alpha+1}^0(X)$ uniformément en X et a . ■

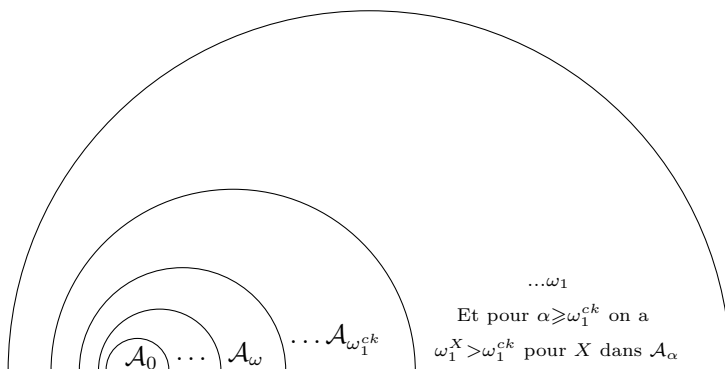


FIGURE 3.8 – Illustration du fait qu’une classe Π_1^1 est une réunion croissante le long de ω_1 de classes $\Sigma_{\alpha+1}^0(Y)$ en n’importe quel Y calculant α . On note que, pour $X \in \mathcal{A} \setminus \mathcal{A}_{\omega_1^{ck}}$, on a nécessairement $\omega_1^X > \omega_1^{ck}$.

Exercice 3.9. (★) (Lusin). La mesure de Lebesgue, bien définie sur les classes boréliennes, peut être étendue de la manière suivante : une classe $\mathcal{A} \subseteq 2^{\mathbb{N}}$ est *mesurable* si elle est égale à $\mathcal{B} \cup \mathcal{C}$, où $\mathcal{B}$ est une classe borélienne et $\mathcal{C}$ est incluse dans une classe borélienne de mesure nulle. La mesure de $\mathcal{A}$ est alors définie comme étant celle de $\mathcal{B}$. Montrer que toute classe Π_1^1 — et donc aussi toute classe Σ_1^1 — est mesurable.

Indication. — Commencer par montrer qu’il existe un ordinal dénombrable α tel que la classe des ensembles X qui calculent un code pour α est de mesure nulle. $\diamond$

4. Analogies entre ensembles Π_1^1 et ensembles c. e.

Il existe une analogie entre les ensembles Π_1^1 et les ensembles c. e. Pour commencer, il devrait être à peu près clair, via le corollaire 3.3, que le $\mathcal{O}$ de Kleene est une version du problème de l’arrêt pour les prédicats Π_1^1 . L’analogie va plus loin que cela : il est possible de considérer les ensembles Π_1^1 comme étant réellement des ensembles calculatoirement énumérables, mais avec des temps de calcul qui peuvent se prolonger le long des ordinaux calculables. Cette manière de voir les choses découle directement du corollaire 3.6 : étant donné $A = \bigcup_{\alpha < \omega_1^{ck}} A_\alpha$, on peut voir les éléments de $A_{\alpha+1} \setminus A_\alpha$ comme étant les éléments « énumérés » dans A à l’étape $\alpha + 1$. Cette idée sera énoncée précisément avec le théorème 4.3.

Nous commençons par énoncer un théorème caractéristique de cette analogie entre ensembles Π_1^1 et ensemble c. e. : supposons que l’on ait une

suite $(A_n)_{n \in \mathbb{N}}$ de sous-ensembles de $\mathbb{N}$, avec chaque A_n c. e. uniformément en n . Alors, il est possible de définir une fonction partielle calculable telle que $A_n \neq \emptyset$ implique $f(n) \in A_n$: il suffit pour tout n de définir $f(n)$ comme étant le premier élément énuméré dans A_n . On peut faire exactement la même chose dans le cas Π_1^1 , mais en considérant cette fois-ci les éléments comme étant énumérés le long de temps de calcul ordinaux.

Théorème 4.1 (Uniformisation Π_1^1 de Kreisel)

Soit $A \subseteq \mathbb{N} \times \mathbb{N}$ une classe Π_1^1 . Alors, il existe une fonction Π_1^1 (c'est-à-dire de graphe Π_1^1) partielle $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que :

$$\forall n (\exists m (n, m) \in A \rightarrow (n, f(n)) \in A).$$

PREUVE. En utilisant le corollaire 3.6, soit $A = \bigcup_{\alpha < \omega_1^{ck}} A_\alpha$, où chaque A_α est un ensemble $\Sigma_{\alpha+1}^0$ uniformément en un code de α .

On définit f de la manière suivante : $f(n) = x$ si x est le plus petit élément tel que $(n, x) \in A_\alpha$ pour α le plus petit ordinal tel que $\{y : (n, y) \in A_\alpha\}$ est non vide. Formellement, $f(n) = x$ s'il existe $\alpha < \omega_1^{ck}$ tel que $(n, x) \in A_\alpha$, tel que $(n, y) \notin A_\beta$ pour tout y et pour tout $\beta < \alpha$, et tel que $(n, y) \notin A_\alpha$ pour tout $y < x$.

Pour voir qu'il s'agit bien d'une définition Π_1^1 , il faut commencer par passer par les codages des ordinaux par des éléments de $\mathcal{O}$. Nous anticipons ensuite un peu sur le théorème 5.1 à venir pour profiter du fait que les prédicats de la forme $x \in A$ pour A un ensemble Σ_α^0 sont uniformément Δ_1^1 : étant donné une réduction many-one h de A à H_{2^a} pour $a \in \mathcal{O}_{=\alpha}$, on a $x \in A$ ssi $\exists X \mathcal{P}(2^a, X) \wedge h(x) \in X$ ssi $\forall X \mathcal{P}(2^a, X) \rightarrow h(x) \in X$. On utilise ici la classe $\Pi_2^0 \mathcal{P}$ du théorème 28-2.1.

Comme c'est le premier exemple de ce genre, nous donnons ici la définition Π_1^1 formelle de notre fonction f . Dans la suite, on s'autorisera les descriptions Π_1^1 de la forme $\exists \alpha < \omega_1^{ck} \dots$. Pour cela, soit h la fonction calculable telle que $h(a, e, n, x) \in H_{2^a}$ ssi (n, x) appartient à l'ensemble $\Sigma_{|a|}^0$ de code e . Soit g la fonction calculable telle que $g(a)$ est le $\Sigma_{|a|}^0$ -code de $A_{|a|}$ pour tout $a \in \mathcal{O}$. Pour finir, étant donné un ensemble X dont on suppose qu'il est égal à H_{2^a} , on note X_b pour $b <_o a$ l'ensemble obtenu de manière calculable à partir de X , qui est égal à H_{2^b} si $X = H_{2^a}$. On a alors $f(n) = x$ si :

$$\exists a \ a \in \mathcal{O} \wedge \forall X \left(\mathcal{P}(2^a, X) \Rightarrow \left(\begin{array}{l} \wedge \quad h(a, g(a), n, x) \in X \\ \wedge \quad \forall b <_o a \ \forall y \ h(b, g(b), n, y) \notin X_b \end{array} \right) \right).$$

Notons que la preuve du théorème d'uniformisation fonctionne aussi pour les classes $\Pi_1^1 \mathcal{A} \subseteq 2^{\mathbb{N}} \times \mathbb{N}$ et des fonctionnelles Π_1^1 partielles $f : 2^{\mathbb{N}} \rightarrow \mathbb{N}$

qui sur un oracle X tel que $\{n : (X, n) \in \mathcal{A}\}$ est non vide, renvoient un entier tel que $(X, f(X)) \in \mathcal{A}$. Nous verrons avec le corollaire 30-5.3 — et c'est bien plus difficile à montrer — que le théorème d'uniformisation Π_1^1 fonctionne même pour des classes $\Pi_1^1 \mathcal{A} \subseteq 2^{\mathbb{N}} \times 2^{\mathbb{N}}$.

En attendant, continuons sur l'analogie entre être Π_1^1 et c. e. Reprenons notre exemple d'ensemble c. e. $(A_n)_{n \in \mathbb{N}}$ et de notre fonction partielle calculable telle que $A_n \neq \emptyset$ implique $f(n) \in A_n$. Supposons un instant qu'aucun ensemble A_n ne soit vide. Alors, notre fonction devient une fonction totale calculable, et donc descriptible de manière Δ_1^0 . En fait, n'importe quelle fonction totale de graphe c. e. admet un graphe calculable, puisque pour décider si $f(n) = x$ il suffit de chercher l'unique y tel que $f(n) = y$ et de voir si $x = y$. Il en va de même pour les fonctions de graphes Π_1^1 .

Proposition 4.2. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ une fonction Π_1^1 totale. Alors, f est Δ_1^1 .★

PREUVE

Il suffit de voir que la relation $f(n) \neq x$ est équivalente à $\exists y \neq x \ f(n) = y$. La relation $f(n) \neq x$ est donc elle aussi Π_1^1 , ce qui rend la fonction Δ_1^1 . ■

Nous verrons dans la prochaine section que les ensembles Δ_1^1 coïncident avec les ensembles hyperarithmétiques. En attendant, voici l'aspect formel du fait que les Π_1^1 peuvent être vus comme un analogue d'être c. e.

Théorème 4.3

Pour tout Π_1^1 non vide $A \subseteq \mathbb{N}$, il existe une fonction Π_1^1 totale $f : \mathcal{O} \rightarrow \mathbb{N}$ telle que pour tout $\mathcal{O}_1 \subseteq \mathcal{O}$ où chaque ordinal calculable admet exactement un code dans $\mathcal{O}_1$, on a $A = \{f(a) : a \in \mathcal{O}_1\}$.

PREUVE. Il s'agit d'une amélioration de la preuve du théorème d'uniformisation. Soit $A = \bigcup_{\alpha < \omega_1^{ck}} A_\alpha$, et soit $y \in A$ un élément fixé. On définit la fonction $f : \mathcal{O} \rightarrow \mathbb{N}$ de la manière suivante. Sur un élément $a \in \mathcal{O}$ codant pour un ordinal α , la fonction commence par calculer le code d'un ordinal β et d'un entier n tel, que $\alpha = \beta \times \omega + n$. Puis, la fonction renvoie le n -ième élément de A_β s'il existe. Sinon, $f(a) = y$.

Il suffit de montrer que

- ▷ (1) pour chaque ordinal α , il existe une unique ordinal β et un unique entier n tels que $\alpha = \beta \times \omega + n$;
- ▷ (2) pour tout $\beta < \omega_1^{ck}$, l'ordinal $\beta \times \omega + n < \omega_1^{ck}$.

Nous laissons ces preuves élémentaires au lecteur.

La fonction f est bien définie par l'unicité de β et n , et est totale car $f(a) = y$ si A_β ne possède pas de n -ième élément. De plus, pour tout $a \in \mathcal{O}$, $f(a) \in A$.

Soit $\mathcal{O}_1 \subseteq \mathcal{O}$ un ensemble contenant exactement un code de chaque ordinal calculable. Montrons que $A \subseteq \{f(a) : a \in \mathcal{O}_1\}$.

Soit $z \in A$, et soient $\beta < \omega_1^{ck}$ et $n \in \mathbb{N}$ tels que z est le n -ième élément de A_β . Par (2), $\beta \times \omega + n < \omega_1^{ck}$, donc il existe un $a \in \mathcal{O}_1$ tel que $|a| = \beta \times \omega + n$, donc $f(a) = z$. Ainsi, $A = \{f(a) : a \in \mathcal{O}_1\}$. ■

L'analogie entre être c. e. et être Π_1^1 n'est pas uniquement formelle. En 1989, Joel David Hamkins et Jeff Kidder imaginent une machine de Turing où le temps de calcul peut être n'importe quel ordinal. Concrètement, la machine de Turing a le même comportement qu'habituellement aux étapes de calcul successeurs. Aux étapes limites, chaque cellule de la machine prend comme valeur la limite inférieure des valeurs aux étapes précédentes, la machine rentre dans un état « limite » et la tête de lecture revient à la première cellule. L'article fondateur [82] sur l'étude de ces machines de Turing à temps infini ne viendra que plus tard, et leurs auteurs y montrent notamment que les réels énumérables par de telles machines, en temps ordinal inférieur à ω_1^{ck} , sont exactement les ensembles Π_1^1 .

5. Théorème d'équivalence de Kleene/Souslin

Nous voyons dans cette section un théorème fondamental : les ensembles/classes Δ_1^1 et hyperarithmétiques coïncident. Ce résultat remonte aux travaux de Souslin au début du XX^e siècle. La notion de calculabilité était alors balbutiante, aussi le résultat de Souslin porte-t-il sur les classes Σ_1^1 , qu'il avait lui-même découvertes un peu plus tôt, via la fameuse « erreur de Lebesgue », dont nous avons déjà parlé dans l'interlude historique au début du chapitre.

5.1. Le théorème d'équivalence pour les ensembles

Nous commençons par montrer la direction simple : si un ensemble est hyperarithmétique, alors il est Δ_1^1 .

Théorème 5.1 (Kleene [116])

Les ensembles hyperarithmétiques sont Δ_1^1 .

PREUVE. Soit $\mathcal{P} \subseteq \mathbb{N} \times 2^{\mathbb{N}}$ la classe Π_2^0 du théorème 28-2.1, telle que pour $a \in \mathcal{O}$ la classe $\{X : \mathcal{P}(a, X)\}$ est la classe Π_2^0 contenant uniquement H_a . Soit X hyperarithmétique. Par le théorème 28-1.12, soient $e \in \mathbb{N}$ et $a \in \mathcal{O}$ tels que $\Phi_e(H_a) = X$. Alors, on a

- (1) $n \in X$ ssi $\exists Y \mathcal{P}(a, Y)$ et $\exists \sigma \prec Y \Phi_e(\sigma, n) \downarrow = 1$;
- (2) $n \notin X$ ssi $\exists Y \mathcal{P}(a, Y)$ et $\exists \sigma \prec Y \Phi_e(\sigma, n) \downarrow = 0$.

On a bien une description Σ_1^1 de X et de son complémentaire. ■

Notons que l'on peut de plus à partir d'un Σ_α^0 -code d'un ensemble A et d'un élément $a \in \mathcal{O}_{=\alpha}$, obtenir uniformément un code Δ_1^1 pour A , c'est-à-dire une paire de codes Σ_1^1 et Π_1^1 pour A . On utilise pour cela la réduction many-one uniforme des ensembles Σ_α^0 aux ensembles H_a pour $a \in \mathcal{O}_{\alpha+1}$.

En particulier, on peut obtenir uniformément en a un code Δ_1^1 de l'ensemble $\mathcal{O}_{<a}$, ce qui sera utilisé pour le prochain lemme.

Théorème 5.2 (Kleene)

Un ensemble $X \subseteq \mathbb{N}$ est Δ_1^1 si, et seulement si, il est hyperarithmétique.

Nous avons déjà la direction X hyperarithmétique implique $X \Delta_1^1$. L'autre direction est plus subtile, et fait appel au *lemme de majoration* Σ_1^1 de Spector.

Lemme 5.3 (Majoration Σ_1^1 , Spector [214]). Soit $A \subseteq \mathcal{O}$ un ensemble Σ_1^1 . Alors, $\sup_{a \in A} |a| < \omega_1^{ck}$. ★

PREUVE. Supposons au contraire que $\sup_{a \in A} |a| = \omega_1^{ck}$. Alors, on peut donner la description Σ_1^1 de $\mathcal{O}$ suivante, en contradiction avec le corollaire 3.4 :

$$a \in \mathcal{O} \iff \exists b \in A \text{ tel que } a \in \mathcal{O}_{<b}.$$

Comme on peut obtenir un code Σ_1^1 pour chaque ensemble $\mathcal{O}_{<b}$ uniformément en b , la description ci-dessus est bien Σ_1^1 , ce qui contredit le fait que $\mathcal{O}$ n'admette pas de description Σ_1^1 . Donc, $\sup_{a \in A} |a| < \omega_1^{ck}$. ■

PREUVE DU THÉORÈME 5.2. D'après le théorème 5.1, les ensembles hyperarithmétiques sont Δ_1^1 . Supposons à présent que $A \subseteq \mathbb{N}$ soit Δ_1^1 . Comme A est Π_1^1 , il existe une fonction calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $n \in A$ ssi $f(n) \in \mathcal{O}$. Comme A est Σ_1^1 , l'ensemble $\{f(n) : n \in A\}$ est Σ_1^1 . D'après le lemme de majoration Σ_1^1 , il existe un ordinal $\alpha < \omega_1^{ck}$ tel que $n \in A$ ssi $f(n) \in \mathcal{O}_{<\alpha}$. Comme $\mathcal{O}_{<\alpha}$ est hyperarithmétique, alors A est aussi hyperarithmétique. ■

5.2. Première conséquence

L'équivalence de Kleene/Souslin nous permet de compléter le théorème 28-2.4, qui stipule que les ensembles arithmétiques ont tous un module. Rappelons qu'une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ est un module d'un ensemble $A \subseteq \mathbb{N}$ si toute fonction dominant f calcule A .

Théorème 5.4 (Groszek et Slaman [79])

Un ensemble $X \subseteq \mathbb{N}$ est hyperarithmétique si, et seulement si, il admet un module.

La première direction est le théorème 28-2.4. L'autre direction du théorème nécessite le lemme suivant qui est intéressant indépendamment de l'utilisation que l'on en fera.

Lemme 5.5. Soit X un ensemble admettant un modulus. Alors, X admet un modulus uniforme : il existe un code e , et une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que pour toute fonction $g \geq f$ on a $\Phi_e(g) = X$. ★

PREUVE. Soit f une fonction telle que, pour tout $g \geq f$, on ait $g \geq_T X$. Supposons par l'absurde que X n'admette pas de modulus uniforme, c'est-à-dire que, pour tout e et toute fonction g , il existe une fonction $h \geq g$ telle que $\Phi_e(h) \neq X$. On construit alors par forcing une fonction $g \geq f$ telle que $g \not\geq_T X$, contredisant le fait que f soit un modulus.

On introduit pour la construction la notation suivante : pour $\sigma \in \mathbb{N}^{<\mathbb{N}}$ et $g \in \mathbb{N}^{\mathbb{N}}$, on note σg la fonction résultant de la concaténation de σ et g , c'est-à-dire la fonction qui à $n < |\sigma|$ associe $\sigma(n)$, et qui à $n \geq |\sigma|$ associe $g(n - |\sigma|)$.

Nos conditions $\mathbb{P}$ sont simplement des couples $\langle \sigma, g \rangle$ tels que $\sigma g \geq f$. L'extension de nos conditions de forcing est définie par $\langle \tau, h \rangle \preceq \langle \sigma, g \rangle$ si $\tau \succeq \sigma$ et $\tau h \geq \sigma g$. De plus, étant donné $\langle \sigma, g \rangle \in \mathbb{P}$ et $\tau \in \mathbb{N}^{<\mathbb{N}}$, on écrit $\tau \in [\sigma, g]$ si $\tau \succeq \sigma$ et $\tau(n) \geq (\sigma g)(n)$ pour tout $n < |\tau|$.

Étant donné une condition $\langle \sigma, g \rangle \in \mathbb{P}$ et un code e , on affirme que l'une des deux possibilités suivantes arrive nécessairement.

- (1) Il existe $m \in \mathbb{N}$ et il existe $\tau \in [\sigma, g]$ tels que $\Phi_e(\tau, m) \downarrow \neq X(m)$
- (2) Il existe $\langle \tau, h \rangle \preceq \langle \sigma, g \rangle$ et il existe $m \in \mathbb{N}$ tel que, pour tout $\rho \in [\tau, h]$, on a $\Phi_e(\rho, m) \uparrow$

Supposons que ni (1) ni (2) n'arrive, alors on a la conjonction des deux énoncés suivants.

- (3) Pour tout entier $m \in \mathbb{N}$ et pour tout $\tau \in [\sigma, g]$, on a $\Phi_e(\tau, m) \downarrow$ implique $\Phi_e(\tau, m) = X(m)$
- (4) Pour tout $\langle \tau, h \rangle \preceq \langle \sigma, g \rangle$ et pour tout $m \in \mathbb{N}$, il existe $\rho \in [\tau, h]$ tel que $\Phi_e(\rho, m) \downarrow$

Il existe par conséquent une unique fonctionnelle Φ telle que, pour toute fonction $h \geq \sigma g$, on a $\Phi(h) = X$. Étant donné un tel h et un entier m , la fonctionnelle cherche $\rho \succeq \sigma$, avec $\rho(n) \geq h(n)$ pour tout $|\sigma| \leq n < |\rho|$ et tel que $\Phi_e(\rho, m) \downarrow = i$. Puis, la fonctionnelle renvoie i . D'après (4), une telle recherche aboutit forcément. D'après (3), la valeur i est nécessairement égale à $X(m)$.

Comme l'existence d'une telle fonctionnelle contredit nos hypothèses, on peut donc pour chaque code e et chaque condition de forcing trouver une

extension satisfaisant (1) ou (2). Dans chacun des cas, on s'assure que notre générique ne calculera pas X via Φ_e . Le générique consistera alors en une fonction $g \geq f$ telle que $X \not\leq_T g$, ce qui contredit le fait que f soit un modulus pour X .

Donc, si X admet un modulus, il admet un modulus uniforme. ■

PREUVE DU THÉORÈME 5.4. Nous avons vu avec le théorème 28-2.4 que tout ensemble hyperarithmétique admet un modulus. Supposons à présent que X admette un modulus. D'après le lemme 5.5, X admet un modulus uniforme via une fonctionnelle Φ_e . Dans ce qui suit, on écrit $\sigma \geq f$ pour signifier $\sigma(m) \geq f(m)$ pour tout $m < |\sigma|$. Soit $\Psi(n)$ le prédicat Σ_1^1 donné par

$$\exists f \text{ tel que } \forall \sigma \geq f \text{ on a } \Phi_e(\sigma, n) \downarrow \rightarrow \Phi_e(\sigma, n) = 1.$$

Si $n \in X$, alors $\Psi(n)$ est clairement vrai en prenant f comme étant le témoin de notre modulus. Si à présent $\Psi(n)$ est vrai pour une certaine fonction f , alors en prenant $g > f$ suffisamment grand pour que $\Phi_e(g) = X$, on a par conséquent $\Phi_e(\sigma, n) \downarrow = X(n)$ pour un certain $\sigma \prec g$. Comme $\Psi(n)$ est vrai avec f , on a en fait $\Phi_e(\sigma, n) \downarrow = 1$, et donc $n \in X$. Il s'ensuit que $n \in X$ ssi $\Psi(n)$ est vrai. Donc, X est Σ_1^1 . On procède de la même manière pour montrer que le complémentaire de X est Σ_1^1 , ce qui implique que X est Δ_1^1 , et donc hyperarithmétique. ■

5.3. Uniformité du théorème d'équivalence

Nous avons vu qu'il est possible d'obtenir le code Δ_1^1 d'un ensemble A à partir de son code hyperarithmétique. Dans l'autre sens, le théorème 5.2 ne nous donne en revanche aucun moyen d'obtenir un code hyperarithmétique d'un ensemble A à partir de son code Δ_1^1 . Le point crucial se situe dans la démonstration du lemme de majoration Σ_1^1 de Spector, qui permet de montrer l'existence d'un majorant pour un ensemble Σ_1^1 d'ordinaux $A \subseteq \mathcal{O}$, mais sans le spécifier.

Nous voyons ici qu'il est possible d'obtenir cette borne uniformément, au prix bien entendu d'une construction plus complexe. Nous utiliserons pour cela des représentations des ordinaux via des arbres bien fondés de l'espace de Baire, et des lemmes de manipulation sur ces derniers.

Définition 5.6. Soient deux arbres $T_1, T_2 \subseteq \mathbb{N}^{<\mathbb{N}}$. On appelle *morphisme* de T_1 vers T_2 une fonction $f : T_1 \rightarrow T_2$ telle que $|f(\sigma)| = |\sigma|$ et telle que $\sigma \preceq \tau$ implique $f(\sigma) \preceq f(\tau)$. ◇

La notation $|\sigma|$ ci-dessus signifie la taille de σ . On commence par une première proposition permettant de comparer les ordinaux codés par des arbres bien fondés via l'existence de morphisme de l'un vers l'autre.

Souvenons-nous pour la preuve qui suit de la notation $|T \upharpoonright_n|$ introduite dans la section 27-5.3.

Proposition 5.7. Étant donné deux arbres bien fondés $T_1, T_2 \subseteq \mathbb{N}^{<\mathbb{N}}$, on a $|T_1| \leq |T_2|$ si, et seulement si, il existe un morphisme de T_1 vers T_2 . ★

PREUVE. On montre la proposition par induction sur les ordinaux codés par T_1 . Si $|T_1| = 0$, alors T_1 est l'ensemble vide, et la proposition est clairement vraie avec n'importe quel arbre T_2 . Supposons à présent la proposition vraie pour n'importe quel arbre T_1 tel que $|T_1| < \alpha$ et n'importe quel arbre T_2 . Considérons un arbre T_1 avec $|T_1| = \alpha$.

Supposons d'abord $|T_1| \leq |T_2|$. Alors, pour chaque nœud $n \in T_1$ — c'est-à-dire chaque nœud de T de taille 1 —, il y a un nœud $m_n \in T_2$ tel que

$$|T_1 \upharpoonright_n| \leq |T_2 \upharpoonright_{m_n}|$$

(s'il y en a plusieurs, on choisit le plus petit). Par hypothèse d'induction, on a donc un morphisme $f_n : T_1 \upharpoonright_n \rightarrow T_2 \upharpoonright_{m_n}$. On définit alors $f : T_1 \rightarrow T_2$ par $f(n\sigma) = m_n f_n(\sigma)$, et l'on vérifie sans peine que c'est un morphisme de T_1 vers T_2 .

Réciproquement, supposons qu'il existe un morphisme de $f : T_1 \rightarrow T_2$. Alors, pour chaque $n \in T_1$, la fonction $f_n : T_1 \upharpoonright_n \rightarrow T_2 \upharpoonright_{f(n)}$ est un morphisme de $T_1 \upharpoonright_n$ vers $T_2 \upharpoonright_{f(n)}$. On obtient par hypothèse d'induction que $|T_1 \upharpoonright_n| \leq |T_2 \upharpoonright_{f(n)}|$. Donc, $|T_1| \leq |T_2|$. ■

On définit à présent des fonctions calculables $\wedge : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ et $\vee : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ sur les paires de codes c. e. d'arbres bien fondés de $\mathbb{N}^{<\mathbb{N}}$.

Lemme 5.8. Il y a une fonction totale calculable $\wedge : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ qui sur deux codes d'arbre c. e. T_1, T_2 renvoie le code d'un arbre c. e. $T_1 \wedge T_2$ bien fondé si, et seulement si, T_1 et T_2 sont tous deux bien fondés, auquel cas on a $|T_1 \wedge T_2| \geq \sup(|T_1|, |T_2|)$. ★

PREUVE. On définit simplement $T_1 \wedge T_2$ comme étant la réunion disjointe de T_1 et T_2 . Concrètement, on met dans $T_1 \wedge T_2$ les nœuds $\langle n, i \rangle \sigma$ pour tout $n\sigma \in T_i$ pour $i \in \{1, 2\}$. L'existence d'un morphisme de T_1 ou T_2 vers $T_1 \wedge T_2$ est claire. On a donc $|T_1 \wedge T_2| \geq \sup(|T_1|, |T_2|)$.

Lemme 5.9. Il existe une fonction totale calculable $\vee$, qui sur deux codes d'arbres c. e. T_1, T_2 renvoie le code d'un arbre c. e. $T_1 \vee T_2$ qui est bien fondé ssi T_1 ou T_2 est bien fondé, auquel cas $|T_1 \vee T_2| \geq \min(|T_1|, |T_2|)$. ★

PREUVE. Dans ce qui suit, on utilise une fonction de couplage sur les chaînes de même taille de $\mathbb{N}^{<\mathbb{N}}$, définie en appliquant successivement la

bijection $\langle \rangle : \mathbb{N}^2 \rightarrow \mathbb{N}$ sur chaque pair d'entiers se trouvant à la même position dans les deux chaînes. Formellement, pour $n = |\sigma_1| = |\sigma_2|$,

$$\langle \sigma_1, \sigma_2 \rangle = \langle \sigma_1(0), \sigma_2(0) \rangle \smallfrown \cdots \smallfrown \langle \sigma_1(n-1), \sigma_2(n-1) \rangle.$$

La définition de $T_1 \vee T_2$ est simple : on énumère un nœud σ dans $T_1 \vee T_2$ si $\sigma = \langle \sigma_1, \sigma_2 \rangle$ où pour $i \in \{1, 2\}$, chaque chaîne σ_i a été énumérée dans T_i . Il est clair que l'on a un chemin infini dans $T_1 \vee T_2$ ssi l'on a un chemin infini dans T_1 et T_2 .

Pour montrer l'inégalité $|T_1 \vee T_2| \geq \min(|T_1|, |T_2|)$, on suppose sans perte de généralité que $|T_1| \leq |T_2|$. Soit $f : T_1 \rightarrow T_2$ un morphisme (notons que, dans le cas où T_2 est mal fondé, un morphisme existe toujours en envoyant tous les nœuds de T_1 vers le chemin infini de T_2). Alors, chaque nœud $\langle \sigma, f(\sigma) \rangle$ appartient à $T_1 \vee T_2$. On peut alors définir le morphisme $g : T_1 \rightarrow |T_1 \vee T_2|$ qui à $\sigma \in T_1$ associe $\langle \sigma, f(\sigma) \rangle$. Il s'ensuit que $|T_1 \vee T_2| \geq |T_1|$. ■

Nous aurons également besoin de combiner une suite infinie d'arbres.

Lemme 5.10. Il existe une fonction totale calculable qui, pour chaque code énumérant une suite infinie $(T_i)_{i \in \mathbb{N}}$ d'arbres c.e., renvoie le code d'un arbre c.e. $\bigwedge_{i \in \mathbb{N}} T_i$, lequel est bien fondé ssi chaque T_i est bien fondé, auquel cas on a $|\bigwedge_{i \in \mathbb{N}} T_i| \geq \sup_i |T_i|$. ★

PREUVE. On définit simplement T comme étant la réunion disjointe des T_i . Concrètement, on met dans T les nœuds $\langle n, i \rangle \sigma$ pour tout $n\sigma \in T_i$. Il est clair que l'on a un morphisme de chaque T_i vers $|\bigwedge_{i \in \mathbb{N}} T_i|$, et donc que $|\bigwedge_{i \in \mathbb{N}} T_i| \geq \sup_i |T_i|$. ■

Exercice 5.11. (★) Montrer l'existence d'une fonction totale calculable $\vee$, qui pour chaque code énumérant une suite infinie $(T_i)_{i \in \mathbb{N}}$ d'arbres c.e., renvoie le code d'un arbre c.e. $\bigvee_{i \in \mathbb{N}} T_i$, lequel est bien fondé ssi au moins l'un des T_i est bien fondé, auquel cas on a

$$\min_i (|T_i|) + \omega > \left| \bigvee_{i \in \mathbb{N}} T_i \right| \geq \min_i (|T_i|).$$

◇

On peut à présent montrer la version uniforme du lemme de majoration Σ_1^1 de Spector.

Lemme 5.12 (Borne uniforme, majoration Σ_1^1 pour les entiers). Il existe une fonction calculable telle que si e est le code de $A \subseteq \mathcal{O}$ un ensemble Σ_1^1 , alors $f(e) \in \mathcal{O}$ est tel que $\sup_{a \in A} |a| \leq |f(e)|$. ★

PREUVE. Soit $A \subseteq \mathcal{O}$ un ensemble Σ_1^1 . D'après la proposition 27-5.18, on a une fonction calculable f tel que $f(e)$ code pour un arbre c.e. T_e tel que T_e est bien fondé ssi $e \in \mathcal{O}$. De plus, si $e \in \mathcal{O}$, on a $|e| \leq |T_e|$.

En utilisant le fait que A soit Σ_1^1 , on obtient via le théorème 3.2 et la proposition 27-5.18 — uniformément en un code de A — une fonction calculable g telle que $g(e)$ code pour un arbre c. e. U_e tel que U_e est bien fondé ssi $e \notin A$. Notons que pour tout $e \in \mathbb{N}$, l'arbre $T_e \vee U_e$ est bien fondé. De plus, d'après le lemme 5.9, si $e \in A \subseteq \mathcal{O}$, on a $|e| \leq |T_e \vee U_e|$.

On prend alors l'arbre c. e. T de code $\bigwedge_{e \in \mathbb{N}} (T_e \vee U_e)$. D'après le lemme 5.10 on a $|T_e \vee U_e| \leq |\bigwedge_{e \in \mathbb{N}} (T_e \vee U_e)|$ pour tout e , et donc $|e| \leq |\bigwedge_{e \in \mathbb{N}} (T_e \vee U_e)|$ pour tout $e \in A \subseteq \mathcal{O}$. On peut alors, pour conclure, retransformer le code de T en un code de $\mathcal{O}$ à l'aide du théorème 27-5.10. ■

On déduit aisément du lemme une borne uniforme permettant de transformer le code Δ_1^1 d'un ensemble en code hyperarithmétique.

5.4. Le théorème d'équivalence pour les classes

Nous voyons à présent le théorème d'équivalence de Kleene/Souslin pour les classes, et nous montrons cette fois-ci directement la version uniforme du théorème.

Proposition 5.13. Si une classe $\mathcal{A} \subseteq 2^{\mathbb{N}}$ est hyperarithmétique, elle est alors Δ_1^1 . ★

PREUVE. Soit $\mathcal{A} \subseteq 2^{\mathbb{N}}$ une classe Σ_α^0 . Soit $\mathcal{P} \subseteq \mathbb{N} \times 2^{\mathbb{N}} \times 2^{\mathbb{N}}$ la classe Π_2^0 du théorème 28-3.4, c'est-à-dire telle que, pour $a \in \mathcal{O}^X$ et $X \in 2^{\mathbb{N}}$, la classe $\{Y : \mathcal{P}(a, X, Y)\}$ est la classe $\Pi_2^0(X)$ contenant uniquement H_a^X .

D'après le théorème 28-4.4, on a un code e tel que $X \in \mathcal{A}$ ssi $e \in H_{2a}^X$ pour $a \in \mathcal{O}_{=\alpha}$. Il est à noter que l'on peut supposer sans perte de généralité $\mathcal{O} \subseteq \mathcal{O}^X$ — afin de satisfaire formellement les hypothèses du théorème 28-3.4. On a la définition Σ_1^1 suivante pour $\mathcal{A}$:

$$\mathcal{A} = \{X : \exists Y \mathcal{P}(2^a, X, Y) \text{ et } e \in Y\},$$

et la définition Σ_1^1 suivante pour $2^{\mathbb{N}} \setminus \mathcal{A}$:

$$2^{\mathbb{N}} \setminus \mathcal{A} = \{X : \exists Y \mathcal{P}(2^a, X, Y) \text{ et } e \notin Y\}.$$

La classe $\mathcal{A}$ est donc Δ_1^1 . ■

Montrons à présent le lemme de majoration Σ_1^1 pour les classes, dû à Spector [214].

Lemme 5.14 (Majoration Σ_1^1 pour les classes)

Soit une classe Σ_1^1 $\mathcal{A} \subseteq 2^{\mathbb{N}}$ telle que $X \in \mathcal{A} \rightarrow e \in \mathcal{O}^X$ pour un certain e . Alors, on peut trouver $a \in \mathcal{O}$ uniformément en e et en un code de $\mathcal{A}$ tels que $X \in \mathcal{A} \rightarrow |e|^X < |a|$. ★

PREUVE. On réutilise ici la fonction de couplage sur les chaînes de $\mathbb{N}^{<\mathbb{N}}$ de même taille, définie dans la preuve du lemme 5.9. Soit U_1^X l'arbre c.e. bien fondé ssi $X \notin \mathcal{A}$. Soit U_2^X l'arbre c.e. bien fondé ssi $e \in \mathcal{O}^X$. On calcule uniformément en X l'arbre bien fondé $T^X = U_1^X \vee U_2^X$. On calcule finalement l'arbre c.e. T suivant : pour chaque chaîne $\sigma \in 2^{<\mathbb{N}}$ tel qu'un nœud m de taille 1 est énuméré dans T^σ , on énumère le nœud $\langle m, b(\sigma) \rangle$ de taille 1 dans T , en utilisant une bijection $b : 2^{<\mathbb{N}} \rightarrow \mathbb{N}$. Pour tout nœud $\langle \tau, b(\sigma_1) \dots b(\sigma_n) \rangle$ énuméré dans T avec $|\tau| = n$, si une chaîne τm est énumérée dans $T^{\sigma_{n+1}}$ pour une chaîne $\sigma_{n+1} \succeq \sigma_n$, on énumère alors

$$\langle \tau m, b(\sigma_1) \dots b(\sigma_n) b(\sigma_{n+1}) \rangle$$

dans T .

L'arbre T est bien fondé : un chemin infini $g \oplus f$ de $[T]$ correspondrait à un élément $g \in [T^Y]$ pour $b^{-1}(f(0)) \preceq b^{-1}(f(1)) \preceq b^{-1}(f(2)) \preceq \dots \preceq Y$, contredisant le fait que chaque T^Y soit bien fondé. De plus, pour tout X , on a un morphisme f de T^X dans T , défini pour un nœud τ de taille n par $f(\tau)$ comme étant $\tau \oplus b(\sigma_1) \dots b(\sigma_n)$ pour les premières chaînes $\sigma_1 \preceq \sigma_2 \preceq \dots \preceq \sigma_n \prec X$ trouvées telles que $\tau \in T^{\sigma_n}$. On en déduit donc $|T^X| \leq |T|$ pour tout X . ■

Théorème 5.15

Une classe $\mathcal{A} \subseteq 2^{\mathbb{N}}$ est Δ_1^1 si, et seulement si, elle est hyperarithmétique.

PREUVE. Nous avons vu avec la proposition 5.13 qu'une classe hyperarithmétique est Δ_1^1 . Supposons à présent que $\mathcal{A} \subseteq 2^{\mathbb{N}}$ soit une classe Δ_1^1 . Comme $\mathcal{A}$ est Π_1^1 , d'après le théorème 3.2 il existe un code e tel que $X \in \mathcal{A}$ si, et seulement si, $e \in \mathcal{O}^X$. Comme $\mathcal{A}$ est Σ_1^1 , la classe $\{X : e \in \mathcal{O}^X\}$ est Σ_1^1 . D'après le lemme de majoration Σ_1^1 pour les réels, il existe $\alpha < \omega_1^{ck}$ tel que $\mathcal{A} = \{X : e \in \mathcal{O}_{<\alpha}^X\}$. D'après le théorème 28-4.5, la classe $\mathcal{A}$ est donc hyperarithmétique. ■

6. Autres théorèmes de majoration

La technique utilisée pour le lemme de majoration Σ_1^1 est très puissante, et peut être exploitée pour mettre en évidence plusieurs autres phénomènes remarquables. Nous en voyons deux pour le moment.

Théorème 6.1

Soit T est un arbre bien fondé Σ_1^1 . Alors, $|T| < \omega_1^{ck}$.

PREUVE. Supposons par l'absurde $|T| \geq \omega_1^{ck}$. On montre alors que l'on a une description Σ_1^1 de $\mathcal{O}$, ce qui est une contradiction.

Soit f la fonction calculable telle que $f(e)$ est un code d'arbre c. e. T_e et tel que T_e est bien fondé ssi $e \in \mathcal{O}$. On peut alors donner une description Σ_1^1 de $\mathcal{O}$ de la manière suivante : $e \in \mathcal{O}$ ssi il existe un morphisme f de T_e vers T . Étant donné f , l'énoncé n'utilise que des questions d'appartenance à T positive (c'est-à-dire de la forme $\sigma \in T$ et non de la forme $\sigma \notin T$), et est donc bien Σ_1^1 :

$$\forall \sigma \in T_e \quad f(\sigma) \in T \wedge |f(\sigma)| = |\sigma| \wedge \sigma \preceq \tau \quad \text{implique} \quad f(\sigma) \preceq f(\tau).$$

On en déduit que $\mathcal{O}$ est Σ_1^1 , ce qui est une contradiction. ■

Le théorème précédent est absolument fascinant : le plus petit ordinal non calculable est le même que le plus petit ordinal non hyperarithmétique ! Toute la puissance conférée par des itérations arbitraires du saut ne permet pas de calculer une représentation de ω_1^{ck} . La preuve que nous avons donnée est non constructive. Il est toutefois possible d'en donner une version constructive, mais avec bien sûr un peu plus de travail.

Exercice 6.2. (★★) Montrer que l'on peut uniformément transformer un Σ_1^1 -code d'un arbre bien fondé T en un code c. e. d'arbre bien fondé T' tel que $|T'| \geq |T|$.

Indication.— On pourra reprendre des éléments de la preuve du lemme 5.12, et utiliser la fonction de l'exercice 5.11. ♦

Voyons à présent notre deuxième théorème, qui s'appuie là encore sur la même technique. Nous commençons par un lemme général.

Lemme 6.3. Soit $\mathcal{A}$ une classe Σ_1^1 telle que $\forall X \in \mathcal{A} \exists a \in \mathcal{O}^X \mathcal{B}(a, X)$ où $\mathcal{B}$ est un prédicat Π_1^1 . Alors, il existe un ordinal $\alpha < \omega_1^{ck}$ tel que

$$\forall X \in \mathcal{A} \exists a \in \mathcal{O}_{<\alpha}^X \mathcal{B}(a, X). \quad \star$$

PREUVE. Supposons par l'absurde que pour tout ordinal $\alpha < \omega_1^{ck}$ il existe $X \in \mathcal{A}$ tel que $\forall a \in \mathcal{O}_{<\alpha}^X \neg \mathcal{B}(a, X)$. Alors, on peut donner la description Σ_1^1 suivante du $\mathcal{O}$ de Kleene : $b \in \mathcal{O}$ si, et seulement si,

$$(1) \quad \exists X \in \mathcal{A} \forall a (a \notin \mathcal{O}^X \vee \neg \mathcal{B}(a, X) \vee b \in \mathcal{O}_{<|a|}).$$

Supposons $b \in \mathcal{O}$. Par hypothèse, il existe $X \in \mathcal{A}$ tel que

$$\forall a \in \mathcal{O}_{<\text{succ}(|b|)}^X \neg \mathcal{B}(a, X).$$

On vérifie alors sans peine que (1) est vrai pour cet élément X . Supposons à présent $b \notin \mathcal{O}$. Soit $X \in \mathcal{A}$. Soit $a \in \mathcal{O}^X$ tel que $\mathcal{B}(a, X)$. Comme $b \notin \mathcal{O}$, on a aussi $b \notin \mathcal{O}_{<|a|}$. Donc, pour tout $X \in \mathcal{A}$, il existe $a \in \mathcal{O}^X$ tel que $\mathcal{B}(a, X)$ et $b \notin \mathcal{O}_{<|a|}$. Ainsi, (1) est faux. On a donc $b \in \mathcal{O}$ ssi (1) est vrai. Donc, $\mathcal{O}$ admet une description Σ_1^1 , ce qui est une contradiction. ■

Théorème 6.4

Soit $\mathcal{A}$ une classe Σ_1^1 ne contenant que des éléments hyperarithmétiques. Alors, il existe $\alpha < \omega_1^{ck}$ tel que tous les éléments de $\mathcal{A}$ sont calculables en $\emptyset^{(\alpha)}$.

PREUVE. Il suffit d'appliquer le lemme précédent avec la formule suivante :

$$\forall X \in \mathcal{A} \quad \exists a \in \mathcal{O} \quad \text{tel que } \emptyset^{(|a|)} \geq_T X.$$

Exercice 6.5. () Soit $\mathcal{A}$ une classe Σ_1^1 ne contenant que des éléments hyperarithmétiques. Montrer que l'on peut trouver uniformément un ordinal $\alpha < \omega_1^{ck}$ tel que tous les éléments de $\mathcal{A}$ sont calculables par $\emptyset^{(\alpha)}$.** $\diamond$

7. Réduction hyperarithmétique

De la même manière que la version relativisée de la calculabilité induit une notion de réduction sur les ensembles, appelée réduction Turing, la relativisation des ensembles hyperarithmétiques induit une réduction sur les ensembles, appelée *réduction hyperarithmétique*. Cette réduction, plus « grossière » que la réduction Turing, est souvent plus appropriée en hypercalculabilité.

Définition 7.1. Un ensemble X est *hyperarithmétiquement réductible* à un autre ensemble Y si X est $\Delta_1^1(Y)$. On écrira alors $X \leq_h Y$, ainsi que $X <_h Y$ si $X \leq_h Y$ et $Y \not\leq_h X$. $\diamond$

De manière équivalente, $X \leq_h Y$ ssi il existe $a \in \mathcal{O}^Y$ tel que $X \leq_T H_a^Y$, ou avec nos notations alternatives, ssi il existe $\alpha < \omega_1^Y$ tel que $X \leq_T Y^{(\alpha)}$. Les deux théorèmes suivants illustrent un peu plus l'analogie entre $\mathcal{O}^X$ et X' .

Théorème 7.2

Soient $X, Y \in 2^{\mathbb{N}}$. Les deux énoncés suivants sont équivalents :

- (1) $X \leq_h Y$;
- (2) $\mathcal{O}^X \leq_m \mathcal{O}^Y$.

PREUVE. Supposons que X soit $\Delta_1^1(Y)$. Comme $\mathcal{O}^X$ est $\Pi_1^1(X)$, alors d'après l'exercice 2.9 $\mathcal{O}^X$ est $\Pi_1^1(Y)$. Comme $\mathcal{O}^Y$ est many-one complet pour les ensembles $\Pi_1^1(Y)$, on a $\mathcal{O}^X \leq_m \mathcal{O}^Y$.

Réciproquement, supposons $\mathcal{O}^X \leq_m \mathcal{O}^Y$. On a alors aussi $X \leq_m \mathcal{O}^X \leq_m \mathcal{O}^Y$ et $\mathbb{N} \setminus X \leq_m \mathcal{O}^X \leq_m \mathcal{O}^Y$. Comme $\mathcal{O}^Y$ est $\Pi_1^1(Y)$, alors X et $\mathbb{N} \setminus X$ sont tous les deux $\Delta_1^1(Y)$. Donc, X est $\Delta_1^1(Y)$. $\blacksquare$

Théorème 7.3

Soit $X \in 2^{\mathbb{N}}$. Alors, on a $X <_h \mathcal{O}^X$. Plus précisément, $X <_m \mathcal{O}^X$ et $\mathcal{O}^X \not\leq_h X$

PREUVE. Il est clair que $X <_m \mathcal{O}^X$. On ne peut avoir $\mathcal{O}^X \leq_h X$, car sinon $\mathcal{O}^X$ serait $\Delta_1^1(X)$, ce qui est une contradiction. ■

Les nombreuses analogies entre X' et $\mathcal{O}^X$ justifient le vocabulaire suivant.

Définition 7.4. Étant donné un ensemble X , l'ensemble $\mathcal{O}^X$ est appelé l'*hypersaut* de X . ◇

Voyons pour finir les rapports entre le calcul du $\mathcal{O}$ de Kleene et le calcul de ω_1^{ck} .

Théorème 7.5

Soient $X, Y \in 2^{\mathbb{N}}$. On a $X \leq_h Y$ implique $\omega_1^X \leq \omega_1^Y$.

PREUVE. Soit X un ensemble $\Delta_1^1(Y)$, et soit α un ordinal X -calculable. Alors, α a une représentation qui est $\Delta_1^1(Y)$. D'après la version relativisée du théorème 6.1, on a donc $\alpha < \omega_1^Y$. Par suite, $\omega_1^X \leq \omega_1^Y$. ■

On verra avec le théorème 30-2.2 que la réciproque n'est pas vraie. En particulier, il existe un ensemble non hyperarithmétique X tel que $\omega_1^X = \omega_1^{ck}$.

Théorème 7.6

Soit $X \in 2^{\mathbb{N}}$. Les deux énoncés suivants sont équivalents :

- (1) $X \geq_h \mathcal{O}$;
- (2) $\omega_1^X > \omega_1^{ck}$.

PREUVE. Supposons $X \geq_h \mathcal{O}$. Alors, $\omega_1^X \geq \omega_1^{\mathcal{O}}$ d'après le théorème 7.5. On a par ailleurs $\omega_1^{\mathcal{O}} > \omega_1^{ck}$: on peut définir la fonctionnelle Φ_e qui à l'aide de l'oracle $\mathcal{O}$ renvoie sur son entrée n la somme des n premiers éléments de $\mathcal{O}$ (via la fonction $+_o$ de l'exemple 27-5.4). L'élément $3 \times 5^e \in \mathcal{O}^{\mathcal{O}}$ codera pour un ordinal plus grand que tous les ordinaux calculables. On a donc $\omega_1^X \geq \omega_1^{\mathcal{O}} > \omega_1^{ck}$, et donc $\omega_1^X > \omega_1^{ck}$. Supposons à présent $\omega_1^X > \omega_1^{ck}$. Soit T un arbre X -calculable tel que $|T| = \omega_1^{ck}$. On procède alors comme dans la preuve du théorème 6.1 pour donner une description $\Sigma_1^1(X)$ de $\mathcal{O}$: étant donné f la fonction calculable telle que $f(e)$ est un code d'arbre c.e. T_e pour lequel T_e est bien fondé ssi $e \in \mathcal{O}$, on a $e \in \mathcal{O}$ ssi il existe un morphisme f de T_e vers T . Il s'agit bien d'une description $\Sigma_1^1(X)$ de $\mathcal{O}$. Ainsi, $\mathcal{O}$ est $\Delta_1^1(X)$, et donc $X \geq_h \mathcal{O}$. ■

Exercice 7.7. (★★) Montrer que pour tout ensemble X qui est Π_1^1 et non Δ_1^1 , on a $X \geq_h \mathcal{O}$. ◇

Chapitre 30

Classes Σ_1^1 et Π_1^1

Nous allons maintenant concentrer notre étude sur les classes Σ_1^1 et Π_1^1 . Suivant l'analogie entre hypercalculabilité et calculabilité classique que nous avons abordée dans la section 29-4, les classes Σ_1^1 jouent informellement le rôle des classes Π_1^0 (voir le chapitre 8). Nous étudierons en particulier un certain nombre de théorèmes de base Σ_1^1 , dont des analogues des théorèmes de base low et d'évitement de cône pour les classes Π_1^0 . Nous verrons que les classes Π_1^1 ont un comportement assez différent : si les singletons Σ_1^1 sont exactement les ensembles hyperarithmétiques, les singletons Π_1^1 peuvent être de complexité bien plus élevée, et contiennent la hiérarchie des hyper-sauts (voir la définition 6.2).

1. Représentation canonique des classes Σ_1^1

Les classes Σ_1^1 se représentent aisément par des arbres calculables de l'espace de Baire. En effet, soit $\mathcal{A} \subseteq 2^{\mathbb{N}}$ une classe Σ_1^1 . Alors, il existe un code e tel que $X \in \mathcal{A}$ ssi $\exists f \Phi_e(f, X) \uparrow$. On définit alors l'arbre

$$T = \{ \langle \sigma, \tau \rangle : |\sigma| = |\tau| = t \text{ et } \Phi_e(\sigma, \tau)[t] \uparrow, \text{ avec } \sigma \in \mathbb{N}^{<\mathbb{N}} \text{ et } \tau \in 2^{<\mathbb{N}} \}.$$

Les éléments de la classe $\mathcal{A}$ sont alors exactement les éléments X tels que $\langle f, X \rangle \in [T]$ pour un certain f . Cette représentation vient directement du théorème de forme normale de Kleene. Voyons-en tout de suite une première conséquence avec la proposition 1.2, à venir : étant donné une collection $\{\mathcal{A}_i\}_{i \in I}$ de classes Σ_1^1 non vides, on peut choisir uniformément un élément dans chaque $\mathcal{A}_i$. En particulier, l'axiome du choix est superflu

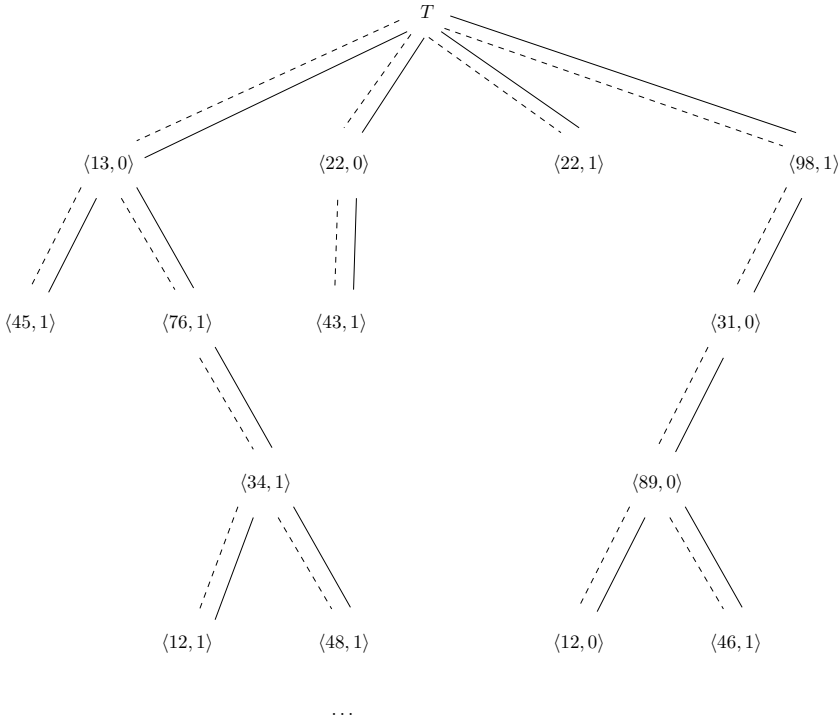


FIGURE 1.1 – Représentation de classes Σ_1^1 par un arbre : chaque élément X de la classe est considéré avec ses fonctions témoins potentiels. Ces fonctions témoins sont représentées par les chemins en pointillé, et les éléments de la classe par les chemins en trait plein.

pour les classes Σ_1^1 . Il convient toutefois de préciser ce que l'on entend par « collection de classes Σ_1^1 non vides ». D'après notre définition, une classe est Σ_1^1 est une classe $\Sigma_1^1(Z)$ pour un certain Z . On suppose ici que la présentation de chaque classe de notre suite est un objet concret, c'est-à-dire un couple (e, Z) où e est la formule $\Sigma_1^1(Z)$ correspondant à notre classe.

Proposition 1.2. On peut choisir un élément dans chaque classe $\Sigma_1^1(Z)$ uniformément en un code $\Sigma_1^1(Z)$ de la classe et en Z . En particulier, l'axiome du choix est superflu pour les classes Σ_1^1 . ★

PREUVE. Soit $\mathcal{A} \subseteq 2^{\mathbb{N}}$ une classe $\Sigma_1^1(Z)$. Soit $T \subseteq \mathbb{N}^{<\mathbb{N}}$ l'arbre Z -calculable canonique tel que $X \in \mathcal{A}$ ssi il existe f tel que $\langle f, X \rangle \in [T]$. Notons que le théorème de forme normale de Kleene est uniforme, et que l'arbre correspondant s'obtient donc lui aussi de manière uniforme.

Il suffit de choisir le plus petit chemin $\langle f, X \rangle$ de $[T]$, pour l'ordre lexicographique. L'élément choisi sera donc X — ainsi que son « témoin » f que l'on peut alors oublier. ■

Insistons une fois de plus sur l'uniformité dans la présentation de nos classes Σ_1^1 , et sans laquelle on ne peut pas faire grand-chose, comme en témoignent les classes d'équivalence de degrés Turing, qui sont toutes Σ_3^0 , mais pour lesquelles on ne peut démontrer l'existence d'une fonction de choix.

2. Théorèmes de base pour les classes Σ_1^1

Quelle est la puissance nécessaire pour calculer un élément d'une classe Σ_1^1 ? D'après la proposition 4.2 à venir, la classe des ensembles non hyperarithmétiques est Σ_1^1 . On a donc besoin de quelque chose qui aille au-delà de la puissance hyperarithmétique. Il n'est pas très difficile de voir que le $\mathcal{O}$ de Kleene suffit : étant donné un arbre $T \subseteq \mathbb{N}^{<\mathbb{N}}$, on cherche à l'aide de $\mathcal{O}$ le premier nœud $\sigma \in T$ de taille 1 tel que $T \upharpoonright_\sigma$ est mal fondé, et l'on recommence inductivement pour calculer petit à petit le chemin infini le plus à gauche de T . Le théorème suivant prouve quelque chose de plus fort : un analogue pour les classes Σ_1^1 du théorème 8-4.3 de la base low des classes Π_1^0 .

2.1. Théorème de la base hyperlow

Rappelons que le $\mathcal{O}$ de Kleene joue le rôle du problème de l'arrêt dans la correspondance entre hypercalculabilité et calculabilité classique. La définition suivante correspond donc à la notion d'ensemble low en hypercalculabilité.

Définition 2.1. Un ensemble X est *hyperlow* si $\mathcal{O}^X \leqslant_T \mathcal{O}$. ◇

Notons que si X est hyperlow, on ne peut pas avoir $\mathcal{O} \leqslant_h X$, car on aurait alors $\mathcal{O}^X \leqslant_h X$, ce qui est impossible par le théorème 29-7.3. En particulier, $\omega_1^X = \omega_1^{ck}$ pour tout X hyperlow.

Théorème 2.2 (Théorème de base de Gandy [72])

Toute classe Σ_1^1 non vide contient un élément X hyperlow. En particulier, avec $\omega_1^X = \omega_1^{ck}$.

PREUVE. Soit $\mathcal{A}$ une classe Σ_1^1 . À l'étape n de la construction, supposons que l'on ait un arbre calculable $T_n \subseteq \mathbb{N}^{<\mathbb{N}}$ tel que $[T_n]$ est non vide et une chaîne σ_n telle que tous les nœuds de T_n sont compatibles avec σ_n .

Supposons que les chemins infinis de $[T_n]$ encodent de manière calculable des éléments de $\mathcal{A}$, via une fonction de décodage Φ_n , telle que pour $f \in [T_n]$ et pour tout m on ait $\Phi_n(f \upharpoonright_m) = X \upharpoonright_m$, où X est l'élément de $\mathcal{A}$ encodé par f . On commence la construction avec T_0 comme étant l'arbre encodant les éléments de $\mathcal{A}$, Φ_0 la fonction de décodage de ces éléments et $\sigma_0 = \epsilon$.

On considère $\mathcal{B}_{n+1}$ la classe Σ_1^1 des $f \in [T_n]$ telles que $n \notin \mathcal{O}^{\Phi_n(f)}$. Il y a deux cas possibles.

Le cas 1. La classe $\mathcal{B}_{n+1}$ est vide, auquel cas on prend $T_{n+1} = T_n \upharpoonright_{(\sigma_n m)}$ pour m le plus petit tel que $[T_n \upharpoonright_{(\sigma_n m)}]$ est non vide et $\sigma_{n+1} = \sigma_n m$. On prend aussi $\Phi_{n+1} = \Phi_n$. Notons que dans ce cas tous les éléments $X \in \mathcal{A}$ encodés dans T_{n+1} sont tels que $n \in \mathcal{O}^X$.

Le cas 2. La classe $\mathcal{B}_{n+1}$ est non vide. Soit alors U l'arbre calculable dont les chemins infinis encodent les éléments de $\mathcal{B}_{n+1}$ et soit Ψ la fonction de décodage de $\mathcal{B}_{n+1}$, c'est-à-dire qui renvoie des éléments de $[T_n]$. Soit τ la plus petite chaîne de U dans l'ordre lexicographique telle que $\Psi(\tau) = \sigma_n$ et telle que $[U \upharpoonright_{(\tau m)}]$ est non vide pour un certain k . Soit m le plus petit de ces entiers k . On prend $T_{n+1} = U \upharpoonright_{\tau m}$ et $\sigma_{n+1} = \tau m$. La fonction de décodage Φ_{n+1} est donnée par $\Phi_{n+1}(\tau) = \Phi_n(\Psi(\tau))$. Notons que dans ce cas tous les éléments $X \in \mathcal{A}$ encodés dans T_{n+1} via Φ_{n+1} sont tels que $n \notin \mathcal{O}^X$.

Notons finalement que $\mathcal{O}$ peut mener à bien la construction de manière calculable, le prédicat « $[T]$ est non vide » pour un arbre calculable T étant Σ_1^1 . On obtient petit à petit un élément

$$X = \Phi_0(\sigma_0) \prec \Phi_1(\sigma_1) \prec \Phi_2(\sigma_2) \prec \dots$$

Il s'agit alors de montrer que ce point limite X appartient bien à $\mathcal{A}$. L'ensemble X est construit petit à petit avec son témoin g , de telle sorte que chaque préfixe de $f_0 = \langle g, X \rangle$ appartient à T_0 . Comme T_0 est un arbre, alors f_0 appartient à $[T_0]$, et donc X à $\mathcal{A}$. Comme vérifié durant la construction, à chaque étape n , si l'on est dans le cas 1 alors $n \in \mathcal{O}^X$, et si l'on est dans le cas 2 alors $n \notin \mathcal{O}^X$. Comme la construction est calculable en $\mathcal{O}$, ce dernier peut donc décider petit à petit pour chaque entier n si celui-ci appartient à $\mathcal{O}^X$ ou non. On conclut dès lors que $\mathcal{O}^X \leq_T \mathcal{O}$. ■

2.2. Espace topologique généré par les classes Σ_1^1

La démonstration du théorème de base de Gandy est très similaire à la démonstration du théorème 8-4.3 (que toute classe Π_1^0 non vide contient un ensemble low). Dans chacune des preuves, on construit une suite décroissante de classes $\mathcal{A}_0 \supseteq \mathcal{A}_1 \supseteq \dots$ telle que $\bigcap_n \mathcal{A}_n$ contienne notre élément, low dans le premier cas et hyperlow dans le second, et où $\mathcal{A}_n$ est Π_1^0 dans le premier cas et Σ_1^1 dans l'autre. Il y a malgré tout une complication dans le cas des classes Σ_1^1 : tandis qu'une intersection décroissante $\bigcap_n \mathcal{A}_n$ de

classes Π_1^0 , où chaque $\mathcal{A}_n$ est non vide, est nécessairement elle-même non vide, ce n'est plus du tout le cas pour les classes Π_2^0 et *a fortiori* Σ_1^1 . On a donc été forcé dans la preuve ci-dessus de « ruser » et de travailler avec l'arbre qui encode les éléments de nos classes Σ_1^1 , en construisant non pas uniquement un élément X commun à chacune de ces classes, mais aussi en construisant pour chacune de ces classes une fonction témoin associée à X . Le théorème suivant est une abstraction qui brode sur cette idée pour dire en termes savants la chose suivante : dans l'espace topologique généré par les classes Σ_1^1 , une intersection d'ouverts denses est dense. Il s'agit d'un résultat important et qui présente de nombreuses applications, en théorie descriptive des ensembles comme en hypercalculabilité. Nous en verrons un exemple avec le théorème 2.4.

Théorème 2.3 (Gandy)

Considérons la classe $\bigcap_n \bigcup_m \mathcal{A}_{n,m}$, où chaque $\mathcal{A}_{n,m}$ est une classe Σ_1^1 telle que pour tout n la classe $\bigcup_m \mathcal{A}_{n,m}$ a une intersection non vide avec toute classe Σ_1^1 non vide. Alors, $\bigcap_n \bigcup_m \mathcal{A}_{n,m}$ a une intersection non vide avec toute classe Σ_1^1 non vide.

PREUVE. On considère $\mathcal{A}$ une classe Σ_1^1 non vide quelconque, et l'on construit un élément dans $\mathcal{A} \cap \bigcap_n \bigcup_m \mathcal{A}_{n,m}$. On procède de manière similaire à la construction du théorème précédent. À l'étape 0, on définit T_0 comme étant l'arbre canonique qui encode les chemins de $\mathcal{A}$, et l'on définit ensuite $\tau_0^0 = \epsilon, \sigma^0 = \epsilon$ et $\eta_0 = \langle \tau_0^0, \sigma^0 \rangle$. À l'étape n de la construction, supposons que l'on ait des arbres calculables $T_0, \dots, T_n$, où $[T_i]$ est non vide pour $i \leq n$, où les nœuds de T_i pour $i \leq n$ sont de la forme $\langle \tau_i, \dots, \tau_0, \sigma \rangle$, avec $\langle \tau_{i+1}, \dots, \tau_0, \sigma \rangle \in T_{i+1}$ implique $\langle \tau_i, \dots, \tau_0, \sigma \rangle \in T_i$ pour $i < n$, et où les chemins infinis $\langle f, X \rangle$ de $[T_0]$ encodent les éléments $X \in \mathcal{A}$ et les chemins infinis $\langle f_{i+1}, \dots, f_0, X \rangle$ de $[T_{i+1}]$ encodent les chemins infinis $\langle f_i, \dots, f_0, X \rangle$ de $[T_i]$ pour $i < n$. On suppose de plus que l'on a un nœud $\eta_n = \langle \tau_n^n, \dots, \tau_0^n, \sigma^n \rangle \in T_n$ de taille n tel que les nœuds de T_n sont tous compatibles avec η_n . Notons que le fait que $[T_n]$ soit non vide, implique, en notant $\eta_i = \langle \tau_i^n, \dots, \tau_0^n, \sigma^n \rangle$ pour $i \leq n$, que la classe $[T_i] \upharpoonright_{\eta_i}$ est également non vide.

L'étape $n+1$ est alors relativement simple : sachant que $\bigcup_m \mathcal{A}_{n,m}$ intersecte toute classe Σ_1^1 non vide, on considère la classe Σ_1^1 des éléments X tels qu'il existe des fonctions $f_n, \dots, f_0$ pour lesquelles $\langle f_n, \dots, f_0, X \rangle \in T_n$. On choisit m tel que l'intersection entre $\mathcal{A}_{n,m}$ et cette classe soit non vide. Soit U l'arbre des nœuds $\langle \tau, \sigma \rangle$ qui encodent les éléments de $\mathcal{A}_{n,m}$, et soit donc T l'arbre de tous les nœuds $\langle \tau_{n+1}, \tau_n, \dots, \tau_0, \sigma \rangle$ pour $\langle \tau_n, \dots, \tau_0, \sigma \rangle \in T_n$ et $\langle \tau_{n+1}, \sigma \rangle \in U$. Notons que par hypothèse sur $\mathcal{A}_{n,m}$, la classe $[T]$ est non vide. On définit alors τ_{n+1} comme étant la première chaîne telle que

pour $\eta = \langle \tau_{n+1}, \tau_n^n, \dots, \tau_0^n, \sigma^n \rangle$ la classe $[T \upharpoonright_\eta]$ est non vide, et l'on définit ensuite η_{n+1} comme étant le premier fils de η dans T tel que $[T \upharpoonright_{\eta_{n+1}}]$ est non vide. On définit enfin $T_{n+1} = T \upharpoonright_{\eta_{n+1}}$.

De la suite $(\eta_n)_{n \in \mathbb{N}}$ on peut extraire pour tout i la suite d'éléments de T_i donnée par $\langle \tau_i^0, \dots, \tau_0^0, \sigma^0 \rangle \prec \langle \tau_i^1, \dots, \tau_0^1, \sigma^1 \rangle \prec \langle \tau_i^2, \dots, \tau_0^2, \sigma^2 \rangle \prec \dots$, qui converge vers $\langle f_i, \dots, f_0, X \rangle$. Comme chaque T_i est un arbre, on a alors

$$\langle f_i, \dots, f_0, X \rangle \in [T_i], \quad \text{pour tout } i.$$

En particulier, $X \in \mathcal{A}$ et $X \in \bigcup_m \mathcal{A}_{n,m}$ pour tout n . Donc, $\mathcal{A} \cap \bigcap_n \bigcup_m \mathcal{A}_{n,m}$ est non vide. Comme $\mathcal{A}$ est arbitraire, la classe $\bigcap_n \bigcup_m \mathcal{A}_{n,m}$ intersecte toute classe Σ_1^1 non vide. ■

2.3. Théorème de base d'évitement de cône hyperarithmétique

Notons que dans le théorème précédent ni l'intersection ni les réunions n'ont besoin d'être effectives. Voyons-en tout de suite une application, avec un autre théorème de base pour les classes Σ_1^1 , celui-ci analogue au théorème 8-4.7 de base d'évitement de cône pour les classes Π_1^0 .

Théorème 2.4 (Théorème de base de Kreisel)

Soit $\mathcal{A}$ une classe Σ_1^1 non vide et soit X non hyperarithmétique. Alors, il existe $Y \in \mathcal{A}$ tel que $Y \not\geq_h X$.

PREUVE. La preuve de ce théorème utilise deux résultats que nous démontrerons un peu plus tard : d'une part, la proposition 4.3 qui implique que la classe $\{Y : \omega_1^Y = \omega_1^{ck}\}$ est Σ_1^1 , et la proposition 3.1 d'autre part qui stipule que si une classe Σ_1^1 ne contient qu'un seul élément, alors cet élément est Δ_1^1 .

D'après le théorème 2.2, il existe un élément $Y \in \mathcal{A}$ tel que $\omega_1^Y = \omega_1^{ck}$. D'après la proposition 4.3 (de la section suivante), la classe $\{Y : \omega_1^Y = \omega_1^{ck}\}$ est Σ_1^1 . Quitte à intersecter $\mathcal{A}$ avec cette classe, on peut donc supposer sans perte de généralité $\omega_1^Y = \omega_1^{ck}$ pour tous les éléments $Y \in \mathcal{A}$. En particulier, si $Y \geq_h X$ pour $Y \in \mathcal{A}$, il existe alors $a \in \mathcal{O}$ tel que $H_a^Y \geq_T X$.

Montrons une première chose : il n'est pas possible d'avoir $\Phi_e(H_a^Y) = X$ pour $a \in \mathcal{O}$, $e \in \mathbb{N}$ et tous les éléments Y d'une classe Σ_1^1 non vide. En effet, si tel était le cas, alors X serait le singleton Σ_1^1 suivant :

$$\{X : \exists Y \in \mathcal{A} \Phi_e(H_a^Y) = X\}$$

(on utilise ici la classe $\mathcal{P}$ du théorème 28-3.4 pour récupérer H_a^Y à partir de Y de manière Σ_1^1). D'après la proposition 3.1, qui arrive sous peu, X serait donc Δ_1^1 , en contradiction avec nos hypothèses.

Soient $e \in \mathbb{N}$, $a \in \mathcal{O}$ et $\mathcal{A}_d$ la classe Σ_1^1 de code d . Pour tout m , soit $\mathcal{A}_{\langle e, a \rangle, \langle d, m \rangle}$ la classe Σ_1^1 des éléments $Y \in \mathcal{A}_d$ tels que $\Phi_e(H_a^Y) \neq X(m)$. D'après ce que l'on vient de montrer, si $\mathcal{A}_d$ est non vide, alors il existe m tel que $\mathcal{A}_{\langle e, a \rangle, \langle d, m \rangle}$ est non vide. En particulier, la réunion $\bigcup_{d, m} \mathcal{A}_{\langle e, a \rangle, \langle d, m \rangle}$ a une intersection non vide avec toute classe Σ_1^1 non vide. Par ailleurs, $\Phi_e(H_a^Y) \neq X$ pour tout élément $Y \in \bigcup_{d, m} \mathcal{A}_{\langle e, a \rangle, \langle d, m \rangle}$. D'après le théorème 2.3, il existe un élément Y dans $\mathcal{A} \cap \bigcap_{e \in \mathbb{N}, a \in \mathcal{O}} \bigcup_{d, m} \mathcal{A}_{\langle e, a \rangle, \langle d, m \rangle}$. On a donc $H_a^Y \not\geq_T X$ pour tout $a \in \mathcal{O}$, et donc $Y \not\geq_h X$. ■

Préservation de ω_1^{ck} et $\leq_h$

Notons que dans le théorème précédent nous avons commencé par restreindre notre classe aux éléments Y tels que $\omega_1^Y = \omega_1^{ck}$. La réduction hyperarithmétique cache un phénomène nouveau par rapport à la réduction Turing : ω_1^{ck} , contrairement à ω , est une notion relative. En particulier, il existe des oracles Y tels que $\omega_1^Y > \omega_1^{ck}$, et donc qui « augmentent » la valeur du plus petit ordinal non calculable.

Dans le théorème 2.4, pour obtenir $Y \not\geq_h X$, il ne suffit donc pas de s'assurer que $H_a^Y \not\geq_T X$ pour tout $a \in \mathcal{O}$. Il se peut par exemple qu'il existe un $a \in \mathcal{O}^Y \setminus \mathcal{O}$ tel que $H_a^Y \geq_T X$. Par exemple, si $\omega_1^Y > \omega_1^{ck}$, il existe $a \in \mathcal{O}^Y$ tel que $|a| = \omega_1^{ck}$, et H_a^Y permet de calculer $\mathcal{O}$, qui peut être vu en quelque sorte comme $\emptyset^{(\omega_1^{ck})}$. Forcer $\omega_1^Y = \omega_1^{ck}$ permet donc de se restreindre aux itérations le long de ω_1^{ck} et non de ω_1^Y , ce qui simplifie la preuve sachant que Y est en cours de construction.

3. L'hypothèse du continu pour les classes Σ_1^1

Nous avons abordé dans la section 9-4.6.2 la question de Cantor, consistant à savoir s'il existait un ensemble $\mathcal{A} \subseteq \mathbb{R}$ tels que $|\mathbb{N}| < |\mathcal{A}| < |2^{\mathbb{N}}|$. Bien que la question dans toute sa généralité soit indépendante de ZFC, nous avons prouvé à travers la proposition 8-2.14 que l'hypothèse du continu restreinte aux fermés de l'espace de Cantor était vraie, au sens où si un fermé $\mathcal{F}$ est indénombrable, alors il existe une injection de $2^{\mathbb{N}}$ dans $\mathcal{F}$, ce qui revient à dire que $\mathcal{F}$ est soit de cardinalité finie, soit dénombrable, soit de cardinalité $|2^{\mathbb{N}}| = |\mathbb{R}|$. Nous allons maintenant étendre la proposition 8-2.14 aux classes Σ_1^1 .

3.1. Les classes Σ_1^1 indénombrables

Nous avons vu que l'axiome du choix dénombrable était superflu pour les classes Σ_1^1 . Voyons à présent que c'est aussi le cas pour l'hypothèse du continu. Nous commençons par une proposition simple : les singletons Σ_1^1 sont Δ_1^1 .

Proposition 3.1. Soit A une classe Σ_1^1 ne contenant qu'un seul élément X . Alors, X est Δ_1^1 . ★

PREUVE. L'ensemble X a la description Σ_1^1 suivante : $n \in X$ ssi $\exists Y \in A$ tel que $n \in Y$. L'ensemble $\mathbb{N} \setminus X$ a la même description : $n \notin X$ ssi $\exists Y \in A$ tel que $n \notin Y$. L'ensemble X est donc Δ_1^1 . ■

Théorème 3.2

Soit $\mathcal{A}$ une classe Σ_1^1 indénombrable.

Alors, il existe une injection $f : 2^{\mathbb{N}} \rightarrow \mathcal{A}$ calculable en $\mathcal{O}$.

PREUVE. Étant donné un arbre calculable $T \subseteq \mathbb{N}^{<\mathbb{N}}$, la question de savoir si $[T]$ est non vide est Σ_1^1 , et $\mathcal{O}$ peut donc y répondre. On considère la classe $\mathcal{A}' = \mathcal{A} \cap \{X : X \text{ n'est pas hyperarithmétique}\}$. Nous verrons avec la proposition 4.2 que la classe $\mathcal{A}'$ reste Σ_1^1 . Comme la classe des ensembles hyperarithmétiques est dénombrable, la classe $\mathcal{A}'$ reste indénombrable.

Soit T l'arbre calculable tel que $[T]$ contient des éléments $\langle f, X \rangle$ pour tout $X \in \mathcal{A}'$. À l'aide de $\mathcal{O}$, on cherche

$$\langle \tau_0, \sigma_0 \rangle \in \mathbb{N}^{<\mathbb{N}} \times 2^{<\mathbb{N}} \quad \text{et} \quad \langle \tau_1, \sigma_1 \rangle \in \mathbb{N}^{<\mathbb{N}} \times 2^{<\mathbb{N}}$$

tels que σ_0 et σ_1 sont incomparables et tels que $[T \upharpoonright_{\langle \tau_0, \sigma_0 \rangle}]$ et $[T \upharpoonright_{\langle \tau_1, \sigma_1 \rangle}]$ sont non vides (c'est-à-dire tels que les arbres correspondants sont mal fondés). Puis, inductivement, supposons que $\langle \tau_\rho, \sigma_\rho \rangle$ soient définis et deux à deux incomparables pour toute chaîne ρ de taille n , et que $[T \upharpoonright_{\langle \tau_\rho, \sigma_\rho \rangle}]$ soit non vide pour chacune d'entre elles.

Supposons par l'absurde que l'un des $[T \upharpoonright_{\langle \tau_\rho, \sigma_\rho \rangle}]$ ne contienne qu'un seul élément. Alors, d'après la proposition 3.1, cet élément est hyperarithmétique, contredisant le fait que $\mathcal{A}'$ ne contienne aucun élément hyperarithmétique. On peut donc continuer : à l'aide de $\mathcal{O}$, on cherche $\langle \tau_{\rho 0}, \sigma_{\rho 0} \rangle \in \mathbb{N}^{<\mathbb{N}} \times 2^{<\mathbb{N}}$ et $\langle \tau_{\rho 1}, \sigma_{\rho 1} \rangle \in \mathbb{N}^{<\mathbb{N}} \times 2^{<\mathbb{N}}$ tels que $\sigma_{\rho i} \succeq \sigma_\rho$, $\tau_{\rho i} \succeq \tau_\rho$ pour $i \in \{0, 1\}$, tels que $\sigma_{\rho 0}, \sigma_{\rho 1}$ sont incomparables et tels que $[T \upharpoonright_{\langle \tau_{\rho 0}, \sigma_{\rho 0} \rangle}]$ et $[T \upharpoonright_{\langle \tau_{\rho 1}, \sigma_{\rho 1} \rangle}]$ sont non vides. La bijection est donnée par $f(X)$ comme étant l'unique élément de $\bigcap_{\rho \prec X} [\sigma_\rho]$. On a bien $X \neq Y$ implique $f(X) \neq f(Y)$. ■

Corollaire 3.3

L'hypothèse du continu est superflue pour les classes Σ_1^1 .

PREUVE. Toute classe Σ_1^1 est $\Sigma_1^1(X)$ pour un certain X . On relativise le théorème précédent à X . ■

Notons qu'il s'agit ici de l'hypothèse du continu telle que posée initialement par Cantor (par opposition à la version de l'hypothèse du continu faisant intervenir les cardinaux, telle que formulée dans la section 27-4.3) : toute classe Σ_1^1 est soit finie, soit dénombrable, soit contient une injection de $2^{\mathbb{N}}$ vers elle-même.

3.2. Les classes Σ_1^1 dénombrables

La technique utilisée dans la preuve du théorème 3.2 nous permet en fait de renforcer considérablement la proposition 3.1 :

Théorème 3.4

Une classe Σ_1^1 dénombrable ne contient que des éléments hyperarithmétiques.

PREUVE. Soit $\mathcal{A}$ une classe Σ_1^1 dénombrable. Soit

$$\mathcal{A}' = \mathcal{A} \cap \{X : X \text{ n'est pas hyperarithmétique}\}.$$

Supposons par l'absurde que $\mathcal{A}'$ est non vide. Alors, on recommence la construction du théorème précédent (le théorème 3.2). Si l'on peut mener cette construction jusqu'au bout, on a une injection de $2^{\mathbb{N}}$ vers $\mathcal{A}'$, ce qui contredit le fait que $\mathcal{A}'$ soit dénombrable. Sinon, la construction bloque au bout d'un moment : il existe une chaîne σ telle que $\mathcal{A}' \cap [\sigma]$ ne contient qu'un seul élément. D'après la proposition 3.1, cet élément est hyperarithmétique, ce qui contredit que $\mathcal{A}'$ ne contient aucun élément hyperarithmétique. Donc, $\mathcal{A}'$ est vide, et donc $\mathcal{A}$ ne contient que des hyperarithmétiques. ■

Notons que d'après le théorème 29-6.4, on peut donc obtenir, uniformément en un code de classe Σ_1^1 dénombrable, un code d'ordinal calculable α tel que $\emptyset^{(\alpha)}$ puisse en calculer tous les éléments. Nous finissons cette section en montrant que les classes Π_1^0 dénombrables de l'espace de Cantor sont déjà suffisantes pour renfermer des éléments hyperarithmétiques de complexité arbitraire.

Proposition 3.5. On considère $T \subseteq \mathbb{N}^{<\mathbb{N}}$ un arbre calculable. Alors, il existe $\mathcal{F} \subseteq 2^{\mathbb{N}}$ une classe Π_1^0 dont les éléments sont à encodage calculable près, les éléments de $[T]$, auxquels il faut ajouter des ensembles finis. ★

PREUVE. Il suffit d'encoder chaque élément de $f \in [T]$ comme étant l'ensemble $X_f = 0^{f(0)}10^{f(1)}10^{f(2)}1 \dots$. De la même manière, on peut associer à chaque $\tau \in T$ la chaîne finie $X_\tau = 0^{\tau(0)}10^{\tau(1)}1 \dots 10^{\tau(|\tau|-1)}1$. Soit $S \subseteq 2^{<\mathbb{N}}$ l'arbre calculable défini par $\sigma \in S$ ssi $\sigma \prec X_\tau$ pour $\tau \in T$. Alors, $[S]$ contient exactement les chemins X_f tels que $f \in [T]$, auxquels il faut rajouter les points limites de suites de tels chemins, qui sont nécessairement des ensembles finis.

Supposons en effet qu'un ensemble infini X appartienne à S . Alors, chaque préfixe de la forme $0^{n_0}10^{n_1}1 \dots 10^{n_k}1$ de X correspond à une chaîne $\tau \in T$ telle que $\tau(0) = n_0, \tau(1) = n_1, \dots, \tau(k) = n_k$. Ces préfixes étant tous comparables et de plus en plus longs, X correspond donc à une fonction $f \in [T]$. ■

Il est intéressant d'examiner les différents théorèmes de base des classes Π_1^0 à la lumière de la proposition précédente. Que ce soit le théorème de base low ou de base calculatoirement dominé —et d'autres encore—, on voit que l'on peut être forcé de choisir des ensembles finis pour les satisfaire, les autres ensembles de la classe pouvant être de n'importe quelle complexité imposable par la puissance définitionnelle Σ_1^1 —par exemple calculant tous $\emptyset^{(\omega)}$.

Corollaire 3.6

Des classes Π_1^0 dénombrables peuvent contenir des éléments hyperarithmétiques de complexité arbitraire.

4. Quelques classes Π_1^1 emblématiques

Nous commençons cette section par la présentation de quelques classes Π_1^1 particulières. Outre l'intérêt propre que l'on peut avoir à étudier ces classes naturelles, celles-ci sont également utiles en tant qu'outil d'analyse des classes Σ_1^1 , comme nous l'avons vu avec la section 2.3 à propos de la base d'évitement de cône pour les classes Σ_1^1 , et la section 3 à propos des classes Σ_1^1 dénombrables.

Définition 4.1. On dénote par HYP la classe des ensembles hyperarithmétiques. ◇

Nous allons maintenant voir que la classe HYP est Π_1^1 , autrement dit que la classe des ensembles non hyperarithmétiques est Σ_1^1 . La proposition 4.2 est utile pour retirer d'une classe Σ_1^1 tous ses membres hyperarithmétiques tout en la conservant Σ_1^1 .

Proposition 4.2. La classe HYP est Π_1^1 . ★

PREUVE

Soit $\mathcal{P} \subseteq \mathbb{N} \times 2^{\mathbb{N}}$ la classe Π_2^0 du théorème 28-2.1 telle que $\{Y : \mathcal{P}(a, Y)\}$ est le singleton H_a pour tout $a \in \mathcal{O}$. La classe des ensembles hyperarithmétiques est alors donnée par

$$\{X : \exists a \in \mathcal{O} \exists e \forall Y (\mathcal{P}(a, Y) \rightarrow \forall n \Phi_e(Y, n) \downarrow = X(n))\}.$$

La classe HYP est donc bien Π_1^1 . ■

Nous avons vu avec le théorème 29-6.4 que HYP n'est pas Σ_1^1 . La classe HYP reste toutefois une classe borélienne : comme il n'y a qu'une quantité dénombrable d'éléments hyperarithmétiques et que chacun d'entre eux est un singleton Π_1^0 , la classe des ensembles hyperarithmétiques est donc Σ_2^0 . Elle est en fait $\Sigma_2^0(\mathcal{O})$.

Nous étudions à présent la classe des ensembles permettant de calculer le plus petit ordinal non calculable, et que nous appellerons $\mathcal{S}$ pour cette section.

Proposition 4.3. La classe $\mathcal{S} = \{X : \omega_1^X > \omega_1^{ck}\}$ est Π_1^1 . ★

PREUVE. Soit R l'ensemble des codes d'ordres c. e. linéaires. Montrons que l'on a $\omega_1^X > \omega_1^{ck}$ si, et seulement si,

$$\exists a \in \mathcal{O}^X \quad \forall f \in \mathbb{N}^{\mathbb{N}} \quad \forall e \in R$$

« f n'est pas un isomorphisme de l'ordre codé par a vers celui codé par e . »

Si $\omega_1^X > \omega_1^{ck}$, soit $a \in \mathcal{O}^X$ tel que $|a| = \omega_1^{ck}$. Alors, pour tout e codant pour un ordre linéaire calculable, soit e correspond à un ordinal $\alpha < \omega_1^{ck}$, soit e correspond à un ordre mal fondé. Dans les deux cas, il n'existe pas d'isomorphisme entre l'ordre codé par a et celui codé par e .

À l'inverse, si $\omega_1^X = \omega_1^{ck}$, alors pour tout élément $a \in \mathcal{O}^X$, l'ordre codé par a est isomorphe à un bon ordre calculable. ■

Nous avons vu avec le théorème 2.2 que la classe $\mathcal{S}$ n'est pas Σ_1^1 , et même plus : elle ne contient aucune sous-classe Σ_1^1 non vide. Toutefois, là encore, cette classe reste borélienne. En voici une description $\Sigma_{\omega_1^{ck}+2}^0(\mathcal{O})$ par l'existence d'un code $3 \times 5^e \in \mathcal{O}_{=\omega_1^{ck}}^X$:

$$\mathcal{S} = \left\{ X : \exists e \left(\begin{array}{l} \forall n \quad \Phi_e(X, n) <_o \Phi_e(X, n+1) \\ \wedge \quad \forall n \quad \exists a \in \mathcal{O} \quad \Phi_e(X, n) \in \mathcal{O}_{<|a|}^X \\ \wedge \quad \forall a \in \mathcal{O} \quad \exists n \quad \Phi_e(X, n) \notin \mathcal{O}_{<|a|}^X \end{array} \right) \right\}.$$

D'après le théorème 2.2, la classe des $\{X : \Phi_e(X, n) \in \mathcal{O}_{<a}^X\}$ est $\Sigma_{|a|+1}^0$, uniformément en $a \in \mathcal{O}$. On obtient donc bien la complexité $\Sigma_{\omega_1^{ck}+2}^0(\mathcal{O})$. Il est possible de montrer en utilisant le *forcing de Steel* [216] que cette classe n'est pas $\Pi_{\omega_1^{ck}+2}^0$.

Exercice 4.4. (★★) (Sacks [189], Tanaka [221]). D'après l'exercice 29-3.9, toute classe Π_1^1 est mesurable. Montrer que la classe suivante est de mesure nulle

$$\mathcal{S} = \{X : \omega_1^X > \omega_1^{ck}\}.$$

◇

Exercice 4.5. (★★) (Keckris [110], Hjorth et Nies [92]). Dédurre de l'exercice précédent qu'il existe une classe Π_1^1 de mesure nulle qui contient toutes les classes Π_1^1 de mesure nulle. ◇

Nous voyons finalement notre troisième exemple de classe Π_1^1 , qui cette fois-ci ne sera pas Σ_1^1 . L'idée est de considérer la classe des X qui sont calculables par $\emptyset^{(\alpha)}$ pour $\alpha < \omega_1^X$. Le problème est que pour $\alpha \geq \omega_1^{ck}$ l'ensemble $\emptyset^{(\alpha)}$ n'est pas bien défini. On pourrait bien sûr itérer la définition de $\emptyset^{(\alpha)}$ pour $\alpha \geq \omega_1^{ck}$, mais sans aide extérieure pour calculer α , la puissance de calcul de l'ensemble $\emptyset^{(\alpha)}$ n'est pas claire. Il est par ailleurs exclu d'utiliser l'oracle X lui-même comme aide extérieure, car il est malaisé de spécifier formellement que X ne peut être utilisé que dans le calcul de α et non dans le calcul de lui-même. On introduit alors la notation suivante.

Notation

Soit un ordinal $\alpha < \omega_1$. On écrit $X \leq_T \emptyset^{(\alpha)}$ pour signifier

$$\forall Y (\alpha < \omega_1^Y \rightarrow X \leq_T Y^{(\alpha)}).$$

On définit ci-après notre dernière classe Π_1^1 .

Définition 4.6. On définit la classe

$$\mathcal{C} = \{X : \exists \alpha < \omega_1^X X \leq_T \emptyset^{(\alpha)}\}.$$



La classe $\mathcal{C}$ contient en particulier tous les ensembles hyperarithmétiques, ainsi que par exemple le $\mathcal{O}$ de Kleene (voir proposition 6.1 et théorème 5.1) et bien d'autres éléments : tous les ensembles X que l'on peut calculer via une itération α du saut Turing, pour α calculable en X . La classe $\mathcal{C}$ semble moins naturelle que HYP ou $\mathcal{S}$. Nous allons néanmoins voir qu'elle possède des propriétés remarquables, que nous étudierons dans la section suivante. Commençons par montrer que cette classe est Π_1^1 .

Proposition 4.7. La classe $\mathcal{C}$ est Π_1^1 .



PREUVE. La classe $\mathcal{C}$ peut s'exprimer de la manière suivante :

$$\mathcal{C} = \{X : \exists b \in \mathcal{O}^X \forall Y (\omega_1^Y \leq |b| \text{ ou } \exists a \in \mathcal{O}_{=|b|}^Y X \leq_T H_a^Y)\}.$$

Le prédicat $b \in \mathcal{O}^X$ est $\Pi_1^1(X)$ uniformément en X . En reprenant les idées de la proposition 4.3, on montre que pour $b \in \mathcal{O}^X$ le prédicat $\omega_1^Y \leq |b|$ est $\Pi_1^1(X \oplus Y)$ uniformément en X et Y . En effet, en considérant l'ensemble R^Y des codes Y -c.e. d'ordres linéaires, on a $\omega_1^Y \leq |b|$ ssi

$$\forall e \in R^Y \forall f \in \mathbb{N}^{\mathbb{N}}$$

« f n'est pas un isomorphisme de l'ordre codé par b vers celui codé par e . »

Le prédicat $\exists a \in \mathcal{O}^Y X \leq_T H_a^Y$ est quant à lui $\Pi_1^1(X \oplus Y)$ uniformément en X et Y . La classe est donc Π_1^1 . ■

5. Étude d'une classe Π_1^1 très spéciale

La classe $\mathcal{C}$ définie précédemment n'est pas une classe Σ_1^1 . Nous le verrons en particulier avec le théorème 5.6. Nous allons voir que $\mathcal{C}$ possède de nombreuses propriétés remarquables, ce qui justifie bien qu'on lui consacre une section.

Théorème 5.1 (Guaspari [80])

La classe $\mathcal{C}$ est une base pour les classes Π_1^1 : toute classe Π_1^1 non vide contient un élément de $\mathcal{C}$.

PREUVE

Soit $\mathcal{A}$ une classe Π_1^1 non vide. D'après la démonstration du théorème 29-3.2 il existe un code e qui est pour tout X le code d'un arbre X -c. e. bien fondé si, et seulement si, $X \in \mathcal{A}$. En utilisant l'ordre de Kleene-Brouwer sur cet arbre, on obtient un code e qui est pour tout X le code d'un ordre linéaire sur un ensemble d'entiers, et tel que $X \in \mathcal{A}$ ssi e code pour un ordre bien fondé. On note $<_e^X$ l'ordre en question, dom_e^X le domaine de $<_e^X$ et $<_e^\sigma$ l'ordre $<_e$ énuméré avec le « morceau d'oracle » σ . Si $<_e^X$ est bien fondé, on note $|e|^X$ l'ordinal $|\langle e^X \rangle|$ et, pour $a \in \text{dom}_e^X$, on note $|a|^X$ l'ordinal correspondant à $<_e^X$ restreint aux éléments strictement plus petits que a .

On commence par construire un arbre T dans l'espace $(\mathbb{N} \times \omega_1 \times 2^{<\mathbb{N}})^{<\mathbb{N}}$ dont les chemins seront des encodages d'éléments de $\mathcal{A}$. Le plus petit chemin de T pour l'ordre lexicographique sera un encodage d'un élément $X \in \mathcal{A} \cap \mathcal{C}$. La technique explicitée dans cette preuve sera réutilisée plusieurs fois dans cette section. On notera $\langle \tau, \rho, (\sigma_1, \dots, \sigma_n) \rangle$ pour $\tau \in \mathbb{N}^{<\mathbb{N}}$ et $\rho \in \omega_1^{<\mathbb{N}}$, avec $|\tau| = |\rho| = n$, les éléments de cet espace pour signifier formellement la concaténation des éléments $(\tau(0), \rho(0), \sigma_1), \dots, (\tau(n-1), \rho(n-1), \sigma_n)$. Étant donné un tel élément $\langle \tau, \rho, (\sigma_1, \dots, \sigma_n) \rangle$, on notera $f_{\tau, \rho} : \mathbb{N} \rightarrow \omega_1$ la fonction partielle, de domaine fini, telle que $f_{\tau, \rho}(\tau(i)) = \rho(i)$ pour $i < n$.

Pour commencer, on met dans T des nœuds $\langle a, \alpha, \sigma \rangle$ de taille 1 pour tout ordinal $\alpha < \omega_1$ et tous a, σ tels que a est le premier élément énuméré dans l'ordre $<_e^\sigma$. Étant donné un nœud $\langle \tau, \rho, (\sigma_1, \dots, \sigma_n) \rangle$ de T , avec

$$\sigma_1 \prec \dots \prec \sigma_n, \quad |\tau| = |\rho| = n,$$

on met pour tout $\alpha < \omega_1$ le nœud $\langle \tau a, \rho \alpha, (\sigma_1, \dots, \sigma_n, \sigma_{n+1}) \rangle$ dans T pour $\sigma_{n+1} \succ \sigma_n$ si a est le $(n+1)$ -ième élément énuméré dans $<_e^{\sigma_{n+1}}$, répétitions non comprises, tel que $f_{\tau a, \rho \alpha}$ est une fonction qui préserve l'ordre.

On notera $\langle f, X \rangle$ la limite $\langle \tau_1, \rho_1, (\sigma_1) \rangle \prec \langle \tau_2, \rho_2, (\sigma_1, \sigma_2) \rangle \prec \dots$ de $[T]$, où $\sigma_1 \prec \sigma_2 \prec \dots$ et $f_{\tau_1, \rho_1} \prec f_{\tau_2, \rho_2} \prec \dots f$. Montrons que $\langle f, X \rangle \in [T]$ pour une certaine fonction f si, et seulement si, $X \in \mathcal{A}$.

Soit $\langle f, X \rangle \in [T]$. Alors, f est une fonction de dom_e^X vers ω_1 qui préserve l'ordre. En particulier, pour $a, b \in \text{dom}_e^X$, on a $a <_e^X b$ ssi $f(a) < f(b)$. Il s'ensuit que $<_e^X$ doit être bien fondé. Donc, $X \in \mathcal{A}$. Réciproquement, si $X \in \mathcal{A}$, alors pour la fonction $f : \text{dom}_e^X \rightarrow \omega_1$ telle que $f(a) = |a|^X$ on doit avoir $\langle f, X \rangle \in [T]$. Comme $\mathcal{A}$ est non vide, alors $[T]$ aussi est non vide.

Soit $\langle f, X \rangle$ le plus petit chemin infini de T pour l'ordre lexicographique. Notons que l'on a forcément $f(a) = |a|^X$ pour un tel chemin infini — car il s'agit du plus petit chemin pour l'ordre lexicographique. En particulier, $|<_e^X| < \omega_1^X$. Notons que bien que la construction de T soit intuitivement effective, il n'est pas possible de parler de calculabilité de T , car on manipule des ordinaux dénombrables arbitraires.

L'objectif est à présent de trouver $\beta < \omega_1^X$ tel que pour tout ensemble Y vérifiant $\beta < \omega_1^Y$, l'ensemble X est calculable en $Y^{(\beta)}$. Soit $\alpha = |<_e^X|$, et soit Y tel que $\alpha < \omega_1^Y$. On recommence alors la construction d'un arbre T_Y dont le but est de recopier T , mais avec des codes d'ordinaux de $\mathcal{O}_{<\alpha}^Y$ à la place des ordinaux eux-mêmes, et en ne gardant donc que la partie de T dont les nœuds mentionnent des ordinaux inférieurs à α . Comme plusieurs éléments de $\mathcal{O}_{<\alpha}^Y$ peuvent coder pour le même ordinal, on travaillera avec l'ensemble $\mathcal{O}_{1,<\alpha}^Y$ qui ne conserve pour chaque ordinal $\beta < \alpha$ que le plus petit code de β (plus petit pour l'ordre sur les entiers). Formellement, $\mathcal{O}_{1,<\alpha}^Y = \{b \in \mathcal{O}_{<\alpha}^Y : \forall c < b \ c \notin \mathcal{O}_{=|b|}^Y\}$. En particulier, $\mathcal{O}_{1,<\alpha}^Y$ est un ensemble $\Delta_1^1(Y)$. L'arbre T_Y est alors un sous-ensemble $\mathcal{O}_{1,<\alpha}^Y$ -calculable de $(\mathbb{N} \times \mathcal{O}_{1,<\alpha}^Y \times 2^{<\mathbb{N}})^{<\mathbb{N}}$. Il est clair que le chemin $\langle f, X \rangle$ le plus à gauche de $[T_Y]$ est le même — à encodage près des ordinaux de la fonction f — que le chemin le plus à gauche de T . Il est clair également que pour Y_1, Y_2 tels que $\alpha \leq \omega_1^{Y_1}$ et $\alpha \leq \omega_1^{Y_2}$, les arbres T_{Y_1} et T_{Y_2} sont les mêmes, à encodage près des ordinaux.

Dans chaque arbre T_Y (pour Y tel que $\alpha \leq \omega_1^Y$), le chemin le plus à gauche est donc le même, et il est calculable en $\mathcal{O}^Y$ en réutilisant la technique du théorème 2.2. Ce n'est pas tout à fait suffisant : ce chemin doit être hyperarithmétique en Y . Pour le montrer, nous allons passer momentanément par l'arbre T_X pour X tel que $\langle f, X \rangle$ est le chemin le plus à gauche de $[T]$. Notons d'abord que la fonction f est hyperarithmétique en X en tant qu'unique fonction telle que $\langle f, X \rangle \in [T_X]$ et telle que $|a|^X = |f(a)|$ pour tout $a \in \text{dom}_e^X$. Donc, $\langle f, X \rangle$ est hyperarithmétique en X . Par ailleurs, chaque nœud situé à gauche de $\langle f, X \rangle$ est le point de départ d'un arbre bien fondé hyperarithmétique en X , et donc codant pour un ordinal inférieur à ω_1^X . À présent, il est possible de calculer à partir de $\langle f, X \rangle$ l'ensemble des codes hyperarithmétiques en X de tous les arbres bien fondés se situant à gauche de $\langle f, X \rangle$. Comme $\langle f, X \rangle$ est hyperarithmétique en X , d'après la

majoration Σ_1^1 de Spector (combinée aux théorèmes 28-1.10 et 29-6.1), le supremum β des ordinaux codés par ces arbres est inférieur à ω_1^X .

À présent, pour un ensemble Y arbitraire tel que $\max(\alpha, \beta) < \omega_1^Y$, l'ordinal β est aussi le supremum des ordinaux codés par des arbres bien fondés se trouvant à gauche de la branche $\langle f, X \rangle \in [T_Y]$. On peut alors calculer en T_Y et $\mathcal{O}_{<\beta}^{T_Y}$ le chemin $\langle f, X \rangle$: étant donné un préfixe $\langle \sigma, \rho \rangle$ de $\langle f, X \rangle$, il suffit de chercher la plus petite extension sous laquelle l'arbre est mal fondé, c'est-à-dire sous laquelle l'arbre ne code pas pour un ordinal de $\mathcal{O}_{<\beta}^{T_Y}$.

Avec une version relativisée du théorème 28-1.10, on a $\mathcal{O}_{<\beta}^{T_Y} \leq_T T_Y^{(\beta+1)}$. Or, $T_Y \leq_T \mathcal{O}_{1, <\alpha}^Y \leq_T Y^{(\alpha+1)}$. Donc, $\mathcal{O}_{<\beta}^{T_Y} \leq_T Y^{(\beta+1+\alpha+1)}$. En particulier, $X \leq_T Y^{(\beta+1+\alpha+1)}$ pour tout Y tel que $\beta + 1 + \alpha + 1 < \omega_1^Y$. Or, $\beta + 1 + \alpha + 1 < \omega_1^X$. Donc, $X \in \mathcal{C}$. ■

Théorème 5.2

Pour toute classe Π_1^1 non vide $\mathcal{A}$, on peut trouver uniformément en un code de $\mathcal{A}$ le code d'un singleton Π_1^1 inclus dans $\mathcal{A}$.

PREUVE. Il suffit essentiellement de s'apercevoir que dans la preuve précédente, le calcul de la branche infinie $\langle f, X \rangle$ est uniforme en $Y^{(\beta+1+\alpha+1)}$. En reprenant les notations de cette preuve-là, soit $T \subseteq (\mathbb{N} \times \omega_1 \times 2^{<\mathbb{N}})^{<\mathbb{N}}$ l'arbre correspondant à la classe $\mathcal{A}$. Soit $\langle f, X \rangle$ le chemin infini le plus à gauche de T et soit $\alpha = |<_e^X|$ où e est le code utilisé pour construire T . Pour Y tel que $\omega_1^Y > \alpha$, soit T_Y défini comme dans la même preuve.

Pour des questions d'uniformité, on a besoin ici de T_Y^γ , qui est l'arbre T_Y qui recopie T_Y en ne gardant que la partie de T_Y dont les nœuds mentionnent des ordinaux inférieurs à γ . Chaque chemin infini à droite de $\langle f, X \rangle$ dans T est de la forme $\langle g, Z \rangle$ pour $g \geq f$. En particulier, pour $\gamma < |\alpha|$, l'arbre T_Y^γ doit être vide et, pour $\gamma \geq |\alpha|$, les arbres T_Y^γ ont tous $\langle f, X \rangle$ comme chemin infini le plus à gauche, et sont tous les mêmes à gauche de $\langle f, X \rangle$ (à codage près des ordinaux).

On a montré dans la preuve précédente que pour un certain $\beta < \omega_1^X$, les arbres — nécessairement bien fondés — se trouvant à gauche de la branche $\langle f, X \rangle \in [T_X^\alpha]$ sont de hauteur au plus β . L'arbre T_Y^α est calculable en $\mathcal{O}_{1, <\alpha}^Y$, lui-même calculable en $Y^{(\alpha+1)}$. Comme $\mathcal{O}_{<\beta}^Z \leq_T Z^{(\beta+1)}$ pour un oracle Z quelconque, en prenant $Z = \mathcal{O}_{1, <\alpha}^Y$, on obtient que les arbres se trouvant à gauche de $\langle f, X \rangle$ dans T_Y^α ont tous un code dans $\mathcal{O}_{<\beta}^{\mathcal{O}_{1, <\alpha}^Y}$, lui-même calculable en $Y^{(\alpha+1+\beta+1)}$. Comme $\mathcal{O}_{<\beta}^{\mathcal{O}_{1, <\alpha}^Y} \subseteq \mathcal{O}_{<\gamma}^{\mathcal{O}_{1, <\alpha}^Y}$ pour $\beta < \gamma$, alors également tout oracle de la forme $Y^{(\alpha+1+\gamma+1)}$ pour $\beta < \gamma$ peut reconnaître uniformément que les arbres à gauche de $\langle f, X \rangle$ sont bien fondés.

On peut alors définir la fonctionnelle Φ_e qui pour n'importe quel Y tel que $\omega_1^Y \geq \omega_1^X$, pour n'importe quel γ_1 avec $\alpha \leq \gamma_1 < \omega_1^Y$ et pour n'importe quel γ_2 avec $\beta \leq \gamma_2 < \omega_1^Y$ sera telle que $\Phi_e(T_Y^{\gamma_1}, Y^{(\gamma_1+1+\gamma_2+1)})$ calcule le chemin $\langle f, X \rangle$, uniformément en γ_1 et γ_2 , et pour $\gamma_1 < \alpha$ ou $\gamma_2 < \beta$, le calcul $\Phi_e(T_Y^{\gamma_1}, Y^{(\gamma_1+1+\gamma_2+1)})$ ne produit pas un objet infini (soit l'arbre $T_Y^{\gamma_1}$ n'a pas de chemin infini, soit $Y^{(\gamma_1+1+\gamma_2+1)}$ ne connaît pas assez de codes d'arbres bien fondés et fait partir le calcul à gauche « trop tôt »).

Le singleton $\Pi_1^1 \mathcal{A}_1 \subseteq \mathcal{A}$ est alors donné par la formule suivante :

$$\left\{ X \in \mathcal{A} : \forall Y \left(\begin{array}{l} \omega_1^Y < \omega_1^X \text{ ou } \exists \gamma_1, \gamma_2 < \omega_1^Y \text{ t.q. } \Phi_e(T_Y^{\gamma_1}, Y^{(\gamma_1+1+\gamma_2+1)}) \\ \text{calcule un chemin infini } \langle f, Z \rangle \text{ avec } Z = X \end{array} \right) \right\}. \blacksquare$$

On a pour corollaire une généralisation du théorème 29-4.1, due tout à la fois à Kondô [123] et Addison [4].

Corollaire 5.3 (Uniformisation des classes Π_1^1)

Pour toute classe $\Pi_1^1 \mathcal{A} \subseteq 2^{\mathbb{N}} \times 2^{\mathbb{N}}$, il existe $f : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$, une fonction partielle Π_1^1 , telle que pour tout X , si $\{Y : (X, Y) \in \mathcal{A}\}$ est non vide, alors $(X, f(X)) \in \mathcal{A}$.

PREUVE. On définit $f(X) = Y$ si Y appartient au $\Pi_1^1(X)$ singleton inclus dans la classe $\{Y : (X, Y) \in \mathcal{A}\}$. $\blacksquare$

Corollaire 5.4

L'axiome du choix est superflu pour les classes Π_1^1 .

Conformément au corollaire 5.3, l'axiome du choix est vérifié tant qu'il y a une uniformité dans la présentation de nos classes Π_1^1 .

Corollaire 5.5

Soit X un singleton Π_1^1 . Alors, $\exists \alpha < \omega_1^X$ tel que $X \leq_T \emptyset^{(\alpha)}$.

Nous reviendrons sur les singletons Π_1^1 dans la section suivante, et nous nous consacrons pour le moment à terminer notre étude de $\mathcal{C}$, en montrant qu'elle ne vérifie pas forcément l'hypothèse du continu, et qu'elle ne contient en particulier aucun fermé parfait. Il s'agit en fait de la plus grosse classe Π_1^1 ne contenant pas de fermé parfait.

Théorème 5.6 (Mansfield [147] Solovay [211])

Soit $\mathcal{A}$ une classe Π_1^1 . Alors, $\mathcal{A}$ contient un fermé parfait si, et seulement si, elle contient un élément $X \notin \mathcal{C}$.

PREUVE. Supposons que $\mathcal{A}$ contient un fermé parfait T . D'après la proposition 4.2 la classe $\mathcal{U}$ des éléments de $[T]$ qui ne sont pas $\Delta_1^1(T)$ est une classe $\Sigma_1^1(T)$ indénombrable. D'après la relativisation du théorème 2.2, la classe $\mathcal{U}$ contient un élément X tel que $\omega_1^{X \oplus T} = \omega_1^T$. On a donc $\omega_1^X \leq \omega_1^T$. En revanche, X n'est pas hyperarithmétique en T . En particulier, $X \notin \mathcal{C}$.

Si $\mathcal{A}$ contient à présent un élément $X \notin \mathcal{C}$, soit $T \subseteq (\mathbb{N} \times \omega_1 \times 2^{<\mathbb{N}})^{<\mathbb{N}}$ l'arbre défini comme dans la preuve du théorème 5.1, c'est-à-dire — en reprenant les notations de la preuve — tel que $X \in \mathcal{A}$ ssi $\langle f, X \rangle \in [T]$ pour une certaine fonction $f : \text{dom}_e^X \rightarrow \omega_1$ (où dom_e^X est défini là encore comme dans la même preuve). Pour un ordinal γ et un ensemble Y avec $\omega_1^Y > \gamma$, on note T_Y^γ l'arbre T restreint aux nœuds ne mentionnant que des ordinaux plus petits que γ et utilisant des codes de $\mathcal{O}_{1, < \gamma}^Y$ pour représenter les ordinaux. Notons qu'à codage près des ordinaux les arbres T_Y^γ sont les mêmes pour tout Y vérifiant $\gamma < \omega_1^Y$. Soit α le plus petit ordinal tel que pour tout Y vérifiant $\alpha < \omega_1^Y$, l'arbre T_Y^α contient un chemin infini $\langle f, X \rangle$ pour un certain $X \in \mathcal{A} \setminus \mathcal{C}$. Notons que l'on a dans ce cas nécessairement $\alpha < \omega_1^X$. Soit $U \subseteq \mathbb{N}^{<\mathbb{N}}$ un arbre calculable tel que $X \notin \mathcal{C}$ ssi $\exists f \langle f, X \rangle \in [U]$. Considérons à présent l'arbre $S_Y \subseteq (\mathbb{N}^{<\mathbb{N}} \times \mathbb{N} \times \alpha \times 2^{<\mathbb{N}})^{<\mathbb{N}}$ tel que

$$\langle \langle \tau_1, \dots, \tau_n \rangle, \tau, \rho, (\sigma_1, \dots, \sigma_n) \rangle \in S$$

ssi $\langle \tau_i, \sigma_i \rangle \in U$ pour tout $i \leq n$ et $\langle \tau, \rho, (\sigma_1, \dots, \sigma_n) \rangle \in T_Y^\alpha$. Notons qu'à codage près des ordinaux les arbres S_Y sont les mêmes pour tout Y tel que $\alpha < \omega_1^Y$. Par hypothèse sur α , l'arbre S_Y est non vide. On prétend que pour tout nœud

$$\eta = \langle \langle \tau_1, \dots, \tau_n \rangle, \tau, \rho, (\sigma_1, \dots, \sigma_n) \rangle \in S_Y$$

tel que $[S_Y \upharpoonright_\eta]$ est non vide, il existe deux nœuds

$$\begin{aligned} \eta_1 &= \langle \langle \tau_1, \dots, \tau_n, \dots, \tau_{m_1}^1 \rangle, \tau_1, \rho_1, (\sigma_1, \dots, \sigma_n, \dots, \sigma_{m_1}^1) \rangle \succ \eta \\ \text{et } \eta_2 &= \langle \langle \tau_1, \dots, \tau_n, \dots, \tau_{m_2}^2 \rangle, \tau_2, \rho_2, (\sigma_1, \dots, \sigma_n, \dots, \sigma_{m_2}^2) \rangle \succ \eta \end{aligned}$$

tels que $\sigma_{m_1}^1$ et $\sigma_{m_2}^2$ sont incompatibles et tels que $[S_Y \upharpoonright_{\eta_1}]$ et $[S_Y \upharpoonright_{\eta_2}]$ sont tous les deux non vides. Si tel est bien le cas, on construit une injection de $2^{\mathbb{N}}$ vers les éléments de $\mathcal{A} \setminus \mathcal{C}$, et la proposition est vérifiée. Si ce n'est pas le cas, alors il existe un nœud $\eta \in S_Y$ tel que $[S_Y \upharpoonright_\eta]$ contient au moins un chemin infini et tel que tous les chemins infinis de $[S_Y \upharpoonright_\eta]$ sont de la forme $\langle f', f, X \rangle$ pour un même ensemble X . Notons que pour Y tel que $\omega_1^Y > \alpha$ l'arbre S_Y est uniformément $\Delta_1^1(Y)$. Pour n suffisamment grand, on a donc $n \in X$ ssi $\exists f', f, Z \langle f', f, Z \rangle \in [S_Y \upharpoonright_\eta] \wedge n \in Z$ et $n \notin X$ ssi $\exists f', f, Z \langle f', f, Z \rangle \in [S_Y \upharpoonright_\eta] \wedge n \notin Z$. Ainsi, X et $\mathbb{N} \setminus X$ sont $\Sigma_1^1(S_Y)$ et donc X est $\Delta_1^1(S_Y)$, et donc $\Delta_1^1(Y)$. La procédure est par ailleurs uniforme en Y . Donc, pour tout Y tel que $\omega_1^Y > \alpha$ il existe β avec $\alpha \leq \beta < \omega_1^Y$ tel que $Y^{(\beta)} \geq_T X$. D'après le lemme 29-6.3 relativisé à X , on a par

conséquent un ordinal β avec $\alpha \leq \beta < \omega_1^X$ tel que $Y^{(\beta)} \geq_T X$ pour tout Y vérifiant $\omega_1^Y > \beta \geq \alpha$. Donc, $X \in \mathcal{C}$, ce qui est une contradiction. ■

Corollaire 5.7

La classe $\mathcal{C}$ est la plus grosse classe Π_1^1 ne contenant pas de fermé parfait.

On ne peut donc pas construire d'injection continue de $2^{\mathbb{N}}$ dans $\mathcal{C}$. Mais la classe $\mathcal{C}$ est-elle elle-même dénombrable ou indénombrable ? Il n'est en fait pas possible de répondre à cette question. On peut montrer que dans le modèle de ZFC des constructibles de Gödel, tout ensemble $X \in 2^{\mathbb{N}}$ est Turing calculable par un élément de $\mathcal{C}$. Dans ce modèle, $\mathcal{C}$ est indénombrable et satisfait l'hypothèse du continu (bien que l'on ne puisse pas en rendre compte via une injection continue de $2^{\mathbb{N}}$ vers $\mathcal{C}$). Via la technique du forcing de Cohen, on peut construire des modèles de ZFC pour lesquels $\mathcal{C}$ peut rester indénombrable et ne pas satisfaire l'hypothèse du continu, ou même devenir dénombrable. On trouvera plus de détails dans la section 4.3 de [35].

6. Les singletons Π_1^1

Par le théorème 5.2, toute classe $\Pi_1^1 \neq \emptyset$ contient un singleton Π_1^1 . On portera donc une attention particulière aux singletons Π_1^1 . En particulier, tout théorème de base pour les Π_1^1 peut se prouver en se restreignant aux classes singletons. Contrairement aux singletons Σ_1^1 , les singletons Π_1^1 ne sont pas forcément hyperarithmétiques. On a, à titre d'exemple, ce qui suit.

Proposition 6.1. Le $\mathcal{O}$ de Kleene est un singleton Π_1^1 . ★

PREUVE. Il est clair que la classe $\mathcal{A}$ suivante est Π_1^1 et contient exactement le $\mathcal{O}$ de Kleene :

$$\left\{ X : \begin{array}{l} \forall e \in X \ e \in \mathcal{O} \wedge 1 \in X \\ \forall a \in \mathbb{N} \ a \in X \rightarrow 2^a \in X \\ \forall e \in \mathbb{N} \left(\forall n \ (\Phi_e(n) \downarrow \in X \wedge \Phi_e(n) <_o \Phi_e(n+1)) \right) \end{array} \right\} \wedge \wedge$$

Notons que la relation $<_o$ utilisée pour la définition de $\mathcal{A}$ est la relation « étendue » à tous les éléments de la forme 2^b ou $3 \cdot 5^b$, comme dans la preuve du théorème 29-3.1. ■

On peut en fait montrer que pour tout A singleton Π_1^1 l'ensemble $\mathcal{O}^A$ est lui aussi un singleton Π_1^1 . Nous le verrons formellement dans la prochaine proposition. Ainsi, $\mathcal{O}, \mathcal{O}^{\mathcal{O}}, \mathcal{O}^{\mathcal{O}^{\mathcal{O}}}$, etc., sont tous des singletons Π_1^1 . Cette considération nous amène à l'itération transfinie de l'hypersaut. Contrairement à la hiérarchie transfinie du saut, on peut cette fois-ci continuer au-delà de ω_1^{ck} , puisque notre oracle alors décode ω_1^{ck} .

Définition 6.2. On définit inductivement la hiérarchie des hypersauts.

- (1) $J_1 = \emptyset$ et $J_2 = \mathcal{O}$. On définit $1 <_{ho} 2$.
- (2) Si a est dans le domaine de $<_{ho}$, alors $J_{2^a} = \mathcal{O}^{J_a}$. On définit $a <_{ho} 2^a$ et $b <_{ho} 2^a$ pour tout $b <_{ho} a$.
- (3) Si a est dans le domaine de $<_{ho}$ et e est le code d'une fonctionnelle telle que $\Phi_e(J_a, 0) \downarrow = a$ et telle que $\Phi_e(J_a, n) \downarrow <_{ho} \Phi_e(J_a, n+1) \downarrow$ pour tout n , alors $J_{3^a \times 5^e} = \bigoplus_{n \in \mathbb{N}} J_{\Phi_e(J_a, n)}$. On définit $b <_{ho} 3^a \times 5^e$ pour tout b tel que $b <_{ho} \Phi_e(J_a, n)$ pour un certain n . $\diamond$

Notons la différence avec l'ordre $<_o$: cette fois-ci les éléments limites sont de la forme $3^a \times 5^e$ et non de la forme 3×5^e , où l'élément a sert comme un code de l'itération de l'hypersaut nécessaire en tant qu'oracle pour déplier l'ordre codé par e .

Proposition 6.3. Soit a dans le domaine de $<_{ho}$. Alors, J_a est un singleton Π_1^1 . $\star$

PREUVE. La preuve est similaire à celle du théorème 28-2.1 : on utilise le théorème du point fixe pour définir une classe $\Pi_1^1 \mathcal{P} \subseteq \mathbb{N} \times 2^{\mathbb{N}}$ telle que pour tout code e dans le domaine de $<_{ho}$ la classe $\{X : (e, X) \in \mathcal{P}\}$ est égale à $\{J_e\}$. Soit Ψ la fonctionnelle telle que $\Psi(\mathcal{O}^X) = X$ pour tout X . La définition de $\mathcal{P}$ est la suivante : $(b, X) \in \mathcal{P}$ si $b = 1$ et $X = \emptyset$, ou si $b = 2^a$ et (en reprenant les éléments de la preuve de la proposition 6.1) si X est dans la classe suivante :

$$\left\{ Y : (a, \Psi(Y)) \in \mathcal{P} \wedge \begin{array}{l} \forall e \in Y \quad e \in \mathcal{O}^{\Psi(Y)} \wedge 1 \in Y \\ \forall e \in \mathbb{N} \quad e \in Y \rightarrow 2^e \in Y \\ \forall e \in \mathbb{N} \quad \left(\begin{array}{l} \forall n \left(\begin{array}{l} \Phi_e(n) \downarrow \in Y \\ \Phi_e(n) <_o \Phi_e(n+1) \end{array} \right) \\ \rightarrow 3 \times 5^e \in Y \end{array} \right) \end{array} \right\},$$

ou si $b = 3^a \times 5^e$ et si X est dans la classe suivante :

$$\left\{ \bigoplus_n Y_n : (a, Y_0) \in \mathcal{P} \wedge \forall n \quad (\Phi_e(Y_0, n), Y_n) \in \mathcal{P} \right\}.$$

Ci-dessus, la relation $<_o$ se calcule à l'aide de l'oracle $\Psi(Y)$. $\blacksquare$

Il est intéressant de se demander jusqu'où va la hiérarchie des hypersauts. L'ordre $<_{ho}$ que l'on a défini conjointement à cette hiérarchie correspond à un bon ordre, qui va bien au-delà de ω_1^{ck} . Quelle est la valeur de cet ordinal ? Pour répondre à cette question, nous introduisons ci-après une hiérarchie d'ordinaux qui va de pair avec celle des hypersauts.

Définition 6.4. On définit la hiérarchie d'ordinaux suivante.

1. Soit ω_1^{ck} le plus petit ordinal non calculable.
2. Supposons ω_α^{ck} défini pour un ordinal dénombrable α . Alors, $\omega_{\alpha+1}^{ck}$ est le plus petit ordinal supérieur à ω_α^{ck} et de la forme ω_1^X pour un certain X .
3. Supposons $\omega_{\alpha_n}^{ck}$ défini pour $\alpha_0 < \alpha_1 < \dots$. Alors, $\omega_{\sup_n \alpha_n}^{ck} = \sup_n \omega_{\alpha_n}^{ck}$.



En utilisant et en itérant le théorème 29-7.6 on montre facilement que $\omega_2^{ck} = \omega_1^{\mathcal{O}}$, $\omega_3^{ck} = \omega_1^{\mathcal{O}^{\mathcal{O}}}$, etc. Notons que $\omega_\omega^{ck} = \sup_n \omega_n^{ck}$ n'est lui-même pas de la forme ω_1^X pour un certain X :

Proposition 6.5. Supposons $\omega_1^X > \omega_n^{ck}$ pour tout n . Alors, $\omega_1^X > \omega_\omega^{ck}$. ★

PREUVE. On va définir une fonction totale $f : \mathbb{N} \rightarrow \mathcal{O}^X$ qui sera $\Pi_1^1(X)$ et telle que $|f(n)| = \omega_n^{ck}$. Comme la fonction est totale sur $\mathbb{N}$ la relation $f(n) = a$ est équivalente à $\forall b \neq a \ f(n) \neq b$. La relation $f(n) = a$ est donc également $\Sigma_1^1(X)$. En particulier, l'image de f est un ensemble $\Sigma_1^1(X)$ inclus dans $\mathcal{O}^X$. D'après le théorème de majoration de Spector, son supremum est donc strictement inférieur à ω_1^X , ce qui nous donne $\omega_1^X > \omega_\omega^{ck}$.

Définissons à présent notre fonction : $f(0) = a$ si

- (1) $a \in \mathcal{O}^X$;
- (2) pour tout e aucune fonction g n'est un isomorphisme entre l'ordre codé par a et celui codé par e ;
- (3) $\forall b <_o^X a \ \exists c \in \mathcal{O} \ b \in \mathcal{O}_{<c}^X$;
- (4) a est le plus petit entier de l'ensemble $\mathcal{O}_{=|a|}^X$.

Chaque condition est $\Pi_1^1(X)$. La première assure que a est bien un ordinal, la deuxième et la troisième que $|a| = \omega_1^{ck}$, et la dernière est là pour l'unicité de a . Notons que l'on peut alors définir un code $\Delta_1^1(X)$ pour $\mathcal{O}$ uniformément en $f(0)$. On note alors H_0 l'ensemble $\mathcal{O}$.

On définit ensuite inductivement $f(n+1) = a$ si :

- (1) $a \in \mathcal{O}^X$;
- (2) pour tout e aucune fonction g n'est un isomorphisme entre l'ordre codé par a et celui codé par e avec oracle H_n ;
- (3) $\forall b <_o^X a \ \exists c \in \mathcal{O}^{H_n} \ b \in \mathcal{O}_{<c}^X$;
- (4) a est le plus petit entier de l'ensemble $\mathcal{O}_{=|a|}^X$.

On définit enfin inductivement H_{n+1} comme étant l'ensemble $\mathcal{O}^{H_n}$ qui est $\Delta_1^1(X)$ uniformément en $f(n+1)$. Chaque condition est $\Pi_1^1(X)$. Pour la condition (2), H_n est simplement une notation pour un ensemble $\Delta_1^1(X)$ dont on connaît le code uniformément en $f(n)$. Le fait que $f(n+1)$ ait

besoin de réutiliser la valeur $f(n)$ n'est pas un problème pour donner une définition $\Pi_1^1(X)$ du prédicat $f(n) = a$. On peut par exemple utiliser le théorème du point fixe pour avoir accès au code du graphe de f dans la définition de f . ■

On peut montrer que pour la plupart des ordinaux limites α , l'ordinal ω_α^{ck} n'est pas de la forme ω_1^X pour un certain X . C'est en fait le cas pour tout ordinal limite $\alpha < \omega_\alpha^{ck}$. Il suffit de répéter la preuve précédente, mais avec une fonction $f : \alpha \rightarrow \mathcal{O}^X$ — on utilise bien sûr en pratique des codes de $\mathcal{O}_{=\alpha}^X$ à la place de α .

Définition 6.6. Un ordinal dénombrable α est *récursivement inaccessible* si α est limite et $\omega_\alpha^{ck} = \omega_1^X$ pour un certain X . ◇

Notons que par la remarque précédant cette définition, tout ordinal récursivement inaccessible α est tel que $\alpha = \omega_\alpha^{ck}$. Il ne s'agit toutefois pas d'une condition suffisante, et le premier ordinal tel que $\alpha = \omega_\alpha^{ck}$ (défini par $\sup_n f^{(n)}(1)$ où $f(\alpha) = \omega_\alpha^{ck}$) n'est pas récursivement inaccessible.

Notons enfin que notre définition d'ordinal récursivement inaccessible n'est pas la définition originale (dont l'équivalence avec la nôtre découle d'un théorème de Sacks [191]), qui nécessite de travailler dans l'univers des constructibles de Gödel et dépasse le cadre de cet ouvrage.

Sacks [190] a montré que le premier ordinal récursivement inaccessible est égal à $|<_{ho}|$, le plus petit ordinal inaccessible par notre hiérarchie des hypersauts. Il est également possible de montrer que le premier ordinal récursivement inaccessible peut être encodé par un singleton Π_1^1 . En particulier, les singletons Π_1^1 vont « au-delà » de la hiérarchie des hypersauts.

Chapitre 31

Les systèmes ATR_0 et $\Pi_1^1\text{-CA}_0$

Les mathématiques à rebours possèdent cinq grands systèmes de référence, à savoir RCA_0 , WKL_0 , ACA_0 , ATR_0 et $\Pi_1^1\text{-CA}_0$, linéairement ordonnés par l'implication logique. Les premiers systèmes, RCA_0 , WKL_0 et ACA_0 , correspondent à des notions calculatoires connues en calculabilité classique : les ensembles calculables, les complétions de l'arithmétique de Peano, et enfin les ensembles arithmétiques. Nous avons donc les outils nécessaires pour étudier ces systèmes dans le chapitre 22.

Les systèmes ATR_0 et $\Pi_1^1\text{-CA}_0$, en revanche, font intervenir des itérations transfinies d'opérateurs arithmétiques, et des quantifications du second ordre. Leurs puissances calculatoires correspondantes se situent donc naturellement plutôt au niveau de l'hypercalculabilité. Profitons donc de cette nouvelle connaissance pour revenir sur l'étude de ces deux systèmes en mathématiques à rebours.

1. Définitions

Rappelons les définitions formelles des systèmes ATR_0 et $\Pi_1^1\text{-CA}_0$, telles que présentées dans le chapitre 22.

Le système ATR_0 . Le système ACA_0 était caractérisé par l'axiome de compréhension sur les formules Σ_1^0 . Le système ATR_0 est quant à lui caractérisé par un axiome d'existence d'ensemble plus puissant, permettant informellement d'utiliser les définitions mathématiques par récurrence transfinie. L'exemple canonique est l'existence de l' α -itération du saut Turing, pour un ordinal α dans le modèle considéré. La formulation précise de ATR_0

soulève un point important : la notion de bon ordre est une notion *relative au modèle*. Plus précisément, considérons un ordre total strict $<$ sur un ensemble $A \subseteq \mathbb{N}$. L'énoncé « $<$ est bien fondé sur A » s'exprime formellement par la formule du second ordre $\text{WO}(<, A)$ suivante :

$$\forall B \subseteq A \ (B \neq \emptyset \rightarrow \exists x \in B \ \forall y \in B \ y \not< x).$$

Autrement dit, $\text{WO}(<, A)$ signifie que tout sous-ensemble non vide de A possède un plus petit élément au sens de l'ordre $<$. En particulier, un modèle $\mathcal{M}$ satisfait $\text{WO}(<, A)$ si tout sous-ensemble non vide B de A *présent dans le modèle* possède un plus petit élément. Il se peut donc qu'un ordre mal fondé $<$ sur A semble être un bon ordre du point de vue de $\mathcal{M}$, mais ne soit pas un bon ordre en absolu, dans le sens où il existe un sous-ensemble de A sans plus petit élément, mais qu'un tel ensemble ne soit pas dans $\mathcal{M}$.

Soit $\theta(x, X)$ une formule arithmétique contenant notamment les variables libres x et X . Cette formule induit un opérateur $\Theta : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ sur les ensembles défini par $\Theta(X) = \{n \in \mathbb{N} : \theta(n, X)\}$. Rappelons le schéma de récursion transfinie présenté dans la section 22-8.3.

Définition 1.1

Le schéma de récursion transfinie énonce, pour toute formule arithmétique à paramètre $\theta(x, X)$, tout ensemble A et tout ordre total strict $< \subseteq A \times A$ tel que $\text{WO}(<, A)$, l'existence d'un ensemble $Y = \bigoplus_{a \in A} Y_a$ défini pour tout $a \in A$ par

$$Y_a = \Theta\left(\bigoplus_{b < a} Y_b\right).$$

◇

Par exemple, si l'on considère la formule $\theta(x, X) = \Phi_x^X(x) \downarrow$ et l'ensemble bien ordonné $(\mathbb{N}, <_{\mathbb{N}})$, alors l'opérateur Θ est le saut Turing défini par $\Theta(X) = X'$. On a

$$Y_0 = \Theta(\emptyset) = \emptyset', \quad Y_1 = \Theta(Y_0) = \emptyset'', \quad Y_2 = \Theta(Y_0 \oplus Y_1) = (\emptyset' \oplus \emptyset'')',$$

et ainsi de suite. L'ensemble Y résultant est de même degré que l' ω -saut Turing de $\emptyset$.

Notation

Le système ATR_0 est RCA_0 augmenté du schéma de récursion transfinie.

Il est clair, au vu de ce qui précède, que le système ATR_0 implique ACA_0 dans RCA_0 . En effet, RCA_0 prouve que l'ordre naturel sur les entiers est un bon ordre. Ainsi, $\text{RCA}_0 + \text{ATR}_0$ prouve l'existence de X' pour tout ensemble X , donc prouve ACA_0 , d'après l'exercice 22-6.2.

Le système $\Pi_1^1\text{-CA}_0$. Le système $\Pi_1^1\text{-CA}_0$ est quant à lui plus simple à définir formellement, à l'aide de formules de l'arithmétique du second

ordre. Rappelons que le schéma de compréhension est donné pour toute formule $F(x)$ par

$$\exists X \forall y (y \in X \leftrightarrow F(y)) \quad (9)$$

Notation

Le système $\Pi_1^1\text{-CA}_0$ est RCA_0 augmenté du schéma de compréhension (9) pour les formules Π_1^1 à paramètre.

Commençons par prouver le résultat qui suit.

Proposition 1.2. Le système $\Pi_1^1\text{-CA}_0$ implique le système ATR_0 . ★

PREUVE. Soit $\theta(x, X)$ une formule arithmétique et soit $< \subseteq A \times A$ un bon ordre sur un ensemble A . Précisons ici que la notation $\bigoplus_{b < a} X_b$ pour $a \in A$ désigne l'ensemble $\{\langle b, n \rangle : n \in X_b\}$. Notons par ailleurs que $\Pi_1^1\text{-CA}_0$ signale le schéma de compréhension pour les formules Σ_1^1 — la compréhension Δ_0 avec paramètre du second ordre permet de montrer que le complémentaire de chaque ensemble existe).

Considérons la formule arithmétique suivante :

$$F(a, X) \leftrightarrow \forall b \leq a \, X_b = \left\{ m : \theta\left(m, \bigoplus_{c < b} X_c\right) \right\}, \text{ où } X = \bigoplus_{b < a} X_b.$$

En supposant $\Pi_1^1\text{-CA}_0$, montrons que pour tout $a \in A$ il existe un unique Z tel que $F(a, Z)$. Supposons par l'absurde que cela ne soit pas le cas. Par hypothèse sur notre ordre $< \subseteq A \times A$, il existe un plus petit $a \in A$ tel que ce n'est pas le cas. En particulier, pour tout $b < a$, il existe un unique ensemble Z^b tel que $F(b, Z^b)$. Montrons que l'ensemble $Z = \bigoplus_{b < a} Z^b$ existe. Considérons la formule

$$F(\langle b, n \rangle) \equiv b < a \wedge \exists Z \, F(b, Z) \wedge n \in Z.$$

D'après le schéma de compréhension Σ_1^1 , il existe un ensemble Z tel que

$$\langle b, n \rangle \in Z \leftrightarrow F(\langle b, n \rangle).$$

Par hypothèse d'induction, cet ensemble est nécessairement unique et égal à $\bigoplus_{b < a} Z^b$. Il suffit à présent d'appliquer le schéma d'axiome de compréhension pour les formules arithmétiques pour obtenir l'existence de

$$Z^a = \left\{ m : \theta\left(m, \bigoplus_{b < a} Z^b\right) \right\},$$

qui vérifie donc $F(a, Z^a)$, ce qui contredit notre hypothèse sur a .

Donc, pour tout $a \in A$, il existe un unique ensemble Z tel que $F(a, Z)$. Pour finir, on montre l'existence de l'ensemble $\bigoplus_{a \in A} Z^a$, en utilisant l'axiome de compréhension Σ_1^1 sur la formule

$$F(\langle a, n \rangle) \equiv a \in A \wedge \exists Z \, F(a, Z) \wedge n \in Z. \quad \blacksquare$$

2. ATR_0 et $\Pi_1^1\text{-CA}_0$ en hypercalculabilité

Tout comme WKL_0 correspond à l'existence de degrés PA relatifs à tout ensemble, ou ACA_0 à l'existence du saut Turing, les systèmes ATR_0 et $\Pi_1^1\text{-CA}_0$ ont également des équivalences calculatoires. Informellement, ATR_0 stipule que, pour tout X , l'ensemble $X^{(\alpha)}$ existe pour tout ordinal $\alpha < \omega_1^X$, et $\Pi_1^1\text{-CA}_0$ stipule quant à lui que $\mathcal{O}^X$ existe pour tout X .

Il convient toutefois de faire attention à la manière d'exprimer formellement ces deux axiomes. De la même manière que la notion de bon ordre est relative au modèle, la notion d'ordinal, en tant qu'ensemble bien ordonné, est sujette aux mêmes problématiques : il se peut qu'un entier a ne code pas en absolu pour un ordinal, mais que tous les sous-ensembles de $\{b : b <_o a\}$ présents dans le modèle aient un plus petit élément. Autrement dit, du point de vue du modèle, a sera le code d'un ordinal valide. Nous verrons que l'une des conséquences de ce phénomène est que HYP n'est pas un ω -modèle de ATR_0 , et qu'un ω -modèle de $\Pi_1^1\text{-CA}_0$ ne contient pas nécessairement le $\mathcal{O}$ de Kleene. En effet, l'axiome « pour tout X l'ensemble $\mathcal{O}^X$ existe » doit être exprimé dans le modèle, et ce sera le $\mathcal{O}$ de Kleene *du point de vue du modèle* qui existera dans le modèle, mais pas nécessairement l'authentique $\mathcal{O}$ de Kleene.

Voyons tout cela plus précisément. Remémorons-nous un instant la preuve que $\mathcal{O}^X$ est $\Pi_1^1(X)$ uniformément en X . La définition se fait comme la conjonction de deux conditions pour que $a \in \mathcal{O}$. La première condition est Π_2^0 : soit A l'ensemble contenant a ainsi que les éléments $b <_o a$. Alors, la condition est la suivante.

- (1) La relation $<_o$ est un ordre linéaire et total restreint aux éléments de A , et ces derniers sont égaux à 1 ou à 2^b pour un certain b , ou à 3×5^e pour un code e tel que Φ_e est total avec $\Phi_e(n) <_o \Phi_e(n+1)$ pour tout n .

Cette condition n'est pas suffisante : il se pourrait qu'un code a la satisfasse mais soit tel que l'énumération de A contienne une suite infinie d'éléments de plus en plus petits : $\dots <_o a_3 <_o a_2 <_o a_1 <_o a$. Il faut alors en plus satisfaire la condition Π_1^1 suivante.

- (2) Pour tout $B \subseteq A$, si B est non vide, alors B contient un plus petit élément pour $<_o$.

Notation

On notera $L_{\mathcal{O}}(a, X)$ le prédicat (1) relativisé à un oracle X , auquel on ajoute la condition que si $b <_o c <_o a$, alors également $2^b \leq_o c$.

Notation

On notera $W_{\mathcal{O}}(a, X)$ le prédicat (2) relativisé à un oracle X .

Remémorons-nous également $\mathcal{P} \subseteq \mathbb{N} \times 2^{\mathbb{N}} \times 2^{\mathbb{N}}$ la classe Π_2^0 , apparaissant dans le théorème 28-2.1, telle que pour tout X et tout $a \in \mathcal{O}^X$ l'ensemble H_a^X est l'unique élément tel que $\mathcal{P}(a, X, H_a^X)$. L'existence de chaque ensemble H_a^X pour $a \in \mathcal{O}^X$ est l'exemple canonique de l'utilisation de ATR_0 . Bien entendu, si l'on est tout à fait formel, le système ATR_0 , appliqué à la formule $\theta(x, X) = \Phi_x^X(x) \downarrow$, montre l'existence des ensembles

$$\emptyset', \emptyset'', (\emptyset' \oplus \emptyset'')', \dots,$$

à la place de $\emptyset', \emptyset'', \emptyset'''$. Mais, on montre aisément par induction dans ACA_0 que chaque H_a est many-one réductible à l'ensemble $Y_a = \Theta(\bigoplus_{b <_a} Y_b)$, où Θ est l'opérateur induit par la formule $\theta(x, X) = \Phi_x^X(x) \downarrow$. Réciproquement il est possible de montrer toujours par induction dans ACA_0 que les ensembles définissables par récursion transfinie sont tous calculables par une certaine itération transfinie du saut. Tout cela nous amène à la proposition suivante.

Proposition 2.1. Dans ACA_0 , le système ATR_0 est équivalent à l'énoncé suivant : « Pour tout X et pour tout $a \in \mathbb{N}$, si $L_{\mathcal{O}}(a, X)$ et $W_{\mathcal{O}}(a, X)$, alors il existe un ensemble Y tel que $\mathcal{P}(a, X, Y)$. » ★

Le lecteur pourra en trouver une preuve détaillée dans [203]. En ce qui concerne $\Pi_1^1\text{-CA}_0$, la preuve que tout ensemble Π_1^1 est many-one réductible à $\mathcal{O}$ se fait sans problème dans ACA_0 , et l'on a donc là aussi une caractérisation calculatoire de $\Pi_1^1\text{-CA}_0$.

Proposition 2.2. Dans ACA_0 , le système $\Pi_1^1\text{-CA}_0$ est équivalent à l'énoncé suivant : « Pour tout X , l'ensemble $\{a \in \mathbb{N} : L_{\mathcal{O}}(a, X) \text{ et } W_{\mathcal{O}}(a, X)\}$ existe. » ★

En particulier, dans chacun des deux cas précédents, le prédicat $W_{\mathcal{O}}(a, X)$ équivalent à : « $\forall B \subseteq \{b : b <_o^X a\}$, si B est non vide, alors B contient un plus petit élément pour $<_o^X$ » doit être interprété dans le modèle, et donc avec la quantification $\forall B$ faite dans le modèle. Ainsi, il se pourrait que a ne soit pas réellement le code d'un ordre bien fondé, mais qu'aucun sous-ensemble $B \subseteq \{b : b <_o^X a\}$ sans plus petit élément n'appartiennent au modèle, et donc que du point de vue du modèle — qui n'a pas assez d'ensembles pour détecter tout ce qui est mal fondé — a soit bien fondé.

Nous allons voir dans la prochaine section une conséquence de ces considérations : HYP n'est pas un modèle d' ATR_0 .

3. HYP n'est pas modèle de ATR_0

Les ensembles hyperarithmétiques sont exactement ceux qui peuvent être obtenus par récursion transfinie d'un opérateur arithmétique le long d'un bon ordre calculable.

En particulier, tout ω -modèle de ATR_0 contient tous les ensembles hyperarithmétiques, et l'on pourrait à première vue conjecturer que HYP forme un modèle de ATR_0 . Nous allons cependant montrer que ce n'est pas le cas, et que certains ω -modèles de ATR_0 possèdent nécessairement des ordres qui semblent être des bons ordres du point de vue du modèle, mais n'en sont pas en absolu. Les axiomes d' ATR_0 vont donc autoriser l'itération d'opérations arithmétiques le long de ces mauvais ordres, et ainsi impliquer l'existence d'ensembles non hyperarithmétiques. Nous introduisons pour cela la notion de pseudo-bon ordre, c'est-à-dire des ordres qui seraient considérés comme des bons ordres dans le modèle HYP .

Définition 3.1. Soit $X \in 2^{\mathbb{N}}$. On dit que a est un X -pseudo bon ordre si $L_{\mathcal{O}}(a, X)$ et si tout sous-ensemble $\Delta_1^1(X)$ de $\{b : b <_o a\}$ possède un plus petit élément. $\diamond$

Notation

On note $\text{PBO}^X \subseteq \mathbb{N}$ l'ensemble des X -pseudos bons ordres, et l'on note PBO l'ensemble des pseudos bons ordres sans relativisation.

Notons que l'on a $\mathcal{O}^X \subseteq \text{PBO}^X$ pour tout X .

Proposition 3.2. Pour tout X , l'ensemble PBO^X est $\Sigma_1^1(X)$. $\star$

PREUVE. On a $a \in \text{PBO}^X$ si, et seulement si,

$$\forall n \in \mathbb{N} \, n \notin \mathcal{O}^X \vee \exists Y \left(\mathcal{P}(n, X, Y) \wedge \begin{array}{l} Y \text{ ne calcule pas de sous-} \\ \text{ensemble de } \{b : b <_o a\} \\ \text{sans plus petit élément.} \end{array} \right),$$

ce qui est bien un prédicat $\Sigma_1^1(X)$. $\blacksquare$

Le théorème suivant dû à Harrison, se base sur un simple argument de définissabilité pour montrer l'existence nécessaire de pseudo bons ordres qui ne sont pas des bons ordres, mais le long desquels il existe quand même une hiérarchie de sauts Turing.

Théorème 3.3 (Harrison [84])

Pour tout X , il existe $a \in \text{PBO}^X \setminus \mathcal{O}^X$ tel qu'il existe Y pour lequel $(a, X, Y) \in \mathcal{P}$.

PREUVE. Soit X un ensemble quelconque. Notons que pour tout $a \in \mathcal{O}^X$ on a bien $L_{\mathcal{O}}(a, X)$ et $\exists Y (a, X, Y) \in \mathcal{P}$. Supposons que cela ne soit le cas pour aucun élément $a \in \text{PBO}^X \setminus \mathcal{O}^X$. Alors, on peut donner la définition $\Sigma_1^1(X)$ de $\mathcal{O}^X$ suivante : $a \in \mathcal{O}^X$ ssi $a \in \text{PBO}^X$ et $\exists Y (a, X, Y) \in \mathcal{P}$. Comme $\mathcal{O}^X$ n'est pas $\Sigma_1^1(X)$, on a là une contradiction. $\blacksquare$

Cela nous amène à la définition suivante.

Définition 3.4. Soit $X \in 2^{\mathbb{N}}$. On dit qu'un code a tel que $L_{\mathcal{O}}(a, X)$ est un *support du saut* de X si $\{Y : \mathcal{P}(a, X, Y)\}$ est non vide. $\diamond$

Notation

Soit $\text{SUPP}_X \subseteq \mathbb{N}$ l'ensemble des supports du saut de X .

Le théorème 3.3 montre donc que l'on a $\mathcal{O}^X \subsetneq \text{PBO}^X \cap \text{SUPP}_X$ pour tout ensemble X . À quoi ressemble alors un élément Y tel que $(a, X, Y) \in \mathcal{P}$ pour $a \in \text{PBO}^X \setminus \mathcal{O}^X$? On peut « déplier » Y le long des codes $b <_o a$; soit Y_b l'élément correspondant au code b . Si $b = 2^c$, on a alors $Y_b = Y'_c$. Si $b = 3 \times 5^e$, on a alors $Y_b = \bigoplus_n Y_{\Phi_e(n)}$. On voit en particulier que pour tout $b <_o c$, on a $Y_b \leq_T Y_c$. En fait, comme $L_{\mathcal{O}}(a, X)$, alors pour tout $b <_o c <_o a$ on a $2^b \leq_o c$, et donc $Y'_b \leq_T Y_c$. Comme $a \notin \mathcal{O}^X$, il doit exister une suite $\dots < a_3 <_o a_2 <_o a_1 <_o a$. Les Y_{a_n} , pour $n \in \mathbb{N}$, forment donc une suite descendante dans les sauts Turing, c'est-à-dire que l'on a : $Y'_{a_{n+1}} \leq_T Y_{a_n}$ pour tout n . Une telle suite ne peut être hyperarithmétique, comme le montre le théorème suivant.

Théorème 3.5 (Enderton, Putnam [56])

Soit une suite $(X_n)_{n \in \mathbb{N}}$ telle que $\forall n, X'_{n+1} \leq_T X_n$. Alors, pour tout n , l'ensemble X_n Turing calcule tous les ensembles hyperarithmétiques.

PREUVE. Soit $\alpha < \omega_1^{ck}$. Supposons que pour tout n l'ensemble X_n calcule $\emptyset^{(\alpha)}$. Montrons que pour tout n l'ensemble X_n calcule $\emptyset^{(\alpha+1)}$. Fixons X_n . Comme $\emptyset^{(\alpha)} \leq_T X_{n+1}$, alors $\emptyset^{(\alpha+1)} \leq_T X'_{n+1}$. Comme $X'_{n+1} \leq_T X_n$, alors $\emptyset^{(\alpha+1)} \leq_T X_n$.

Supposons à présent que pour $\alpha = \sup_n \beta_n$ — pour une suite calculable d'ordinaux $\beta_1 < \beta_2 < \dots$ — on ait $\emptyset^{(\beta_n)} \leq_T X_n$ pour tout n . Fixons X_n . Notons que X_{n+2} calcule aussi $\emptyset^{(\beta_n)}$ pour tout β_n . En utilisant X''_{n+2} , on peut lister tous les éléments Y_e qui sont calculés par une fonctionnelle Φ_e totale sur X_{n+2} . On peut ensuite décider uniformément en un code de β_n en utilisant Y''_e — et donc en utilisant X''_{n+2} — si Y_e est un élément du singleton Π_2^0 contenant $\emptyset^{(\beta_n)}$. En utilisant X''_{n+2} on peut donc reconstruire petit à petit $\bigoplus_n \emptyset^{(\beta_n)}$. Comme $X''_{n+2} \leq_T X_n$ on en déduit, que X_n calcule $\emptyset^{(\alpha)}$. ■

Le théorème précédent se relativise sans problème à n'importe quel oracle. C'est l'objet du corollaire qui suit.

Corollaire 3.6

Soit $X \in 2^{\mathbb{N}}$, et soit $a \in \text{SUPP}_X \setminus \mathcal{O}^X$.

Alors, pour tout $Y \in 2^{\mathbb{N}}$ tel que $\mathcal{P}(a, X, Y)$, l'ensemble Y calcule tous les ensembles hyperarithmétiques en X .

PREUVE. Il suffit de remarquer que pour un tel ensemble Y il existe une suite $(Y_n)_{n \in \mathbb{N}}$ telle que $Y'_{n+1} \leq_T Y_n$ avec $Y_0 = Y$. ■

Corollaire 3.7 (Enderton, Putnam [56])

Il existe des ensembles X qui Turing calculent tous les ensembles hyperarithmétiques et avec $\omega_1^X = \omega_1^{ck}$ — en particulier avec $X \not\leq_h \mathcal{O}$.

PREUVE. Soit $a \in \text{SUPP}_X \setminus \mathcal{O}^X$. Comme la classe $\{Y : \mathcal{P}(a, \emptyset, Y)\}$ est Π_2^0 , et donc Σ_1^1 , elle contient d'après le théorème 30-2.2 un élément X tel que $\omega_1^X = \omega_1^{ck}$. Dans le même temps, X calcule tous les ensembles hyperarithmétiques d'après le corollaire 3.6. ■

Nous avons vu avec le théorème 28-2.1 que pour tout code $a \in \mathcal{O}^X$, la classe $\{Y : \mathcal{P}(a, X, Y)\}$ est un singleton Π_2^0 . La situation est bien différente lorsque $a \in \text{SUPP}_X \setminus \mathcal{O}^X$. Intuitivement, dès lors que a ne code pas pour un ordre bien fondé, il existe une suite infinie décroissante, et il existe de nombreuses possibilités d'itérer le saut Turing le long de cette suite. Le corollaire suivant appuie cette intuition.

Corollaire 3.8

Soit $X \in 2^{\mathbb{N}}$, et soit $a \in \text{SUPP}_X \setminus \mathcal{O}^X$. Alors, $\{Y : \mathcal{P}(a, X, Y)\}$ est indénombrable.

PREUVE. Comme la classe $\{Y : \mathcal{P}(a, X, Y)\}$ ne contient que des éléments calculant tous les hyperarithmétiques, elle ne contient donc aucun élément hyperarithmétique. Comme cette classe est Σ_1^1 , elle doit être indénombrable d'après le théorème 30-3.4. ■

Corollaire 3.9

La classe HYP n'est pas un modèle de ATR_0 .

PREUVE. Soit $X = \emptyset$ et supposons par l'absurde que HYP est un modèle de ATR_0 . Au vu du théorème 3.3, il existe un code $a \in \text{PBO}^X \cap \text{SUPP}_X \setminus \mathcal{O}^X$. Comme a est dans PBO^X , du point de vue du modèle HYP, a code bien pour un ordinal. Comme HYP est modèle de ATR_0 , il doit forcément exister $Y \in \text{HYP}$ tel que $(a, X, Y) \in \mathcal{P}$. Comme $a \in \text{SUPP}_X$, d'après le théorème 3.5 aucun élément Y tel que $(a, X, Y) \in \mathcal{P}$ n'est hyperarithmétique, ce qui est une contradiction. Donc, HYP n'est pas modèle de ATR_0 . ■

4. Codes d'ordinaux non standard

L'ensemble $\text{PBO} \setminus \mathcal{O}$ est en quelque sorte un ensemble d'*ordinaux non standard* : il ne s'agit pas de bons ordres, mais il est impossible de s'en apercevoir même avec la puissance de calcul hyperarithmétique. Il s'agit d'une classe d'objets fascinante et qui n'a probablement pas encore révélé tous ses secrets. Voici un petit résumé de ce que l'on a réussi à en observer jusqu'ici.

Théorème 4.1 (Harrison [84])

Supposons $e \in \text{PBO}^X \setminus \mathcal{O}^X$. Alors, le type d'ordre de e est $\omega_1^X + \omega_1^X \nu + \alpha$, pour ν le type d'ordre de $\mathbb{Q}$ et $\alpha < \omega_1^X$.

PREUVE. Nous montrons le théorème pour $X = \emptyset$; il se relativise sans problème à n'importe quel ensemble X . Soit $e \in \text{PBO} \setminus \mathcal{O}$. Pour $a <_o e$, soient $A_{<a} = \{b : b <_o a\}$ et $A_{>a} = \{b : a <_o b <_o e\}$. Pour $a <_o b <_o e$, soit $A_{]a,b[} = \{c : a <_o c <_o b\}$. Si $A_{]a,b[}$ est bien ordonné, on note $|A_{]a,b[}|$ l'ordinal qui lui correspond.

Fait 1. Soit $a <_o e$ tel que $A_{>a}$ n'est pas bien ordonné pour $<_o$. Montrons que

$$\omega_1^{ck} = \sup\{|A_{]a,b[}| : b \in A_{>a} \text{ tel que } A_{]a,b[} \text{ est bien ordonné}\}.$$

Supposons que l'ensemble $A_{>a}$ ne soit pas bien ordonné pour $<_o$. Soit alors $\alpha = \sup\{|A_{]a,b[}| : b \in A_{>a} \text{ tel que } A_{]a,b[} \text{ est bien ordonné}\}$. Notons que l'on a nécessairement $\alpha \leq \omega_1^{ck}$, car dans le cas inverse on aurait un élément b tel que $A_{]a,b[}$ est bien ordonné et tel que $|A_{]a,b[}| = \omega_1^{ck}$, ce qui donnerait un calcul de ω_1^{ck} . Montrons que l'on a nécessairement $\alpha = \omega_1^{ck}$. Supposons par l'absurde $\alpha < \omega_1^{ck}$. Les deux ensembles

$$Z = \{b \in A_{>a} : A_{]a,b[} \text{ est bien ordonné}\},$$

$$Z_1 = \{b \in A_{>a} : \text{il existe un morphisme de } |A_{]a,b[}| \text{ vers } \alpha\}$$

sont égaux. L'ensemble Z_1 a une définition Σ_1^1 , tandis que l'ensemble Z a une définition Π_1^1 naturelle, et est donc Δ_1^1 . On peut à présent calculer via Z un ensemble mal fondé d'éléments de $A_{>a}$: il suffit de prendre un élément c_0 tel que $A_{]a,c_0[}$ est mal fondé, puis de chercher un élément $c_1 <_o c_0$ tel que $a <_o c_1$ et $c_1 \notin Z$, puis de chercher $c_2 <_o c_1$ tel que $a <_o c_2$ et $c_2 \notin Z$, etc. Comme Z est hyperarithmétique, cela contredit $e \in \text{PBO}$. Donc, $\alpha = \omega_1^{ck}$. Cela termine la preuve du fait 1.

On peut à présent appliquer cela à l'ordinal 1 codant pour le plus petit élément de e . On a ainsi

$$\omega_1^{ck} = \sup\{|A_{]1,b[}| : b \in A_{>1} \text{ tel que } A_{]1,b[} \text{ est bien ordonné}\}.$$

L'ordre de e commence donc par un segment initial bien fondé de taille ω_1^{ck} .

Fait 2. Soit $a <_o e$, Montrons que l'ensemble

$$Z = \{b : A_{]b,a[} \text{ est bien ordonné}\}$$

possède un plus petit élément. Supposons par l'absurde que ce n'est pas le cas. On définit à présent l'ensemble Π_1^1

$$B = \{(n, b) : n, b \in Z \text{ et } b <_o n <_o a\}.$$

En utilisant le théorème 29-4.1 d'uniformisation Π_1^1 , soit g une fonction partielle Π_1^1 telle que si $\{b : (n, b) \in B\}$ est non vide, alors $(n, g(n)) \in B$. On définit alors une fonction Π_1^1 totale f , avec $f(0)$ un élément de Z quelconque, puis $f(n+1) = g(f(n))$. On a alors $f(n+1) <_o f(n)$ pour tout n . Comme f est totale, elle est donc Δ_1^1 d'après la proposition 29-4.2. Son image est donc Δ_1^1 , ce qui contredit $e \in \text{PBO}$. Donc, pour tout $a <_o e$, l'ensemble $Z = \{b : A_{]b,a[} \text{ est bien ordonné}\}$ possède un plus petit élément. Cela prouve le fait 2.

Pour $a <_o e$, soit b_a un tel élément. Si $A_{>b_a}$ est bien ordonné, alors il s'agit d'un ordinal calculable α , ce qui correspond au segment final de l'ordre. Sinon, pour tout $a <_o e$, on a $\omega_1^{ck} = \sup_{b \in A} \{|A_{]b_a, b[} : A_{]b_a, b[} \text{ est bien ordonné}\}$.

Fait 3. Soient $a_1 <_o a_2 < e$, chacun étant les débuts d'une partie bien fondée (c'est-à-dire de la forme b_a pour un certain a). Montrons qu'il existe un $c <_o a_2$ tel que $c \notin \{b : A_{]a_1, b[} \text{ est bien ordonné}\}$.

Sachant que $a_1 <_o a_2$ sont les débuts d'une partie bien fondée, on a forcément $a_2 \notin \{b : A_{]a_1, b[} \text{ est bien ordonné}\}$. Alors, de même, on a forcément $c <_o a_2$ tel que $c \notin \{b : A_{]a_1, b[} \text{ est bien ordonné}\}$, car sinon a_2 permettrait de calculer ω_1^{ck} étant donné que a_1 démarre une partie bien fondée de taille ω_1^{ck} .

Le fait 1 montre que le type d'ordre de e commence par ω_1^{ck} . Les faits 1 et 2 ensemble montrent que chaque élément est soit dans un sous-ordre de type ω_1^{ck} , soit appartient à la partie finale $\alpha < \omega_1^{ck}$. Enfin, le fait 3 montre qu'entre deux copies de ω_1^{ck} , ou entre une copie de ω_1^{ck} et α , il existe une autre copie de ω_1^{ck} . Le type d'ordre de e est donc $\omega_1^{ck} + \omega_1^{ck}\nu + \alpha$. ■

Corollaire 4.2 (Feferman et Spector [58])

Il existe un ensemble $\mathcal{O}_1 \subseteq \mathcal{O}$ qui est Π_1^1 et bien ordonné par $<_o$, et dont le supremum est ω_1^{ck} .

PREUVE. Soit $e \in \text{PBO}^X \setminus \mathcal{O}^X$. Alors, en reprenant les notations du théorème précédent, on a $\mathcal{O}_1 = \{b <_o e : A_{]1, b[} \text{ est bien fondé}\}$. ■

L'ensemble $\mathcal{O}_1$ peut être utile pour s'abstraire encore plus des systèmes de codage : au lieu de considérer que l'on travaille avec le code d'un ordinal α , on peut alors faire comme si l'on travaillait directement avec l'ordinal α lui-même, via son unique code dans $\mathcal{O}_1$. Nous voyons à présent un

théorème remarquable sur la relation entre les ensembles PBO^X et SUPP^X . Nous allons voir en particulier que si un ordre mal fondé est le support d'une hiérarchie de saut, alors nécessairement cet ordre ne contient pas de sous-ensemble hyperarithmétique sans plus petit élément. En d'autres termes, $\text{SUPP}^X \subseteq \text{PBO}^X$. Nous utilisons pour cela le lemme suivant.

Lemme 4.3 (Steel). Soit Φ une fonctionnelle calculable. Alors, il n'existe aucune suite $(X_n)_{n \in \mathbb{N}}$ telle que $\Phi(X_n) = X'_{n+1}$ pour tout n . ★

PREUVE. Soit $(X_n)_{n \in \mathbb{N}}$ une suite telle que $\Phi(X_n) = X'_{n+1}$ pour tout n . On construit facilement une fonctionnelle Ψ telle que $\Psi(X_n, m) = X'_{n+m}$ pour tous $n, m \in \mathbb{N}$ avec $m > 0$. En utilisant le théorème du point fixe, soit e le code d'un programme qui s'arrête sur l'oracle X si $\exists m > 0 \ e \notin \Psi(X, m)$.

Supposons $e \notin X'_0$. Alors, pour tout $m > 0$ on a $e \in \Psi(X_0, m)$ c'est-à-dire $e \in X'_m$ pour tout $m > 0$. En particulier, on a $e \in X'_1$, et donc $\exists k > 0$ tel que $e \notin \Psi(X_1, k)$. On a donc $e \notin X'_{1+k}$, ce qui contredit $e \in X'_m$ pour tout $m > 0$.

Supposons à présent $e \in X'_0$. Alors, $\exists m > 0$ tel que $e \notin \Psi(X_0, m)$ et donc tel que $e \notin X'_m$. On peut donc répéter l'argument précédent mais avec la suite $Y_0 = X_m, Y_1 = X_{m+1}, \dots$. On aboutit dans tous les cas à une contradiction. ■

On peut à présent montrer le résultat attendu.

Théorème 4.4 (Friedman [68])

Pour tout oracle X , on a $\text{SUPP}^X \subseteq \text{PBO}^X$.

PREUVE. On montre le résultat pour $X = \emptyset$, la preuve se relativisant sans problème à tout oracle X . Supposons que la classe $\{Y : \mathcal{P}(e, \emptyset, Y)\}$ soit non vide pour $e \notin \mathcal{O}$, et soit Y l'un de ses membres. Supposons par l'absurde qu'il existe un ensemble hyperarithmétique $Z \subseteq \{a : a <_o e\}$ ne possédant pas de plus petit élément. On va alors construire une suite d'ensembles $(X_n)_{n \in \mathbb{N}}$ et une fonctionnelle Φ telle que $\Phi(X''_n) = X'''_{n+1}$. Il suffira d'appliquer le lemme 4.3 à la suite $(X''_n)_{n \in \mathbb{N}}$ pour obtenir une contradiction.

On peut déplier l'ensemble Y le long des codes $a <_o e$. Soit Y_a l'ensemble correspondant au code a . D'après le théorème 3.5, chaque ensemble Y_a pour $a \in Z$ calcule tous les hyperarithmétiques et donc en particulier Z . Soit $a \in \mathcal{O}$ tel que $\Psi(H_a) = Z$ pour une fonctionnelle Ψ . La fonctionnelle Φ procède comme suit : sur un oracle de la forme $(1^b 0 Y_b)''$ pour $b \in Z$, la fonctionnelle liste toutes les fonctions totales pour Y_b et regarde si le résultat du calcul est bien un membre de la classe $\{X : \mathcal{P}(a, \emptyset, X)\}$, auquel cas il s'agit nécessairement de H_a . Une fois H_a identifié, la fonctionnelle s'en sert pour calculer Z , puis pour chercher $c \in Z$ avec $c <_o b$ et tel que $2^c \leq_o b$.

Une fois cet élément trouvé, la fonctionnelle peut alors calculer $Y_{2^c} = Y'_c$ à partir de Y_b , puis calculer $(1^c 0 Y_c)' \leq_T Y'_c \leq_T Y_b$ puis finalement renvoyer $(1^c 0 Y_c)'''$ en utilisant Y_b'' . ■

A-t-on égalité entre SUPP^X et PBO^X ? Friedman [66] a montré que ce n'était pas le cas, en construisant un élément $e \in \text{PBO} \setminus \text{SUPP}$.

Théorème 4.5 (Friedman)

Pour tout oracle X , on a $\text{SUPP}^X \subsetneq \text{PBO}^X$.

PREUVE. On montre le théorème pour $X = \emptyset$, la preuve se relativisant sans problème à tout oracle X . Soit f une fonction totale calculable, définie à l'aide du théorème 27-5.10 et de la proposition 27-5.18 telle que $e \in \mathcal{T}$ ssi $f(e) \in \mathcal{O}$. Soit g la fonction totale calculable qui sur le code e d'un arbre c.e. $T_e \subseteq \mathbb{N}^{<\mathbb{N}}$ renvoie le code de l'arbre $T_{g(e)}$ de l'espace de Baire qui encode les éléments de la classe Π_2^0 suivante :

$$\{X : L_{\mathcal{O}}(f(e), \emptyset) \text{ et } (f(e), \emptyset, X) \in \mathcal{P}\}.$$

Expliquons un peu plus concrètement l'arbre $T_{g(e)}$: étant donné $\bigcap_n \mathcal{U}_n$ une description Π_2^0 de la classe $\{X : (f(e), \emptyset, X) \in \mathcal{P}\}$ et $\bigcap_n A_n$ une description Π_2^0 de l'ensemble $\{a : L_{\mathcal{O}}(a, \emptyset)\}$, on procède ainsi : supposons un nœud ρ de taille $n + 1$ énuméré dans $T_{g(e)}$ et correspondant à des chaînes $\sigma_0 \prec \dots \prec \sigma_n$, où σ_i est énuméré dans $\mathcal{U}_i$. Avant d'énumérer des fils de ρ dans $T_{g(e)}$, on cherche si $f(e) \in A_{n+1}$. Si cela est bien vérifié, on énumère ρm dans $T_{g(e)}$ pour tout m correspondant à une extension $\sigma_{n+1} \succ \sigma_n$ avec σ_{n+1} énuméré dans $\mathcal{U}_{n+1}$. Il est clair que

$$L_{\mathcal{O}}(f(e), \emptyset) \text{ et } \{X : (f(e), \emptyset, X) \in \mathcal{P}\}$$

est non vide si, et seulement si, l'arbre $T_{g(e)}$ a un chemin infini, auquel cas les chemins infinis de $T_{g(e)}$ sont exactement les encodages des éléments de $\{X : (f(e), \emptyset, X) \in \mathcal{P}\}$.

À l'aide du théorème du point fixe, soit à présent e tel que $T_e = T_{g(e)}$. Supposons par l'absurde que T_e ne contienne pas de chemin infini. Alors, $f(e) \in \mathcal{O}$ et dans ce cas l'arbre $T_{g(e)}$ doit contenir un chemin infini, ce qui est une contradiction. Donc, T_e est mal fondé. Comme $T_{g(e)}$ est mal fondé, on en déduit $L_{\mathcal{O}}(f(e), \emptyset)$ et $\{X : (f(e), \emptyset, X) \in \mathcal{P}\}$ non vide, ce qui implique $f(e) \in \text{SUPP} \setminus \mathcal{O}$. En particulier, $f(e) \in \text{PBO}$. Notons que tout élément X tel que $(f(e), \emptyset, X) \in \mathcal{P}$ calcule un élément de $[T_e]$ — sa propre représentation —, et donc calcule un sous-ensemble de $Z_X \subseteq \{a : a <_o f(e)\}$ qui n'a pas de plus petit élément. Considérons à présent l'ordre $f(e) +_o f(e)$. La preuve que l'on a $L_{\mathcal{O}}(f(e) +_o f(e), \emptyset)$ est laissée aux bons soins du lecteur. Tout élément $a <_o f(e)$ possède son analogue $f(e) +_o a$ qui est tel que $f(e) <_o f(e) +_o a <_o f(e) +_o f(e)$.

Réciproquement, pour tout élément a tel que $f(e) <_o a <_o f(e) +_o f(e)$, on peut chercher un élément $b <_o a$ tel que $a = f(e) +_o b$. Il est clair que $f(e) +_o f(e) \in \text{PBO}$: en effet, tout sous-ensemble Δ_1^1

$$Z \subseteq \{a : a <_o f(e) +_o f(e)\}$$

et sans plus petit élément est cofini dans $\{a : a <_o f(e)\}$ — auquel cas $f(e) \notin \text{PBO}$ — ou cofini dans $\{a : a <_o f(e) +_o f(e)\}$ auquel cas Z peut calculer son analogue sous $f(e)$, et là encore $f(e) \notin \text{PBO}$. On prétend qu'en revanche $f(e) +_o f(e) \notin \text{SUPP}$. Supposons par l'absurde qu'il existe un ensemble Y tel que $(f(e) +_o f(e), \emptyset, Y) \in \mathcal{P}$. On peut alors « déplier » Y le long des codes de pseudo ordinaux et pour $a <_o f(e) +_o f(e)$ on note Y_a l'ensemble correspondant à a dans ce dépliage. Notons que $Y_{f(e)}$ — qui est tel que $(f(e), \emptyset, Y_{f(e)}) \in \mathcal{P}$ — calcule l'ensemble $Z_{Y_{f(e)}} \subseteq \{a : a <_o f(e)\}$ sans plus petit élément, et donc un ensemble $Z \subseteq \{a : f(e) <_o a <_o f(e) + f(e)\}$ sans plus petit élément. On va alors définir une fonction calculable Φ et une suite $(X_n)_{n \in \mathbb{N}}$ qui seront telles que $\Phi(X_n) = X'_n$, en contradiction avec le lemme 4.3. La suite $(X_n)_{n \in \mathbb{N}}$ est simplement donnée par $X_n = 1^{a_n} 0 Y_{a_n}$ pour une suite $\dots < a_3 <_o a_2 < a_1 \in Z$. La fonctionnelle Φ agit comme suit sur l'ensemble $1^{a_n} 0 Y_{a_n}$ avec $f(e) <_o a_n$ et $a_n \in Z$: elle calcule $Y_{f(e)}$ à partir de Y_{a_n} et de la connaissance de a_n , puis elle calcule Z à partir de $Y_{f(e)}$ et enfin elle cherche $a_{n+1} \in Z$ avec $a_{n+1} <_o a_n$ tel que $2^{a_{n+1}} \leq_o a_n$. Notons que l'on a $Y_{2^{a_{n+1}}} = Y'_{a_{n+1}}$. À partir de Y_{a_n} , la fonctionnelle peut donc calculer $Y'_{a_{n+1}}$ puis calculer et renvoyer l'ensemble $(1^{(a_{n+1})} 0 Y_{a_{n+1}})'$. On a alors une contradiction avec le lemme 4.3. Il s'ensuit que notre hypothèse est fautive, et donc que $f(e) +_o f(e) \notin \text{SUPP}$. ■

Nous terminons cette section par un théorème d'Harrington : PBO et SUPP sont tous les deux Σ_1^1 -complets.

Théorème 4.6 (Harrington (non publié))

Soit A un ensemble Σ_1^1 tel que $\mathcal{O} \subseteq A \subseteq \text{PBO}$. Alors, A est Σ_1^1 -complet.

PREUVE. Nous établissons d'abord le résultat pour l'ensemble $\mathcal{T}$ des codes c. e. d'arbres bien fondés de l'espace de Baire, et pour l'ensemble PBT des codes c. e. d'arbres sans chemins infinis hyperarithmétiques.

On suppose donc que A , ensemble Σ_1^1 , vérifie $\mathcal{T} \subseteq A \subseteq \text{PBT}$. Montrons alors que l'on peut réduire par une réduction many-one l'ensemble des codes c. e. d'arbres mal fondés à A . Comme A est Σ_1^1 , il existe une fonction f telle que $n \in A$ ssi $f(n)$ est un code d'arbre c. e. mal fondé. En utilisant le théorème du point fixe, on définit une fonction g telle que pour tout e , la valeur $g(e)$ est l'arbre c. e. des morphismes (voir la définition 29-5.6) de l'arbre c. e. codé par e vers celui codé par $f(g(e))$.

Montrons d'abord que $g(e)$ code pour tout e pour un arbre mal fondé. Supposons par l'absurde que $g(e)$ code pour un arbre bien fondé. Alors, $g(e)$ appartient à A , et donc $f(g(e))$ code pour un arbre mal fondé. Il existe alors nécessairement un morphisme de l'arbre codé par e vers celui codé par $f(g(e))$, et donc $g(e)$ — qui code pour l'ensemble de ces morphismes — doit être mal fondé.

Montrons à présent que g est la réduction attendue, c'est-à-dire telle que e code pour un arbre mal fondé ssi $g(e) \in A$. Supposons que e code pour un arbre mal fondé. Comme $g(e)$ code toujours pour un arbre mal fondé, il existe un morphisme de l'arbre codé par e vers celui codé par $f(g(e))$. Donc, $f(g(e))$ code aussi pour un arbre mal fondé, et donc $g(e) \in A$ par définition de f . Supposons à présent que e code pour un arbre bien fondé. Soit $\alpha < \omega_1^{ck}$ la hauteur de cet arbre. Nous savons qu'il existe un morphisme de l'arbre codé par e vers celui codé par $f(g(e))$, car $g(e)$ code pour un arbre mal fondé. D'après la proposition 28-1.7, l'ensemble $\emptyset^{(\alpha+1)}$ peut obtenir uniformément en $\beta \leq \alpha$ l'ensemble $\mathcal{T}_{<\beta}$ des codes d'arbre c. e. bien fondés et de hauteur inférieure à β . On peut alors à l'aide de $\emptyset^{(\alpha+1)}$ calculer un tel morphisme : il suffit d'envoyer des nœuds commençant des arbres de hauteurs β vers des nœuds de même taille commençant des arbres qui ne sont pas de hauteur inférieure à β (tout en restant cohérent avec la partie du morphisme déjà définie). La connaissance de l'ensemble des codes d'arbres de hauteurs $\beta \leq \alpha$ est donc suffisante. L'arbre codé par $g(e)$ contient ainsi des chemins infinis hyparithmétiques, si bien qu'il n'appartient alors pas à PBO, et donc pas non plus à A .

Pour obtenir le résultat pour un A ensemble Σ_1^1 , avec $\mathcal{O} \subseteq A \subseteq \text{PBO}$, il suffit à présent de considérer avec l'aide du théorème 27-5.10 et de la proposition 27-5.18 une fonction calculable f telle que e est un code c. e. pour un arbre bien fondé — c'est-à-dire $e \in \mathcal{T}$ — ssi $f(e) \in \mathcal{O}$, et de considérer l'ensemble $B = \{e \in \mathbb{N} : f(e) \in A\}$. Un tel ensemble est Σ_1^1 . Il est clair par définition de f que $\mathcal{T} \subseteq B$ puisque $\mathcal{O} \subseteq A$. De plus, on montre facilement que si e code pour un arbre avec un chemin hyperarithmétique, alors si l'on a $L_{\mathcal{O}}(f(e), \emptyset)$, l'ensemble $\{a : a <_o f(e)\}$ contient un sous-ensemble hyperarithmétique qui n'a pas de plus petit élément. On a donc $e \notin \text{PBT}$ implique $f(e) \notin \text{PBO}$, donc $f(e) \in \text{PBO}$ implique $e \in \text{PBT}$. Comme $A \subseteq \text{PBO}$, alors $B \subseteq \text{PBT}$. On réduit alors tout ensemble Σ_1^1 à B , qui lui-même se réduit à A via la fonction f . ■

Corollaire 4.7

Les ensembles PBO et SUPP sont Σ_1^1 -complets.

PREUVE. Nous avons vu avec la proposition 3.2 que l'ensemble PBO est Σ_1^1 . L'ensemble SUPP est également Σ_1^1 : $a \in \text{SUPP}$ ssi $\exists Y \mathcal{P}(a, \emptyset, Y)$. Ils sont donc tous les deux Σ_1^1 -complets. ■

5. Séparation entre ATR_0 et $\Pi_1^1\text{-CA}_0$

Si certains modèles d' ATR_0 et de $\Pi_1^1\text{-CA}_0$ peuvent être complexes à appréhender en raison d'ordres qu'ils ne peuvent détecter comme étant mal fondés, il ne doit pas forcément en être ainsi. On peut tout à fait définir des modèles qui reflètent bien ce qu'il se passe à l'extérieur d'eux-mêmes.

Définition 5.1. Soit $\mathcal{M} \subseteq 2^{\mathbb{N}}$. Alors, $\mathcal{M}$ est un β -modèle si pour tout X dans $\mathcal{M}$ et pour tout $\mathcal{A}$ une classe $\Sigma_1^1(X)$ non vide, il existe $Y \in \mathcal{A} \cap \mathcal{M}$. $\diamond$

Notons que l'on suppose implicitement que les β -modèles sont toujours des ω -modèles, pour lesquels on ne spécifie donc que la partie du second ordre.

Proposition 5.2. Soit $\mathcal{M}$ un β -modèle clos par jointure Turing. Alors, $\mathcal{M}$ est un modèle de ATR_0 . $\star$

PREUVE. Pour chaque $X \in \mathcal{M}$ et chaque $Y \leq_T X$, l'élément Y est un singleton $\Pi_1^0(X)$, et donc $Y \in \mathcal{M}$. Ainsi, $\mathcal{M}$ est clos par jointure et par réduction Turing, donc est un idéal Turing. On a donc $\mathcal{M} \models \text{RCA}_0$. Par ailleurs, pour tout $X \in \mathcal{M}$ la classe $\{X'\}$ est $\Sigma_1^1(X)$, donc $X' \in \mathcal{M}$. Ainsi, $\mathcal{M} \models \text{ACA}_0$.

Par la proposition 2.1, il suffit de montrer que si $X \in \mathcal{M}$ et $a \in \mathbb{N}$ est tel que $L_{\mathcal{O}}(a, X)$ et $W_{\mathcal{O}}(a, X)$, alors il existe un $Y \in \mathcal{M}$ tel que $(a, X, Y) \in \mathcal{P}$. Soit $X \in \mathcal{M}$, et soit $a \in \mathbb{N}$ tel que $L_{\mathcal{O}}(a, X)$ et $W_{\mathcal{O}}(a, X)$. Montrons alors que $a \in \mathcal{O}^X$. Si $a \notin \mathcal{O}^X$, la classe des sous-ensembles de $\{b : b <_o a\}$ sans plus petit élément est en effet un $\Sigma_1^1(X)$ non vide, et il existe donc un témoin de ce fait dans $\mathcal{M}$, donc $\mathcal{M} \not\models W_{\mathcal{O}}(a, X)$, contradiction. Si $a \in \mathcal{O}^X$, la classe $\{Y : (a, X, Y) \in \mathcal{P}\}$ est alors le singleton $\Pi_2^0 H_a^X$, et donc $H_a^X \in \mathcal{M}$, car $\mathcal{M}$ est un β -modèle. $\blacksquare$

Proposition 5.3. Il existe un β -modèle minimal de $\Pi_1^1\text{-CA}_0$: la classe des éléments Turing réductibles à une itération finie de l'hypersaut. $\star$

PREUVE. Par la proposition 2.2, n'importe lequel des β -modèles de $\Pi_1^1\text{-CA}_0$ contient le vrai $\mathcal{O}$ de Kleene, et par conséquent $\mathcal{O}^{\mathcal{O}}$, $\mathcal{O}^{\mathcal{O}^{\mathcal{O}}}$, et ainsi de suite. Comme il est clos par réduction Turing, il contient tous les ensembles Turing réductibles à une itération finie de l'hypersaut.

Réciproquement, soit $\mathcal{M}$ le modèle des éléments Turing réductibles à une itération finie de l'hypersaut. Montrons alors que $\mathcal{M}$ est un idéal Turing. Pour tout $X \in \mathcal{M}$, il existe un n tel que $X \leq_T J_n$, où J_n est la n -ième itération de l'hypersaut. Si $Y \leq_T X$, alors $Y \leq_T J_n$, donc $Y \in \mathcal{M}$. De plus, pour tous $X, Y \in \mathcal{M}$, on a $X \leq_T J_n$ et $Y \leq_T J_m$ pour $n, m \in \mathbb{N}$, et donc $X \oplus Y \leq_T J_{\max(n, m)}$. Ainsi, $X \oplus Y \in \mathcal{M}$. On a donc, $\mathcal{M} \models \text{RCA}_0$.

Montrons que pour tout $X \in \mathcal{M}$, $\mathcal{O}^X \in \mathcal{M}$. Par définition de $\mathcal{M}$, il existe un $n \in \mathbb{N}$ tel que $X \leq_T J_n$. En particulier, $\mathcal{O}^X \leq_T J_{n+1}$, et donc $\mathcal{O}^X \in \mathcal{M}$. Ainsi, $\mathcal{M} \models \Pi_1^1\text{-CA}_0$. Notons au passage que $\mathcal{M}$ est un β -modèle, car pour une classe $\Sigma_1^1(X)$ non vide, un élément de cette classe est d'après le théorème 30-2.2 calculable à l'aide de $\mathcal{O}^X$, et donc appartient à $\mathcal{M}$. ■

On montre à présent notre théorème de séparation entre ATR_0 et $\Pi_1^1\text{-CA}_0$.

Théorème 5.4

Il existe des β -modèles d' ATR_0 ne contenant pas $\mathcal{O}$.

PREUVE. Soit $e_1, e_2, \dots$ une suite, où e_i est un code de classe Σ_1^1 et où chaque code possible est répété une infinité de fois. À l'étape n , supposons que l'on dispose de n ensembles $X_1, \dots, X_n$ avec $\mathcal{O}^{X_1 \oplus \dots \oplus X_n} \leq_T \mathcal{O}$ et tels que X_{i+1} est un élément de la classe $\Sigma_1^1(X_1 \oplus \dots \oplus X_i)$ de code e_{i+1} pour tout $i < n$, si cette classe est non vide. On considère alors la classe $\Sigma_1^1(X_1 \oplus \dots \oplus X_n)$ de code e_{n+1} . Si celle-ci est non vide, on définit à l'aide du théorème 30-2.2 X_{n+1} comme étant un élément de cette classe tel que $\mathcal{O}^{X_1 \oplus \dots \oplus X_n \oplus X_{n+1}} \leq_T \mathcal{O}^{X_1 \oplus \dots \oplus X_n} \leq_T \mathcal{O}$. Si cette classe est vide, on définit $X_{n+1} = \emptyset$. Notre modèle $\mathcal{M}$ est donné par $\{X_n : n \in \mathbb{N}\}$.

Il est clair que $\mathcal{M}$ ne contient que des éléments X tels que $\mathcal{O}^X \leq_T \mathcal{O}$, et ne contient donc pas $\mathcal{O}$. De plus, $\mathcal{M}$ est un β -modèle, car pour $X_n \in \mathcal{M}$ et une classe $\Sigma_1^1(X_n)$, il existe un code e_m pour $m > n$ tel que e_m correspond avec un oracle $X_1 \oplus \dots \oplus X_n \oplus \dots \oplus X_{m-1}$ à la classe $\Sigma_1^1(X_n)$ (et ce indépendamment de m). Pour un tel e_m , on aura donc un élément X_m de cette classe dans notre modèle $\mathcal{M}$. Enfin, $\mathcal{M}$ est clos par jointure Turing, car pour tous n_1, n_2 il existe un code e_m pour $m > n_1, n_2$ tel que $X_{n_1} \oplus X_{n_2}$ est le seul élément de la classe $\Sigma_1^1(X_1 \oplus \dots \oplus X_{n_1} \oplus \dots \oplus X_{n_2} \oplus \dots \oplus X_{m-1})$ de code e_m . D'après la proposition 5.2, $\mathcal{M}$ est donc un β -modèle de ATR_0 . ■

Corollaire 5.5

Le système ATR_0 n'implique pas $\Pi_1^1\text{-CA}_0$.

PREUVE. On a un β -modèle d' ATR_0 ne contenant pas $\mathcal{O}$. Comme il s'agit d'un β -modèle, il ne peut être modèle de $\Pi_1^1\text{-CA}_0$, car tout β -modèle de $\Pi_1^1\text{-CA}_0$ contient $\mathcal{O}$. ■

Il est tentant de faire un parallèle entre les systèmes ATR_0 et WKL_0 ainsi qu'entre les systèmes ACA_0 et $\Pi_1^1\text{-CA}_0$. Là où ACA_0 affirme l'existence du saut Turing de tout ensemble, $\Pi_1^1\text{-CA}_0$ affirme l'existence de l'hypersaut Turing pour tout ensemble.

Là où l'on sépare WKL_0 de ACA_0 en construisant un modèle de WKL_0 ne contenant pas le saut, on sépare ATR_0 de $\Pi_1^1\text{-CA}_0$ en construisant un β -modèle d' ATR_0 ne contenant pas l'hypersaut.

La correspondance a toutefois ses limites. Il est ainsi possible de construire un modèle dénombrable de WKL_0 dont la représentation est low, mais il est impossible de construire un β -modèle dénombrable d' ATR_0 qui soit hyperlow.

Définition 5.6. Un ensemble $M \subseteq \mathbb{N}$ est un *code* d'une classe dénombrable $\mathcal{M} = \{X_n : n \in \mathbb{N}\}$ si $M = \bigoplus_n X_n$. $\diamond$

Notons qu'une même classe dénombrable possède une infinité de codes, car l'ordre de ses éléments n'est pas fixé.

Théorème 5.7

Soit $\mathcal{M}$ un β -modèle dénombrable du système ATR_0 codé par M . Alors, $\mathcal{O} \leq_T M''$.

PREUVE. Il est facile de donner une définition arithmétique en $\mathcal{M}$, de l'ensemble des codes d'arbres mal fondés. Il s'agit de l'ensemble des codes e codant pour un arbre T_e tel qu'il existe $X \in \mathcal{M}$ appartenant à $[T_e]$. La quantification existentielle du second ordre sur X se transforme en quantification existentielle du premier ordre M . Le prédicat $X \in [T_e]$ est quant à lui $\Pi_1^0(X)$, et donc $\Pi_1^0(M)$. Ainsi, l'ensemble des codes d'arbres mal fondés est calculable en M'' . On en déduit $\mathcal{O} \leq_T M''$. ■

Nous terminons ce chapitre en montrant finalement que les modèles de ATR_0 ou $\Pi_1^1\text{-CA}_0$ ne sont pas nécessairement des β -modèles.

Théorème 5.8

Il existe des modèles de $\Pi_1^1\text{-CA}_0$ ne contenant pas $\mathcal{O}$.

PREUVE. Il suffit de considérer la classe des ensembles X qui codent pour des modèles dénombrables de $\Pi_1^1\text{-CA}_0$. Les quantifications universelles ou existentielles du second ordre se transforment en quantifications du premier ordre sur les éléments $X_1 \oplus X_2 \oplus \dots$ qui sont codés par X . Il s'agit donc d'une classe arithmétique, et en particulier Σ_1^1 . Comme il existe des modèles dénombrables de $\Pi_1^1\text{-CA}_0$, cette classe est non vide. D'après le théorème 30-2.2, elle contient un élément X tel que $\omega_1^X = \omega_1^{ck}$. Il s'ensuit que X code pour un modèle dénombrable de $\Pi_1^1\text{-CA}_0$ ne pouvant pas contenir $\mathcal{O}$.

Corollaire 5.9

Il existe des modèles de $\Pi_1^1\text{-CA}_0$ — et donc aussi de ATR_0 — qui ne sont pas des β -modèles.

PREUVE. Le modèle du théorème précédent ne peut pas être un β -modèle, d'après la proposition 5.3. ■

Correction des exercices

Chapitre 2

Solution 3.4. On définit $g : \mathbb{N} \rightarrow \mathbb{N}$ de la manière suivante : $g(0)$ est le plus petit entier x tel que $f(x) \in B$. Supposons $g(n)$ défini. Alors, on définit $g(n+1)$ comme étant le plus petit entier x strictement plus grand que $g(n)$ et tel que $f(x) \in B$. Comme B est infini alors g est bien défini partout. On définit alors

$$h(n) = f(g(n)).$$

Par construction, h est bijective.

Solution 3.7.

1. Montrons que $(x_1, y_1) \neq (x_2, y_2)$ implique $\alpha_2(x_1, y_1) \neq \alpha_2(x_2, y_2)$. Supposons $(x_1, y_1) \neq (x_2, y_2)$. Supposons d'abord que $x_1 + y_1 < x_2 + y_2$ (le cas $x_2 + y_2 < x_1 + y_1$ étant symétrique). Alors,

$$\begin{aligned} & (\sum_{i=0}^{x_1+y_1} i) + x_1 + y_1 + 1 \leq \sum_{i=0}^{x_2+y_2} i \\ \rightarrow & y_1 + \sum_{i=0}^{x_1+y_1} i < \sum_{i=0}^{x_2+y_2} i \\ \rightarrow & y_1 + \sum_{i=0}^{x_1+y_1} i < y_2 + \sum_{i=0}^{x_2+y_2} i \\ \rightarrow & \alpha_2(x_1, y_1) < \alpha_2(x_2, y_2). \end{aligned}$$

Supposons à présent que $x_1 + y_1 = x_2 + y_2$. Notez que l'on a alors $y_1 = y_2$ implique $x_1 = x_2$. Comme $(x_1, y_1) \neq (x_2, y_2)$, alors nécessairement $y_1 \neq y_2$.

On a donc :

$$\begin{aligned} & (\sum_{i=0}^{x_1+y_1} i) = \sum_{i=0}^{x_2+y_2} i \\ \rightarrow & y_1 + (\sum_{i=0}^{x_1+y_1} i) \neq y_2 + \sum_{i=0}^{x_2+y_2} i \\ \rightarrow & \alpha_2(x_1, y_1) \neq \alpha_2(x_2, y_2). \end{aligned}$$

Montrons à présent que α_2 est surjective. Pour cela, montrons d'abord que :

$$\begin{aligned}\alpha_2(x-1, y+1) - \alpha_2(x, y) &= 1 \quad \text{pour } x > 0 \\ \alpha_2(y+1, 0) - \alpha_2(0, y) &= 1.\end{aligned}$$

Dans le premier cas, on a :

$$\begin{aligned}\alpha_2(x-1, y+1) - \alpha_2(x, y) &= (y+1 + \sum_{i=0}^{x-1+y+1} i) - (y + \sum_{i=0}^{x+y} i) \\ &= (y+1 + \sum_{i=0}^{x+y} i) - (y + \sum_{i=0}^{x+y} i) \\ &= 1.\end{aligned}$$

Dans le deuxième cas, on a :

$$\begin{aligned}\alpha_2(y+1, 0) - \alpha_2(0, y) &= (\sum_{i=0}^{y+1} i) - (y + \sum_{i=0}^y i) \\ &= (y+1 + \sum_{i=0}^y i) - (y + \sum_{i=0}^y i) \\ &= 1.\end{aligned}$$

Montrons à présent par récurrence que pour tout n , il existe (x, y) tel que

$$\alpha_2(x, y) = n.$$

On a bien $\alpha_2(0, 0) = 0$.

Supposons à présent que $\alpha_2(x, y) = n$. Si jamais $x > 0$, alors on a donc

$$\alpha_2(x-1, y+1) = n+1.$$

Si jamais $x = 0$, on a donc $\alpha_2(y+1, 0) = n+1$. On en déduit que α_2 est surjective.

2. Trivial.

Solution 3.9. On vérifie sans peine que $f : \mathbb{N} \rightarrow \mathbb{Z}$ donnée par $f(2n) = n$ et $f(2n+1) = -n-1$ est une bijection. On en déduit à l'aide du corollaire 3.8 que $\mathbb{Z} \times \mathbb{Z}$ est dénombrable.

On dispose d'une injection de $\mathbb{Q}$ vers $\mathbb{Z} \times \mathbb{Z}$ en assignant à $r \in \mathbb{Q}$ le couple (p, q) avec $p \in \mathbb{Z}$ et $q \in \mathbb{N}^*$, et où p/q est la fraction réduite égale à r . On en déduit donc que $|\mathbb{Q}| \leq |\mathbb{Z} \times \mathbb{Z}|$. Comme $\mathbb{Z} \times \mathbb{Z}$ est dénombrable, on a aussi $|\mathbb{Z} \times \mathbb{Z}| \leq |\mathbb{N}|$. Comme $\mathbb{Q}$ est infini, $|\mathbb{N}| \leq |\mathbb{Q}|$, ce qui donne $|\mathbb{Q}| = |\mathbb{N}|$ par le théorème 2.3 de Cantor-Bernstein.

Solution 3.10. Soit $g : A \rightarrow \mathbb{N}$ la fonction qui à $y \in A$ associe le plus petit élément n tel que $f(n) = y$. La fonction g est injective, car si $x \neq y \in A$, alors $\{n : f(n) = x\} \cap \{n : f(n) = y\} = \emptyset$, donc $g(x) \neq g(y)$. Ainsi, A est subpotent à $\mathbb{N}$. Comme A est infini, par la proposition 3.2, $\mathbb{N}$ est subpotent à A , donc par le théorème 2.3 de Cantor-Bernstein, A et $\mathbb{N}$ sont équipotents.

Question cachée : la proposition 3.2 utilise l'axiome du choix. Comment peut-on résoudre l'exercice de manière constructive, sans faire appel à la proposition 3.2 ?

Solution 3.11. On définit la fonction $h : \mathbb{N} \times \mathbb{N} \rightarrow B$ par $h(n, m) = f_n(m)$. La fonction est clairement surjective (bijective si les ensembles B_n sont deux à deux disjoints). Par la proposition 3.5, il existe une bijection $b : \mathbb{N} \rightarrow \mathbb{N} \times \mathbb{N}$. La fonction $h \circ b : \mathbb{N} \rightarrow B$ est donc une fonction surjective. Comme B est infini, l'exercice précédent permet de conclure.

Chapitre 3

Solution 2.3. D'après l'exercice 2-3.7, la bijection de couplage et ses inverses peuvent se calculer de la manière suivante.

```
int couplage(int a, int b) {
    return (a+b+1)(a+b)/2;
}

int invere1(int c) {
    for (int i = 0; i <= c; i = i + 1)
        for (int j = 0; j <= c; j = j + 1)
            if (couplage(i,j) == c)
                return i;
}

int invere2(int c) {
    for (int i = 0; i <= c; i = i + 1)
        for (int j = 0; j <= c; j = j + 1)
            if (couplage(i,j) == c)
                return j;
}
```

Solution 2.4.

```
int A(int a, int b) {
    // Fonction pour calculer A
}

int ins1(int x, int y) {
    for (int i = 0; i < y; i = i + 1)
        if (A(x, i) == 0)
            return 0;
    return 1;
}

int ins2(int x, int y) {
    for (int i = 0; i < y; i = i + 1)
        if (A(x, i) == 1)
            return 1;
    return 0;
}
```

Solution 6.4.

1. Soient $x_1 \in A$ et $x_2 \notin A$. On définit f par $f(x) = x_2$ si $x \in A$, et $f(x) = x_1$ sinon.
2. D'après le théorème du point fixe, il existe e tel que $\Phi_{f(e)} = \Phi_e$.
3. Trivial.
4. Trivial.

Solution 7.3. Soit A un ensemble calculable. Alors, on définit le code d'une fonction e qui s'arrête sur son entrée x si $x \in A$, et qui fait une boucle infinie si $x \notin A$.

Solution 7.9. On crée le programme qui sur une entrée n , énumère A jusqu'à ce qu'un élément plus grand que n soit énuméré dans A . Comme A est infini, cela arrivera nécessairement. À ce moment, si n fait partie des éléments énumérés dans A , alors $n \in A$, sinon $n \notin A$ (car seuls des éléments strictement supérieurs à n seront énumérés par la suite).

Solution 7.10. Soit A est un ensemble c. e. infini. On définit un ensemble c. e. B vérifiant $B \subseteq A$ et que l'on énumère dans l'ordre, de la manière suivante : on énumère x_0 dans B pour x_0 le premier élément énuméré dans A . Supposons que $x_0 < x_1 < \dots < x_n$ aient été énumérés dans B , alors l'élément x_{n+1} énuméré dans B sera le prochain élément énuméré dans A qui est plus grand que x_{n+1} . Comme A est infini, cela arrive forcément.

Solution 7.11. À l'étape de calcul t , pour tout $e \leq t$ tel que e n'est énuméré ni dans A ou B , si $\Phi_e(e)[t] \downarrow \in \{0, 1\}$, alors on énumère e dans A si $\Phi_e(e)[t] \downarrow = 0$ et l'on énumère e dans B si $\Phi_e(e)[t] \downarrow = 1$. Ainsi, si Φ_e calcule un ensemble C , alors soit $A \not\subseteq C$, soit $C \cap B \neq \emptyset$.

Solution 7.12. Soit C un ensemble c. e. Nous allons énumérer dans l'ordre deux ensembles A, B (qui seront donc calculables) tels que $x \in C$ ssi $2^x \in D_{A,B}$.

Supposons A_t, B_t énumérés à l'étape t . Soit x un nouvel élément énuméré dans C à l'étape $t+1$. Soit k le plus petit entier tel que $2^k > x$ et tel que 2^k est plus grand que tous les éléments de A_t et B_t . On énumère alors 2^{k+1} dans A et $2^{k+1} - x$ dans B . On a bien $2^{k+1} - (2^{k+1} - x) = x$ dans l'ensemble $D_{A,B}$. De plus, les autres éléments rajoutés dans $D_{A,B}$ à cette occasion sont de la forme $2^{k+1} - b$ pour $b \in B_t$ et de la forme $a - (2^{k+1} - x)$ pour $a \in A_t$. Comme $b < 2^k$, on a $2^k < 2^{k+1} - b < 2^{k+1}$. Donc, $2^{k+1} - b$ n'est pas une puissance de 2. Comme $a, x < 2^k$, alors $a - (2^{k+1} - x)$ est négatif, et n'appartient donc pas à $D_{A,B}$. Les seuls éléments de $D_{A,B}$ qui sont des puissances de 2 encodent alors les éléments de C . Donc, $D_{A,B}$ permet de calculer C , et donc si $D_{A,B}$ était calculable, C le serait aussi.

Solution 7.13. Montrons que le processus uniforme donné dans l'indication existe. Si W_e énumère un premier élément a_0 , alors on énumère $[0, a_0[$ dans W_d .

Puis, pour tout n , si W_e énumère un élément $a_{n+1} > a_n$, on énumère $]a_n, a_{n+1}[$ dans W_d . Cela termine la description de W_d . Si W_e est fini, l'ensemble W_d est fini aussi, et donc de complémentaire infini. Si W_e est infini, la suite $(a_n)_{n \in \mathbb{N}}$ telle que $a_n < a_{n+1}$ est bien définie, et l'ensemble W_d consiste alors exactement en le complémentaire de cette suite (le complémentaire de W_d est donc bien infini). Il est clair que si $W_d \subseteq W_e$ et W_e est infini, alors $W_e = \mathbb{N}$.

On dira que « $\mathcal{R}_e$ est satisfait » si l'ensemble W_d énuméré vérifie bien

$$W_d \subseteq W_e \rightarrow (|\mathbb{N} \setminus W_e| < \infty \vee |W_e \setminus W_d| < \infty).$$

La difficulté consiste à présent en la coordination de différentes stratégies afin que chaque $\mathcal{R}_e$ soit satisfait pour *le même ensemble* W_d . L'idée est la suivante : on effectue bien le processus décrit plus haut pour construire W_d de manière à satisfaire $\mathcal{R}_0$. Ce processus énumère une suite d'entiers $a_0^0 < a_1^0 < \dots <$ qui va constituer le complémentaire de W_d pour le moment. Il y a deux possibilités.

▷ Cas 1. La suite $(a_n^0)_{n \in \mathbb{N}}$ est infinie, auquel cas on satisfait $\mathcal{R}_1$ de la même manière en considérant $(a_n^0)_{n \in \mathbb{N}}$ comme notre nouvel espace de travail.

▷ Cas 2. Il existe un élément a_k^0 maximal, auquel cas on peut satisfaire $\mathcal{R}_1$ en considérant $\{m : m > a_k^0\}$ comme notre nouvel espace de travail. Le problème est que l'on doit satisfaire $\mathcal{R}_1$ uniformément, et que l'on ne sait pas si la satisfaction de $\mathcal{R}_0$ va déboucher sur le cas « infini » ou sur le cas « fini ». L'idée est alors de traiter les deux possibilités simultanément. Pour satisfaire $\mathcal{R}_1$, on considère la possibilité « finie », et ainsi pour chaque a_k^0 qui est le plus grand élément pour le moment, on construit pour satisfaire $\mathcal{R}_1$ des éléments $a_1^1 < a_1^1 < \dots < a_n^1$ à l'intérieur de $\{m : m > a_k^0\}$. Notons que cela impacte le travail pour satisfaire $\mathcal{R}_0$, car la satisfaction de $\mathcal{R}_1$ contribue aussi à l'énumération de W_d ; en pratique, ce n'est pas un problème. Dans le même temps, on va considérer pour satisfaire $\mathcal{R}_1$ la possibilité « infinie » et tenter, à mesure de l'énumération de $a_0^0 < a_1^0 < \dots < a_n^0$, de travailler dans cet espace. Il faut alors itérer cette idée pour satisfaire chaque $\mathcal{R}_e$.

La formalisation de l'argument ci-dessus se fait naturellement via une méthode de priorité avec blessures infinies, comme exposé de manière détaillée dans le chapitre 13. Toutefois, pour cette construction précise, il est possible de donner un algorithme étonnamment simple qui implémente « automatiquement », ce qui a été expliqué. Étant donné x , on note $\sigma_e(x)[s]$ le nombre $\sum_{i \leq e, x \in W_i[s]} 2^{e-i}$. L'idée est que $\sigma_e(x)[s]$ est un poids que l'on donne à x , qui augmente à mesure que x appartient à de plus en plus d'ensembles W_i pour $i \leq e$. De plus, l'appartenance de x à W_i a plus de valeur que l'appartenance de x à l'ensemble des W_j pour $i < j \leq e$. L'algorithme est le suivant. À l'étape 0, soit $a_n^0 = n$ pour tout n . À l'étape $s+1$, on définit a_0^{s+1} comme le plus petit entier $x \in [0, s+1] \cap \overline{W_d}$ tel que $\sigma_e(x)[s]$ est maximal — ici, $\overline{W_d}$ dénote le complémentaire de W_d à cette étape de calcul — et l'on énumère $[0, a_0^{s+1}[$ dans W_d . Puis, inductivement, si a_e^{s+1} est défini, on définit a_{e+1}^{s+1} comme le plus petit entier $x \in]a_e^{s+1}, s+1] \cap \overline{W_d}[s]$, s'il existe, tel que $\sigma_e(x)[s]$ est maximal, puis on énumère $]a_e^{s+1}, a_{e+1}^{s+1}[$ dans W_d . Cela conclut la construction.

Notons que, pour tout e, x , on a $\sigma_e(x)[s] \leq \sigma_e(x)[s+1]$. Il y a par ailleurs e valeurs possibles pour $\sigma_e(x)[s]$. Donc, $\lim_s \sigma_e(x)[s]$ existe. On en déduit que $a_0 = \lim_s a_0^s$ existe aussi, et par induction que $a_e = \lim_s a_e^s$ existe pour tout e . Il est alors clair que W_d est le complémentaire de $(a_e)_{e \in \mathbb{N}}$, et que $\overline{W_d}$ est donc infini. Supposons par l'absurde qu'il existe e tel que $W_d \subseteq W_e$, $|W_e \setminus W_d| = \infty$ et $|\mathbb{N} \setminus W_e| = \infty$. Soit e le plus petit de ces entiers. En particulier, il existe un entier a_n à partir duquel pour tout $i < e$, soit $a_m \in W_i$ pour tout $m > n$, soit $a_m \notin W_i$ pour tout $m > n$. Soit $a_{m_3} > a_{m_2} > a_{m_1} > a_n$ tels que $a_{m_2} \notin W_e$ et $a_{m_1}, a_{m_3} \in W_e$. Alors, $\sigma_{m_2}(a_{m_2}) < \sigma_{m_2}(a_{m_3})$ (souvenons-nous que l'appartenance de a_{m_3} à W_e a plus de poids que son appartenance ou non à W_i pour $e < i \leq m_2$). En particulier, à un certain moment on devrait énumérer $]a_{m_1}, a_{m_3}[$ dans W_d , ce qui est une contradiction.

Chapitre 4

Solution 4.3. Si $Y(n) = \Phi_a(X, n)$ pour tout n et $Z(n) = \Phi_b(Y, n)$ pour tout n , alors on peut créer la fonctionnelle Φ_e qui à partir de l'oracle X , cherche pour tout entier n une chaîne σ telle que $\sigma(i) = \Phi_a(X, i)$ pour $i < |\sigma|$ et telle que $\Phi_b(\sigma, n) \downarrow$.

Solution 5.3. Trivial d'après l'exercice 4.3.

Solution 5.4. Supposons $Y =^* X$. Pour calculer Y à partir de X , il suffit d'encoder les variations de $X \ll$ en dur $\gg$ dans un programme (une suite finie d'instructions), qui sur l'oracle X va répéter en sortie son oracle, sauf pour le nombre fini d'éléments spécifiés dans le programme lui-même.

Solution 5.8. Soit e_1, e_2 tels que $\Phi_{e_1}(Y, n) \downarrow = X(n)$ et $\Phi_{e_2}(B, n) \downarrow = A(n)$. Alors, on définit la fonctionnelle Φ_e telle que $\Phi_e(Z_0 \oplus Z_1, 2n) = \Phi_{e_1}(Z_0, n)$ et telle que $\Phi_e(Z_0 \oplus Z_1, 2n+1) = \Phi_{e_2}(Z_1, n)$.

Solution 6.2. Soit Ψ la fonctionnelle telle que $\Psi(Y, n) \downarrow = X(n)$ pour tout n . On note $\Psi(Y)$ pour l'ensemble $\{n \in \mathbb{N} : \Psi(Y, n) \downarrow = 1\}$. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ la fonction calculable telle que $\Phi_{f(n)}(Y, m) = \Phi_n(\Psi(Y), n) \downarrow$ pour tout m . En particulier, $f(n) \in Y'$ ssi $n \in X'$. Donc, $X' \leq_T Y'$.

Solution 6.4. Soit $g : \mathbb{N} \rightarrow \mathbb{N}$ définie comme dans la preuve de la proposition 6.3. La fonctionnelle Φ_e , sur l'oracle Z et l'entrée n , regarde simplement si $g(n) \in Z$. Si c'est le cas, alors elle renvoie 1, sinon elle renvoie 0.

Solution 7.5. Clairement, $\emptyset''$ est $\emptyset'$ -c.e. et $\emptyset'$ est $\emptyset$ -c.e., mais $\emptyset''$ n'est pas est $\emptyset$ -c.e. car tout ensemble $\emptyset$ -c.e. est calculable à la limite et donc calculable en $\emptyset'$, mais d'après la proposition 6.3 $\emptyset''$ n'est pas calculable en $\emptyset'$.

Solution 7.8. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $f(n) \geq \mu_A(n)$. Pour calculer $A(n)$, on lance l'énumération de A jusqu'à l'étape $f(n)$. On a alors $A_{f(n)}(n) = A(n)$.

Solution 10.5. Il suffit de répéter la même preuve, mais en changeant la condition (P2) comme suit :

$$\forall (x, y) \in \text{dom } h \setminus \text{dom } g, \quad x < m \Rightarrow h(x, y) = B(x).$$

Solution 10.6. La construction de la proposition 10.2 est calculable en $\emptyset''$; on peut toutefois utiliser le fait que $\emptyset''$ soit calculatoirement énumérable en $\emptyset'$ pour que la construction devienne effective en $\emptyset'$. À une étape t , on satisfait le contrat S_e à partir d'un couple (g, e) pour une fonction partielle g , sur la base de ce que l'on croit être le préfixe correct de $\emptyset''$ de taille e à l'étape t .

Quand $\emptyset'$ détecte que k est énuméré dans $\emptyset''$ pour un certain k , les contrats S_e pour $e \geq k$ deviennent caduques, car ils étaient satisfaits sur la base d'un mauvais préfixe. On peut toutefois re-satisfaire ces contrats autant de fois que nécessaire, en utilisant le fait que l'énumération de $\emptyset''$ converge petit à petit.

Solution 10.7. Il suffit de répéter la preuve de la proposition 8.1, mais avec une quantité dénombrable d'ensembles : supposons qu'à l'étape t on ait défini des préfixes $\sigma_{0,t}, \dots, \sigma_{t,t}$ de $A_0, \dots, A_t$, tout en s'assurant qu'aucune fonctionnelle Φ_e pour $e \leq t$ ne puisse calculer $\sigma_{i,t}$ à partir de $\sigma_{j,t}$ pour $i \neq j \leq t$. À l'étape $t+1$, on définit des extensions $\sigma_{i,t+1}$ de chaque $\sigma_{i,t}$, ainsi qu'une extension $\sigma_{t+1,t+1}$ du mot vide, de manière à ce que pour tout $e \leq t+1$, la fonctionnelle Φ_e ne puisse calculer $\sigma_{i,t+1}$ à partir de $\sigma_{j,t+1}$ pour $i \neq j \leq t+1$. Les extensions se font petit à petit en diagonalisant sur les fonctionnelles Φ_e pour $e \leq t+1$ et sur tous les couples (i, j) possibles pour $i, j \leq t+1$.

Chapitre 5

Solution 1.8. Il n'existe qu'un nombre dénombrable d'ensembles dans la hiérarchie arithmétique, donc il existe des ensembles non Σ_n^0 . Soit A un tel ensemble. En particulier $A = \bigcup_{n \in A} \{n\}$. Chaque singleton étant calculable, et *a fortiori* Σ_n^0 , A est une réunion dénombrable d'ensembles Σ_n^0 , mais n'est pas Σ_n^0 .

Solution 5.7. Pour la direction $X \equiv_T Y \rightarrow X' \equiv_m Y'$, il suffit de reprendre l'exercice 4-6.2 et de s'apercevoir que la réduction définie est en fait une réduction many-one.

Supposons à présent $X' \geq_m Y'$. Soit $e_{n,0}$ et $e_{n,1}$ les codes des programmes qui s'arrêtent sur leur entrées si respectivement $Z(n) = 0$ ou $Z(n) = 1$ pour l'oracle courant Z . En particulier, pour n'importe quel oracle Z on a $Z'(e_{n,0}) \neq Z'(e_{n,1})$ et au moins une des deux valeurs est à 1. À présent, à partir de X et de la fonction f telle que $x \in Y'$ ssi $f(x) \in X$, sur une entrée n , on énumère X'

jusqu'à trouver $f(e_{n,0}) \in X'$ ou $f(e_{n,1}) \in X'$ si le premier événement arrive alors $Y(n) = 0$ et sinon $Y(n) = 1$.

Solution 7.2. On a $e \in \text{TOT}$ ssi $\forall n \exists t \Phi_e(n)[t] \downarrow$, ce qui est un prédicat Π_2^0 . Soit à présent un ensemble Π_2^0

$$A = \{n \in \mathbb{N} : \forall x_1 \exists x_2 R(n, x_1, x_2)\},$$

où R est un prédicat calculable. Étant donné n , on définit la fonctionnelle e_n telle que $\Phi_{e_n}(m)$ s'arrête si $\exists t R(n, m, t)$. Il est clair que $e_n \in \text{TOT}$ ssi $n \in A$.

Solution 7.5. Si c'était le cas, l'arrêt serait calculable.

Solution 7.9. Trivial.

Solution 7.10. Il suffit de créer à l'aide du théorème du point fixe le code e d'une fonctionnelle qui calcule un ensemble X en ne choisissant que des 0 pour les premiers bits de X , jusqu'à ce que $f(e)$ renvoie une valeur : si cette valeur est 0 le code e décide de mettre un des bits encore non calculé à 1, sinon il continue de ne mettre que des 0. Plus formellement, soit g la fonction calculable telle que :

1. $\Phi_{g(e)}(t) = 1$, si t est le plus petit tel que $f(e)[t] \downarrow$ et si $f(e)[t] \downarrow = 0$;
2. $\Phi_{g(e)}(t) = 0$, sinon.

D'après le théorème du point fixe, il existe un code a tel que $\Phi_a = \Phi_{g(a)}$. Notons que $g(e)$ est toujours le code Δ_1^0 d'un ensemble fini. C'est donc le cas pour $g(a)$, et donc aussi pour a . Par hypothèse sur f , on doit donc avoir $f(a) \downarrow = n$ pour un certain n . Si $n = 0$, alors e code pour un ensemble à un élément. Dans le cas contraire, e code pour l'ensemble vide.

Solution 7.11. Soit b un entier tel que, pour tout oracle X , on a $\Phi_b(X, b) \downarrow$ ssi X est non vide, et tel que $\Phi_b(X, b) \uparrow$ sinon. On définit la fonction $g : \mathbb{N} \rightarrow \mathbb{N}$ telle que :

1. $\Phi_{g(e)}(\emptyset', t) = 0$ si $\Phi_{f(e)}(\emptyset', b)[t] \uparrow$;
2. $\Phi_{g(e)}(\emptyset', t) = 1$ si $\Phi_{f(e)}(\emptyset', b)[t] \downarrow = 0$;
3. $\Phi_{g(e)}(\emptyset', t) = 0$ si $\Phi_{f(e)}(\emptyset', b)[t] \downarrow = 1$.

D'après le théorème du point fixe, il existe un code a tel que $\Phi_{g(a)} = \Phi_a$. Notons que $g(e)$ code toujours pour une fonction totale en $\emptyset'$, et est donc un code Δ_2^0 . Par conséquent, $g(a)$ et a le sont également. Notons enfin que l'ensemble de code a est soit vide, soit n'a qu'un nombre fini de 0. Il est donc calculable, et en particulier low. Donc, par hypothèse sur f , $f(a)$ est un code Δ_2^0 , et donc $\Phi_{f(a)}$ est totale sur l'oracle $\emptyset'$ et il existe t tel que $\Phi_{f(a)}(\emptyset', b)[t] \downarrow = i$. Si $i = 0$, alors a code pour un ensemble X non vide, et l'on devrait avoir $X'(b) = 1 \neq \Phi_{f(a)}(\emptyset', b)$. Si $i = 1$, alors a code pour un ensemble X vide, et $X'(b) = 0 \neq \Phi_{f(a)}(\emptyset', b)$ comme il se devrait.

Exercice caché : donner à présent une description informelle de la procédure décrite ci-dessus, comme celle de la solution précédente.

Chapitre 6

Solution 3.10. Soit m tel que notre programme utilise au plus les registres

$$R_0, \dots, R_m.$$

Il suffit de recopier au début du programme les registres de R_1 à R_m dans les registres de R_{m+1} à R_{m+m} . Ensuite, le programme est le même en remplaçant tous les registres R_i par R_{i+m} pour $i > 0$, puis enfin en ajoutant des instructions à la fin qui remettent à zéro tous les registres de R_{m+1} à R_{m+m} .

Solution 3.12. L'addition est définie en utilisant les schémas de composition et récursion primitive. La multiplication et l'exponentielle en utilisant le schéma de récursion primitive.

Addition :

$$\begin{aligned} \text{add}(a, 0) &= p_1^2(a, 0) \\ \text{add}(a, b+1) &= \text{succ}(p_2^2(a, \text{add}(a, b))) \end{aligned}$$

Multiplication :

$$\begin{aligned} \text{mult}(a, 0) &= 0 \\ \text{mult}(a, b+1) &= \text{add}(a, \text{mult}(a, b)) \end{aligned}$$

Exponentielle :

$$\begin{aligned} \text{exp}(a, 0) &= 1 \\ \text{exp}(a, b+1) &= \text{mult}(a, \text{exp}(a, b)) \end{aligned}$$

Solution 3.13. Les fonctions sont définies en utilisant le schéma de récursion primitive.

Prédécesseur :

$$\begin{aligned} \text{pred}(0) &= 0 \\ \text{pred}(n+1) &= p_1^2(n, \text{pred}(n)) \end{aligned}$$

Soustraction :

$$\begin{aligned} -(a, 0) &= p_1^2(a, 0) \\ -(a, b+1) &= \text{pred}(p_2^2(a, -(a, b))) \end{aligned}$$

Solution 3.14. Les fonctions sont définies à l'aide du schéma de récursion primitive.

sg :

$$\begin{aligned} \text{sg}(0) &= 0 \\ \text{sg}(x+1) &= c_1^2(x, \text{sg}(x)) \end{aligned}$$

$\overline{\text{sg}}$:

$$\begin{aligned} \overline{\text{sg}}(0) &= 1 \\ \overline{\text{sg}}(x+1) &= c_0^2(x, \overline{\text{sg}}(x)) \end{aligned}$$

Solution 3.19. Soient $P_1, P_2 \subseteq \mathbb{N}^n$ des prédicats primitifs rékursifs. Alors,

$$\begin{aligned} P_1(a_1, \dots, a_n) \wedge P_2(a_1, \dots, a_n) &= P_1(a_1, \dots, a_n) \times P_2(a_1, \dots, a_n) \\ P_1(a_1, \dots, a_n) \vee P_2(a_1, \dots, a_n) &= \text{sg}(P_1(a_1, \dots, a_n) + P_2(a_1, \dots, a_n)) \\ \neg P_1(a_1, \dots, a_n) &= \overline{\text{sg}}(P_1(a_1, \dots, a_n)). \end{aligned}$$

Soit $P \subseteq \mathbb{N}^{n+1}$ un prédicat primitif récursif. Alors,

$$\begin{aligned}\exists y \leq z \ P(x_1, \dots, x_n, y) &= Q_1(x_1, \dots, x_n, z) \\ \forall y \leq z \ P(x_1, \dots, x_n, y) &= Q_2(x_1, \dots, x_n, z)\end{aligned}$$

où

$$\begin{aligned}Q_1(x_1, \dots, x_n, 0) &= P(x_1, \dots, x_n, 0) \\ Q_1(x_1, \dots, x_n, z+1) &= P(x_1, \dots, x_n, \text{succ}(z)) \vee Q_1(x_1, \dots, x_n, z) \\ &\text{et} \\ Q_2(x_1, \dots, x_n, 0) &= P(x_1, \dots, x_n, 0) \\ Q_2(x_1, \dots, x_n, z+1) &= P(x_1, \dots, x_n, \text{succ}(z)) \wedge Q_2(x_1, \dots, x_n, z).\end{aligned}$$

Solution 3.20. On utilise le schéma de récursion primitive, la définition par cas et la clôture des prédicats primitifs récursifs par conjonction et quantification bornée :

$$\begin{aligned}g(x_1, \dots, x_p, 0) &= 0 \\ g(x_1, \dots, x_p, n+1) &= \text{succ}(n) && \text{si } f(x_1, \dots, x_p, \text{succ}(n)) = 0 \\ & && \text{et } \forall t \leq n \ f(x_1, \dots, x_p, t) \neq 0 \\ &= g(x_1, \dots, x_p, n) && \text{sinon.}\end{aligned}$$

Solution 3.23. L'ensemble A peut être défini de manière calculable par récurrence. On pose $A_0 = \{e_0\}$, où e_0 est le code de la fonction successeur. Supposons A_n défini avec $n = \langle k, a \rangle$. Alors, A_{n+1} est l'ensemble A_n auquel on ajoute un code pour toutes les possibilités d'application du schéma de composition et de récursion primitive des codes de A_n , ainsi que des codes pour la fonction constante $f(x_1, \dots, x_k) = a$ et pour la projection $f(x_1, \dots, x_k) = x_a$ dans le cas où $a \leq k$. On prendra bien garde à ce que les nouveaux codes de A_{n+1} soient tous supérieurs aux codes de A_n , via le lemme de remplissage. L'ensemble $A = \bigcup_n A_n$ est alors calculable.

Par diagonalisation, en utilisant le fait qu'une fonction primitive récursive est toujours totale; on calcule alors une fonction Φ_e qui n'a pas de code dans A . Soient $a_0 < a_1 < a_2 < \dots$ les éléments de A codant pour des fonctions prenant exactement un paramètre. On peut alors par exemple définir $\Phi_e(n) = 0$ si $\Phi_{a_n}(n) \neq 0$, et $\Phi_e(n) = 1$ sinon. Notons qu'il est nécessaire pour tout n de lancer l'exécution de $\Phi_{a_n}(n)$ jusqu'à ce qu'elle s'arrête (ce qui arrive nécessairement car a_n code pour une fonction primitive récursive).

Solution 3.25.

- (1) On a $A_0(x) = 2^x > x$, donc (1) est vrai pour A_0 . Supposons (1) pour A_n , et montrons (1) pour A_{n+1} . On procède par récurrence sur x . On a $A_{n+1}(0) = 1 > 0$. Supposons $A_{n+1}(x) > x$ pour x . Alors, $A_{n+1}(x+1) = A_n(A_{n+1}(x))$. Comme (1) est vrai pour A_n , alors $A_n(A_{n+1}(x)) > A_{n+1}(x)$. Par hypothèse de récurrence, $A_{n+1}(x) > x$. Donc, $A_{n+1}(x+1) > x+1$.
- (2) La fonction A_0 est clairement strictement croissante. Ensuite, pour tout n , on a $A_{n+1}(x+1) = A_n(A_{n+1}(x))$. En utilisant (1), on a $A_n(A_{n+1}(x)) > A_{n+1}(x)$, et donc $A_{n+1}(x+1) > A_{n+1}(x)$. Donc, pour tout n , la fonction A_{n+1} est strictement croissante.

- (3) Pour $x=0$, pour tout n on a $A_{n+1}(0)=1$ et $A_n(0)=1$. Donc, $A_{n+1}(0) \geq A_n(0)$.
 Pour $x > 0$, on a $A_{n+1}(x) = A_n(A_{n+1}(x-1))$.

D'après (1), on a $A_{n+1}(x-1) > x-1$, donc $A_{n+1}(x-1) \geq x$. Aussi, comme la fonction A_n est croissante, on a $A_n(A_{n+1}(x-1)) \geq A_n(x)$. On a donc $A_{n+1}(x) \geq A_n(x)$, ce qui implique que la fonction $n \mapsto A_n(x)$ est croissante.

- (4) Par récurrence sur y . Pour $y=0$, on a $A_{n+1}(x) \geq x$, d'après (1).
 Supposons (4) pour y , afin de le montrer pour $y+1$. On a

$$A_{n+1}(x+y+1) = A_n(A_{n+1}(x+y)).$$

Par l'hypothèse de récurrence, $A_{n+1}(x+y) \geq A_n^{(y)}(x)$. On a donc

$$A_{n+1}(x+y+1) \geq A_n(A_n^{(y)}(x)) = A_n^{(y+1)}(x),$$

car A_{n+1} est croissante.

- (5) D'après (4), on a $A_{n+1}(2(x+1)) \geq A_n^{(x+1)}(x+1)$.

Il suffit de montrer $A_{n+1}(x) > A_n(2(x+1))$ pour presque tout x pour avoir (5). Par (3), pour tout $n \in \mathbb{N}$, on a $A_{n+1}(x) \geq A_0(x) = 2^x > 2x+3$ pour presque tout x . Par définition, pour tout $n \in \mathbb{N}$, on a

$$A_{n+1}(x) = A_n(A_{n+1}(x-1)) \geq A_n(2(x-1)+3),$$

pour presque tout x (car $n \mapsto A_n$ est croissante par (3)).

- (6) Pour les fonctions primitives récursives de base (projections, fonctions constantes et successeur), $P(f)$ est clairement vérifié. Supposons que l'on ait $P(h)$ et $P(g_1), \dots, P(g_n)$ pour les fonctions primitives récursives $h : \mathbb{N}^n \rightarrow \mathbb{N}$ et $g_1, \dots, g_n : \mathbb{N}^m \rightarrow \mathbb{N}$. Montrons $P(f)$ pour la fonction

$$f(x_1, \dots, x_m) = h(g_1(x_1, \dots, x_m), \dots, g_n(x_1, \dots, x_m)).$$

Il existe $k \in \mathbb{N}$ tel que

$$h(x_1, \dots, x_n) < A_k(\max(x_1, \dots, x_n)) \text{ et } g_i(x_1, \dots, x_m) < A_k(\max(x_1, \dots, x_m))$$

presque partout et pour tout $i \leq n$. Comme A_k soit croissante, on a

$$h(g_1(x_1, \dots, x_m), \dots, g_n(x_1, \dots, x_m)) \leq A_k(A_k(\max(x_1, \dots, x_m))).$$

D'après (5), pour presque tout $x_1, \dots, x_m$, on a

$$A_k^{(2)}(\max(x_1, \dots, x_m)) < A_{k+2}(\max(x_1, \dots, x_m)).$$

Donc, $P(f)$ est vérifié.

Supposons à présent que $P(h)$ et $P(g)$ soit vrais pour les fonctions primitives récursives $h : \mathbb{N}^{n+2} \rightarrow \mathbb{N}$ et $g : \mathbb{N}^n \rightarrow \mathbb{N}$. Montrons que $P(f)$ est vrai pour la fonction

$$\begin{aligned} f(x_1, \dots, x_n, 0) &= g(x_1, \dots, x_n) \\ f(x_1, \dots, x_n, x+1) &= h(x_1, \dots, x_n, x, f(x_1, \dots, x_n, x)). \end{aligned}$$

En particulier, il existe k tel que

$$g(x_1, \dots, x_n) < A_k(\max(x_1, \dots, x_n)) \text{ et } h(x_1, \dots, x_n, z, x) < A_k(\max(x_1, \dots, x_n, z, x))$$

pour presque tous $x_1, \dots, x_n, z, x$. On laisse au lecteur le soin de montrer par récurrence sur x que l'on a $f(x_1, \dots, x_n, x) < A_k^{(x+1)}(\max(x_1, \dots, x_n))$ pour

tout x . On a également

$$A_k^{(x+1)}(\max(x_1, \dots, x_n)) \leq A_k^{(\max(x_1, \dots, x_n, x+1))}(\max(x_1, \dots, x_n, x+1));$$

or,

$$A_k^{(\max(x_1, \dots, x_n, x+1))}(\max(x_1, \dots, x_n, x+1)) \leq A_{k+2}(\max(x_1, \dots, x_n, x))$$

pour presque tout $x_1, \dots, x_n, x$ (d'après (5)).

Ainsi, $\forall^\infty x \ f(x_1, \dots, x_n, x) < A_{k+2}(\max(x_1, \dots, x_n, x))$, et donc $P(f)$. On a dès lors $P(f)$ pour toute fonction primitive récursive f . On en déduit que la fonction d'Arckermann domine presque partout toutes les fonctions récursives.

Chapitre 7

Solution 1.5. Si un ensemble A n'est pas immune il contient un ensemble infini c. e. $x_0, x_1, \dots$. On peut supposer sans perte de généralité $x_i \neq x_j$ pour $i \neq j$ et en posant $F_i = \{x_i\}$ on obtient un tableau c. e. $F_0, F_1, \dots$ pour lequel $A \cap F_n \neq \emptyset$ pour tout n . Donc, A n'est pas hyperimmune.

Solution 2.2. Soit X un ensemble calculant une fonction DNC $f : \mathbb{N} \rightarrow \mathbb{N}$. Par le lemme de remplissage, il existe un ensemble infini calculable $A = \{e_0 < e_1 < \dots\}$ de codes de la fonction définie nulle part. Soit $g : \mathbb{N} \rightarrow \mathbb{N}$ la fonction définie par $g(x) = f(x)$ si $x \notin A$, et $g(e_n) = X(n)$ sinon. La fonction g est X -calculable. Inversement, on peut re-calculer X à partir de g en regardant les valeurs de g aux positions de A . Ainsi, g est dans le degré de X . Montrons que g est DNC. Si $x \notin A$, alors $g(x) = f(x) \neq \Phi_x(x)$. Si $x \in A$, alors Φ_x n'est définie nulle part, donc $g(x) \neq \Phi_x(x)$.

Solution 2.9. On construit un ensemble A comme limite de chaînes

$$\sigma_0 \prec \sigma_1 \prec \sigma_2 \prec \dots,$$

via une construction calculable en $\emptyset'$. La manière de rendre A non calculable est identique à celle de la preuve de la proposition 4-9.1.

Pour obtenir A non DNC, étant donné une fonctionnelle Φ et une chaîne σ_n , soit il existe $m \in \mathbb{N}$ tel que $\Phi(\sigma_n \tau, m) \uparrow$ pour toute chaîne τ , auquel cas $\Phi(A, m)$ sera partielle, soit pour tout m on a $\Phi(\sigma_n \tau, m) \downarrow$ pour une chaîne τ , auquel cas on peut calculer la fonction totale $f(m) = \Phi(\sigma_n \tau, m)$ pour la première chaîne τ trouvée telle que $\Phi(\sigma_n \tau, m) \downarrow$. Il existe alors a tel que $\Phi_a(a) \downarrow = f(a)$. On choisit $\sigma_{n+1} = \sigma_n \tau$ pour τ telle que $\Phi(\sigma_n \tau, a) \downarrow = f(a) = \Phi_a(a)$, et l'on aura $\Phi(A, a) \downarrow = \Phi_a(a) \downarrow$.

Solution 3.2. On procède comme dans l'indication. Soit il existe n tel que la fonction $m \mapsto g(a_{n,m})$ est DNC relativement à C , soit pour tout n , il existe m tel que $\Phi_m(C, m) \downarrow = g(a_{n,m})$. On définit alors la fonction $g \oplus C$ -calculable $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $f(n)$ renvoie le plus petit s tel que $\Phi_m(C, m)[s] \downarrow = g(a_{n,m})$ pour $m \leq s$. Supposons par l'absurde qu'il existe n tel que $\emptyset'(n) = 1$ et tel que $\emptyset'(n)[f(n)] = 0$. Alors, $g(a_{n,m}) = \Phi_{a_{n,m}}(a_{n,m})$, ce qui contredit que g est DNC. Donc, f borne le temps d'entrée de n dans $\emptyset'$.

Solution 4.2. Une direction est triviale. Pour l'autre direction, supposons qu'il existe une fonction calculable g et un entier y pour lesquels $g(x) > f(x)$ pour tout $x \geq y$. Alors, la fonction $g'(x)$ définie par $g'(x) = f(x) + 1$ si $x < y$, et par $g'(x) = g(x)$ sinon, est calculable et domine f partout. Donc, f n'est pas hyperimmune.

Solution 4.5. Soit $X \geq_T f$ avec f hyperimmune. Alors, $X \geq_T g$ avec $g : \mathbb{N} \rightarrow \mathbb{N}$ définie par $g(n) = \langle f(n), X(n) \rangle$. Comme on a $g > f$, alors g est hyperimmune, et il est clair que g calcule X .

Solution 4.6. Soit $(f_n)_{n \in \mathbb{N}}$ une énumération des fonctions calculables totales. On définit $\sigma_0 = \epsilon$. Supposons σ_n défini.

Alors, on définit σ_{n+1} comme étant $\sigma_n 0^{f_n(n)+1} 1$. L'ensemble $\{X\} = \bigcap_n [\sigma_n]$ est hyperimmune via la fonction qui à n associe le n -ième élément de X .

Solution 5.8. Supposons $T(\sigma)$ défini pour toute chaîne σ de taille n , telle que pour tout $e \leq n$ on a $\Phi_e(T(\sigma)) \not\leq T(\tau)$. Pour tous σ, τ de taille n . Soit $(\sigma_{j,0})_{j \leq 2^{n+1}}$ une énumération des chaînes de la forme $T(\sigma)i$ pour une chaîne σ de taille n et $i \in \{0, 1\}$. Soit $((a_j, b_j))_{j \in k}$ une énumération des couples distincts d'éléments inférieurs à 2^{n+1} . Supposons $(\sigma_{j,t})_{j \leq 2^{n+1}}$ défini à l'étape t . À l'étape $t+1$, on définit $\sigma_{a_{t+1}, t+1}$ et $\sigma_{b_{t+1}, t+1}$ comme étant des extensions de $\sigma_{a_{t+1}, t}$ et $\sigma_{b_{t+1}, t}$ telles que $\Phi_{n+1}(\sigma_{a_{t+1}, t+1}) \not\leq \sigma_{b_{t+1}, t+1}$. On définit $\sigma_{c, t+1} = \sigma_{c, t}$ pour $c \neq a_{t+1}$ et $c \neq b_{t+1}$. Une fois que tous les couples $((a_j, b_j))_{j \in k}$ ont été considérés, on définit finalement $T(\sigma i)$ comme étant la chaîne $\sigma_{j,k}$ qui étend $T(\sigma)$ pour $i \in \{0, 1\}$.

Solution 6.3. Montrons $(1) \Rightarrow (2)$. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ une fonction qui domine presque partout toute fonction calculable. On procède comme dans $(2) \Rightarrow (3)$ de le théorème 6.2, mais pour les fonctionnelles à valeur dans $\mathbb{N}$.

Montrons $(2) \Rightarrow (3)$. On transforme la liste $(f_n)_{n \in \mathbb{N}}$ en remplaçant chaque fonction f telle que $f(n) < f(n+1)$ par l'ensemble X tel que $f(n)$ est le n -ième élément de X .

Si pour une fonction f on s'aperçoit à un moment $f(n+1) \leq f(n)$, on complète alors l'ensemble X correspondant par un ensemble calculable infini fixé à l'avance.

Montrons $(3) \Rightarrow (1)$. Il suffit de calculer la fonction $g : \mathbb{N} \rightarrow \mathbb{N}$ définie par $g(n)$ comme étant la somme des n -ième éléments des ensembles X_i pour $i \leq n$.

Solution 6.5. Soit $f : \mathbb{N} \rightarrow \mathbb{N}$ la fonction X -calculable qui à n associe le plus petit m tel que $\mathbb{N} \setminus X$ contienne $2n$ éléments plus petits que m . Supposons par l'absurde qu'il existe une fonction calculable g telle que $\exists^\infty n \ g(n) > f(n)$. On peut supposer sans perte de généralité que $g(n) < g(n+1)$.

Montrons tout d'abord qu'il existe une infinité de valeurs n telles que $\mathbb{N} \setminus X$ contient au moins deux éléments dans $[g(n), g(n+1)[$. En effet, dans le cas contraire, soit n_0 le plus petit entier tel que pour tout $n \geq n_0$, $[g(n), g(n+1)[$ contienne au plus un élément de $\mathbb{N} \setminus X$. Soit k le nombre d'éléments de $\mathbb{N} \setminus X$

inférieurs à $g(n_0)$. Pour tout $n > n_0$, le nombre d'éléments de $\mathbb{N} \setminus X$ inférieurs à $g(n)$ est au plus $k+n-n_0$. Pour n suffisamment grand, on a $k+n-n_0 < 2n$, autrement dit il existe moins de $2n$ éléments plus petits que $g(n)$, donc $g(n) < f(n)$, ce qui contredit notre hypothèse.

Montrons maintenant qu'il existe un ensemble c. e. $Y \supseteq X$ tel que $Y \setminus X$ et $\mathbb{N} \setminus Y$ sont tous les deux infinis. On recopie X dans Y , et en plus de cela à chaque temps de calcul s on énumère dans Y le plus petit entier de l'intervalle $[g(n), g(n+1)[$ (pour $n \leq s$) qui n'est pas énuméré dans X à l'étape s . Notons que pour chaque intervalle $[g(n), g(n+1)[$ contenant au moins deux éléments de $\mathbb{N} \setminus X$, on énumérera un élément dans Y qui n'est pas dans X , et l'on gardera un élément de $\mathbb{N} \setminus X$ hors de Y . Le complémentaire de Y sera donc infini, et une infinité d'éléments de Y ne seront pas dans X , ce qui contredit l'hypothèse de maximalité sur X . Donc, X est high.

Chapitre 8

Solution 1.7. Trivial.

Solution 1.8. Une fois un nœud $\sigma_n \in T$ calculé, ce nœud a au moins une extension dans T . On calcule alors σ_{n+1} comme étant l'extension la plus à gauche de σ_n dans T (c'est-à-dire $\sigma_n 0$ si $\sigma_n 0 \in T$, et $\sigma_n 1$ sinon).

Solution 4.2. À l'aide de $\emptyset'$, on peut calculer le sous-arbre $T' \subseteq T$ des nœuds extensibles de T . D'après l'exercice 1.8, cet arbre contient un chemin calculable en l'arbre et donc $\emptyset'$ -calculable.

Solution 5.2. Pas de difficulté particulière : il s'agit de dupliquer la construction faite dans la preuve du théorème 7-5.6, de la même manière que la construction d'un degré calculatoirement dominé dans une classe Π_1^0 est dupliquée dans la preuve du théorème 5.1.

Solution 5.3. En reprenant les éléments de la preuve du théorème 5.1, à une étape n , on a pour chaque chaîne $\sigma \in 2^{<\mathbb{N}}$ de taille n des classes Π_1^0 deux à deux disjointes et non vides $\mathcal{P}_\sigma \subseteq \mathcal{P}$. Soit Φ_n la fonctionnelle de code n , et soient σ_0, σ_1 deux chaînes distinctes de taille n .

Soit il existe X tel que $\Phi_n(Y) = X$ pour tout $Y \in \mathcal{P}_{\sigma_0}$, auquel cas X est calculable et donc $\Phi_n(Y) \notin \mathcal{P}_{\sigma_1}$ pour tout $Y \in \mathcal{P}_{\sigma_0}$, soit il existe m tel que

$$\{Y \in \mathcal{P}_0 : \Phi_n(Y, m) \uparrow \neq 0\} \neq \emptyset \quad \text{et} \quad \{Y \in \mathcal{P}_0 : \Phi_n(Y, m) \uparrow \neq 1\} \neq \emptyset.$$

Au moins une des deux classes parmi $\{Y \in \mathcal{P}_1 : Y(m) = 0\}$ et $\{Y \in \mathcal{P}_1 : Y(m) = 1\}$ est non vide. On peut donc toujours trouver deux classes Π_1^0 non vides $\mathcal{P}'_{\sigma_0} \subseteq \mathcal{P}_{\sigma_0}$ et $\mathcal{P}'_{\sigma_1} \subseteq \mathcal{P}_{\sigma_1}$ telles que $\Phi_n(Y) \notin \mathcal{P}'_{\sigma_1}$ pour tout $Y \in \mathcal{P}'_{\sigma_0}$.

Il suffit d'itérer cette idée pour toutes les paires σ_0, σ_1 de taille n , et de recommencer en alternant ces étapes avec celles de la preuve du théorème 5.1, pour diagonaliser contre toutes les fonctionnelles Φ_n .

Solution 5.4. La différence est que l'on n'a ici pas d'effectivité. On montre le lemme suivant : étant donné deux chaînes σ et τ qui sont chacune respectivement le premier nœud branchant de f -arbres S_σ et S_τ , et étant donné une fonctionnelle Φ_e , il existe des extensions $\sigma' \succeq \sigma$, $\tau' \succeq \tau$ et un sous f -arbre $S'_\sigma \subseteq S_\sigma$ dont σ' est le premier nœud branchant, et tel que $\Phi_e(X) \not\preceq \tau'$ pour tout $X \in [S'_\sigma]$.

Soient ρ_0, ρ_1 telle que $\tau\rho_0, \tau\rho_1 \in \text{Im } S_\tau$ sont incompatibles. Alors, on a deux possibilités. Soit il existe $\sigma' \succeq \sigma$ avec $\sigma' \in \text{Im } S_\sigma$ telle que $\Phi_e(\sigma') \succeq \tau\rho_0$; dans ce cas, on trouve un sous-arbre de S_σ tel qu'aucun de ses nœuds qui étend σ' n'est envoyé vers $\tau\rho_1$. Soit pour tout $\sigma' \succeq \sigma$ on a $\Phi_e(\sigma') \not\preceq \tau\rho_0$; dans ce cas, aucun nœud de S_σ qui étend σ n'est envoyé vers $\tau\rho_0$.

À présent, étant donné un f -arbre quelconque T , il suffit de construire un sous f -arbre S de T en itérant petit à petit le lemme sur chaque feuille du morceau fini de S que l'on est en train de construire.

Solution 5.5. Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre calculable tel que $\mathcal{P} = [T]$. À l'aide de l'arrêt, on calcule l'ensemble $T' \subseteq T$ des nœuds extensibles de T . On peut ensuite trouver un chemin de $[T']$ en diagonalisant contre tous les ensembles calculables (sachant que $\emptyset'$ peut lister uniformément tous les ensembles calculables).

Solution 6.10. Étant donné n , on définit le code e_n de la fonctionnelle telle que $\Phi_{e_n}(e_n) = 0$ si $n \in A$ et $\Phi_{e_n}(e_n) = 1$ si $n \in B$. Si aucun des deux cas ne se produit, alors $\Phi_{e_n}(e_n) \uparrow$. Soit à présent $f : \mathbb{N} \rightarrow \{0, 1\}$ une fonction X -calculable telle que $f(n) \neq \Phi_n(n)$ pour tout n . On calcule alors $C(n) = f(e_n)$.

Solution 7.6. Si Y un ensemble $\text{PA}(X)$, il peut calculer un chemin dans toute classe $\Pi_1^0(X)$ non vide, et donc en particulier dans une classe $\Pi_1^0(X)$ ayant X pour seul chemin.

Solution 7.11. La clôture par préfixe d'un ensemble de chaînes est un arbre. Si cet ensemble de chaînes est infini, il s'agit alors d'un arbre infini, qui admet un chemin par le lemme faible de König.

Solution 7.12. Soit $\sigma_t = \emptyset'[t] \upharpoonright_t$ et $S = \{\sigma_t : t \in \mathbb{N}\}$. Alors, l'unique chemin de la clôture par préfixe de S est $\emptyset'$.

Solution 7.13. Soit $f : 2^{<\mathbb{N}} \times \mathbb{N} \rightarrow \{0, 1\}$ une approximation Δ_2^0 de T , c'est-à-dire telle que pour tout $\sigma \in 2^{<\mathbb{N}}$, $\lim_y f(\sigma, y)$ existe, et $\lim_y f(\sigma, y) = 1$ ssi $\sigma \in T$. On supposera de plus que pour tout y , $\{\sigma \in 2^{<\mathbb{N}} : f(\sigma, y) = 1\}$ est clos par préfixe et infini.

Nous allons définir une suite calculable de chaînes $\sigma_0, \sigma_1, \dots$ telle que $|\sigma_n| = n$ et telle que l'ensemble des chemins infinis de sa clôture par préfixe est $[T]$. Initialement, $\sigma_0 = \epsilon$. Supposons que l'on a défini $\sigma_0, \dots, \sigma_n$.

Pour toute chaîne τ de longueur au plus n , on définit son *ancienneté* $\mu_n(\tau)$ à l'étape n comme le plus grand entier $s \leq n$ tel que $\tau \preceq \sigma_s$ s'il existe. Sinon, $\mu_n(\tau) = 0$. Soit σ_{n+1} une chaîne de longueur $n+1$ telle que $f(\sigma_{n+1}, n+1) = 1$, et telle que pour tout $s \leq n$, $\sigma_{n+1} \upharpoonright_s$ est d'ancienneté minimale parmi

$$\{\rho \in 2^{\mathbb{N}} : |\rho| = s \wedge \rho \upharpoonright_{s-1} = \sigma_{n+1} \upharpoonright_{s-1}\}.$$

Soit $S = \{\sigma_n : n \in \mathbb{N}\}$. Montrons que $[\widehat{S}] \subseteq [T]$. Soit $P \in [\widehat{S}]$, et soit $\tau \prec P$. Alors, il existe une infinité de n tels que $\tau \prec \sigma_n$.

Comme $f(\sigma_n, n) = 1$ et $\{\rho \in 2^{<\mathbb{N}} : f(\rho, n) = 1\}$ est clos par préfixes, il existe une infinité de n tels que $f(\tau, n) = 1$, donc $\lim_n f(\tau, n) = 1$, autrement dit $\tau \in T$. Comme tout segment initial de P est dans T , $P \in [T]$.

Montrons que $[T] \subseteq [\widehat{S}]$. Soit $P \in [T]$. Supposons par l'absurde que $P \notin [\widehat{S}]$, et soit $\tau \prec P$ de longueur minimale tel que $[\tau] \cap [\widehat{S}] = \emptyset$. Soit n_0 tel que

- (1) pour tout $n > n_0$, $\sigma_n \upharpoonright_{|\tau|}$ est extensible en un chemin infini dans $\widehat{S}$;
- (2) pour tout $\rho \in 2^{<\mathbb{N}}$ de longueur $|\tau|$ extensible en un chemin infini dans $\widehat{S}$, il existe $n < n_0$ tel que $\rho \prec \sigma_n$;
- (3) pour tout $n > n_0$, $f(\tau, n) = 1$.

Par minimalité de τ , il existe $n > n_0$ tel que $\tau \upharpoonright_{|\tau|-1} \prec \sigma_n$. Par le choix de σ_n , $\sigma \upharpoonright_{|\tau|}$ est une chaîne étendant $\sigma \upharpoonright_{|\tau|-1}$ d'ancienneté minimale. Par (2), toutes les chaînes de longueur $|\tau|$ extensibles dans $\widehat{S}$ sont d'ancienneté strictement positive; par (3), il existe une chaîne étendant $\sigma \upharpoonright_{|\tau|-1}$ d'ancienneté 0; et, par (1), $\sigma \upharpoonright_{|\tau|}$ est extensible en un chemin infini. On obtient une contradiction. Donc, $P \notin [\widehat{S}]$.

Chapitre 10

Solution 2.8. Supposons que $U^<$ soit dense dans $2^{<\mathbb{N}}$. Soit $\sigma \in 2^{<\mathbb{N}}$. Montrons que $[\sigma] \cap \bigcup_{\tau \in U} [\tau] \neq \emptyset$. Par densité de $U^<$, il existe $\rho \succeq \sigma$ tel que $\rho \in U^<$. Il existe donc un préfixe $\rho' \preceq \rho$ tel que $\rho' \in U$. Soit $\sigma \preceq \rho'$ auquel cas $[\rho'] \subseteq [\sigma]$, soit $\rho' \preceq \sigma$ auquel cas $[\sigma] \subseteq [\rho']$. Dans tous les cas, $[\rho'] \cap [\sigma] \neq \emptyset$, et donc $[\sigma] \cap \bigcup_{\tau \in U} [\tau] \neq \emptyset$.

Supposons maintenant que l'ouvert $\bigcup_{\tau \in U} [\tau]$ soit dense dans l'espace de Cantor. Soit $\sigma \in 2^{<\mathbb{N}}$. En particulier, $[\sigma] \cap \bigcup_{\tau \in U} [\tau] \neq \emptyset$, donc il existe $A \in [\sigma] \cap \bigcup_{\tau \in U} [\tau]$. Soit $\tau \in U$ tel que $A \in [\tau]$. Soit $n > \max(|\sigma|, |\tau|)$. En particulier, $A \upharpoonright_n \succeq \tau$, donc par la clôture par suffixe de $U^<$, $A \upharpoonright_n \in U^<$, et $A \in [\sigma]$, donc $A \upharpoonright_n \succeq \sigma$. La chaîne σ a donc une extension dans $U^<$.

Solution 2.12. Par l'exercice 2.8, pour tout n , comme $W_n^<$ est dense dans $2^{<\mathbb{N}}$ et clos par suffixe, alors $[W_n]$ est dense dans l'espace de Cantor. Ainsi, $\bigcap_n [W_n]$ est une intersection dénombrable d'ouverts denses, et la classe est donc co-maigre.

Solution 2.13. Soit G un ensemble $\vec{W}$ -générique. En particulier, G rencontre W_n pour tout n , et il existe donc $\sigma \prec G$ tel que $\sigma \in W_n$; par suite, $G \in [W_n]$, et donc $G \in \bigcap_n [W_n]$. Inversement, soit $G \in \bigcap_n [W_n]$. Alors, pour tout n , $G \in [W_n]$,

et il existe donc $\sigma \prec G$ tel que $\sigma \in W_n$. Autrement dit, G rencontre W_n pour tout n , de sorte que G est $\tilde{W}$ -générique.

Solution 3.5. Comme dans (3) $\rightarrow$ (2) du théorème 3.2, mais relativisé à A .

Solution 3.6. Soit $f \leq_T G$ une fonction égale infiniment souvent à toute fonction A -calculable. Alors, $f + 1$ est infiniment souvent au-dessus de toute fonction A -calculable.

Solution 3.11. La construction est similaire à celle de la preuve du théorème 3.9. On construit pour tout e une fonction $\emptyset''$ -calculable f_e , qui différera de $\Phi_e(X)$ pour tout X dans un arbre $T : \mathbb{N}^{<\mathbb{N}} \rightarrow \mathbb{N}^{<\mathbb{N}}$, que l'on construit petit à petit.

Supposons à l'étape $s - 1$ que T est déjà défini sur les chaînes $\sigma_0, \dots, \sigma_s$, et que l'on a des réservoirs c. e. infinis $V_{t,s} \subseteq \mathbb{N}^{<\mathbb{N}}$ pour $t \leq s$. À l'étape $s = \langle e, i \rangle$, à la sous-étape $t \leq s$, on demande à $\emptyset''$ s'il existe une infinité de chaînes $\mu \in V_{t,s}$ ayant une extension $\mu' \succeq \mu$ pour laquelle $\Phi_e(\mu', i) \downarrow$ est égale à un nombre dont le t -ième bit est 0. Si la réponse est oui, alors $V_{t,s+1}$ est l'ensemble de ces extensions, et sinon $V_{t,s+1}$ est l'ensemble $V_{t,s}$ auquel on retire un nombre fini d'éléments de manière à ce que toutes les extensions possibles μ' restantes sur lesquelles $\Phi_e(\mu', i)$ s'arrête soient envoyées sur un nombre dont le t -ième bit est 1. Une fois la réponse obtenue pour tout $t \leq s$, on construit un nombre de s bits qui diffère de tous les calculs possibles par un oracle de notre arbre. On définit alors $f_e(i)$ comme étant ce nombre.

Solution 3.22. Soit A un ensemble calculable, et soit W l'ensemble Σ_1^0

$$\{(A \upharpoonright_n)^\frown (1 - A(n)) : n \in \mathbb{N}\}.$$

Par construction, A ne possède aucun segment initial dans W , mais aucun segment initial de A n'évite W non plus. Ainsi, A ne possède pas de segment initial dans $W \cup W^\perp$, et A n'est donc pas 1-générique.

Solution 3.23. Étant donné une chaîne σ et un ensemble c. e. $U \subseteq 2^{<\mathbb{N}}$, l'arrêt $\emptyset'$ peut répondre à la question de savoir si σ a une extension dans U est Σ_1^0 . On construit ainsi aisément un ensemble générique G $\emptyset'$ -calculable. Comme G est 1-générique, alors $G' \leq_T G \oplus \emptyset' \equiv \emptyset'$. Donc, G est low.

Solution 3.25. Rappelons que la fonction principale d'un ensemble infini X donné par $\{a_0 < a_1 < \dots\}$ est la fonction p_X qui à i associe a_i . Soit Φ_e une fonctionnelle, et i un entier. Considérons l'ouvert

$$\mathcal{U}_{e,i} = \left\{ \bigoplus_{n \in \mathbb{N}} X_n : \exists m \Phi_e \left(\bigoplus_{j \neq i} X_j, m \right) \downarrow < p_{X_i}(m) \right\}.$$

Supposons qu'il existe une chaîne σ telle que $[\sigma] \cap \mathcal{U}_{e,i} = \emptyset$. Montrons que l'on a nécessairement un entier m tel que $\Phi_e(\bigoplus_{j \neq i} X_j, m) \uparrow$ pour tout $X = \bigoplus_{n \in \mathbb{N}} X_n$ tel que $\sigma \prec X$. Supposons, par l'absurde, le contraire. Soit $m > |\sigma|$. Alors, il existe $X = \bigoplus_{n \in \mathbb{N}} X_n$ avec $\sigma \prec X$ tel que $\Phi_e(\bigoplus_{j \neq i} X_j, m) \downarrow = k \in \mathbb{N}$.

Comme $[\sigma] \cap \mathcal{U}_{e,i} = \emptyset$, on a nécessairement $p_{X_i}(m) \leq k$. Il suffit à présent de considérer $Y = \bigoplus_{n \in \mathbb{N}} Y_n$ tel que $Y_j = X_j$ pour $j \neq i$, $Y_i(m') = 0$ si $m \leq m' \leq k$, et $Y_i(m') = X_i(m')$ sinon. On a alors $\Phi_e(\bigoplus_{j \neq i} Y_j, m) \downarrow = k < p_{Y_i}(m)$, donc $Y \in \mathcal{U}_{e,i}$, et comme $Y \succ \sigma$, cela contredit notre hypothèse sur σ .

Ainsi, si $G = \bigoplus_{n \in \mathbb{N}} G_n$ est 1-générique, soit $G \in \mathcal{U}_{e,i}$, soit il existe un préfixe $\sigma \prec G$ tel que $[\sigma] \cap \mathcal{U}_{e,i} = \emptyset$, auquel cas il existe m tel que $\Phi_e(\bigoplus_{j \neq i} G_j, m) \uparrow$.

Solution 3.26. La preuve est une adaptation de la direction (1) $\rightarrow$ (3) du théorème 3.2. Pour tout $i \in \mathbb{N}$, soit f_i la fonction principale de X_i , c'est-à-dire la fonction qui à n associe le n -ième élément de X_i . Notons que f_i est croissante, et hyperimmune relativement à $\bigoplus_{j \neq i} X_j$. On calcule à partir de X un ensemble 1-générique $G \in 2^{\mathbb{N}}$. Soit $(W_e)_{e \in \mathbb{N}}$ une énumération des sous-ensembles Σ_1^0 de $2^{<\mathbb{N}}$. On construit G par approximations successives $\sigma_0 \preceq \sigma_1 \preceq \sigma_2 \preceq \dots$.

On décrit d'abord une procédure récursive à effectuer à chaque fois que l'on veut concaténer une chaîne τ à une chaîne σ que l'on a jusqu'à présent calculé. Cette procédure, que nous nommerons R , prend un troisième paramètre : un entier e qui correspond au plus petit entier tel que $\sigma\tau$ est énuméré dans W_e à l'étape de calcul $f_e(|\sigma|)$. On notera $R(\sigma, \tau, e)$ pour le résultat de l'appel à cette procédure. Notons enfin que certains entiers sont marqués comme « satisfaits » au moment où la procédure est appelée : ce sont les entiers e tels que σ étend une chaîne de W_e à l'étape de calcul courante.

La procédure $R(\sigma, \tau, e)$ fait la chose suivante : pour chaque préfixe $\tau' \preceq \tau$ dans l'ordre, elle cherche le plus petit entier $e' < e$ qui n'est pas satisfait et tel qu'une chaîne de la forme $\sigma\tau'\rho$ soit énumérée dans $W_{e'}[f_{e'}(|\sigma\tau'|)]$. Si un tel entier est trouvé, la procédure renvoie alors le résultat de l'appel récursif à $R(\sigma\tau', \rho, e')$. Sinon, elle renvoie $\sigma\tau$. Notons que la diminution de la valeur du dernier paramètre dans les appels récursifs fait que la procédure s'arrête nécessairement.

À l'étape 0, on définit $\sigma_0 = \epsilon$. Supposons σ_t défini à l'étape t . À l'étape $t+1$, on cherche le plus petit entier $e \leq t+1$ non satisfait tel qu'une chaîne de la forme $\sigma_t\tau$ soit énumérée dans $W_e[f_e(|\sigma_t|)]$. Si l'on trouve un tel entier e , on définit σ_{t+1} comme étant $R(\sigma_t, \tau, e)$. Sinon, σ_{t+1} comme étant σ_0 . Cela conclut la construction.

Supposons par l'absurde que G n'est pas 1-générique. Soit e le plus petit entier tel que W_e est un ensemble dense le long de G . Dans ce cas, la fonction f_e n'a aucun impact sur la construction, qui est alors $\bigoplus_{j \neq e} X_j$ -calculable. En particulier, f_e est G -hyperimmune. Soit h la fonction totale G -calculable qui à n associe le plus petit temps de calcul t tel que W_e énumère une chaîne étendant $G \upharpoonright_n$. Comme f_e est G -hyperimmune, il existe une infinité de valeurs n telles que $f_e(n) > h(n)$.

Soit t tel que tous les entiers $e' < e$ qui sont satisfaits à un moment de la construction sont satisfaits au temps t . Soit n le plus petit entier supérieur ou égal à $|\sigma_t|$ tel que $f_e(n) > h(n)$. Soit $s \geq t$ le plus petit entier tel que $|\sigma_s| \leq n < |\sigma_{s+1}|$. Si $|\sigma_s| = n$, alors par minimalité de e l'algorithme définit $\sigma_{s+1} = \sigma_s\tau$ avec $\sigma_s\tau$ dans $W_e[f_e(n)]$. Sinon, alors par construction au moment de définir $\sigma_{s+1} = \sigma_s\tau$

pour une certaine chaîne τ , l'algorithme vérifie pour tout préfixe $\tau' \preceq \tau$, que l'on n'a pas une extension de $\sigma_s \tau'$ énumérée dans $W_e[f_e(|\sigma_s \tau'|)]$, et en particulier pour le préfixe τ' tel que $|\sigma_s \tau'| = n$. Si c'est le cas, l'algorithme est relancé sur cette extension. Comme c'est effectivement le cas par hypothèse, et par minimalité de e , on aura en fait $\sigma_s \tau \in W_e[f_e(n)]$ pour $\sigma_{s+1} = \sigma_s \tau$. On en conclut que W_e n'est pas dense le long de $G = \sigma_0 \prec \sigma_1 \prec \sigma_2 \prec \dots$, donc que l'ensemble G est bien 1-générique.

Solution 3.30. Montrons qu'il existe un ensemble faiblement 1-générique relativement à A et approchable par la gauche relativement à A . Soit $(W_n)_{n \in \mathbb{N}}$ une énumération des ensembles de chaînes $\Sigma_1^0(A)$.

À l'étape s , on calcule à l'aide de A une chaîne $\sigma_{n,s}$ pour tout $n \leq s$, telle que $\sigma_{n,s} \prec \sigma_{n+1,s}$ pour tout $n < s$ et telle que $\sigma_{n,s-1}$ est lexicographiquement plus petit ou égal à $\sigma_{n,s}$. À l'étape $s = 0$, on définit $\sigma_{0,0}$ comme étant la chaîne 0. Supposons $\sigma_{n,s}$ défini à l'étape s pour tout $n \leq s$. À l'étape $s + 1$, soit $n \leq s$ le plus petit tel que $\sigma_{n,s}$ est de la forme $\tau 0$ et tel qu'il existe une extension $\tau 1 \rho$ de $\tau 1$ dans $W_n[s + 1]$. Si aucun tel entier n n'existe, alors $\sigma_{n,s+1} = \sigma_{n,s}$ pour $n \leq s$, et l'on définit $\sigma_{s+1,s+1} = \sigma_{s,s+1} 0$.

Sinon, on définit $\sigma_{i,s+1} = \sigma_{i,s+1}$ pour $i < n$, $\sigma_{n,s+1} = \tau 1 \rho$ et $\sigma_{i+1,s+1} = \sigma_{i,s+1} 0$ pour $s + 1 \geq i \geq n$. On laisse au lecteur le soin de vérifier que $\lim_{s \rightarrow +\infty} \sigma_{n,s} = \sigma_n$ est bien défini et que l'unique élément $G \in \bigcap_n [\sigma_n]$ est bien approchable par la gauche relativement à A et faiblement 1-générique relativement à A .

Montrons qu'aucun ensemble faiblement 1-générique relativement à A n'est approchable par la gauche relativement à A . Soit $(X_s)_{s \in \mathbb{N}}$ une approximation par la gauche relativement à A d'un ensemble X . On peut supposer sans perte de généralité $X_s \neq X_{s+1}$. Pour tout s , soit n_s le plus petit entier tel que

$$X_s \upharpoonright_{n_s+1} \neq X_{s+1} \upharpoonright_{n_s+1}.$$

Notons que l'on a nécessairement $X_s(n_s) = 0$ et $X_{s+1}(n_s) = 1$. On énumère dans W — un ensemble A -c. e. — la chaîne $X_s \upharpoonright_{n_s+1}$ pour tout s . On laisse au lecteur le soin de vérifier que W est dense le long de G sans jamais rencontrer G .

Chapitre 11

Solution 1.7. Soit $Q = \{d \in \mathbb{P} : d \text{ et } c \text{ sont incompatibles}\}$. Montrons que $D \cup Q$ est dense. Soit $d \in \mathbb{P}$.

Cas 1. Les conditions d et c sont incompatibles, auquel cas $d \in Q$.

Cas 2. Il existe $d' \leq d, c$. Sachant que D est dense sous c , il existe $e \leq d'$ tel que $e \in D$. En particulier, $e \leq d$ et $e \in D \cup Q$. Ainsi, $D \cup Q$ est dense. Sachant que F est suffisamment générique, il intersecte $D \cup Q$. S'il contient c , alors par définition d'un filtre, tous les éléments de F sont compatibles avec c , donc $F \cap Q = \emptyset$. Il s'ensuit que F intersecte D .

Solution 2.4. Soit F un filtre suffisamment générique contenant c , et soit

$$D = \{d \in \mathbb{P} : d \text{ force } \mathcal{R}\}.$$

Par l'exercice 1.7, F intersecte l'ensemble D , donc $\dot{F}$ satisfait $\mathcal{R}$. Il s'ensuit que c force $\mathcal{R}$.

Solution 2.8. Par induction sur la complexité de $\mathcal{R}$.

Cas 1. Le contrat $\mathcal{R}$ est Σ_1^0 ou Π_1^0 . Par définition, pour tout $X \in [c]$, $\mathcal{R}(X)$ est vrai. Comme $d \leq c$, alors $[d] \subseteq [c]$, donc pour tout $X \in [d]$, $\mathcal{R}(X)$ est vrai. Ainsi, $d \Vdash^* \mathcal{R}$.

Cas 2. Le contrat $\mathcal{R}$ est de la forme $\exists x \mathcal{S}(x)$ où $\mathcal{S}(x)$ est Π_k^0 pour $k \geq 1$. Par définition, il existe $n \in \mathbb{N}$ tel que $c \Vdash^* \mathcal{S}(n)$. Par hypothèse d'induction, $d \Vdash^* \mathcal{S}(n)$, donc $d \Vdash^* \exists x \mathcal{S}(x)$.

Cas 3. Le contrat $\mathcal{R}$ est de la forme $\forall x \mathcal{S}(x)$, où $\mathcal{S}(x)$ est Σ_k^0 pour $k \geq 1$. Alors, $c \Vdash^* \mathcal{R}$ implique $\forall e \leq c \ e \Vdash^* \neg \mathcal{R}$ implique $\forall e \leq d \ e \Vdash^* \neg \mathcal{R}$ implique $d \Vdash^* \mathcal{R}$.

Solution 2.9. Par induction sur la complexité de $\mathcal{R}$, et soit

$$U = \{c \in \mathbb{P} : c \Vdash^* \mathcal{R} \text{ ou } c \Vdash^* \neg \mathcal{R}\}.$$

On peut supposer que $\mathcal{R}$ est Σ_n^0 (sinon, on répète l'argument avec $\neg \mathcal{R}$ à la place de $\mathcal{R}$). Étant donné une condition c , soit il existe une extension de $d \leq c$ telle que $d \Vdash^* \mathcal{R}$, soit $\forall d \leq c \ d \Vdash^* \neg \mathcal{R}$ auquel cas $c \Vdash^* \neg \mathcal{R}$.

Solution 2.10. Par induction sur la complexité de $\mathcal{R}$.

Cas 1. Le contrat $\mathcal{R}$ est Σ_1^0 ou Π_1^0 . Supposons que $c \Vdash^* \mathcal{R}$. Pour tout filtre maximal F contenant c , comme $\dot{F} \in [c]$, alors $\dot{F}$ satisfait $\mathcal{R}$, donc $c \Vdash \mathcal{R}$.

Cas 2. Le contrat $\mathcal{R}$ est de la forme $\exists x \mathcal{S}(x)$. Supposons que $c \Vdash^* \exists x \mathcal{S}(x)$. Par définition, il existe un $n \in \mathbb{N}$ tel que $c \Vdash^* \mathcal{S}(n)$. Par hypothèse d'induction, c force $\mathcal{S}(n)$, donc $c \Vdash \exists x \mathcal{S}(x)$.

Cas 3. Le contrat $\mathcal{R}$ est de la forme $\forall x \mathcal{S}(x)$. Supposons que $c \Vdash^* \forall x \mathcal{S}(x)$. Par définition, pour tout $d \leq c$ et $n \in \mathbb{N}$, $d \Vdash^* \neg \mathcal{S}(n)$. Par l'exercice 2.9, pour tout n , l'ensemble $\{d \in \mathbb{P} : d \Vdash^* \mathcal{S}(n)\}$ est dense sous c . Soit F un filtre suffisamment générique contenant c . En particulier, pour tout $n \in \mathbb{N}$, il existe $d \in F$ tel que $d \Vdash^* \mathcal{S}(n)$. Par hypothèse d'induction, $d \Vdash \mathcal{S}(n)$, donc $\dot{F}$ satisfait $\mathcal{S}(n)$. Il s'ensuit que $\dot{F}$ satisfait $\forall x \mathcal{S}(x)$, donc $c \Vdash \forall x \mathcal{S}(x)$.

Solution 3.3. Il suffit de montrer que G est différent d'un ensemble A quelconque fixé à l'avance. Soit $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ un f-arbre calculable. En particulier, $T(0)$ et $T(1)$ sont incomparables, donc il existe $i < 2$ tel que $T(i)$ est incomparable avec A . Le f-arbre $S : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ défini par $S(\sigma) = T(i\sigma)$ est un sous-f-arbre calculable de T tel que pour tout $P \in [S]$, $P \neq A$.

Solution 3.5. Soit $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ une condition du forcing de Sacks et soit Φ_e une fonctionnelle. Supposons que pour tout n il existe i_n tel que pour tout $\sigma \in$

$2^{<\mathbb{N}}$, on a $\Phi_e(T(\sigma), n) \downarrow \rightarrow \Phi(T(\sigma), n) \downarrow = i_n$. Alors, si Φ_e est totale sur $X \in [T]$, on a nécessairement $\Phi_e(X) = Y$, avec $Y(n) = i_n$ pour tout n . Un tel ensemble Y est calculable, car il suffit pour connaître $Y(n)$ de chercher $\sigma \in 2^{<\mathbb{N}}$ tel que $\Phi_e(T(\sigma), n) \downarrow$. Donc, $\Phi_e(X) \neq A$. En particulier, $\Phi_e(X) \neq A$, pour tout $X \in [T]$.

Sinon, il existe n et $\sigma_1, \sigma_2 \in 2^{<\mathbb{N}}$ tels que $\Phi_e(T(\sigma_1), n) \nmid \neq \Phi_e(T(\sigma_2), n) \downarrow$. On peut considérer sans perte de généralité $\Phi_e(T(\sigma_1), n) \nmid \neq A(n)$. On prend alors l'extension $S \leq T$ définie par $S(\tau) = T(\sigma_1\tau)$.

L'ensemble des conditions T telles que

$$T \Vdash^* \exists n \Phi_e(G, n) \uparrow \quad \text{ou} \quad \exists n T \Vdash^* \Phi_e(G, n) \nmid \neq A(n)$$

est dense. Donc, si G est suffisamment générique pour le forcing de Sacks, il ne calcule pas A .

Solution 3.6.

1. Soit $T : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}}$ une condition du forcing de Sacks et soit Φ_e une fonctionnelle. Si $\exists n \exists \sigma \forall \tau \succeq \sigma \Phi_e(T(\tau), n) \uparrow$, on considère l'extension $S \leq T$ définie par $S(\rho) = T(\sigma\rho)$. On force ainsi la partialité de Φ_e .

Sinon, $\forall n \forall \sigma \exists \tau \succeq \sigma \Phi_e(T(\tau), n) \downarrow$, on construit une condition $S \leq T$ telle que $\Phi_e(S(\sigma), |\sigma|) \downarrow$ pour tout $\sigma \in 2^{<\mathbb{N}}$. La taille de l'ensemble T_n des valeurs calculées par des chaînes $S(\sigma)$, pour σ de taille n , est bornée par 2^n .

2. Soit X calculatoirement traçable avec borne calculable h . Soit h' comme dans l'énoncé. Soit $(A_n)_{n \in \mathbb{N}}$ une suite d'intervalles consécutifs formant une partition de $\mathbb{N}$ et telle que $h(\min A_n) < h'(\max A_n)$. On note $\langle f(x) \rangle_{x \in A_n}$ pour l'entier $\langle f(x_0), \dots, f(x_k) \rangle$, où $x_0, \dots, x_k$ sont les éléments de A_n .

Étant donné $f \leq_T X$, on considère $g \leq_T f$ défini par $g(n) = \langle f(x) \rangle_{x \in A_n}$. Soit $(T_n)_{n \in \mathbb{N}}$ une trace de g avec $|T_n| \leq h(n)$. On peut décomposer la trace $(T_n)_{n \in \mathbb{N}}$ en une trace $(S_n)_{n \in \mathbb{N}}$ pour la fonction f qui sera alors telle que $|S_n| \leq h'(n)$ pour $n > 0$ (car on a $h(\min A_n) < h'(\max A_n)$).

3. Pour chaque condition T , l'ensemble $[T]$ des chemins de T contient des fonctions $f : \mathbb{N} \rightarrow \mathbb{N}$ avec $f(n) < 2^n$. L'objectif est de montrer que si une telle fonction g est suffisamment générique, alors elle sera calculatoirement dominée, et pour toute trace $(T_n)_{n \in \mathbb{N}}$ telle que $|T_n| \leq n$, on aura $g(m) \notin T_m$ pour un certain m . D'après (2), cela est suffisant pour montrer que l'ensemble $X_g \in 2^{\mathbb{N}}$ qui encode g n'est pas calculatoirement traçable, car X_g calcule g qui n'admet de pas trace $(T_n)_{n \in \mathbb{N}}$ avec $|T_n| \leq n$.

La manière de forcer l'ensemble à être calculatoirement dominé est similaire à celle du forcing de Sacks : soit il existe $n \in \mathbb{N}$ et il existe $\sigma \in T$ tels que $\forall \tau \succeq \sigma$ avec $\tau \in T$ on a $\Phi_e(\tau, n) \uparrow$, auquel cas $S = T \upharpoonright_\sigma$ est une extension qui force la partialité, soit pour tout $n \in \mathbb{N}$ et pour tout $\sigma \in T$ il existe $\tau \succeq \sigma$ avec $\tau \in T$ tel que $\Phi_e(\tau, n) \downarrow$. Dans ce cas, pour tout nœud $\sigma \in T$ à branchement maximal on peut trouver pour tout $i < 2^{|\sigma|}$ des extensions $\tau_i \succeq \sigma i$ telles que $\Phi_e(\tau_i, n) \downarrow$. On utilise cela pour construire une extension $S \subseteq T$ telle que Φ_e est totale sur les chemins de $[S]$, ainsi qu'une fonction calculable bornant $n \mapsto \Phi(X, n)$ pour tout $X \in [S]$.

Par nature des conditions de forcing, étant donné une trace calculable $(T_n)_{n \in \mathbb{N}}$ avec $|T_n| \leq n$, on a une chaîne $\tau \in T$ telle que $\tau i \in T$ et $i \notin T_{|\tau|}$. On peut alors étendre T en la condition $S \subseteq T$ des éléments de T compatibles avec τi , afin de forcer à ne pas être tracé par $(T_n)_{n \in \mathbb{N}}$.

Solution 4.4. (a) Soit $T \in \mathbb{P}$ et soit $\sigma \in 2^{<\mathbb{N}}$ tels que $[T] \cap [\sigma] \neq \emptyset$. Soit $S = \{\tau \in T : \tau \succeq \sigma \vee \tau \prec \sigma\}$. Alors, S est une extension de T telle que $[S] \subseteq [\sigma]$. Ainsi, $(\mathbb{P}, \leq)$ est un forcing de Cantor.

(b) Montrons que pour tout filtre $F \subseteq \mathbb{P}$ suffisamment générique pour $\mathbb{P}$, le filtre $\widehat{F} = \{[T] : T \in F\}$ est suffisamment générique pour le forcing de Jockusch-Soare. Soit $\widehat{D}$ un ensemble dense dans le forcing de Jockusch-Soare. Montrons que $D = \{T \in \mathbb{P} : [T] \in \widehat{D}\}$ est dense dans $\mathbb{P}$. Soit $T \in \mathbb{P}$. Par densité de $\widehat{D}$, il existe une classe $\Pi_1^0 \mathcal{C} \subseteq [T]$ telle que $\mathcal{C} \in \widehat{D}$. Par la proposition 8-3.3, il existe un arbre infini calculable $S \subseteq 2^{<\mathbb{N}}$ tel que $[S] = \mathcal{C}$. Notons que $S \cap T$ est également un arbre infini calculable tel que $[S \cap T] = \mathcal{C}$ et tel que $S \cap T$ est une extension de T . Donc, D est dense dans $\mathbb{P}$. Il s'ensuit que tout ensemble suffisamment générique pour $\mathbb{P}$ est suffisamment générique pour le forcing de Jockusch-Soare.

Montrons que pour tout filtre $\widehat{F}$ suffisamment générique pour le forcing de Jockusch-Soare, $F = \{T \in \mathbb{P} : [T] \in \widehat{F}\}$ est suffisamment générique pour $\mathbb{P}$. Soit $D \subseteq \mathbb{P}$ un ensemble dense pour $\mathbb{P}$. Montrons que $\widehat{D} = \{[T] : T \in D\}$ est dense pour le forcing de Jockusch-Soare. Soit $\mathcal{C} \subseteq 2^{<\mathbb{N}}$ une classe Π_1^0 non vide. Par la proposition 8-3.3, il existe un arbre infini calculable $T \subseteq 2^{<\mathbb{N}}$ tel que $[T] = \mathcal{C}$. Par densité de D , il existe une extension S de T dans D . En particulier, $[S]$ est une extension de $\mathcal{C}$ dans $\widehat{D}$, donc $\widehat{D}$ est dense. Il s'ensuit que tout ensemble suffisamment générique pour le forcing de Jockusch-Soare est suffisamment générique pour $\mathbb{P}$.

(c) Soit $T \in \mathbb{P}$ un arbre calculable infini, et soit $S = \{\sigma \in T : \forall n < |\sigma| \ \Phi(\sigma, n) \uparrow\}$. Si S est infini, alors S est une extension de T tel que $S \Vdash^\circ \forall n \Phi(G, n) \uparrow$. Si S est fini, soit t suffisamment grand tel qu'aucune chaîne de longueur t est dans S . Alors, pour tout $\sigma \in T$ de longueur t , $\exists n < t \ \Phi(\sigma, t) \downarrow$, donc $T \Vdash^\circ \exists n \Phi(G, n) \downarrow$.

(d) Immédiat.

Solution 4.7.

1. Nous allons définir à chaque étape s un arbre parfait $T_s \subseteq 2^{<\mathbb{N}}$ calculable uniformément en s , avec $T_{s+1} \subseteq T_s$. Étant donné un arbre parfait T , on définit $T \upharpoonright_n$ comme étant la clôture par préfixe de l'ensemble des chaînes $\sigma \in T$ ayant exactement n préfixes branchants dans T . Ainsi, $T \upharpoonright_n$ est un arbre fini avec 2^n feuilles.

Soit $(\mathcal{P}_e)_{e \in \mathbb{N}}$ une énumération des classes Π_1^0 . À l'étape 0, on définit $T_0 = 2^{<\mathbb{N}}$. Supposons T_s défini à l'étape s . À l'étape $s+1$, on considère le plus petit entier $e \leq s+1$ tel que pour une feuille $\sigma \in T_s \upharpoonright_e$ on a $\mathcal{P}_e[s+1] \cap [\sigma] \neq \emptyset$ et $\mathcal{P}_e[s+1] \cap [\sigma] \subsetneq [T_s] \cap [\sigma]$. Si un tel e et une telle chaîne σ existent, on définit alors τ comme étant une extension de σ appartenant à T_s et telle que $\mathcal{P}_e[s+1] \cap [\tau] = \emptyset$, puis on définit T_{s+1} comme la réunion de toutes les

chaînes de T_s incomparables avec σ , et de toutes les chaînes de T_s comparables avec τ . On dit alors que « T_s est modifié à cause de $\mathcal{P}_e$ ». Cela conclut la construction.

Étant donné un entier n , on notera s_n le plus petit temps de calcul tel que $T_{s_n} \upharpoonright_n = T_t \upharpoonright_n$ pour tout $t \geq s_n$. Montrons par induction sur n que s_n existe pour tout n . Une seule modification peut arriver à cause de $\mathcal{P}_0$: celle du premier nœud branchant $\sigma \in T_t \upharpoonright_0$. Si la modification arrive bien à une étape t , on a alors $\mathcal{P}_0[t+1] \cap [\sigma] = \emptyset$ pour σ l'unique feuille de $T_{t+1} \upharpoonright_0$, et par construction aucune autre modification n'arrivera à cause de $\mathcal{P}_0$, parce que $\mathcal{P}_0[t] \cap [\sigma] = \emptyset$. De plus, une modification due à $\mathcal{P}_e$ pour $e > 0$ se fait sur des feuilles de $T_t \upharpoonright_e$, et laisse donc $T_t \upharpoonright_0$ inchangé. Donc, $s_0 = t + 1$. En itérant cette idée et en utilisant le fait que $\mathcal{P}_e$ ne peut apporter au plus que 2^e modifications sur $T_t \upharpoonright_e$ pour $t \geq s_{e-1}$, on arrive à la conclusion. On en déduit que $T = \bigcap_s T_s$ est un arbre parfait, et que $[T]$ est donc une classe Π_1^0 parfaite.

Montrons que $[T]$ est fine. Soit $\mathcal{P}_e$ une classe Π_1^0 telle que $\mathcal{P}_e \subseteq [T]$. Supposons par l'absurde $\mathcal{P}_e \cap [\sigma_i] \neq \emptyset$ et $\mathcal{P}_e \cap [\sigma_i] \subsetneq [T] \cap [\sigma_i]$ pour un certain $\sigma_i \in T \upharpoonright_e$. Alors, cela doit être détecté à une certaine étape $t > s_e$, auquel cas par construction $T_{t+1} \upharpoonright_e \neq T_t \upharpoonright_e$, ce qui contredit la définition de s_e .

2. Soit $\mathcal{P}$ une classe Π_1^0 fine, et soit $X \in \mathcal{P}$. Étant donné une fonctionnelle Φ_e soit $\mathcal{Q} = \{Y \in \mathcal{P} : \Phi_e(Y, e) \uparrow\}$. À l'aide de $\emptyset'$, on détermine d'abord si $\mathcal{Q}$ est vide. Si c'est bien le cas, alors $e \in Y'$. Sinon, on cherche à l'aide de $\emptyset''$ des chaînes $\sigma_0, \dots, \sigma_n$ telles que $\mathcal{Q} = \mathcal{P} \cap ([\sigma_0] \cup \dots \cup [\sigma_n])$. Une fois de telles chaînes trouvées, on vérifie si l'une d'elles est un préfixe de Y . Si c'est le cas, alors $e \notin Y'$, sinon $e \in Y$.

Solution 4.8. Notons d'abord que toute classe de contenant que des aléatoires de Martin-Löf est de mesure positive. L'idée est de répéter la preuve du théorème 4.6 en utilisant le lemme 18-3.3, afin de construire dans toute classe Π_1^0 $\mathcal{P}$ de mesure positive une sous-classe Π_1^0 $\mathcal{S} \subseteq \mathcal{P}$ parfaite telle que, pour toute chaîne σ pour laquelle $\mathcal{S} \cap [\sigma] \neq \emptyset$, il existe $\tau \succeq \sigma$ pour laquelle $\mathcal{S} \cap [\tau] = \emptyset$ et $\mathcal{P} \cap [\tau] \neq \emptyset$.

Solution 4.17. Montrons que $c? \vdash \mathcal{R}$ satisfait les propriétés (1) et (2) de la définition d'une question de forcing.

- (1) Cette propriété est vérifiée par définition de $c? \vdash \mathcal{R}$.
- (2) Si $c? \not\vdash \mathcal{R}$, alors pour toute extension $d \leq c$, il vient $d \not\vdash \mathcal{R}$. Comme l'ensemble $\{d : d \Vdash \mathcal{R} \text{ ou } d \Vdash \neg \mathcal{R}\}$ est dense, il existe une extension $d \leq c$ telle que $d \Vdash \neg \mathcal{R}$. Autrement dit, $c? \vdash \neg \mathcal{R}$.

Solution 4.26. La preuve est exactement celle de la proposition 4.25, où l'on ne se restreint plus aux fonctions à valeurs dans $\{0, 1\}$. On évite donc de calculer une fonction DNC à valeurs arbitraires. Dans le cas 1, on se retrouve avec une quantité dénombrable de propriétés $\neg(\Phi_e^{G^{(n)}}(m) \downarrow = v)$ pour $v \in \mathbb{N}$ à forcer simultanément, ce qui est possible, car la question de forcing est Π - ω -fusionnable.

Chapitre 14

Solution 1.10. Il suffit de considérer le chemin X le plus à gauche de la classe. Ce chemin-ci est approchable par la gauche, et il permet de calculer l'ensemble c. e. $\{\sigma \in 2^{<\mathbb{N}} : \sigma \text{ est lexicographiquement à gauche de } X\}$. Notons que cet ensemble permet de recalculer X . Si une classe Π_1^0 ne contient que des ensembles de degrés minimaux, alors son chemin le plus à gauche est non calculable, et permet donc de calculer un ensemble c. e. non calculable, qui n'est alors pas de degré minimal : contradiction !

Solution 1.12. Il suffit de considérer la classe

$$\{X \oplus Y : \forall t \forall e \Phi_e(e)[t] \neq X(e) \wedge \Phi_e(Y, e)[t] \neq Y(e)\}.$$

Les membres $X \oplus Y$ de cette classe sont tous ceux tels que X est de degré DNC_2 et Y de degré DNC_2 relativement à X .

En particulier, par l'exercice 8-7.6, $Y \geq_T X$, mais $X \not\geq_T Y$.

Chapitre 16

Solution 1.7. Il s'agit d'un cas particulier du lemme 4-7.2 de Schoenfield, mais appliqué à des fonctions plutôt que des ensembles d'entiers.

Solution 1.9. Soit $f \geq \Sigma$. Alors, $U(\sigma) \downarrow$ ssi $U(\sigma)[f(|\sigma|)] \downarrow$. Donc, f permet de calculer U , et donc aussi $\emptyset'$.

Solution 1.10. Montrons que pour n suffisamment grand, il existe une chaîne de taille $2n$, de complexité inférieure à $2n$ mais telle qu'aucun programme de taille inférieure à $2n$ ne peut la calculer en un temps inférieur à $\Sigma(n)$. Étant donné n , soit σ_n le « castor affairé » pour n , c'est-à-dire avec $|\sigma_n| \leq n$ et $\Sigma(n) = t$, où t est le plus petit temps de calcul tel que $U(\sigma_n)[t] \downarrow$. Pour n fixé, on considère la plus petite chaîne τ_n correspondant au programme qui cherche le plus petit temps de calcul t tel que $U(\sigma_n)[t] \downarrow$, qui calcule $U(\rho)[t]$ pour toute chaîne ρ de taille strictement inférieure à $2n$, et écrit finalement une chaîne de taille $2n$ qui n'est pas produite par un tel programme. Comme $\sum_{i < 2n} 2^i = 2^{2n} - 1$, une telle chaîne de taille $2n$ existe nécessairement.

Pour n suffisamment grand, la taille de τ_n est plus petite que $2n$, puisque l'information nécessaire pour ce programme est une chaîne de taille inférieure ou égale à n , la connaissance de n qui est de l'ordre de $\log_2(n)$, et le reste du programme qui est constant pour tout n . En particulier, pour σ tel que $U(\tau_n) \downarrow = \sigma$, on a donc $C(\sigma) < 2n$. Aussi, par construction, aucun programme de taille inférieure à $2n$ ne peut produire σ en temps de calcul inférieur à $\Sigma(n)$. On peut donc calculer une borne de $\Sigma(n)$ à partir de la connaissance de C , en regardant pour tout n suffisamment grand la complexité de toutes les chaînes de taille $2n$, en cherchant pour chacune d'entre elles le plus petit programme permettant de les produire,

puis en considérant le plus petit temps de calcul utilisé alors. Ce temps de calcul borne nécessairement $\Sigma(n)$. Par (1), on peut donc calculer U .

Solution 1.11. Soit c la constante telle que $C(\sigma) \leq |\sigma| + c$, pour toute chaîne σ . Il suffit alors d'utiliser le théorème de la base low sur la classe Π_1^0 suivante :

$$\mathcal{P} = \{I : 2^{<\mathbb{N}} \rightarrow 2^{<\mathbb{N}} : (\forall \sigma |I(\sigma)| \leq |\sigma| + c) \wedge (\forall t \forall \sigma |I(\sigma)| \leq C(\sigma)[t])\}.$$

Notons que la condition $\forall \sigma |I(\sigma)| \leq |\sigma| + c$ n'est pas en soi absolument nécessaire, mais il s'agit d'insister sur le fait que l'arbre calculable dont les chemins infinis sont les éléments de $\mathcal{P}$ est calculatoirement borné (voir la définition 8-7.7).

Solution 1.14. Soit M le programme qui sur une chaîne σ renvoie la chaîne $\sigma'U(\sigma)$, où σ' est la représentation binaire de $|U(\sigma)|$, et soit c tel que $C(\sigma) < C_M(\sigma) + c$ pour toute chaîne σ . Soit d tel que $C(\sigma) \leq |\sigma| + d$ pour toute chaîne σ .

Soit σ une chaîne de taille supérieure à $c + d + k + 2^{k+c+d}$, et soit $\tau \prec \sigma$ un préfixe de taille $c + d + k$. Soit n l'entier codé par τ . Soit ρ de taille n tel que $\tau\rho \preceq \sigma$. Alors, $C_M(\tau\rho) = C(\rho) \leq |\rho| + d = (|\tau\rho| - k - c - d) + d = |\tau\rho| - k - c$. Donc, $C(\tau\rho) \leq |\tau\rho| - k$.

Solution 1.15. Fixons k , et trouvons des chaînes σ, τ telles que

$$C(\tau\rho) \geq C(\tau) + C(\rho) + k.$$

Soit c la constante telle que $C(\sigma) \leq |\sigma| + c$, pour toute chaîne σ . Considérons alors une chaîne σ telle que $C(\sigma) \geq |\sigma|$ et pour laquelle on a un préfixe τ tel que $C(\tau) \leq |\tau| - k - c$, et soit ρ tel que $\sigma = \tau\rho$. On a alors

$$C(\tau\rho) \geq |\sigma| = |\tau| + |\rho| \geq (C(\tau) + k + c) + C(\rho) - c = C(\tau) + C(\rho) + k.$$

Enfin, il est clair que $C(\langle \tau, \rho \rangle) \geq^+ C(\tau\rho)$ pour toutes chaînes σ, τ . Donc, également pour tout k il existe des chaînes τ, ρ telles que $C(\langle \tau, \rho \rangle) \geq C(\tau) + C(\rho) + k$.

Solution 2.3. On définit la machine M telle que $M(\sigma) = f(U(\sigma))$. Ainsi,

$$K(f(\tau)) \leq^+ K_M(f(\tau)) \leq^+ K(\tau),$$

pour toute chaîne τ .

Solution 2.6. On définit la machine M telle que $M(\sigma)$ cherche τ, ρ pour lesquelles $\tau\rho = \sigma$ et telles que $U(\tau) = |\rho|$. La machine renvoie alors ρ . Notons que si de telles chaînes τ, ρ existent, elles sont uniques, car U est sans préfixe. La machine M est elle aussi sans préfixe : si $M(\tau\rho) \downarrow$, cela signifie $U(\tau) = |\rho|$. Si à présent $M(\tau\rho') \downarrow$ pour $\rho' \prec \rho$ ou $\rho \prec \rho'$, cela signifierait aussi $U(\tau) = |\rho'| \neq |\rho|$, ce qui est impossible. On obtient enfin l'inégalité $K(\sigma) \leq^+ |\sigma| + K(|\sigma|)$ via la machine M .

Soit $\log_2^{(n)}(\sigma)$ l'application n fois de $\log_2$ à σ . Si l'on répète l'inégalité récursivement, alors pour tout n on a $K(\sigma) \leq^+ |\sigma| + \sum_{k=1}^n \log^{(n)}(|\sigma|) + K(\log^{(n)}(|\sigma|))$,

pour tout σ . Pour $n = 1$, cela nous donne

$$\begin{aligned} K(\sigma) &\leq^+ |\sigma| + \log_2(|\sigma|) + K(\log_2(|\sigma|)) \\ &\leq^+ |\sigma| + \log_2(|\sigma|) + \log_2(\log_2(|\sigma|)) + 2\log_2(\log_2(\log_2(|\sigma|))), \end{aligned}$$

ce qui est bien une amélioration.

Solution 2.7. Supposons $K(\sigma) < |\sigma| + c$, pour toute chaîne σ . Pour n fixé, on a $\sum_{|\sigma|=n} 2^{-K(\sigma)} > \sum_{|\sigma|=n} 2^{-|\sigma|-c} = 2^{-c} \sum_{|\sigma|=n} 2^{-|\sigma|} = 2^{-c}$. Pour $n = 2^c + 1$, on a alors $\sum_{|\sigma| \leq n} 2^{-K(\sigma)} = n2^{-c} > 1$, ce qui est une contradiction.

Solution 2.11. Soit T un arbre calculable tel que $[T]$ sont les chemins de $\mathcal{P}$. Alors, le plus petit chemin de $[T]$ pour l'ordre lexicographique est approchable par la gauche : il est approximé par $\sigma_t 0^\infty$ pour σ_t le nœud de taille t le plus à gauche de T .

Solution 4.7. Cela découle simplement de la proposition 2.4, qui implique

$$K(\sigma_0 \oplus \sigma_1) =^+ K(\langle \sigma_0, \sigma_1 \rangle) \leq^+ K(\sigma_0) + K(\sigma_1),$$

pour toutes chaînes σ_0, σ_1 .

Chapitre 17

Solution 3.4. On définit la fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ qui sur $\langle n, i, e \rangle$ pour $n \geq 2$ renvoie $\langle n, 1 - i, e' \rangle$ où e' est tel que $W_{e'} = \{f(a) : a \in W_e\}$, et qui sur $\langle 1, i, e \rangle$ renvoie $\langle 1, 1 - i, e \rangle$.

Solution 3.6. Un examen attentif de la preuve montre que l'on a l'uniformité.

Solution 3.9. Il suffit de voir que toute classe $\Pi_2^0(\emptyset')$ dense contient un ensemble low, par la méthode des extensions finies.

Solution 3.10. Considérons un ouvert $\mathcal{U} = \bigcup_{\tau \in W} [\tau]$ contenant tous les éléments qui ne sont pas des sauts Turing, et des chaînes σ, τ telles que $X' \succeq \sigma$ pour tout $X \succeq \tau$. Montrons qu'il existe une extension $\sigma^* \succeq \sigma$, avec $\sigma^* \in W$, et une extension $\tau^* \succeq \tau$ telle que $X' \succeq \sigma^*$ pour tout $X \succeq \tau^*$.

Soit $\mathcal{V}_m$ une suite décroissante de classes Σ_1^0 telle que $\bigcap_m \mathcal{V}_m$ est vide et telle que $\mathcal{V}_m \cap [\tau]$ est non vide pour tout m . Soit $a_m > |\sigma|$ le code de la machine qui s'arrête sur l'entrée a_m ssi son oracle appartient à $\mathcal{V}_m$.

Soient $\sigma_0 = \sigma$ et $\mathcal{W}_0 = [\tau]$. Soit aussi $e = |\sigma|$. Supposons définies des chaînes $\sigma_n \succeq \dots \succeq \sigma_0 = \sigma$ telles que $|\sigma_i| = |\sigma| + i$, ainsi que des ouverts $\mathcal{W}_i$ pour $i < n$ tels que $\mathcal{W}_{i+1} = \{X : \Phi_{e+i}(X, e+i) \downarrow\}$ si $\sigma_n(e+i) = 1$, et tels

que $\mathcal{W}_{i+1} \subseteq \{X : \Phi_{e+i}(X, e+i) \uparrow\}$ si $\sigma_n(e+i) = 0$. Notons que $\bigcap_{k \leq n} \mathcal{W}_k$ ne contient que des éléments X tels que $\sigma_n \prec X'$. Supposons aussi que

(1) la classe $(\bigcap_{k \leq n} \mathcal{W}_k) \cap \mathcal{V}_m$ est non vide pour tout m .

Notons que (1) est vérifié pour $n = 0$. Construisons σ_{n+1} et $\mathcal{W}_{n+1}$. Si $e+n$ est égal à a_m pour un certain m , alors $\sigma_{n+1} = \sigma_n 1$ et

$$\mathcal{W}_{n+1} = \mathcal{V}_m = \{X : \Phi_{e+n}(X, e+n) \downarrow\}.$$

On vérifie facilement (1). Sinon, on a deux possibilités :

▷ si $\{X : \Phi_{e+n}(X, e+n) \downarrow\} \cap \bigcap_{k \leq n} \mathcal{W}_k \cap \mathcal{V}_m$ est non vide pour tout m , alors $\sigma_{n+1} = \sigma_n 1$, $\mathcal{W}_{n+1} = \{X : \Phi_{e+n}(X, e+n) \downarrow\}$, et (1) est toujours vérifié ;

▷ sinon, cela signifie $\bigcap_{k \leq n} \mathcal{W}_k \cap \mathcal{V}_m \subseteq \{X : \Phi_{e+n}(X, e+n) \uparrow\}$ pour un certain m .

On définit alors $\sigma_{n+1} = \sigma_n 0$ et $\mathcal{W}_{n+1} = \bigcap_{k \leq n} \mathcal{W}_k \cap \mathcal{V}_m$, et l'on a toujours (1).

Soit $Y = \sigma_1 \prec \sigma_2 \prec \dots$. Montrons que Y n'est pas un saut Turing. Supposons que $Y = X'$ pour un ensemble X . Soit $m \in \mathbb{N}$. Comme $Y(a_m) = 1$, alors $\Phi_{a_m}(X, a_m) \downarrow$, et donc $X \in \mathcal{V}_m$. Il s'ensuit que $X \in \bigcap_m \mathcal{V}_m$, mais $\bigcap_m \mathcal{V}_m = \emptyset$, ce qui amène à une contradiction. Donc, $Y \in \mathcal{U}$ par hypothèse sur $\mathcal{U}$. On a ainsi un préfixe $\sigma_n \prec Y$ tel que $\sigma_n \in W$. Par (1), la classe $\bigcap_{k \leq n} \mathcal{W}_k \subseteq [\tau]$ est non vide. Elle est aussi ouverte, et il existe donc une extension $\tau_n \succeq \tau$ telle que $X' \succeq \sigma_n$ pour tout $X \succeq \tau_n$.

Pour finir, soit $\bigcap_n \mathcal{U}_n$ une classe Π_2^0 contenant tous les éléments qui ne sont pas des sauts Turing. En utilisant ce qui précède, on construit alors aisément par extension finie des éléments X, Y tels que $Y \in \bigcap_n \mathcal{U}_n$ et tels que $Y = X'$. Donc, $\bigcap_n \mathcal{U}_n$ contient aussi des éléments qui sont des sauts Turing, et la classe de tous les éléments qui ne sont pas des sauts n'est donc pas Π_2^0 .

Solution 3.11. Introduisons quelques notations. Pour une chaîne σ , on écrira $r(\sigma)$ pour désigner le ratio de 1 dans σ par rapport à sa taille. On notera $P_m(\sigma, \varepsilon)$ le prédicat $\ll r(\sigma \upharpoonright_n) < \varepsilon$ pour tout n tel que $m < n \leq |\sigma| \gg$.

Soit $\bigcup_n \mathcal{F}_n$ une classe Σ_2^0 contenant tous les éléments de densité supérieure positive, et soit $\bigcap_n \mathcal{U}_n$ le complémentaire de $\bigcup_n \mathcal{F}_n$. Soit σ une chaîne quelconque et soit ε un rationnel. Montrons qu'il doit exister une extension $\tau \succeq \sigma$ et un entier n tels que :

(1) $P_{|\sigma|}(\tau, \varepsilon)$;

(2) $\{X \succeq \tau : \forall m > |\sigma| \ P_{|\sigma|}(X \upharpoonright_m, \varepsilon)\} \subseteq \mathcal{F}_n$.

On fixe pour cette preuve un autre rationnel $\varepsilon' < \varepsilon$. Soit $\tau_0 = \sigma$. Notons que l'on a (1) avec τ_0 . Donc, si l'on a (2) avec τ_0 et $n = 0$, alors c'est terminé. Sinon, il doit exister $X \in \mathcal{U}_0$, avec $X \succeq \tau_0$ tel que $\forall m > |\sigma| \ P_{|\sigma|}(X \upharpoonright_m, \varepsilon)$. Soit $\tau_0 \preceq \rho_0 \prec X$ tel que $[\rho_0] \subseteq \mathcal{U}_0$. Supposons τ_n, ρ_n définis. Alors, on trouve une extension $\tau_{n+1} \succeq \tau_n$ telle que $P_{|\sigma|}(\tau_{n+1}, \varepsilon)$ et telle que $r(\tau_{n+1}) > \varepsilon'$. Si l'on a (2) avec τ_{n+1} et $n+1$, alors c'est terminé. Sinon, il doit exister $X \in \mathcal{U}_{n+1}$, avec $X \succeq \tau_{n+1}$ tel que $\forall m > |\sigma| \ P_{|\sigma|}(X \upharpoonright_m, \varepsilon)$. Soit $\tau_{n+1} \preceq \rho_{n+1} \prec X$ tel que $[\rho_{n+1}] \subseteq \mathcal{U}_{n+1}$. Supposons que l'on ne soit jamais dans le cas (2). Alors, on construit une suite $Y = \tau_0 \prec \tau_1 \prec \dots$ telle que $Y \in \bigcap_n \mathcal{U}_n$ et telle que $r(\tau_n) > \varepsilon'$

pour tout n . En particulier, la suite Y est de densité supérieure positive, et donc $Y \in \bigcup_n \mathcal{F}_n$, ce qui est une contradiction. On arrive donc nécessairement dans le cas (2), ce qui conclut notre démonstration intermédiaire.

Soit à présent $\bigcap_k \bigcup_n \mathcal{F}_{k,n}$ une classe Π_3^0 contenant tous les éléments de densité supérieure positive. Soit σ_0 tel que $P_0(\sigma_0, 2^{-1})$, et soit n_0 tel que

$$\{X \succeq \sigma_0 : \forall m > 0 \ P_0(X \upharpoonright_m, 2^{-1})\} \subseteq \mathcal{F}_{0,n_0}.$$

Supposons la chaîne σ_k et l'entier n_k définis. Alors, on définit $\sigma_{k+1} \succeq \sigma_k$ et n_{k+1} tels que $P_{|\sigma_k|}(\sigma_{k+1}, 2^{-k-2})$ et tels que

$$\{X \succeq \sigma_{k+1} : \forall m > |\sigma_k| \ P_{|\sigma_k|}(X \upharpoonright_m, 2^{-k-2})\} \subseteq \mathcal{F}_{k+1,n_{k+1}}.$$

Soit $Y = \sigma_0 \preceq \sigma_1 \preceq \dots$. Pour tout k , on a par construction

$$\forall m > |\sigma_k| \ P_{\sigma_k}(Y \upharpoonright_m, 2^{-k-1}).$$

Par ailleurs, $Y \in \mathcal{F}_{n_k}$ pour tout k . Ainsi, $Y \in \bigcap_k \bigcup_n \mathcal{F}_{k,n}$. Il est de plus clair que la densité supérieure de Y est 0. Par suite, $\bigcap_k \bigcup_n \mathcal{F}_{k,n}$ contient aussi des éléments de densité supérieure 0, et la classe des éléments de densité supérieure positive n'est donc pas Π_3^0 .

Solution 4.7. Pour tout n , l'ensemble $\emptyset^{(n)}$ est Σ_n^0 . Considérons la classe générée par la réunion des cylindres $[0^m 1]$ pour $m \in \emptyset^{(n)}$. Il s'agit un ouvert qui admet une description effective Σ_n^0 et dont la représentation binaire de la mesure est égale à $\emptyset^{(n)}$.

Chapitre 18

Solution 1.3. En utilisant l'indication, pour $q = n$ et pour $a < 2^n$, l'ensemble des $X \in q^{\mathbb{N}}$ tels que la limite inférieure de la fréquence d'apparition de a dans ses préfixes ne converge pas vers $1/q$ est égale à $\bigcup_\varepsilon \bigcap_n \bigcup_{m \geq n} C_{\varepsilon,m}^{q,a}$. D'après l'inégalité de Hoeffding, pour chaque ε , la classe $\bigcap_n \bigcup_{m \geq n} C_{\varepsilon,m}^{q,a}$ est une classe Π_2^0 de mesure 0, et telle que la mesure de chaque composante $\bigcup_{m \geq n} C_{\varepsilon,m}^{q,a}$ est bornée par $2a_n/(1 - a_1)$, où $a_m = e^{-2m\varepsilon^2}$ (voir le développement fait au début du chapitre 18).

Chapitre 19

Solution 3.3. Dans une direction, il suffit de transformer une classe $\Pi_2^0(\emptyset^{(n)})$ en une classe Π_{n+2}^0 . Chaque ouvert $\Sigma_1^0(\emptyset^{(n)})$ est décrit par un ensemble de chaînes $\Sigma_1^0(\emptyset^{(n)})$ qui est aussi Σ_{n+1}^0 , d'après le théorème 5-5.5. On peut donc transformer chaque ouvert uniformément en classe Σ_{n+1}^0 , l'intersection étant alors Π_{n+2}^0 .

Pour l'autre direction, il suffit d'appliquer le théorème 17-4.4.

Solution 3.6. Étant donné une classe $\Pi_1^0 \mathcal{P}$ et une fonctionnelle Φ_e , on définit à l'aide du théorème du point fixe le code a_e de la fonction $\emptyset'$ -calculable qui fait la chose suivante : cherche une chaîne σ telle que $\mathcal{P} \cap [\sigma] \neq \emptyset$ et telle que $\Phi_e(\sigma, a_e) \downarrow$. Si une telle chaîne σ est trouvée, alors a_e code pour la fonction telle que $\Phi_{a_e}(\emptyset', a_e) = \Phi_e(\sigma, a_e) \downarrow$. Sinon, a_e reste partielle. Notons que le code a_e lui-même est uniformément calculable, et cela sans l'aide de $\emptyset'$.

Dans le même temps, et cette fois-ci sans l'aide de $\emptyset'$, on cherche la première chaîne σ telle que $\mathcal{P} \cap [\sigma]$ est pour le moment non vide et telle que $\Phi_e(\sigma, a_e) \downarrow$, et l'on énumère dans une classe Σ_2^0 le code de la classe $\Pi_1^0 \mathcal{P} \cap [\sigma]$. Si l'on s'aperçoit que $\mathcal{P} \cap [\sigma] = \emptyset$, on cherche une autre chaîne σ telle que $\mathcal{P} \cap [\sigma]$ est pour le moment non vide et telle que $\Phi_e(\sigma, a_e) \downarrow$, etc. Entre chaque énumération de codes de la forme $\mathcal{P} \cap [\sigma]$, on énumère dans notre classe Σ_2^0 un code correspondant à la classe $\mathcal{P}$, mais que l'on videra complètement une fois la prochaine chaîne σ trouvée. Il y a plusieurs deux possibilités :

▷ soit $\Phi(X, a_e) \uparrow$ pour tout $X \in \mathcal{P}$, auquel cas on arrivera à un moment où aucune chaîne σ ne sera trouvée, et notre classe Σ_2^0 contiendra un code de $\mathcal{P}$ qui ne sera jamais vidé ;

▷ soit $\Phi(X, a_e) \downarrow$ pour un élément $X \in \mathcal{P}$, auquel cas on finira par trouver une chaîne σ telle que $\mathcal{P} \cap [\sigma]$ est non vide et telle que $\Phi(\sigma, a_e) \downarrow$. Il suffit alors de synchroniser notre recherche avec le calcul effectué à l'aide de l'arrêt pour s'assurer que $\Phi_{a_e}(\emptyset', a_e) = \Phi_e(\sigma, a_e) \downarrow$ pour la même chaîne σ . À ce moment notre classe Σ_2^0 ne contient plus que des codes de classes vides, ainsi que le code d'une sous-classe Π_1^0 de $\mathcal{P}$ telle que $\Phi_e(X, a_n) \downarrow = \Phi_{a_e}(\emptyset', a_e)$ pour tout X dans cette sous-classe.

On construit de cette manière pour tout e une classe Σ_2^0 qui intersecte chaque classe Π_1^0 non vide (et qui contient donc tous les ensembles calculatoirement dominés) et telle que pour tout X dans cette classe, soit $\Phi(X, a_e) \uparrow$, soit $\Phi(X, a_e) \downarrow = \Phi_{a_e}(a_e)$. L'intersection de ces classes Σ_2^0 contient tous les ensembles calculatoirement dominés, et aucun d'entre eux n'est de degré DNC($\emptyset'$).

Solution 4.9. Soit $\mathcal{F}$ une classe Π_1^0 de mesure positive, et soit $X \in \mathcal{F}$ un ensemble pour lequel il existe n et $\varepsilon < 1$ tels que pour tout $m \geq n$ on a

$$\lambda(\mathcal{F} \mid [X \upharpoonright_m]) < 1 - \varepsilon.$$

On définit le test suivant. Soit $U_0 = \{X \upharpoonright_n\}$ et supposons $U_k \subseteq 2^{<\mathbb{N}}$ défini tel qu'un préfixe de Y est dans U_k . Pour tout $\sigma \in U_k$, on cherche t tel que $\lambda(\mathcal{F}[t] \mid [\sigma])$ soit $< 1 - \varepsilon$. Si cela arrive, on énumère dans U_{k+1} un ensemble fini de chaînes générant l'ouvert/fermé $\mathcal{F}[t] \cap [\sigma]$. Par induction, un préfixe de X est nécessairement dans U_{k+1} et, par définition, $\lambda([U_{k+1}]) < (1 - \varepsilon)\lambda([U_k])$. Donc, X est capturé par $\bigcap_k [U_k]$, qui est un test de Martin-Löf.

Chapitre 20

Solution 1.2. Il s'agit de la même preuve, sauf que l'on crée un ensemble borné de requêtes pour « copier » la machine U^A par une machine sans oracle. Ainsi, à la place de chercher $n > 2e$ tel que $n \in W_e[t+1]$ et tel que

$$\sum_{\substack{U(\tau)[t+1] \downarrow = m, \\ \text{avec } m \geq n}} 2^{-|\tau|} < 2^{-e-1},$$

on cherche $n > 2e$ tel que $n \in W_e[t+1]$ et tel que

$$\sum_{\substack{U^{At+1}m(\tau)[t+1] \downarrow, \\ \text{avec } m \geq n}} 2^{-|\tau|} < 2^{-e-1}.$$

Si l'on trouve un tel n , on l'énumère dans A à l'étape $t+1$, puis pour tout τ tel que $U^{At+1}m(\tau)[t+1] \downarrow = x$ pour $m \geq n$, on énumère $(x, |\tau|)$ dans L à l'étape $t+1$.

Solution 1.5. Soit A un ensemble, et soit $g : \mathbb{N} \rightarrow 2^{<\mathbb{N}}$ une fonction partielle A -calculable telle que si $\Phi_e^A(e) \downarrow$, alors $g(e) \prec A$ et $\Phi_e^{g(e)}(e) \downarrow$. En particulier,

$$K^A(g(e)) \leq^+ K(e) \leq^+ 2 \log_2(e).$$

Si de plus A est low-pour- K , alors $K(g(e)) \leq^+ K^A(g(e))$. Soit $c \in \mathbb{N}$ tel que pour tout e , $K(g(e)) \leq 2 \log_2(e) + c$. Soit U une machine universelle telle que $K_U = K$. Pour décider si $\Phi_e^A(e) \downarrow$, il suffit de considérer chaque chaîne σ de longueur au plus $2 \log_2(e) + c$, de tester à l'aide de $\emptyset'$ si $U(\sigma) \downarrow$ et $U(\sigma) \prec A$ (ce qui est possible, car A est Δ_2^0) et si $\Phi_e^{U(\sigma)}(e) \downarrow$. Si c'est le cas pour une chaîne σ , alors $\Phi_e^A(e) \downarrow$. Réciproquement, si $\Phi_e^A(e) \downarrow$, alors $g(e) \downarrow$; or, $K(g(e)) \leq 2 \log_2(e) + c$, si bien qu'il existe une chaîne σ de longueur au plus $2 \log_2(e) + c$ telle que $U(\sigma) = g(e)$. Cette chaîne σ satisfait la propriété désirée.

Solution 3.3. D'après le théorème de la base low, il existe un ensemble MLR low, et donc en particulier incomplet et Δ_2^0 . Cet élément étant MLR, il n'est évidemment pas K -trivial.

Chapitre 22

Solution 2.2. Soit $F(x)$ une formule de l'arithmétique du second ordre telle que $F(0)$ est vrai et telle que pour tout x , $F(x) \rightarrow F(x+1)$. Par le schéma de compréhension (9), l'ensemble $X = \{n \in \mathbb{N} : F(x)\}$ existe. On a notamment $0 \in X$, et pour tout x , si $x \in X$ alors $x+1 \in X$. Par le schéma d'induction (10), pour tout x , on a $x \in X$, donc pour tout x , $F(x)$ est vrai.

Solution 5.3. Il suffit de voir que la formule $n \in X$ pour une variable d'ensemble X est Δ_0^0 .

Solution 5.7. Soit X un ensemble low non calculable. On construit à l'aide du théorème 10-3.31 un ensemble 1-générique $G \leq_T \emptyset'$ tel que $X \oplus G$ calcule $\emptyset'$. Notons que $G' \leq_T G \oplus \emptyset'$, donc G est low.

Solution 5.8. On lance en parallèle la construction avec des arbres calculables de deux ensembles X_0, X_1 calculatoirement dominés. On encode dans X_0 l'information nécessaire pour passer de l'étape n à $n+1$ dans la construction de X_1 , puis alternativement dans X_1 l'information nécessaire pour passer de l'étape n à $n+1$ dans la construction de X_0 . L'encodage est fait par rapport au choix du chemin (aller à gauche ou aller à droite) de X_0 ou X_1 dans leurs arbres calculables respectifs. L'ensemble $X_0 \oplus X_1$ peut alors décoder la construction : il suffira d'y inclure un encodage d'un ensemble arbitraire Z .

Solution 5.10. Soit $X \in \mathcal{I}$, et soit $Y \leq_T X$. Il existe en particulier n tel que $X \in \mathcal{I}_n$. Par clôture de $\mathcal{I}_n$ par réduction Turing, $Y \in \mathcal{I}_n$, donc $Y \in \mathcal{I}$. Soient $X, Y \in \mathcal{I}$. En particulier, il existe n_0, n_1 tels que $X \in \mathcal{I}_{n_0}$ et $Y \in \mathcal{I}_{n_1}$. Soit $n = \max(n_0, n_1)$. On a $X, Y \in \mathcal{I}_n$, donc par clôture de $\mathcal{I}_n$ par jointure, $X \oplus Y \in \mathcal{I}_n$, donc $X \oplus Y \in \mathcal{I}$.

Solution 5.11. Soit $\mathcal{I}_0 \subseteq \mathcal{I}_1 \subseteq \dots$ la suite définie par $\mathcal{I}_n = \{X \in 2^{\mathbb{N}} : X \leq_T Z_n\}$. En particulier, $\mathcal{I}_n$ est un idéal Turing, donc par l'exercice 5.10, $\mathcal{I} = \bigcup_n \mathcal{I}_n$ est un idéal Turing.

Solution 6.2. La formule $F(e) \equiv \Phi_e^X(e) \downarrow$ étant Σ_1^0 avec paramètre X , ACA_0 prouve l'existence de l'ensemble $Y = \{e : \Phi_e^X(e) \downarrow\}$ à l'aide du schéma de compréhension arithmétique.

Réciproquement, par la proposition 6.1, il suffit de montrer que $\text{RCA}_0 + \forall X \exists Y Y = X'$ prouve le schéma de compréhension Σ_1^0 . Soit $F(x)$ une formule Σ_1^0 avec paramètres $X_0, \dots, X_{k-1}$. En particulier, avec le schéma de compréhension Δ_1^0 , l'ensemble $X_0 \oplus \dots \oplus X_{k-1}$ existe, et l'ensemble $Y = (X_0 \oplus \dots \oplus X_{k-1})'$ existe donc également. Soit $Z = \{x : F(x)\}$. L'ensemble Z est $X_0 \oplus \dots \oplus X_{k-1}$ -c.e., donc Y -calculable. Par le schéma de compréhension Δ_1^0 avec paramètre Y , l'ensemble Z existe.

Solution 6.6. Pour tout $\sigma \in 2^{\mathbb{N}}$, soit $I_\sigma \subseteq [0, 1]$ l'intervalle des réels dont l'expansion binaire commence par $0.\sigma$. Soit

$$\mathcal{C} = \{X \in 2^{\mathbb{N}} : \forall n \exists m > n \ x_m \in I_{\upharpoonright n}\}.$$

La classe $\mathcal{C}$ est $\Pi_1^0(\emptyset')$, non vide et contient exactement les $X \in 2^{\mathbb{N}}$ tels que $0.X$ est un point de convergence d'une sous-suite de $(x_n)_{n \in \mathbb{N}}$. Par la proposition 8-3.3 relativisée à $\emptyset'$, il existe un arbre $\emptyset'$ -calculable $T \subseteq 2^{<\mathbb{N}}$ tel que $[T] = \mathcal{C}$.

Soit $T \subseteq 2^{<\mathbb{N}}$ un arbre $\emptyset'$ -calculable infini. Par l'exercice 8-7.13, il existe un ensemble calculable $S = \{\sigma_0, \sigma_1, \dots\} \subseteq 2^{<\mathbb{N}}$ contenant exactement une chaîne de chaque longueur, et telle que $[T] = [\widehat{S}]$, où $\widehat{S}$ est la clôture par préfixe de S . La suite $(x_n)_{n \in \mathbb{N}}$ définie par $x_n = 0.\sigma_n$ est telle que les points de convergence de ses sous-suites sont exactement de la forme $0.X$, où $X \in [T]$.

Solution 7.4. La construction est exactement celle de la proposition 7.3, mais en appliquant le théorème de base Π_1^0 calculatoirement dominée à la place du théorème d'évitement de cône.

Solution 7.5. La construction est exactement celle de la proposition 7.3, mais en appliquant le théorème de base Π_1^0 low à la place du théorème d'évitement de cône.

Solution 7.7. Pour tout n , soit $T_n = \{\sigma \in T : |\sigma| = n\}$, et soit $\mathcal{U}_n = \bigcup_{\sigma \in T_n} [\sigma]$. Notons que $[T] = \bigcap_n \mathcal{U}_n$. Comme pour tout entier n , on a $\mathcal{U}_{n+1} \subseteq \mathcal{U}_n$, alors $\lambda([T]) = \liminf_n \lambda(\mathcal{U}_n)$. Comme T_n est sans préfixe, on a

$$\lambda(\mathcal{U}_n) = \sum_{\sigma \in T_n} \lambda([\sigma]) = \sum_{\sigma \in T_n} \frac{1}{2^n} = \frac{|\{\sigma \in T : |\sigma| = n\}|}{2^n}.$$

Chapitre 23

Solution 2.2. Il suffit de reprendre l'exemple 2.1 en ayant cette fois deux éléments ∞_1 et ∞_2 tels que

$$\begin{aligned} \infty_1 + \infty_2 &= \infty_2, \quad \infty_2 + \infty_1 = \infty_1, \\ \infty_1 \times \infty_2 &= \infty_1, \quad \infty_2 \times \infty_1 = \infty_2, \quad \infty_1 + \infty_1 = \infty_1, \quad \infty_2 + \infty_2 = \infty_2, \end{aligned}$$

ainsi que $i + n = n + i = i$ pour $i \in \{\infty_1, \infty_2\}$ et $n \in \omega$, de même que $i \times n = n \times i = i$ pour $i \in \{\infty_1, \infty_2\}$ et $n \in \omega$ avec $n \neq 0$, et finalement $i \times 0 = 0 \times i = 0$ pour $i \in \{\infty_1, \infty_2\}$.

Solution 2.4.

- (1) La stratégie est la même que pour montrer $x + y = y + x$. On montre d'abord $0 \times x = 0$, puis $(x + 1) \times y = x \times y + y$. Cette étape utilise l'associativité et la commutativité de l'addition. On utilise ces deux résultats pour montrer $x \times y = y \times x$. Les détails sont laissés aux lecteurs.
- (2) On a $x \times (y + 0) = x \times y = x \times y + x \times 0$ par (4) et (6) de $\mathbf{Q}$. Supposons $x \times (y + z) = (x \times y) + (x \times z)$. Alors, $x \times (y + (z + 1)) = x \times ((y + z) + 1) = (x \times (y + z)) + x = (x \times y) + (x \times z) + x = x \times y + x \times (z + 1)$ par (7) de $\mathbf{Q}$, par l'hypothèse d'induction, et par l'associativité et la commutativité de l'addition.
- (3) On a $(x \times y) \times 0 = 0 = x \times (y \times 0)$ par (6) de $\mathbf{Q}$. Supposons $(x \times y) \times z = x \times (y \times z)$. Alors, $(x \times y) \times (z + 1) = ((x \times y) \times z) + (x \times y) = x \times (y \times z) + (x \times y) = x \times (y \times z + y) = x \times (y \times (z + 1))$ par (7) de $\mathbf{Q}$, par l'hypothèse d'induction et la distributivité.

Solution 2.6. Supposons $x < y$ et soit $z \neq 0$. On a $x + a = y$ pour $a \neq 0$. Donc, $(x + a) \times z = y \times z$. Par distributivité de la multiplication, on a

$$x \times z + a \times z = y \times z.$$

Comme $a \neq 0$ et $z \neq 0$, on a $a \times z \neq 0$ par injectivité de la multiplication. Donc, $x \times z < y \times z$.

Solution 2.7. Montrons l'existence par induction sur a . Soit $b \neq 0$.

On a $0 = 0 \times b + 0$. Supposons $\exists q, r \leq a$ ($0 \leq r < b \wedge a = qb + r$). Si $r < b - 1$, alors $a + 1 = qb + (r + 1)$. Si $r = b - 1$, alors $a + 1 = (q + 1)b$. Dans tous les cas, $\exists q, r \leq a + 1$ ($0 \leq r < b \wedge a + 1 = qb + r$). Par ID_0^0 , l'existence est assurée pour tout a .

Montrons l'unicité. Supposons $a = qb + r = q'b + r'$ avec $0 \leq r, r' < b$. Supposons par l'absurde $q \neq q'$. On peut supposer sans perte de généralité $q + 1 \leq q'$. Par l'exercice 2.6, on a $qb + b \leq q'b$. Comme $r < b$, on a $qb + r < qb + b \leq q'b \leq q'b + r'$. Donc, $qb + r < q'b + r'$, ce qui contredit $qb + r = q'b + r'$. Ainsi, $q = q'$, ce qui implique $r = r'$.

Solution 3.6. Nous allons le montrer dans le cas Σ_n^0 . Le cas Π_n^0 est similaire. Soit P_n le schéma d'induction ordonné pour les formules Σ_n^0 .

Montrons $\text{Q} \vdash \text{IS}_n^0 \rightarrow \text{P}_n$. Soit $F(x)$ une formule Σ_n^0 telle que

$$\forall x[(\forall y < x F(y)) \rightarrow F(x)],$$

et soit z fixé. Montrons que $F(z)$ est vrai. Soit $G(x) \equiv (x \leq z \rightarrow \forall y < x F(y))$. Clairement, $G(0)$ est vrai, car $\forall y < 0 F(y)$ est vrai. De plus, si $G(x)$ est vrai, alors $G(x + 1)$ est vrai. En effet, si $x \leq z$, alors $\forall y < x F(y)$ est vrai, donc $F(x)$ est vrai, donc $\forall y < x + 1 F(y)$ est vrai, donc $G(x + 1)$ est vrai. Si $x > z$, alors trivialement $G(x + 1)$ est vrai. Il s'ensuit que G satisfait les prémisses de IS_n^0 , donc $\forall x G(x)$ est vrai. En particulier, $G(z)$ est vrai, donc $\forall y < z F(y)$ est vrai, donc $F(z)$ est vrai.

Montrons $\text{Q} \vdash \text{P}_n \rightarrow \text{IS}_n^0$. Soit $F(x)$ une formule Σ_n^0 telle que $F(0)$ et $\forall x(F(x) \rightarrow F(x + 1))$. En particulier, $\forall x[(\forall y < x F(y)) \rightarrow F(x)]$, donc par P_n , $\forall x F(x)$ est vrai.

Solution 5.4. (i) La relation $x \in y$ est Δ_0^0 pour tout y ; l'ensemble $\{x \in \mathbb{N} : x \in y\}$ existe donc par le schéma de compréhension Δ_1^0 .

(ii) Soit X un ensemble et soit $b \in \mathbb{N}$. La fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ définie par $f(0) = 0$ et $f(n + 1) = f(n) + 2^n$ si $n \in X$, et $f(n + 1) = f(n)$ si $n \notin X$ est primitive récursive en X , donc prouvable dans RCA_0 . Par construction, $f(b)$ est le code canonique de l'ensemble X .

Solution 5.7. Le cas $n = 0$ est trivial, car RCA_0 prouve les deux énoncés.

Supposons ID_n^0 . Soient $F(x, y)$, $G(x, y)$ des formules respectivement Π_{n-1}^0 , Σ_{n-1}^0 telles que $\forall x(\exists y F(x, y) \leftrightarrow \forall y G(x, y))$ et soit $a \in \mathbb{N}$. Montrons qu'avec ces hypothèses-là l'ensemble $A = \{x \in \mathbb{N} : x < a \wedge \exists y F(x, y)\}$ existe. On a

$$\forall x \exists y (F(x, y) \vee \neg G(x, y)).$$

Par le théorème 3.9, $\text{ID}_n^0 \rightarrow \text{BS}_n^0$, et par BS_n^0 , il existe un $b \in \mathbb{N}$ tel que

$$\forall x \leq a \exists y \leq b (F(x, y) \vee \neg G(x, y)).$$

Ainsi, $A = \{x \in \mathbb{N} : x < a \wedge \exists y \leq b F(x, y)\}$, donc est Π_{n-1}^0 . Par le théorème 3.7,

on a $\text{B}\Sigma_n^0 \rightarrow \text{I}\Sigma_{n-1}^0$, et par la proposition 5.6, $\text{I}\Sigma_{n-1}^0 \rightarrow \text{BC}\Sigma_{n-1}^0$, or $\text{BC}\Sigma_{n-1}^0 \leftrightarrow \text{BC}\Pi_{n-1}^0$, donc l'ensemble A existe.

Supposons $\text{BC}\Delta_n^0$. Soit $F(x)$ un prédicat Δ_n^0 tel que $F(0)$ est vrai et pour tout entier n , on a $F(n) \rightarrow F(n+1)$. Soit $a \in \mathbb{N}$. Montrons que $F(a)$ est vrai. Par $\text{BC}\Delta_n^0$, l'ensemble $X = \{x \leq a : F(x)\}$ existe. Par le schéma de compréhension Δ_1^0 , l'ensemble $Y = X \cup \{y \in \mathbb{N} : y > x\}$ existe. On a $0 \in Y$ et $\forall x (x \in Y \rightarrow x+1 \in Y)$. Par l'axiome d'induction (10), qui prouvable dans RCA_0 , $\forall x, x \in Y$. En particulier, $a \in Y$, donc $F(a)$ est vrai.

Solution 6.9. Cette transformation purement syntaxique se fait par induction sur la structure des formules, en remplaçant les formules basiques $x \in X$ par la formule $\Sigma_1 F_{\exists}(x)$, et $x \notin X$ par la formule $\Sigma_1 \neg F_{\forall}(x)$, où $F_{\exists}$ et $F_{\forall}$ définissent l'ensemble $X \in \mathcal{S}^{\mathcal{N}}$ à l'aide de paramètres dans $\mathcal{M}$. Les quantificateurs existentiels sortent de la formule à l'aide de $\text{B}\Sigma_1^0$, qui est satisfait par $\mathcal{M}$.

Solution 6.16. Il suffit de montrer l'exercice dans le cas où F est Δ_0^0 . On imposera de plus que $H(\sigma)$ soit monotone sur σ , c'est-à-dire que si $H(\sigma)$ est vrai, $H(\tau)$ est vrai pour tout $\tau \succeq \sigma$. On procède par induction sur F .

Les cas suivants sont immédiats.

Si $F(X) \equiv x \in X$, alors $H(\sigma) \equiv x < |\sigma| \wedge \sigma(x) = 1$.

Si $F(X) \equiv x \notin X$, alors $H(\sigma) \equiv x < |\sigma| \wedge \sigma(x) = 0$.

Si $F(X)$ est parmi $t_1 = t_2$, $t_1 \neq t_2$, ou $t_1 < t_2$, alors la variable X n'apparaît pas dans F , et H est donc identique à F .

Si $F(X)$ est parmi $F_1(X) \wedge F_2(X)$ ou $F_1(X) \equiv F_2(X)$, alors par hypothèse d'induction, il existe des formules Δ_0^0 monotones $H_1(\sigma)$ et $H_2(\sigma)$ telles que pour tout ensemble $G \subseteq M$, $\mathcal{M} \cup \{G\} \models F_i(G) \leftrightarrow \exists k H_i(G \upharpoonright_k)$. Alors, si $H(\sigma)$ est défini par $H_1(\sigma) \wedge H_2(\sigma)$ ou $H_1(\sigma) \vee H_2(\sigma)$ en fonction du cas.

Si $F(X) \equiv \exists x < t F_1(x, X)$, alors soit $H_1(x, \sigma)$ la formule Δ_0^0 correspondant à F_1 par hypothèse d'induction. La formule $H(\sigma) \equiv \exists x < t H_1(x, \sigma)$ satisfait l'induction.

Le cas non trivial est la quantification universelle bornée : si

$$F(X) \equiv \forall x < t F_1(x, X),$$

alors soit $H_1(x, \sigma)$ la formule Δ_0^0 correspondant à F_1 par hypothèse d'induction. Soit $H(\sigma) \equiv \forall x < t H_1(x, \sigma)$. Montrons que $H(\sigma)$ satisfait l'induction. Soit $G \subseteq M$. Si $\mathcal{M} \cup \{G\} \models \exists k H(G \upharpoonright_k)$, alors par définition, il existe $k \in M$ tel que $\mathcal{M} \cup \{G\} \models \forall x < t H_1(x, G \upharpoonright_k)$, donc $\mathcal{M} \cup \{G\} \models \forall x < t \exists k H_1(x, G \upharpoonright_k)$. Par hypothèse d'induction, $\mathcal{M} \cup \{G\} \models \forall x < t F_1(x, G)$ et $\mathcal{M} \cup \{G\} \models F(G)$.

Réciproquement, supposons que $\mathcal{M} \cup \{G\} \models F(G)$.

Alors, par définition, $\mathcal{M} \cup \{G\} \models \forall x < t F_1(x, G)$, donc par hypothèse d'induction, $\mathcal{M} \cup \{G\} \models \forall x < t \exists k F_1(x, G \upharpoonright_k)$. Par $\text{B}\Sigma_1^0$ et monotonie de F_1 , $\mathcal{M} \cup \{G\} \models \exists k \forall x < t F_1(x, G \upharpoonright_k)$.

Chapitre 24

Solution 1.4. Montrons $WKL \leq_\omega KL$. Les idéaux Turing satisfaisant WKL sont les idéaux de Scott, tandis que ceux satisfaisant KL sont les idéaux de saut. Sachant que pour tout X , X' est de degré PA relativement à X , tout idéal de saut est un idéal de Scott.

Montrons $KL \not\leq_\omega WKL$. Soit $\mathcal{I}$ un idéal de Scott ne contenant que des ensembles low (voir l'exercice 22-7.5). Alors, $\mathcal{I}$ n'est pas un idéal de saut, puisque $\emptyset \in \mathcal{I}$ mais $\emptyset' \notin \mathcal{I}$. Ainsi, $\mathcal{I} \models WKL$, mais $\mathcal{I} \not\models KL$.

Solution 2.2. Soit $\mathcal{I}$ un idéal Turing satisfaisant Q . Montrons que $\mathcal{I} \models P$. Soit X une instance de P dans $\mathcal{I}$. Comme $P \leq_c Q$, il existe une instance $\tilde{X}$ de Q telle que pour toute Q -solution $\tilde{Y}$ de $\tilde{X}$, $X \oplus \tilde{Y}$ calcule une P -solution de X . Comme $\mathcal{I}$ est clos par réduction Turing, $\tilde{X} \in \mathcal{I}$. Comme $\mathcal{I} \models Q$, il existe une Q -solution $\tilde{Y}$ de $\tilde{X}$ dans $\mathcal{I}$. Par clôture de $\mathcal{I}$ par jointure effective, $X \oplus \tilde{Y} \in \mathcal{I}$, et enfin par clôture de $\mathcal{I}$ par réduction Turing, $\mathcal{I}$ contient une P -solution de X . Donc, $\mathcal{I} \models P$.

Solution 2.3. Soit T , un arbre infini calculable à branchement fini, dont tous les chemins sont de degré PA relativement à $\emptyset'$ (voir la proposition 8-7.4). Soit $\tilde{X} \leq_T T$ une instance de J . En particulier, $\tilde{X}$ est calculable, donc $\tilde{X}' \equiv_T \emptyset'$, et $\tilde{X}'$ n'est donc pas de degré PA relativement à $\emptyset'$. Comme $\tilde{X}'$ est l'unique solution de $\tilde{X}$ et ne T -calcule pas de chemin infini de T , $KL \not\leq_c J$.

Solution 2.7. Supposons que P est calculatoirement vrai. Soit X une instance de P . En particulier, X est une instance de Id , donc l'unique solution est X ; or, $X \oplus X \geq X$ et X calcule une P -solution de X , donc $X \oplus X$ calcule une solution de X .

Supposons que $P \leq_c Id$. Soit X une instance de P . Soit $\tilde{X}$ une instance X -calculable de Id telle que pour toute Id -solution $\tilde{Y}$ de $\tilde{X}$, $X \oplus \tilde{Y}$ calcule une solution de P . Comme $\tilde{X}$ est l'unique Id -solution $\tilde{Y}$ et $X \geq_T X \oplus \tilde{X}$, alors X calcule une P -solution de X , donc P est calculatoirement vrai.

Solution 3.3. Supposons que P est uniformément vrai à l'aide de la fonctionnelle Φ . Soit Ψ la fonctionnelle définie par $\Psi(A \oplus B) = \Psi(B)$. Alors, $P \leq_W Id$ comme en témoignent les fonctionnelles Φ et Ψ .

Réciproquement, supposons que $P \leq_W Id$ à l'aide des fonctionnelles Φ et Ψ . Soit Γ la fonctionnelle définie par $\Gamma(X) = \Psi(X \oplus \Phi(X))$. Pour toute instance X de P , $\Gamma(X)$ est une solution de X .

Solution 3.7. Montrons que $LLPO \leq_W LPO$. Soit Φ la fonctionnelle définie par $\Phi^X(n) = 1 - X(n)$. Soit Ψ la fonctionnelle telle $\Psi^{X \oplus i}(0)$ cherche un entier n tel que $X(n) = 1$ si $i = 1$, et renvoie $(n+1) \bmod 2$. Si $i = 0$, la fonctionnelle renvoie 0 (par convention, on aurait pu choisir 1). Si X est une instance de $LLPO$, alors Φ^X est une instance de LPO dont la solution est 1 ssi il existe n tel que $X(n) = 1$.

Ainsi, si $i = 1$, l'instance X de LLPO n'aura qu'une seule solution, et la recherche de $\Psi^{X \oplus i}(0)$ s'arrête et renvoie la bonne solution. Si $i = 0$, les deux valeurs sont des solutions de X , $\Psi^{X \oplus i}(0) = 0$ est donc une solution valide. Ainsi, Φ et Ψ témoignent de $\text{LLPO} \leq_W \text{LPO}$.

Montrons que $\text{LPO} \not\leq_W \text{LLPO}$. Raisonnons par l'absurde. Supposons que $\text{LPO} \leq_W \text{LLPO}$ est témoigné par des fonctionnelles Φ et Ψ . La suite 1^∞ est une instance de LPO dont la solution est 0. Ainsi, $\Phi(1^\infty)$ est une instance de LLPO.

Cas 1. $\Phi(1^\infty) = 0^\infty$, autrement dit, les valeurs 0 et 1 sont toutes deux LLPO-solutions de $\Phi(1^\infty)$. Alors, $\Psi^{1^\infty \oplus 0}(0) \downarrow = \Psi^{1^\infty \oplus 1}(0) \downarrow = 0$. Soit n la taille maximale de l'usage de ces deux calculs. Alors, $\Psi^{0^n 1^\infty \oplus 0}(0) \downarrow = \Psi^{0^n 1^\infty \oplus 1}(0) \downarrow = 0$, ce qui n'est pas la LPO-solution de $0^n 1^\infty$.

Cas 2. $\Phi(1^\infty)(s) = 1$ pour un $s \in \omega$. Alors, $i = (s + 1) \bmod 2$ est l'unique solution de la LLPO-instance $\Phi(1^\infty)$ et $\Psi^{1^\infty \oplus i}(0) \downarrow = 0$. Soit n plus grand que l'usage de $\Phi(1^\infty)(s)$ et de $\Psi^{1^\infty \oplus i}(0)$. En particulier, $\Phi(1^n 0^\infty)(s) = 1$ et $\Psi^{1^n 0^\infty \oplus i}(0) \downarrow = 0$; or, i est la LLPO-solution de $\Phi(1^n 0^\infty)$, mais 0 n'est pas la LPO-solution de $1^n 0^\infty$. Contradiction !

Solution 3.8. Supposons que LLPO est uniformément vrai, avec Φ comme fonctionnelle Turing. En particulier, 0^∞ est une instance de LLPO, donc $\Phi(0^\infty)(0) \downarrow = i$ pour un $i < 2$. Soit n la longueur de l'usage de ce calcul.

En particulier, $X_0 = 0^n 10^\infty$ et $X_1 = 0^n 010^\infty$ sont deux instances de LLPO telles que $\Phi(X_0) = \Phi(X_1) = i$ par la propriété de l'usage, mais X_0 et X_1 ont des solutions opposées. Contradiction !

Solution 3.10. Soient Φ, Ψ des fonctionnelles Turing qui témoignent de $\text{P} \leq_W \text{Q}$. Soit Φ_1 la fonctionnelle définie par $\Phi_1(\bigoplus_n X_n) = \bigoplus_n \Phi(X_n)$ et soit Ψ_1 la fonctionnelle définie par $\Psi_1(\bigoplus_n X_n, \bigoplus_n Y_n) = \bigoplus_n \Psi(X_n, Y_n)$. Les fonctionnelles Φ_1 et Ψ_1 témoignent de $\widehat{\text{P}} \leq_W \widehat{\text{Q}}$.

Solution 3.12. Montrons que $\text{WKL} \leq_W \widehat{\text{LLPO}}$. Pour tout $\sigma \in 2^{<\mathbb{N}}$, soit Γ_σ la fonctionnelle Turing définie par $\Gamma_\sigma^T(2s + i) = 1$ ssi s est le plus petit entier tel que $\forall \tau \in 2^{|\sigma|+s+1}$ ($\sigma i \preceq \tau \rightarrow \tau \notin T$). Autrement dit, soit $T \subseteq 2^{<\mathbb{N}}$ un arbre binaire, et $\sigma \in T$ un nœud dont la sous-branche dans T est infinie. Si $i < 2$ est une solution de Γ_σ^T vue comme une instance de LLPO, σi est un nœud dont la sous-branche dans T est encore infinie. Soit Φ la fonctionnelle Turing définie par $\Phi^X = \bigcup_{\sigma \in 2^{<\mathbb{N}}} \Gamma_\sigma^X$. Soit Ψ la fonctionnelle Turing définie pour tout n par

$$\Psi\left(T \oplus \left(\bigoplus_{\sigma} i_{\sigma}\right)\right)(n) = i_{\Psi(T \oplus (\bigoplus_{\sigma} i_{\sigma})) \upharpoonright n}.$$

Les fonctionnelles Φ et Ψ témoignent de $\text{WKL} \leq_W \widehat{\text{LLPO}}$.

Montrons que $\widehat{\text{LLPO}} \leq_W \text{WKL}$. Soit Φ la fonctionnelle Turing définie par

$$\Phi \oplus_n X_n = \{\sigma \in 2^{<\mathbb{N}} : \forall n, s < |\sigma| X_n(2s + \sigma(n)) = 0\}.$$

Notons que $\Phi \oplus_n X_n$ est un arbre binaire infini dont tous les chemins sont des $\widehat{\text{LLPO}}$ -solutions de $\bigoplus_n X_n$. Soit Ψ la fonctionnelle définie par $\Psi(X \oplus Y) = Y$. Les fonctionnelles Φ et Ψ témoignent de $\widehat{\text{LLPO}} \leq_W \text{WKL}$.

Solution 4.4. Montrons que $KL \leq_\omega^2 J$. Soit T une instance de KL jouée par le joueur 1. Autrement dit, T est un arbre infini à branchement fini. Le joueur 2 joue T comme une instance de J . Le joueur 1 n'a pas d'autre choix que de jouer l'unique J -solution de T , à savoir T' . Le joueur 2 joue T' comme une instance J . Le joueur 1 joue alors T'' , et le joueur 2 T'' -calcule une solution de T , car tout arbre infini à branchement fini admet une solution calculable en son double saut. Le joueur 2 a donc une stratégie qui le fait gagner en au plus trois tours.

Montrons que $J \not\leq_\omega^1 KL$. Le joueur 1 joue un arbre T infini à branchement fini calculable n'ayant pas de chemin $\emptyset'$ -calculable. Le joueur 2 doit alors jouer une J -instance $X \leq T$, et le joueur 1 joue alors la J -solution X' . Comme $X' \leq_T T'$ et T n'admet pas de solution T' -calculable, le joueur 2 ne gagne pas non plus au second tour.

Solution 5.6. Notons que pour tout problème Q , si $P \leq_{sW} Q$, alors $P \leq_W Q$, car il suffit de ne pas tenir compte de l'instance X de P .

Supposons que Q est un cylindre pour $\leq_{sW}$ et que $P \leq_W Q$, comme en témoignent des fonctionnelles Φ_0, Ψ_0 . Soient alors Φ_1, Ψ_1 les fonctionnelles témoignant de $Id \times Q \leq_{sW} Q$. Soit Φ_2 la fonctionnelle définie par $\Phi_2(X) = \Phi_1(X \oplus \Phi_0(X))$ et Ψ_2 la fonctionnelle définie par $\Psi_2(Y) = \Psi_0(\Psi_1(Y))$. Les fonctionnelles Φ_2 et Ψ_2 témoignent de $P \leq_{sW} Q$.

Supposons maintenant que pour tout problème P , si $P \leq_W Q$ alors $P \leq_{sW} Q$. En particulier, pour P le problème $Id \times Q$, on a bien $Id \times Q \leq_W Q$, donc $Id \times Q \leq_{sW} Q$. Autrement dit, Q est un cylindre pour $\leq_{sW}$.

Chapitre 25

Solution 1.4. Voir le théorème 2.5.

Solution 2.15. Soit $h : \mathbb{N} \rightarrow \mathbb{N}$ un module de $\emptyset'$. On définit la fonction h -calculable $f : [\mathbb{N}]^2 \rightarrow 2$ en associant à $x < y$ la couleur 1 ssi $h(x) < y$. Tout ensemble infini H homogène pour f est homogène de couleur 1, et la fonction qui à n associe le n -ième élément de H domine h . La proposition 2.11 permet de conclure.

Solution 3.5. Pour une formule $\Delta_0^0 \Phi(x, y, n)$ et un paramètre du premier ordre k l'énoncé est

$$\exists x < k \forall y \exists n \Phi(x, y, n) \vee \exists z \forall x < k \exists y < z \forall n \Phi(x, y, n).$$

Il est clair que si $\exists y < z \forall n \Phi(x, y, n)$, alors $\forall a \exists y < z \forall n < a \Phi(x, y, n)$. Inversement, si $\forall y < z \exists n \neg \Phi(x, y, n)$, alors par $B\Sigma_1^0$ (prouvable dans RCA_0)

$$\exists a \forall y < z \exists n < a \neg \Phi(x, y, n).$$

Ainsi, on a

$$\exists y < z \forall n \Phi(x, y, n) \Leftrightarrow \forall a \exists y < z \forall n < a \Phi(x, y, n).$$

Solution 3.7. Trivial.

Solution 3.9. Voir la preuve du lemme 14-4.6.

Solution 3.20. Soient A et C fixés. Par le corollaire 11-4.21, il existe un idéal de Scott $\mathcal{I}$ tel que pour tout $X \in \mathcal{I}$, C n'est pas $\Sigma_1^0(X)$. Supposons (H1), car sinon nous avons déjà la solution désirée. Nous allons construire deux ensembles infinis $G^0 \subseteq A^0$, et $G^1 \subseteq A^1$ satisfaisant pour tout $e_0, e_1 \in \mathbb{N}$ le contrat

$$\mathcal{R}_{e_0, e_1} : W_{e_0}^{G^0} \neq C \vee W_{e_1}^{G^1} \neq C.$$

Les ensembles G^0 et G^1 vont être construits par le forcing de Dzhafarov-Jockusch avec l'idéal $\mathcal{I}$. Montrons le lemme suivant.

« Soit $c = (\sigma_0, \sigma_1, X)$ une condition et soient $e_0, e_1 \in \mathbb{N}$. Il existe une extension (τ_0, τ_1, Y) de c forçant $\mathcal{R}_{e_0, e_1}$. »

PREUVE DU LEMME. Soit $W = \{x \in \mathbb{N} : c \Vdash x \in W_{e_0}^{G^0} \vee x \in W_{e_1}^{G^1}\}$. Par le lemme 3.15, W est $\Sigma_1^0(X)$ avec $X \in \mathcal{I}$, et C n'est pas $\Sigma_1^0(X)$, donc $W \neq C$. Deux cas se présentent.

- ▷ Cas 1. Il existe $x \in W \setminus C$. Par la proposition 3.17, il existe une extension (τ_0, τ_1, Y) de c et $i < 2$ tel que $(\tau_i, Y) \Vdash x \in W_{e_i}^G$.
- ▷ Cas 2. Il existe $x \in C \setminus W$. Par la proposition 3.17, il existe une extension (τ_0, τ_1, Y) de c et $i < 2$ tel que $(\tau_i, Y) \Vdash x \notin W_{e_i}^G$.

Soit F un filtre suffisamment générique pour le forcing de Dzhafarov-Jockusch, et soit $(G^0, G^1) = \dot{F}$. Par le lemme 3.13, G^0 et G^1 sont tous les deux infinis. Par définition d'une condition de forcing, $G^0 \subseteq A^0$ et $G^1 \subseteq A^1$. Par le lemme ci-dessus, (G^0, G^1) satisfont $\mathcal{R}_{e_0, e_1}$ pour tout $e_0, e_1 \in \mathbb{N}$. Donc, soit C n'est pas $\Sigma_1^0(G^0)$, soit C n'est pas $\Sigma_1^0(G^1)$. Ainsi, soit G^0 , soit G^1 satisfait le théorème, ce qui conclut la preuve.

Solution 3.21. Soient A et f fixés. Par l'exercice 22-7.4, il existe un idéal de Scott $\mathcal{I}$ qui ne contient que des ensembles calculatoirement dominés. En particulier, pour tout $X \in \mathcal{I}$, la fonction f est X -hyperimmune. Supposons (H1), car sinon nous avons déjà la solution désirée. Nous allons construire deux ensembles infinis $G^0 \subseteq A^0$, $G^1 \subseteq A^1$ satisfaisant pour tous $e_0, e_1 \in \mathbb{N}$ le contrat

$$\mathcal{R}_{e_0, e_1} : \Phi_{e_0}^{G^0} \not\geq f \vee \Phi_{e_1}^{G^1} \not\geq f.$$

Les ensembles G^0 et G^1 vont être construits par le forcing de Dzhafarov-Jockusch avec l'idéal $\mathcal{I}$. Montrons le lemme suivant.

« Soit $c = (\sigma_0, \sigma_1, X)$ une condition et soient $e_0, e_1 \in \mathbb{N}$. Il existe une extension (τ_0, τ_1, Y) de c forçant $\mathcal{R}_{e_0, e_1}$. »

PREUVE DU LEMME. Soit $g : \mathbb{N} \rightarrow \mathbb{N}$ la fonction qui sur tout n cherche un ensemble fini $F \subseteq \mathbb{N}$ tel que $c \Vdash \Phi_{e_0}^{G^0}(n) \downarrow \in F \vee \Phi_{e_1}^{G^1}(n) \downarrow \in F$. Si un tel ensemble est trouvé, $g(n) = \max F$, sinon $g(n)$ n'est pas définie. Par le lemme 3.15, la fonction g est partielle X -calculable avec $X \in \mathcal{I}$. Deux cas se présentent.

- ▷ Cas 1. g est totale. Alors, comme f est X -hyperimmune, il existe $n \in \mathbb{N}$ tel que $g(n) < f(n)$. Par définition de g , il existe un ensemble F satisfaisant $\max F < f(n)$ et tel que $c \upharpoonright F \vdash \Phi_{e_0}^{G^0}(n) \downarrow \in F \vee \Phi_{e_1}^{G^1}(n) \downarrow \in F$. Par la proposition 3.17, il existe une extension (τ_0, τ_1, Y) de c et $i < 2$ tel que $(\tau_i, Y) \Vdash \Phi_{e_i}^{G^i}(n) \downarrow \in F$.
- ▷ Cas 2. Il existe $n \in \mathbb{N}$ tel que $g(n)$ n'est pas définie. Par définition de g , pour tout ensemble fini F , $c \upharpoonright F \not\vdash \Phi_{e_0}^{G^0}(n) \downarrow \in F \vee \Phi_{e_1}^{G^1}(n) \downarrow \in F$. Par compacité, $c \upharpoonright \Phi_{e_0}^{G^0}(n) \downarrow \vee \Phi_{e_1}^{G^1}(n) \downarrow$. Par la proposition 3.17, il existe une extension (τ_0, τ_1, Y) de c et $i < 2$ tel que $(\tau_i, Y) \Vdash \Phi_{e_i}^{G^i}(n) \uparrow$.

Soit F un filtre suffisamment générique pour le forcing de Dzhafarov-Jockusch, et soit $(G^0, G^1) = \dot{F}$. Par le lemme 3.13, G^0 et G^1 sont tous les deux infinis. Par définition d'une condition de forcing, $G^0 \subseteq A^0$ et $G^1 \subseteq A^1$. Par le lemme ci-dessus, (G^0, G^1) satisfont $\mathcal{R}_{e_0, e_1}$ pour tous $e_0, e_1 \in \mathbb{N}$. Donc, soit f est G^0 -hyperimmune, soit f est G^1 -hyperimmune. Ainsi, soit G^0 , soit G^1 satisfait le théorème, ce qui conclut la preuve.

Solution 3.22. Il s'agit exactement de la construction de l'exercice 3.21, en forçant les contrats suivants pour tous $e_0, e_1 \in \mathbb{N}$:

$$\mathcal{R}_{e_0, e_1} : \Phi_{e_0}^{G^0} \not\geq f_0 \vee \Phi_{e_1}^{G^1} \not\geq f_0 ;$$

$$\mathcal{S}_{e_0, e_1} : \Phi_{e_0}^{G^0} \not\geq f_1 \vee \Phi_{e_1}^{G^1} \not\geq f_1 ;$$

$$\mathcal{T}_{e_0, e_1} : \Phi_{e_0}^{G^0} \not\geq f_2 \vee \Phi_{e_1}^{G^1} \not\geq f_2 .$$

Soit F un filtre suffisamment générique pour le forcing de Dzhafarov-Jockusch, et soit $(G^0, G^1) = \dot{F}$. Par le lemme 3.13, G^0 et G^1 sont tous les deux infinis. Par définition d'une condition de forcing, $G^0 \subseteq A^0$ et $G^1 \subseteq A^1$. Par le lemme de l'exercice 3.21, (G^0, G^1) satisfont $\mathcal{R}_{e_0, e_1}$, $\mathcal{S}_{e_0, e_1}$ et $\mathcal{T}_{e_0, e_1}$ pour tous $e_0, e_1 \in \mathbb{N}$. Donc, pour tout $i < 3$, soit f_i est G^0 -hyperimmune, soit f_i est G^1 -hyperimmune. Par le principe des tiroirs, au moins deux fonctions f_0, f_1 et f_2 sont simultanément G^0 -hyperimmunes ou G^1 -hyperimmunes. Ainsi, soit G^0 , soit G^1 satisfait le théorème, ce qui conclut la preuve.

Solution 3.35. La classe

$$\mathcal{C} = \{X : \forall e \forall s (W_{e,s} \subseteq X \vee W_{e,s} \subseteq \overline{X}) \rightarrow |W_{e,s}| \leq 2e + 2\}$$

est une classe Π_1^0 non vide, qui ne contient que des ensembles effectivement bi-immunes. En particulier, $\mathcal{C}$ contient un membre low, donc Δ_2^0 .

Solution 3.36. Il s'agit d'une généralisation immédiate de la proposition 3.34. Nous définissons une suite $\emptyset'$ -calculable $\sigma_0 \prec \sigma_1 \prec \dots$ de chaînes dans $k^{<\mathbb{N}}$ en alternant la concaténation de longues suites de i pour chaque $i < k$ de telle sorte que la fonction principale de $\mathbb{N} \setminus A_i$ diagonalise contre toutes les fonctions calculables.

Supposons σ_{ke+j} définie pour $j < k$, $\sigma_{ke+j+1} = \sigma_{ke+j}j^{\ell+1}0$ si $\Phi_e(|\sigma_{ke+j}|+1) \downarrow = \ell$, et $\sigma_{ke+k} = \sigma_2e0$ sinon. Aussi vérifie-t-on sans peine que si Φ_e est totale, alors pour $m = |\sigma_{ke+j}| + 1$ on a $p_A(m) \geq |\sigma_{ke+k+1}| - 1 > \Phi_e(m)$.

Solution 3.37. Soit $A_0 \sqcup A_1 \sqcup A_2 = \mathbb{N}$ une 3-partition Δ_2^0 telle que pour tout $i < 3$, $\mathbb{N} \setminus A_i$ est hyperimmune. Son existence est assurée par l'exercice 3.36. On peut voir cette 3-partition comme une instance g de RT_3^1 définie par $g(x) = i$ tel que $x \in A_i$.

Rappelons que la *fonction principale* d'un ensemble infini $H \subseteq \mathbb{N}$ est la fonction $p_H : \mathbb{N} \rightarrow \mathbb{N}$ qui à n associe le $(n+1)$ -ième élément de H . Rappelons également que H est hyperimmune ssi p_H n'est bornée par aucune fonction calculable. Pour tout $i < 3$, soit f_i la fonction principale de $\mathbb{N} \setminus A_i$. Notons que f_i est hyperimmune pour tout $i < 3$, et que pour tout ensemble G infini g -homogène de couleur i , G est un sous-ensemble de A_i , donc G est un sous-ensemble de $\mathbb{N} \setminus A_j$ pour tout $j \neq i$. Il s'ensuit que sa fonction principale p_G domine f_j pour tout $j \neq i$, donc que aucun des f_j n'est G -hyperimmune pour $j \neq i$.

Soit $h : \mathbb{N} \rightarrow 2$ une instance de RT_2^1 . Par l'exercice 3.22, il existe un ensemble infini h -homogène H tel que au moins deux parmi les trois fonctions f_0, f_1 et f_2 sont H -hyperimmunes. Il s'ensuit par l'observation précédente que H ne calcule pas d'ensemble infini g -homogène.

Solution 4.2. Montrons tout d'abord que tout ensemble suffisamment générique pour $\mathbb{P}$ est infini. Soit $\mathcal{W}_x \subseteq \mathbb{P}$ l'ensemble des conditions de Mathias (σ, X) telles que $\sigma(y) = 1$ pour un $y > x$. Montrons que $\mathcal{W}_x$ est dense dans $\mathbb{P}$. Soit $(\sigma, X) \in \mathbb{P}$. En particulier, X est infini, donc il existe $y > x$ tel que $y \in X$.

La condition $(\sigma \cup \{y\}, X \setminus \{0, \dots, y\})$ est une extension de (σ, X) dans $\mathcal{W}_x$. Ainsi, pour tout filtre F suffisamment générique, $F \cap \mathcal{W}_x \neq \emptyset$, et donc $\dot{F}$ sera infini.

Soit Z un ensemble calculable, et soit $\mathcal{W}_Z \subseteq \mathbb{P}$ l'ensemble des conditions de Mathias (σ, X) telles que $X \subseteq Z$ ou $X \subseteq \bar{Z}$. Montrons que $\mathcal{W}_Z$ est dense dans $\mathbb{P}$. Soit $(\sigma, X) \in \mathbb{P}$. Alors, $X \cap Z$ et $X \cap \bar{Z}$ sont tous les deux calculables, et au moins l'un des deux est infini. Alors, soit $(\sigma, X \cap Z)$, soit $(\sigma, X \cap \bar{Z})$ est une condition de Mathias valide étendant (σ, X) , et dans tous les cas, cette extension sera dans $\mathcal{W}_Z$. Pour tout filtre F suffisamment générique, $F \cap \mathcal{W}_Z \neq \emptyset$, et donc $\dot{F} \subseteq^* Z$ ou $\dot{F} \subseteq^* \bar{Z}$. Ainsi, $\dot{F}$ est un ensemble cohésif pour tous les ensembles calculables.

Solution 4.11. Soit C un ensemble non Σ_1^0 . Soit $\mathbb{P}$ l'ensemble des conditions de Mathias (σ, X) telles que X est un ensemble calculable. Montrons que pour toute condition (σ, X) et tout $e \in \mathbb{N}$, il existe une extension (τ, Y) forçant $W_e^G \neq C$. Soit (σ, X) une condition, et soit $U = \{x \in \mathbb{N} : (\exists \rho \subseteq X)x \in W_e^{\sigma \cup \rho}\}$. L'ensemble U est Σ_1^0 contrairement à C . S'il existe $x \in C \setminus U$, alors par définition de U , (σ, X) force déjà $x \notin W_e^G$, donc force $W_e^G \neq C$. Sinon, il existe $x \in U \setminus C$. Soit $\rho \subseteq X$ tel que $x \in W_e^{\sigma \cup \rho}$. Alors, la condition $(\sigma \cup \rho, X \setminus \{0, \dots, |\rho|\})$ est

une extension de (σ, X) forçant $x \in W_e^G$, donc forçant $W_e^G \neq C$. Ainsi, pour tout ensemble G suffisamment générique pour $\mathbb{P}$, C ne sera pas $\Sigma_1^0(G)$. De plus, pour toute condition (σ, X) et tout n , soit $(\sigma, X \cap R_n)$, soit $(\sigma, X \cap \bar{R}_n)$ est une extension valide, donc si G est un ensemble suffisamment générique pour $\mathbb{P}$, alors G sera cohésif pour $(R_n)_{n \in \mathbb{N}}$.

Solution 4.18. Soit $(R_n)_{n \in \mathbb{N}}$ la suite calculable donnée par $R_x = \{y : f(\{x, y\}) = 1\}$. Par l'exercice 4.11, il existe un ensemble D cohésif pour la suite $(R_n)_{n \in \mathbb{N}}$ tel que C n'est pas $\Sigma_1^0(D)$. En particulier, pour tout $x \in D$, $\lim_{y \in D} f(\{x, y\})$ existe. Soit $g : D \rightarrow 2$ le coloriage défini par $g(x) = \lim_{y \in D} f(\{x, y\})$. Par la proposition 2.2 et l'exercice 3.20, il existe un ensemble infini $H \subseteq D$ homogène pour g tel que C n'est pas $\Sigma_1^0(H \oplus D)$. Par le fait 4.1, $H \oplus D$ calcule un ensemble infini Y homogène pour f . En particulier, C n'est pas $\Sigma_1^0(Y)$.

Chapitre 27

Solution 3.7.

- (1) Après avoir écrasé ω Borks, il reste ω Borks. À l'étape $\omega \times \omega = \omega + \omega + \omega + \dots$, tous les Borks apparus durant l'écrasement du n -ième bloc de ω Borks seront anéantis durant l'écrasement du $(n+1)$ -ième bloc de ω Borks. Il n'y a donc plus de Borks à l'étape $\omega \times \omega$.

- (2) Soit $f(\alpha)$ le nombre de Borks restant après avoir écrasé α Borks. On a clairement $f(\omega) = \omega^2$, et de manière générale $f(\omega^n) = \omega^{(n+1)}$. En particulier,

$$f(\omega^\omega) = f(\sup_n \omega^n) = \sup_n f(\omega^n) = \sup_n \omega^{(n+1)} = \omega^\omega.$$

Il y a donc ω^ω Borks après les ω^ω premières étapes. On répète alors l'argument de (1) pour voir qu'il n'y a plus de Borks à l'étape $\omega^\omega \times \omega$.

- (3) Étant donné les Borks présents à l'étape α , soit A_α le sous-ensemble de Borks qui finiront par être foudroyés à une certaine étape (A_α est défini via l'axiome de compréhension). Soit à présent $f(\alpha)$ la plus petite étape à laquelle tous les Borks de A_α sont foudroyés (l'axiome de remplacement implique que le supremum des étapes auxquels les Borks de A_α sont foudroyés est bien un ordinal). On définit $\alpha_0 = \omega$ puis, pour tout n , on définit $\alpha_{n+1} = f(\alpha_n)$. Si $\alpha_n = \alpha_{n+1}$ pour un certain n , cela signifie que plus aucun Bork ne sera foudroyé à partir de l'étape α_n et donc forcément qu'il n'y a plus de Bork à cette étape. Dans ce cas, tous les Borks seront anéantis.

Sinon, $\alpha_n < \alpha_{n+1}$ pour tout n . Soit $\beta = \sup_n \alpha_n$. On a alors

$$f(\beta) = f(\sup_n \alpha_n) = \sup_n f(\alpha_n) = \sup_n \alpha_{n+1} = \beta.$$

Comme $f(\beta) = \beta$ cela signifie que plus aucun Bork ne sera foudroyé à partir de l'étape β et donc forcément qu'il n'y a plus de Bork à cette étape. Dans tous les cas, tous les Borks finiront par disparaître.

Solution 5.3. Il s'agit de montrer par induction que si $|a| = \alpha$ pour $a \in \mathcal{O}$, alors

$$\{|b| : b <_o a\} = \{\beta : \beta < \alpha\}.$$

Solution 5.5. La preuve se fait par induction sur b via l'ordre bien fondé $<_o$.

Si $b = 1$, alors par définition $|a +_o b| = |a| = |a| + |b|$.

Si $b = 2^c$, alors $|a +_o b| = |2^{a+_oc}| = \text{succ}(|a +_o c|) = \text{succ}(|a| + |c|) = |a| + |b|$.

Si $b = 3 \times 5^e$, alors $|a +_o b| = \sup_n |a +_o \Phi_e(n)| = \sup_n (|a| + |\Phi_e(n)|) = |a| + |b|$.

Solution 5.12. La fonction f sur l'entrée n renvoie le code d'une énumération

$$0 < 1 < 2 < \dots$$

Si à un moment on s'aperçoit que $n \in \emptyset'$, la fonction choisit un nombre k pour lequel rien n'a encore été décidé et énumère dorénavant des axiomes tels que $m < k$ pour tout autre entier m , tout en gardant l'ordre usuel sur les entiers différents de k .

Solution 5.17. On a $0 = |T| = |T|_{\text{KB}}$ pour l'arbre vide T . Soit α un ordinal et supposons $|T| \leq |T|_{\text{KB}}$ pour tout arbre T tel que $|T| < \alpha$. Soit T un arbre tel que $|T| = \alpha$. Alors, pour chaque nœud σ de T de taille 1, on a $|T \upharpoonright_\sigma| < \alpha$. Par hypothèse d'induction, on a donc $|T \upharpoonright_\sigma| \leq |T \upharpoonright_\sigma|_{\text{KB}}$ pour chaque $\sigma \in T$ de taille 1. Dans le contexte de T , on a $|T \upharpoonright_\sigma| = |\sigma|$ et $|T \upharpoonright_\sigma|_{\text{KB}} \leq |\sigma|_{\text{KB}}$ (car l'ordre $<_{\text{KB}}$ sur $T \upharpoonright_\sigma$ se plonge dans l'ordre $<_{\text{KB}}$ sur T). Par définition, on a donc

$$|T| = \sup\{|\sigma| + 1 : \sigma \in T \text{ de taille } 1\} \leq \sup\{|\sigma|_{\text{KB}} + 1 : \sigma \in T \text{ de taille } 1\} = |T|_{\text{KB}}.$$

Chapitre 28

Solution 1.11. Montrons que chaque ensemble $\mathcal{T}_{<\omega(\alpha+k)}$ est $\Sigma_{\alpha+2k}^0$ et chaque ensemble $\mathcal{T}_{\leq\omega(\alpha+k)+p}$ est $\Pi_{\alpha+2k+1}^0$. Pour tout $p \in \mathbb{N}$, l'ensemble $\mathcal{T}_{\leq p}$ est Π_1^0 uniformément en p : c'est l'ensemble des codes qui énumèrent des arbres de hauteurs inférieures ou égales à p .

Supposons que pour $\alpha = 0$ ou limite, pour $k \in \mathbb{N}$ et pour tout $p \in \mathbb{N}$ l'ensemble $\mathcal{T}_{\leq\omega(\alpha+k)+p}$ soit $\Pi_{\alpha+2k+1}^0$ uniformément en p, k et un code de α . Alors,

$$\mathcal{T}_{<\omega(\alpha+k+1)} = \bigcup_p \mathcal{T}_{\leq\omega(\alpha+k)+p}$$

est $\Sigma_{\alpha+2(k+1)}^0$ uniformément en $k+1$ et en un code de α . De même, si $\mathcal{T}_{\leq\omega(\alpha+k)}$ est $\Pi_{\alpha+2k+1}^0$ uniformément en k et un code de α , pour $\alpha = 0$ ou limite et pour tout $k \in \mathbb{N}$, alors $\mathcal{T}_{<\omega(\alpha+\omega)} = \bigcup_k \mathcal{T}_{\leq\omega(\alpha+k)}$ est $\Sigma_{\alpha+\omega}^0$.

Supposons à présent que pour $\alpha = 0$ ou limite, pour $k \in \mathbb{N}$ l'ensemble $\mathcal{T}_{<\omega(\alpha+k)}$ est $\Sigma_{\alpha+2k}^0$ uniformément en k et en un code de α . Alors, pour $p \in \mathbb{N}$, l'ensemble $\mathcal{T}_{\leq\omega(\alpha+k)+p}$ est $\Pi_{\alpha+2k+1}^0$, car il s'agit de l'ensemble des codes d'arbres c. e. tels que pour tout nœud σ de taille $p+1$ énuméré, le code de l'arbre $T \upharpoonright_\sigma$ appartient à $\mathcal{T}_{<\omega(\alpha+k)}$, ce qui par hypothèse d'induction est une condition $\Pi_{\alpha+2k+1}^0$ uniformément en p, k et en un code de α .

On montre finalement l'existence d'une fonction $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ telle que :

- ▷ pour tout $\Sigma_{\alpha+2k}^0$ -code e d'un ensemble S_e , la fonction $n \mapsto f(e, n)$ est totale et vérifie $n \in S_e$ implique $f(e, n) \in \mathcal{T}_{<\omega(\alpha+k)}$, et $n \notin S_e$ implique $f(e, n)$ mal fondé ;
- ▷ pour tout $\Pi_{\alpha+2k+1}^0$ -code e d'un ensemble S_e , la fonction $n \mapsto f(e, n)$ est totale et vérifie $n \in S_e$ implique $f(e, n) \in \mathcal{T}_{\leq \omega(\alpha+k)}$ et $n \notin S_e$ implique $f(e, n)$ mal fondé.

Si e est le Π_1^0 -code d'un ensemble S_e , alors $f(e, n)$ renvoie le code d'un arbre c. e. vide tant que $n \in S_e$, et mal fondé sinon ($n \notin S_e$ est un événement Σ_1^0). Si e est le Σ_α^0 -code d'un ensemble $\bigcup_m F_{a_m}$, où chaque F_{a_m} est de code a_n , alors $f(e, n)$ renvoie le résultat de la fonction OR de l'exercice 29-5.11 sur $\{f(a_m, n) : m \in \mathbb{N}\}$.

Si e est le Π_α^0 -code d'un ensemble $\bigcap_m F_{a_m}$, où chaque F_{a_m} est de code a_n , alors $f(e, n)$ renvoie le résultat de la fonction AND du lemme 29-5.10 sur l'ensemble

$$\{f(a_m, n) : m \in \mathbb{N}\}.$$

On laisse au lecteur le soin de vérifier que f a bien la propriété demandée.

Solution 4.6. Il faut reprendre les relations du lemme 1.3 pour montrer que les réunions et intersections finies de classes Σ_α^0 sont uniformément Σ_α^0 . On peut ensuite utiliser cela pour obtenir à partir d'un Σ_α^0 -code de la classe $\bigcup_n \mathcal{B}_n$, un Σ_α^0 -code de la classe $\bigcup_n (\bigcup_{m \leq n} \mathcal{B}_m)$.

Solution 4.7. D'après l'exercice précédent, on peut supposer sans perte de généralité que nos classes sont croissantes. Il faut montrer par induction que l'on a une fonction calculable f telle que pour tout Σ_α^0 -code e d'une classe $\mathcal{A}$, alors $f(e)$ est le Σ_α^0 -code de l'ensemble $\{q \in \mathbb{Q} : \lambda(\mathcal{A}) > q\}$. Pour le montrer, nous aurons en fait besoin de créer des codes de fonctions $f_{i,\varepsilon}$ pour $i \in \{-1, 1\}$ et $\varepsilon \in \mathbb{Q}$ tels que pour tout Σ_α^0 -code e d'une classe $\mathcal{A}$, alors $f_{i,\varepsilon}(e)$ est le Σ_α^0 -code de l'ensemble

$$\{q \in \mathbb{Q} : \lambda(\mathcal{A}) > i \times q + \varepsilon\}.$$

L'idée est la suivante. Si $\mathcal{A} = \bigcup_m \mathcal{B}_m$, où chaque $\mathcal{B}_m$ est Π_β^0 pour $\beta < \alpha$, alors

$$\begin{aligned} \{q \in \mathbb{Q} : \lambda(\mathcal{A}) > q\} &= \bigcup_m \{q \in \mathbb{Q} : \lambda(\mathcal{B}_m) > q\} = \bigcup_{m,n} \{q \in \mathbb{Q} : \lambda(\mathcal{B}_m) > q + 2^{-n}\} \\ &= \bigcup_{m,n} \{q \in \mathbb{Q} : \lambda(2^{\mathbb{N}} \setminus \mathcal{B}_m) \leq 1 - (q + 2^{-n})\}. \end{aligned}$$

Notons que chaque ensemble $\{q \in \mathbb{Q} : \lambda(2^{\mathbb{N}} \setminus \mathcal{B}_m) \leq 1 - (q + 2^{-n})\}$ est le complémentaire de l'ensemble

$$\{q \in \mathbb{Q} : \lambda(2^{\mathbb{N}} \setminus \mathcal{B}_m) > 1 - (q + 2^{-n})\} = \{q \in \mathbb{Q} : \lambda(2^{\mathbb{N}} \setminus \mathcal{B}_m) > -q + (1 - 2^{-n})\},$$

qui a par induction un Σ_β^0 -code via $f_{-1, 1-2^{-n}}$.

Les détails sont laissés au lecteur.

Chapitre 29

Solution 2.9. Comme X est $\Sigma_1^1(Y)$ et Y est $\Delta_1^1(Z)$, on a

$$\begin{aligned} X &= \{n \in \mathbb{N} : \exists f \mathcal{A}(Y, f, n)\} \\ Y &= \{n \in \mathbb{N} : \exists f_1 \mathcal{A}_1(Z, f_1, n)\} \\ \mathbb{N} \setminus Y &= \{n \in \mathbb{N} : \exists f_2 \mathcal{A}_2(Z, f_2, n)\}, \end{aligned}$$

où $\mathcal{A}_1$ et $\mathcal{A}_2$ sont arithmétiques. Alors,

$$X = \left\{ n \in \mathbb{N} : \exists Y \forall m \left[\begin{aligned} &(m \in Y \wedge \exists f_1 \mathcal{A}_1(Z, f_1, m)) \vee \\ &(m \notin Y \wedge \exists f_2 \mathcal{A}_2(Z, f_2, m)) \end{aligned} \right] \wedge \exists f \mathcal{A}(Y, f, n) \right\}.$$

Solution 3.9. Pour chaque e , d'après le théorème 28-4.5, la classe

$$\mathcal{B}_{e,\alpha} = \{X : e \in \mathcal{O}_{=\alpha}^X\}$$

est borélienne. Pour e fixé, si $\alpha_1 \neq \alpha_2$ alors les classes $\mathcal{B}_{e,\alpha_1}$ et $\mathcal{B}_{e,\alpha_2}$ sont disjointes. Donc, par additivité dénombrable de la mesure, pour chaque rationnel $q > 0$ et chaque e il n'y a qu'un nombre fini d'ordinaux α tels que $\lambda(\mathcal{B}_{e,\alpha}) > q$. Soit $\alpha_{e,q}$ le supremum de ces ordinaux. Soit $\beta = \sup_{e,q} \alpha_{e,q}$ et $\mathcal{B} = \bigcup_e \mathcal{B}_{e,\beta}$. On a nécessairement $\beta < \omega_1$ et $\lambda(\mathcal{B}) = 0$. Notons que $\mathcal{B}$ est une classe borélienne.

Soit $\mathcal{A}$ une classe Π_1^1 . Alors, $\mathcal{A} = \mathcal{A} = \mathcal{A} = \bigcup_{\alpha < \omega_1} \{Y : e \in \mathcal{O}_{<\alpha}^Y\}$ pour un certain e . La classe $\bigcup_{\alpha \leq \beta} \{Y : e \in \mathcal{O}_{<\alpha}^Y\}$ est borélienne, et la classe $\bigcup_{\beta < \alpha < \omega_1} \{Y : e \in \mathcal{O}_{<\alpha}^Y\}$ est incluse dans $\mathcal{B}$ puisque si $e \in \mathcal{O}_{<\alpha}^Y$ pour $\alpha > \beta$, alors il existe e' tel que $e' \in \mathcal{O}_{=\beta}^Y$.

Solution 5.11. On définit une suite d'arbres $(U_i)_{i \in \mathbb{N}}$. $U_0 = T_0$. Si U_i est défini, alors U_{i+1} est obtenu en y mettant tous les nœuds de U_i taille inférieure ou égale à $i + 1$, et pour chacun de ces nœuds σ , on y ajoute les nœuds $\sigma\tau$ pour tout $\tau \in T_{i+1} \vee U_i \upharpoonright \sigma$. On renvoie finalement un code pour l'arbre $T = \lim_i U_i$.

Solution 6.2. Soit U un arbre Σ_1^1 bien fondé. Soit U_σ l'arbre bien fondé ssi $\sigma \notin U$. On calcule uniformément en X l'arbre bien fondé $T^X = \bigvee_{\sigma \prec X} U_\sigma$. On calcule finalement l'arbre c.e. T suivant : pour chaque chaîne $\sigma \in 2^{<\mathbb{N}}$ telle qu'un nœud m de taille 1 est énuméré dans T^σ , on énumère le nœud $\langle m, b(\sigma) \rangle$ de taille 1 dans T , en utilisant une bijection $b : 2^{<\mathbb{N}} \rightarrow \mathbb{N}$. Pour tout nœud $\langle \tau, b(\sigma_1) \dots b(\sigma_n) \rangle$ énuméré dans T avec $|\tau| = n$, si une chaîne τm est énumérée dans $T^{\sigma_{n+1}}$ pour une chaîne $\sigma_{n+1} \succeq \sigma_n$, on énumère alors $\langle \tau m, b(\sigma_1) \dots b(\sigma_n) b(\sigma_{n+1}) \rangle$ dans T .

Solution 6.5. Il suffit de montrer que le lemme 6.3 peut être obtenu uniformément. Soit $\mathcal{A}$ une classe Σ_1^1 telle que $\forall X \in \mathcal{A} \exists a \in \mathcal{O}^X \mathcal{B}(a, X)$, où $\mathcal{B}$ est un prédicat Π_1^1 . Soit e le code de l'arbre X -c.e. T_e^X qui est bien fondé ssi $X \notin \mathcal{A}$. Soit w_1 le code de l'arbre X -c.e. $T_{w_1}^X$ qui est bien fondé ssi $a \in \mathcal{O}^X$. On peut supposer de plus $|T_{w_1}^X| \geq |a|$ pour $a \in \mathcal{O}^X$. Soit w_2 le code de l'arbre X -c.e. $T_{w_2}^X$, qui est bien

fondé ssi $\mathcal{B}(a, X)$. Soit T^X l'arbre c. e. défini par

$$T_e^X \vee \bigvee_{a \in \omega} T_{w_1}^X \wedge T_{w_2}^X.$$

Notons que T^X est bien fondé pour tout X . On définit finalement en utilisant une bijection $b : 2^{<\mathbb{N}} \rightarrow \mathbb{N}$:

$$T = \{ \langle \sigma, b(\tau_1) \dots b(\tau_k) \rangle : |\sigma| = k, \tau_1 \preceq \dots \preceq \tau_k \text{ et } \sigma \in T^{\tau_k} \}.$$

On laisse au lecteur le soin de vérifier que l'ordinal $|T|$ vérifie bien la propriété demandée.

Solution 7.7. Il suffit de montrer $\omega_1^X > \omega_1^{ck}$. On utilise le théorème 4.3 pour voir X comme un ensemble que l'on énumère le long des ordinaux calculables. On définit la fonction $f : \mathbb{N} \rightarrow \omega_1^{ck}$ qui à n associe le plus petit ordinal $\alpha < \omega_1^{ck}$ tel que $X[\alpha] \upharpoonright_n = X \upharpoonright_n$. Ici, $X[\alpha]$ est « l'énumération » de X jusqu'à l'étape α . On a nécessairement $\sup_n f(n) = \omega_1^{ck}$, car sinon l'énumération de X serait achevée à une étape $\alpha < \omega_1^{ck}$, et X serait alors Δ_1^1 . Comme f est totale, son image est $\Delta_1^1(X)$. D'après le lemme 5.3 de majoration de Spector relativisé à X , on a donc $\omega_1^{ck} < \omega_1^X$.

Chapitre 30

Solution 4.4. Soit $\mathcal{A} = \{X : \forall n \exists \alpha < \omega_1^{ck} \Phi_e(X, n) \in \mathcal{O}_{<\alpha}^X\}$ pour $e \in \mathbb{N}$. Supposons $\lambda(\mathcal{A}) \geq r$ pour $r \in \mathbb{Q}$. Soit la fonction f qui à n associe le plus petit $\alpha < \omega_1^{ck}$ tel que $\lambda(\bigcap_{m \leq n} \{X : \Phi_e(X, m) \in \mathcal{O}_{<\alpha}^X\}) \geq r - 2^{-n}$. En utilisant l'exercice 28-4.7, la fonction f est Π_1^1 . Par l'hypothèse $\lambda(\mathcal{A}) \geq r$, la fonction f est totale, et donc de graphe Δ_1^1 . Donc, son image est Δ_1^1 et, par le lemme 29-5.3 de majoration de Spector, $\sup_n f(n) = \alpha < \omega_1^{ck}$. On a alors

$$\lambda\left(\bigcap_n \{X : \Phi_e(X, n) \in \mathcal{O}_{<\alpha}^X\}\right) \geq r.$$

Comme c'est le cas pour tout rationnel $r < \lambda(\mathcal{A})$, on en déduit que les éléments $X \in \mathcal{A}$ pour lesquels $(\Phi_e(X, n))_{n \in \mathbb{N}}$ est co-final dans ω_1^{ck} , est de mesure nulle. Comme c'est le cas pour tout e alors l'ensemble des éléments X capables de calculer une suite $(a_n)_{n \in \mathbb{N}}$ d'éléments de $\mathcal{O}_{<\omega_1^{ck}}^X$ qui soit co-final dans ω_1^{ck} est de mesure nulle. Si $\omega_1^X > \omega_1^{ck}$ alors X doit être capable de calculer une telle suite, donc $\lambda(\{X : \omega_1^X > \omega_1^{ck}\}) = 0$.

Solution 4.5. Soit $\mathcal{S} = \{X : \omega_1^X > \omega_1^{ck}\}$. D'après le corollaire 29-3.7, une classe $\Pi_1^1 \mathcal{A}$ est de la forme $\bigcup_{\alpha < \omega_1} \mathcal{A}_\alpha$. On peut de plus supposer que les classes $\mathcal{A}_\alpha$ sont deux à deux disjointes. On s'assure ainsi que pour tout $\alpha \geq \omega_1^{ck}$ et $X \in \mathcal{A}_\alpha$, on a $X \notin \bigcup_{\beta < \omega_1^{ck}} \mathcal{A}_\beta$, donc $\omega_1^X > \omega_1^{ck}$. Autrement dit, $\bigcup_{\omega_1^{ck} \leq \alpha < \omega_1} \mathcal{A}_\alpha \subseteq \mathcal{S}$. La classe $\mathcal{A}$ est de mesure nulle ssi chaque $\mathcal{A}_\alpha$ pour $\alpha < \omega_1^{ck}$ est de mesure nulle. Notons que chaque classe $\mathcal{A}_\alpha$ pour $\alpha < \omega_1^{ck}$ est Δ_1^1 .

Soit $A \subseteq \mathbb{N}$ l'ensemble Π_1^1 des paires $\langle a, n \rangle \in \mathbb{N}$ telles que $a \in \mathcal{O}$ et telles que $\lambda(\{X : n \in H_{2^a}^X\}) = 0$. Notons que d'après le théorème 28-4.4 toute classe Δ_1^1 de mesure nulle a un code dans A .

La plus grande classe Π_1^1 de mesure nulle est alors donnée par

$$\mathcal{S} \cup \bigcup_{\langle a, n \rangle \in A} \{X : n \in H_{2^a}^X\}.$$

Bibliographie

- [1] *Cabal Seminar 76-77*, volume 689, 1976.
- [2] Uri ABRAHAM et Richard A. SHORE : Initial Segments of the Degrees of Size $\aleph_1$. *Israel Journal of Mathematics*, 53(1), 1986.
- [3] Wilhelm ACKERMANN : Zum Hilbertschen Aufbau Der Reellen Zahlen. *Math. Ann.*, 99(1):118–133, 1928.
- [4] J. W. ADDISON : Separation Principles in the Hierarchies of Classical and Effective Descriptive Set Theory. *Fundamenta Mathematicae*, 46: 123–135, 1958. Publisher : Instytut Matematyczny Polskiej Akademii Nauk.
- [5] Klaus AMBOS-SPIES, Carl G. JOCKUSCH, Richard A. SHORE et Robert I. SOARE : An Algebraic Decomposition of the Recursively Enumerable Degrees and the Coincidence of Several Degree Classes with the Promptly Simple Degrees. *Transactions of the American Mathematical Society*, 281(1):109–109, janvier 1984.
- [6] Klaus AMBOS-SPIES, Bjørn KJOS-HANSSEN, Steffen LEMPP et Theodore A. SLAMAN : Comparing DNR and WWKL. *Journal of Symbolic Logic*, pages 1089–1104, 2004.
- [7] Uri ANDREWS, Peter GERDES et Joseph S. MILLER : The Degrees of Bi-Hyperhyperimmune Sets. *Ann. Pure Appl. Logic*, 165(3):803–811, 2014.
- [8] Marat Mirzaevich ARSLANOV : Some Generalizations of a Fixed-Point Theorem. *Izvestiya Vysshikh Uchebnykh Zavedenii. Matematika*, (5):9–16, 1981. Publisher : Kazan (Volga region) Federal University.
- [9] Theodore BAKER, John GILL et Robert SOLOVAY : Relativizations of the $P=NP$ Question. *SIAM Journal on computing*, 4(4):431–442, 1975.
- [10] Jack BARONE et Albert NOVIKOFF : A History of the Axiomatic Formulation of Probability from Borel to Kolmogorov : Part I. *Archive for History of Exact Sciences*, 18(2):123–190, 1978. Publisher : Springer.
- [11] James E. BAUMGARTNER : A Short Proof of Hindman’s Theorem. *Journal of Combinatorial Theory, Series A*, 17(3):384–386, 1974.
- [12] Jean-Pierre BELNA : *Cantor*, volume 20. Belles Lettres, 2000.

- [13] Laurent BIENVENU, Adam R DAY, Noam GREENBERG, Antonín KUČERA, Joseph S. MILLER, André NIES et Dan TURETSKY : Computing K-Trivial Sets by Incomplete Random Sets. *The Bulletin of Symbolic Logic*, pages 80–90, 2014. Publisher : JSTOR.
- [14] Laurent BIENVENU et Rod DOWNEY : Kolmogorov Complexity and Solovay Functions. In *26th International Symposium on Theoretical Aspects of Computer Science*, 2009.
- [15] Laurent BIENVENU, Noam GREENBERG, Antonin KUČERA, André NIES et Dan TURETSKY : K-Triviality, Oberwolfach Randomness, and Differentiability. 2012.
- [16] Laurent BIENVENU, Noam GREENBERG, Antonín KUČERA, Joseph S. MILLER, André NIES et Dan TURETSKY : Joining Non-Low c.e. Sets with Diagonally Non-Computable Functions. *J. Logic Comput.*, 23(6):1183–1194, 2013.
- [17] Laurent BIENVENU, Noam GREENBERG, Antonín KUČERA, André NIES et Dan TURETSKY : Coherent Randomness Tests and Computing the K-Trivial Sets. *J. Eur. Math. Soc. (JEMS)*, 18(4):773–812, 2016.
- [18] Laurent BIENVENU, Rupert HÖLZL, Joseph S MILLER et André NIES : The Denjoy Alternative for Computable Functions. In *29th International Symposium on Theoretical Aspects of Computer Science*, page 543, 2012.
- [19] Laurent BIENVENU, Rupert HÖLZL, Joseph S. MILLER et André NIES : Denjoy, Demuth and Density. *Journal of Mathematical Logic*, 14(01):1450004, 2014. Publisher : World Scientific.
- [20] Laurent BIENVENU, Wolfgang MERKLE et Andre NIES : Solovay Functions and K-Triviality. In *28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011)*, 2011.
- [21] Andreas R. BLASS, Jeffry L. HIRST et Stephen G. SIMPSON : Logical Analysis of some Theorems of Combinatorics and Topological Dynamics. *Logic and Combinatorics*, S. Simpson, ed., *Contemporary Math*, 69:125–156, 1987.
- [22] Émile BOREL : Les «paradoxes» de la théorie des ensembles. In *Annales Scientifiques De L'École Normale Supérieure*, volume 25, pages 443–448, 1908.
- [23] M Émile BOREL : Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 27(1):247–271, 1909. Publisher : Springer.
- [24] V. BRATTKA, G. GHERARDI et A. PAULY : Weihrauch Complexity in Computable Analysis. *ArXiv*, abs/1707.03202, 2017.

- [25] Vasco BRATTKA et Tahina RAKOTONIAINA : On the Uniform Computational Content of Ramsey's Theorem. *J. Symb. Log.*, 82(4):1278–1316, 2017.
- [26] Cristian S. CALUDE, Peter H. HERTLING, Bakhadyr KHOUSSAINOV et Yongge WANG : Recursively Enumerable Reals and Chaitin's Numbers. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 596–606. Springer, 1998.
- [27] Georg CANTOR : Fondements d'une théorie générale des ensembles. *Acta Mathematica*, 2(1):381–408, 1883. Publisher : Springer.
- [28] Olivier CARTON : *Langages formels, calculabilité et complexité*, volume 28. Vuibert, 2008.
- [29] Gregory J. CHAITIN : A Theory of Program Size Formally Identical to Information Theory. *Journal of the ACM (JACM)*, 22(3):329–340, 1975. Publisher : ACM New York, NY, USA.
- [30] Gregory J. CHAITIN : Information-Theoretic Characterizations of Recursive Infinite Strings. *Theoretical Computer Science*, 2(1):45–48, 1976. Publisher : Elsevier.
- [31] Gregory J. CHAITIN : Incompleteness Theorems for Random Reals. *Advances in Applied Mathematics*, 8(2):119–146, 1987. Publisher : Elsevier.
- [32] David G. CHAMPERNOWNE : The Construction of Decimals Normal in the Scale of Ten. *Journal of the London Mathematical Society*, 1(4):254–260, 1933.
- [33] Peter A. CHOLAK, Carl G. JOCKUSCH et Theodore A. SLAMAN : On the Strength of Ramsey's Theorem for Pairs. *The Journal of Symbolic Logic*, 66(1):1–55, 2001. Publisher : Cambridge University Press.
- [34] Peter A. CHOLAK et Ludovic PATEY : Thin Set Theorems and Cone Avoidance. *Transactions of the AMS*. To appear., 2019.
- [35] Chi Tat CHONG et Liang YU : *Recursion Theory : Computational Aspects of Definability*, volume 8. Walter de Gruyter GmbH & Co KG, 2015.
- [36] Alonzo CHURCH et Stephen C. KLEENE : Formal Definitions in the Theory of Ordinal Numbers. *Fundamenta Mathematicae*, 28(1):11–21, 1937.
- [37] Paul J. COHEN : The Independence of the Continuum Hypothesis, II. *Proceedings of the National Academy of Sciences of the United States of America*, 51(1):105, 1964. Publisher : National Academy of Sciences.
- [38] Joshua A COLE et Stephen G SIMPSON : Mass Problems and Hyperarithmeticality. *Journal of Mathematical Logic*, 7(02):125–143, 2007.

- [39] W Wistar COMFORT : Ultrafilters : Some Old and some New Results. *Bulletin of the American Mathematical Society*, 83(4):417–455, 1977.
- [40] Barry S. COOPER : Degrees of Unsolvability Complementary between Recursively Enumerable Degrees. *Annals of Mathematical Logic*, 4(1): 31–73, 1972.
- [41] Barry S. COOPER : *Computability Theory*. CRC Press, 2003.
- [42] René CORI et Daniel LASCAR : *Logique mathématique, Volume II*. Masson, 1993.
- [43] Adam R. DAY et Joseph S. MILLER : Density, Forcing, and the Covering Problem. *Mathematical Research Letters*, 22(3):719–727, 2015. Publisher : International Press of Boston.
- [44] Patrick DEHORNOY : La théorie des ensembles. *Calvage et Mounet, Paris*, 2017.
- [45] Osvald DEMUTH et Antonín KUČERA : Remarks on 1-Genericity, Semigenericity and Related Concepts. *Commentationes Mathematicae Universitatis Carolinae*, 28(1):85–94, 1987. Publisher : Charles University in Prague, Faculty of Mathematics and Physics.
- [46] Rod DOWNEY, Noam GREENBERG, Matthew HARRISON-TRAINOR, Ludovic PATEY et Dan TURETSKY : Relationships between Computability-Theoretic Properties of Problems, 2019.
- [47] Rod DOWNEY, Denis HIRSCHFELDT, Steffen LEMPP et Reed SOLOMON : A Δ_2^0 Set with no Infinite Low Subset in either it or its Complement. *Journal of Symbolic Logic*, 66(3):1371–1381, 2001.
- [48] Rod DOWNEY, André NIES, Rebecca WEBER et Liang YU : Lowness and Π_2^0 Nullsets. *Journal of Symbolic Logic*, 71(3):1044–1052, 2006.
- [49] Rodney G. DOWNEY : Abstract Dependence, Recursion Theory, and the Lattice of Recursively Enumerable Filters. *Bulletin of the Australian Mathematical Society*, 27(3):461–464, 1983.
- [50] Rodney G. DOWNEY et Denis R. HIRSCHFELDT : *Algorithmic Randomness and Complexity*. Springer Science & Business Media, 2010.
- [51] Arnaud DURAND et Paul ROZIÈRE : Calculabilité et incomplétude - Notes de cours.
- [52] Damir D. DZHAFAROV : Cohesive Avoidance and Strong Reductions. *Proceedings of the American Mathematical Society*, 143(2):869–876, 2014.
- [53] Damir D. DZHAFAROV, Denis R. HIRSCHFELDT et Sarah C. REITZES : Reduction Games, Provability, and Compactness, 2020.
- [54] Damir D DZHAFAROV et Carl G. JOCKUSCH : Ramsey’s Theorem and Cone Avoidance. *The Journal of Symbolic Logic*, 74(2):557–578, 2009. Publisher : Cambridge University Press.

- [55] Calvin C. ELGOT et Abraham ROBINSON : Random-Access Stored-Program Machines, an Approach to Programming Languages. *In Selected Papers*, pages 17–51. Springer, 1982.
- [56] Herbert ENDERTON et Hilary PUTNAM : A Note on the Hyperarithmetical Hierarchy. *The Journal of Symbolic Logic*, 35(3):429–430, 1970. Publisher : Cambridge University Press.
- [57] Andrei Petrovich ERSHOV : On Operator Algorithms. *In Doklady Akademii Nauk*, volume 122, pages 967–970. Russian Academy of Sciences, 1958.
- [58] Solomon FEFERMAN et Clifford SPECTOR : Incompleteness along Paths in Progressions of Theories 1. *The Journal of Symbolic Logic*, 27(4):383–390, 1962. Publisher : Cambridge University Press.
- [59] Michael R. FELLOWS : *Computer Science and Mathematics in the Elementary Schools*. Citeseer, 1991.
- [60] Lance FORTNOW et Michael SIPSER : Are there Interactive Protocols for Co-Np Languages? *Information Processing Letters*, 28(5):249–251, 1988.
- [61] Johanna FRANKLIN et Keng Meng NG : Difference Randomness. *Proceedings of the American Mathematical Society*, 139(1):345–360, 2011.
- [62] Richard FRIEDBERG : A Criterion for Completeness of Degrees of Unsolvability. *The Journal of Symbolic Logic*, 22(2):159–160, 1957.
- [63] Richard M. FRIEDBERG : Two Recursively Enumerable Sets of Incomparable Degrees of Unsolvability (Solution of Post’s Problem, 1944). *Proceedings of the National Academy of Sciences of the United States of America*, 43(2):236–238, 1957. Publisher : National Academy of Sciences.
- [64] Richard M. FRIEDBERG : Three Theorems on Recursive Enumeration. I. Decomposition. II. Maximal Set. III. Enumeration without Duplication. *J. Symbolic Logic*, 23:309–316, 1958.
- [65] Harvey FRIEDMAN : Some Systems of Second Order Arithmetic and their Use. *In Proceedings of the International Congress of Mathematicians (Vancouver, BC, 1974)*, volume 1, pages 235–242. Citeseer, 1975.
- [66] Harvey FRIEDMAN : Uniformly Defined Descending Sequences of Degrees. *The Journal of Symbolic Logic*, 41(2):363–367, 1976. Publisher : JSTOR.
- [67] Harvey M. FRIEDMAN : Higher Set Theory and Mathematical Practice. *Mathematical Logic in the 20th Century*, 2(3):49, 2003.
- [68] Harvey Martin FRIEDMAN : *Subsystems of Set Theory and Analysis*. PhD Thesis, Massachusetts Institute of Technology, 1967.

- [69] Peter GÁCS : On the Symmetry of Algorithmic Information. *In Doklady Akademii Nauk*, volume 218, pages 1265–1267. Russian Academy of Sciences, 1974. Issue : 6.
- [70] Péter GÁCS : Every Sequence is Reducible to a Random One. *Information and Control*, 70(2/3):186–192, 1986.
- [71] GALILÉE : *Discours et démonstrations mathématiques concernant deux sciences nouvelles*. Masson Éditeur, 1972. Introduction, traduction, notes et index de Maurice Clavelin.
- [72] Robin O. GANDY : On a Problem of Kleene's. *Bulletin of the American Mathematical Society*, 66(6):501–502, 1960.
- [73] Guido GHERARDI et Alberto MARCONE : How Incomputable is the separable Hahn-Banach Theorem? *Notre Dame J. Form. Log.*, 50(4): 393–425 (2010), 2009.
- [74] Kurt GÖDEL : The Consistency of the Axiom of Choice and of the Generalized Continuum-Hypothesis. *Proceedings of the National Academy of Sciences of the United States of America*, 24(12):556, 1938. Publisher : National Academy of Sciences.
- [75] Herman Heine GOLDSTINE et John VON NEUMANN : Planning and Coding of Problems for an Electronic Computing Instrument. 1947.
- [76] Noam GREENBERG et Joseph S. MILLER : Lowness for Kurtz Randomness. *The Journal of Symbolic Logic*, 74(2):665–678, 2009. Publisher : Cambridge University Press.
- [77] Marcia J. GROSZEK et Theodore A. SLAMAN : Independence Results on the Global Structure of the Turing Degrees. *Transactions of the American Mathematical Society*, 277(2):579–588, 1983.
- [78] Marcia J. GROSZEK et Theodore A. SLAMAN : Π_1^0 Classes and Minimal Degrees. *Annals of Pure and Applied Logic*, 87(2):117–144, 1997.
- [79] Marcia J GROSZEK et Theodore A SLAMAN : Moduli of Computation (Talk). *Buenos Aires, Argentina*, 2007.
- [80] David GUASPARI : A Note on the Kondo-Addison Theorem. *The Journal of Symbolic Logic*, 39(3):567–570, 1974. Publisher : JSTOR.
- [81] Petr HÁJEK et Pavel PUDLÁK : *Metamathematics of First-Order Arithmetic*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1998.
- [82] Joel David HAMKINS et Andy LEWIS-PYE : Infinite Time Turing Machines. *Journal of Symbolic Logic*, pages 567–604, 2000. Publisher : JSTOR.
- [83] Leo HARRINGTON et Saharon SHELAH : The Undecidability of the Recursively Enumerable Degrees. *Bulletin of the American Mathematical Society*, 6, 1982.

- [84] Joseph HARRISON : Recursive Pseudo-Well-Orderings. *Transactions of the American Mathematical Society*, 131(2):526–543, 1968. Publisher : JSTOR.
- [85] Leon HENKIN : The Completeness of the First-Order Functional Calculus. *The Journal of Symbolic Logic*, 14(3):159–166, 1949.
- [86] Neil HINDMAN : Finite Sums From Sequences within Cells of a Partition of N . *J. Combinatorial Theory Ser. A*, 17:1–11, 1974.
- [87] Denis R HIRSCHFELDT : Slicing the Truth. *Lecture Notes Series, Institute for Mathematical Sciences, National University of Singapore*, 28, 2015. Publisher : World Scientific Publishing.
- [88] Denis R. HIRSCHFELDT et Carl G. JOCKUSCH : On Notions of Computability-Theoretic Reduction between Π_2^1 Principles. *J. Math. Log.*, 16(1):1650002, 59, 2016.
- [89] Denis R. HIRSCHFELDT, Carl G. JOCKUSCH, Bjørn KJOS-HANSEN, Steffen LEMPP et Theodore A. SLAMAN : The Strength of some Combinatorial Principles Related to Ramsey’s Theorem for Pairs. *Computational Prospects of Infinity, Part II : Presented Talks, World Scientific Press, Singapore*, pages 143–161, 2008.
- [90] Denis R HIRSCHFELDT, André NIES et Frank STEPHAN : Using Random Sets as Oracles. *Journal of the London Mathematical Society*, 75(3):610–622, 2007. Publisher : Oxford University Press.
- [91] Jeffry L. HIRST : *Combinatorics in Subsystems of Second Order Arithmetic*. Thèse de doctorat, Pennsylvania State University, août 1987.
- [92] Greg HJORTH et André NIES : Randomness via Effective Descriptive Set Theory. *Journal of the London Mathematical Society*, 75(2):495–508, 2007.
- [93] Wassily Hoeffding : Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. Publisher : Taylor & Francis Group.
- [94] Mathieu HOYRUP et Cristóbal ROJAS : Applications of Effective Probability Theory to Martin-Löf Randomness. In *International Colloquium on Automata, Languages, and Programming*, pages 549–561. Springer, 2009.
- [95] Shamil ISHMUKHMETOV : Weak Recursive Degrees and a Problem of Spector. In *Recursion Theory and Complexity (Kazan, 1997)*, volume 2 de *De Gruyter Ser. Log. Appl.*, pages 81–87. de Gruyter, Berlin, 1999.
- [96] Carl JOCKUSCH et Robert SOARE : Degrees of Members of Π_1^0 Classes. *Pacific Journal of Mathematics*, 40:605–616, 1972.

- [97] Carl G. JOCKUSCH : Relationships between Reducibilities. *Transactions of the American Mathematical Society*, 142:229–237, 1969.
- [98] Carl G JOCKUSCH : Degrees in which the Recursive Sets are Uniformly Recursive. *Canadian Journal of Mathematics*, 24(6):1092–1099, 1972. Publisher : Cambridge University Press.
- [99] Carl G. JOCKUSCH : Ramsey’s Theorem and Recursion Theory. *The Journal of Symbolic Logic*, 37(2):268–280, 1972. Publisher : Cambridge University Press.
- [100] Carl G. JOCKUSCH : Degrees of Generic Sets. *Recursion Theory : Its Generalizations and Applications*, pages 110–139, 1980. Publisher : Cambridge Univ. Press Cambridge.
- [101] Carl G. JOCKUSCH, Manuel LERMAN, Robert I. SOARE et Robert M. SOLOVAY : Recursively Enumerable Sets modulo Iterated Jumps and Extensions of Arslanov’s Completeness Criterion. *The Journal of Symbolic Logic*, 54(4):1288–1323, 1989. Publisher : Cambridge University Press.
- [102] Carl G. JOCKUSCH et Richard A. SHORE : Pseudo-Jump Operators. II : Transfinite Iterations, Hierarchies and Minimal Covers. *The Journal of Symbolic Logic*, 49(4):1205–1236, 1984. Publisher : JSTOR.
- [103] Carl G. JOCKUSCH et Frank STEPHAN : A Cohesive Set which is not High. *Mathematical Logic Quarterly*, 39(1):515–530, 1993. Publisher : Wiley Online Library.
- [104] Carl G. JOCKUSCH, Jr. et Robert I. SOARE : Π_1^0 Classes and Degrees of Theories. *Trans. Amer. Math. Soc.*, 173:33–56, 1972.
- [105] Carl G JOCKUSCH JR : Degrees of Functions with no Fixed Points. *In Studies in Logic and the Foundations of Mathematics*, volume 126, pages 191–201. Elsevier, 1989.
- [106] Akihiro KANAMORI : Mathias and Set Theory. *Mathematical Logic Quarterly*, 62(3):278–294, 2016.
- [107] Loren KANTOR, Jean-Michel et Graham : *Au nom de l’infini. Une histoire vraie de mysticisme religieux et de création mathématique*. Belin, 2010.
- [108] Steven M. KAUTZ : *Degrees of Random Sets*. PhD Thesis, Citeseer, 1991.
- [109] Richard KAYE : *Models of Peano Arithmetic*. 1991.
- [110] Alexander S. KECHRIS : The Theory of Countable Analytical Sets. *Transactions of the American Mathematical Society*, 202:259–297, 1975.
- [111] Mushfeq KHAN et Joseph S. MILLER : Forcing with Bushy Trees. *Bulletin of Symbolic Logic*, 23(2):160–180, 2017. Publisher : Cambridge University Press.

- [112] Bjørn KJOS-HANSSEN, Wolfgang MERKLE et Frank STEPHAN : Kolmogorov Complexity and the Recursion Theorem. *Transactions of the American Mathematical Society*, 363(10):5465–5480, 2011.
- [113] Bjørn KJOS-HANSSEN, Wolfgang MERKLE et Frank STEPHAN : Kolmogorov Complexity and the Recursion Theorem. *Transactions of the American Mathematical Society*, 363(10):5465–5480, 2011.
- [114] Stephen C. KLEENE : On Notation for Ordinal Numbers. *The Journal of Symbolic Logic*, 3(4):150–155, 1938. Publisher : JSTOR.
- [115] Stephen C. KLEENE : Arithmetical Predicates and Function Quantifiers. *Transactions of the American Mathematical Society*, 79(2):312–340, 1955. Publisher : JSTOR.
- [116] Stephen C. KLEENE : Hierarchies of Number-Theoretic Predicates. *Bulletin of the American Mathematical Society*, 61(3):193–213, 1955.
- [117] Stephen C. KLEENE : On the Forms of the Predicates in the Theory of Constructive Ordinals (Second Paper). *American Journal of Mathematics*, 77(3):405–428, 1955. Publisher : JSTOR.
- [118] Stephen C. KLEENE et Emil L. POST : The Upper Semi-Lattice of Degrees of Recursive Unsolvability. *Annals of Mathematics*, pages 379–407, 1954. Publisher : JSTOR.
- [119] Andrej KOLMOGOROFF : Zur Deutung der Intuitionistischen Logik. *Mathematische Zeitschrift*, 35(1):58–65, 1932.
- [120] Andrei KOLMOGOROV : Logical Basis for Information Theory and Probability Theory. *IEEE Transactions on Information Theory*, 14(5):662–664, 1968.
- [121] Andrei N. KOLMOGOROV : On Tables of Random Numbers. *Sankhya The Indian Journal of Statistics, Series A*, pages 369–376, 1963. Publisher : JSTOR.
- [122] Andrei N. KOLMOGOROV : Three Approaches to the Quantitative Definition Ofinformation’. *Problems of Information Transmission*, 1(1):1–7, 1965.
- [123] Motokiti KONDÔ : Sur l’uniformisation des complémentaires analytiques et les ensembles projectifs de la seconde classe. In *Japanese Journal of Mathematics : Transactions and Abstracts*, volume 15, pages 197–230. The Mathematical Society of Japan, 1939. Issue : 0.
- [124] G. KREISEL : Analysis of the Cantor-Bendixson Theorem by Means of the Analytic Hierarchy. *Bull. Acad. Polon. Sci. Sér. Sci. Math. Astr. Phys.*, 7:621–626. (unbound insert), 1959.
- [125] Masahiro KUMABE et Andrew LEWIS-PYE : A Fixed-Point-Free Minimal Degree. *Journal of the London Mathematical Society*, 80(3), 2009.

- [126] Stuart A. KURTZ : *Randomness and Genericity in the Degrees of Unsolvability*. PhD Thesis, 1982.
- [127] Antonín KUČERA et Theodore SLAMAN : Randomness and Recursive Enumerability. *SIAM Journal on Computing*, 31(1):199–211, 2001. Publisher : SIAM.
- [128] Antonín KUČERA : Measure, Π_1^0 -Classes and Complete Extensions of PA. In *Recursion Theory Week*, pages 245–259. Springer, 1985.
- [129] A. H. LACHLAN : Lower Bounds for Pairs of Recursively Enumerable Degrees. *Proceedings of the London Mathematical Society*, s3-16(1): 537–569, 1966.
- [130] Alistair H. LACHLAN : Complete Recursively Enumerable Sets. *Proceedings of the American Mathematical Society*, 19(1):99–102, 1968.
- [131] Alistair H. LACHLAN : Embedding Nondistributive Lattices in the Recursively Enumerable Degrees. In *Conference in Mathematical Logic—London 70*, pages 149–177. Springer, 1972.
- [132] Alistair H. LACHLAN : Uniform Enumeration Operations. *Journal of Symbolic Logic*, pages 401–409, 1975.
- [133] Alistair H. LACHLAN et Robert LEBEUF : Countable Initial Segments of the Degrees of Unsolvability. *The Journal of Symbolic Logic*, 41(2): 289–300, 1976.
- [134] Alistair H. LACHLAN et Robert I. SOARE : Not Every Finite Lattice is Embeddable in the Recursively Enumerable Degrees. *Advances in Mathematics*, 37(1):74–82, 1980.
- [135] Joachim LAMBEK : How to Program an Infinite Abacus. *Canadian Mathematical Bulletin*, 4(3):295–302, 1961.
- [136] Henri LEBESGUE : Sur une généralisation de l'intégrale définie. *CR Acad. Sci. Paris*, 132:1025–1028, 1901.
- [137] Steffen LEMPP, André NIES et Theodore A. SLAMAN : The Π_3^0 -Theory of the Computably Enumerable Turing Degrees is Undecidable. *Transactions of the American Mathematical Society*, pages 2719–2736, 1998.
- [138] Manuel LERMAN : *Degrees of Unsolvability*, volume 11. Cambridge University Press, 2017.
- [139] Leonid A. LEVIN : *Some Theorems on the Algorithmic Approach to Probability Theory and Information Theory*. Dissertation in Mathematics, Université de Moscou, 1971.
- [140] Leonid A. LEVIN : On the Notion of a Random Sequence. In *Soviet. Math. Dokl.*, volume 14, pages 1413–1416, 1973. Issue : 5.
- [141] Leonid A. LEVIN : The Concept of a Random Sequence. In *Dokl. Akad. Nauk SSSR*, volume 212, pages 2–2, 1973. Issue : 548-550.

- [142] Leonid A. LEVIN : Laws of Information Conservation (Nongrowth) and Aspects of the Foundation of Probability Theory. *Problemy Peredachi Informatsii*, 10(3):30–35, 1974. Publisher : Russian Academy of Sciences.
- [143] Andrew LEWIS-PYE : Strong Minimal Covers and a Question of Yates : The Story so Far. In *Logic Colloquium 2006*, volume 32 de *Lect. Notes Log.*, pages 213–228. Assoc. Symbol. Logic, Chicago, IL, 2009.
- [144] Ming LI et Paul VITÁNYI : *An Introduction to Kolmogorov Complexity and its Applications*, volume 3. Springer, 2008.
- [145] J. E. LITTLEWOOD : *Lectures on the Theory of Functions*. Oxford University Press, 1944.
- [146] Jiayi LIU : RT_2^2 does not Imply WKL_0 . *The Journal of Symbolic Logic*, pages 609–620, 2012. Publisher : JSTOR.
- [147] Richard MANSFIELD : Perfect Subsets of Definable Sets of Real Numbers. *Pacific Journal of Mathematics*, 35(2):451–457, 1970. Publisher : Mathematical Sciences Publishers.
- [148] Donald A. MARTIN : Classes of Recursively Enumerable Sets and Degrees of Unsolvability. *Mathematical Logic Quarterly*, 12(1):295–310, 1966. Publisher : Wiley Online Library.
- [149] Donald A. MARTIN : Borel Determinacy. *Annals of Mathematics*, pages 363–371, 1975.
- [150] Per MARTIN-LÖF : The Definition of Random Sequences. *Information and Control*, 9(6):602–619, 1966. Publisher : Elsevier.
- [151] A. R. David MATHIAS : *On a Generalization of Ramsey’s Theorem*. Thèse de doctorat, University of Cambridge, 1969.
- [152] A. R. David MATHIAS : Happy Families. *Ann. Math. Logic*, 12(1):59–111, 1977.
- [153] Kenneth MCALOON : Paris-Harrington Incompleteness and Progressions of Theories. In *Recursion Theory (Ithaca, N.Y., 1982)*, volume 42 de *Proc. Sympos. Pure Math.*, pages 447–460. Amer. Math. Soc., Providence, RI, 1985.
- [154] Yuri T. MEDVEDEV : Degrees of Difficulty of the Mass Problem. In *Doklady Akademii Nauk SSSR, Ns*, volume 104, pages 501–504, 1955.
- [155] Zdzisław Alexander MELZAK : An Informal Arithmetical Approach to Computability and Computation. *Canadian Mathematical Bulletin*, 4(3):279–293, 1961.
- [156] Joseph R. MILETI : Partition Theorems and Computability Theory. *The Bulletin of Symbolic Logic*, 11(3):411–427, 2005. Publisher : JSTOR.

- [157] Webb MILLER et Donald A. MARTIN : The Degrees of Hyperimmune Sets. *Mathematical Logic Quarterly*, 14(7-12):159–166, 1968. Publisher : Wiley Online Library.
- [158] Benoît MONIN : Higher Randomness and Forcing with Closed Sets. *Theory of Computing Systems*, 60(3):421–437, 2017.
- [159] Benoît MONIN et Ludovic PATEY : Pigeons do not jump high. *Advances in Mathematics*, 352:1066–1095, 2019.
- [160] Benoît MONIN et Ludovic PATEY : SRT22 does not imply COH in Omega-Models, 2019.
- [161] Benoît MONIN et Ludovic PATEY : The Weakness of the Pigeonhole Principle under Hyperarithmetical Reductions, 2019.
- [162] Yiannis John MOSCHOVAKIS : Many-One Degrees of the Predicates $\text{Ha}(X)$. *Pacific Journal of Mathematics*, 18(2):329–342, 1966. Publisher : Mathematical Sciences Publishers.
- [163] Albert A. MUCHNIK : On the Unsolvability of the Problem of Reducibility in the Theory of Algorithms. In *Dokl. Akad. Nauk SSSR*, volume 108, pages 194–197, 1956. Issue : 1.
- [164] Albert A. MUCHNIK : Solution of Post’s Reduction Problem and of Certain Other Problems in the Theory of Algorithms. *Trudy Moskovskogo Matematicheskogo Obshchestva*, 7:391–405, 1958.
- [165] Albert A. MUCHNIK : Strong and Weak Reducibility of Algorithmic Problems 1. *Computability : The Journal of the Association CiE*, 5(1):49–59, 2016. Traduction de l’article original de 1963.
- [166] André NIES : Lowness Properties and Randomness. *Advances in Mathematics*, 197(1):274–305, 2005.
- [167] André NIES : *Computability and Randomness*, volume 51. Oxford University Press, 2009.
- [168] André NIES, Richard A. SHORE et Theodore A. SLAMAN : Interpretability and Definability in the Recursively Enumerable Degrees. *Proceedings of the London Mathematical Society*, 77(2):241–291, 1998.
- [169] Piergiorgio ODIFREDDI : *Classical Recursion Theory : The Theory of Functions and Sets of Natural Numbers*. Elsevier, 1992.
- [170] J. B. PARIS et L. A. S. KIRBY : Σ_n -Collection Schemas in Arithmetic, 1978.
- [171] Charles PARSONS : On a Number Theoretic Choice Schema and its Relation to Induction. In *Intuitionism and Proof Theory (Proc. Conf., Buffalo, N.Y., 1968)*, pages 459–473. North-Holland, Amsterdam, 1970.
- [172] Ludovic PATEY : The Weakness of Being Cohesive, Thin or Free in Reverse Mathematics. *Israel J. Math.*, 216(2):905–955, 2016.

- [173] Ludovic PATEY et Keita YOKOYAMA : The Proof-Theoretic Strength of Ramsey's Theorem for Pairs and Two Colors. *Adv. Math.*, 330: 1034–1070, 2018.
- [174] Rózsa PÉTER : Programmierung und Partiell-Rekursive Funktionen. *Acta Mathematica Academiae Scientiarum Hungarica*, 14(3-4):373–401, 1963.
- [175] Von Rózsa PÉTER : Graphschemata und Rekursive Funktionen. *Dialectica*, 12(3-4):373–393, 1958.
- [176] H. POINCARÉ : *Dernières pensées*. Flammarion, 1917.
- [177] Henri POINCARÉ : L'intuition et la logique en mathématiques. *La valeur de la Science*.
- [178] Henri POINCARÉ : *Science et méthode*. Flammarion, 1908.
- [179] Christopher P. PORTER : *Mathematical and Philosophical Perspectives on Algorithmic Randomness*. University of Notre Dame, 2012.
- [180] David B. POSNER et Robert W. ROBINSON : Degrees Joining to $0'$. *Journal of Symbolic Logic*, pages 714–722, 1981. Publisher : JSTOR.
- [181] Emil L. POST : Recursively Enumerable Sets of Positive Integers and their Decision Problems. *Bulletin of the American Mathematical Society*, 50(5):284–316, 1944.
- [182] Tibor RADO : On Non-Computable Functions. *Bell System Technical Journal*, 41(3):877–884, 1962.
- [183] Hartley ROGERS, Jr. : Gödel Numberings of Partial Recursive Functions. *J. Symbolic Logic*, 23:331–341, 1958.
- [184] Barkley ROSSER : An Informal Exposition of Proofs of Godel's Theorems and Church's Theorem. *Journal of Symbolic Logic*, 4(2):53–60, 1939. Publisher : Association for Symbolic Logic.
- [185] Gerald E. SACKS : A Minimal Degree less than $0'$. *Bull. Amer. Math. Soc.*, 67:416–419, 1961.
- [186] Gerald E. SACKS : On Suborderings of Degrees of Recursive Unsolvability. *Mathematical Logic Quarterly*, 7(1-5):46–56, 1961.
- [187] Gerald E. SACKS : A Maximal Set which is not Complete. *Michigan Mathematical Journal*, 11(3):193–205, 1964.
- [188] Gerald E. SACKS : *Degrees of Unsolvability*. Numéro 55. Princeton University Press, 1966.
- [189] Gerald E. SACKS : Measure-Theoretic Uniformity in Recursion Theory and Set Theory. *Transactions of the American Mathematical Society*, 142:381–420, 1969.
- [190] Gerald E. SACKS : Forcing with Perfect Closed Sets. *In Axiomatic Set Theory*, volume 1, pages 331–355. Amer. Math. Soc Providence, RI, 1971.

- [191] Gerald E SACKS : Countable Admissible Ordinals and Hyperdegrees. *Advances in Mathematics*, 20(2):213–262, 1976. Publisher : Academic Press.
- [192] Gerald E. SACKS : *Higher Recursion Theory*, volume 2. Cambridge University Press, 2017.
- [193] Claus-Peter SCHNORR : The Process Complexity and Effective Random Tests. In *Proceedings of the Fourth Annual ACM Symposium on Theory of Computing*, pages 168–176, 1972.
- [194] Dana SCOTT : Algebras of Sets Binumerable in Complete Extensions of Arithmetic. In *Proc. Sympos. Pure Math*, volume 5, pages 117–121, 1962.
- [195] David SEETAPUN et Theodore SLAMAN : On the Strength of Ramsey’s Theorem. *Notre Dame Journal of Formal Logic*, 36(4):570–582, 1995. Publisher : University of Notre Dame.
- [196] Adi SHAMIR : $IP = PSPACE$. *Journal of the ACM (JACM)*, 39(4):869–877, 1992.
- [197] John C. SHEPHERDSON et Howard E. STURGIS : Computability of Recursive Functions. *Journal of the ACM (JACM)*, 10(2):217–255, 1963.
- [198] J. R. SHOENFIELD : A Theorem on Minimal Degrees. *J. Symbolic Logic*, 31:539–544, 1966.
- [199] Richard A. SHORE : On the $\forall\exists$ -Sentences of α -Recursion Theory. In *Studies in Logic and the Foundations of Mathematics*, volume 94, pages 331–353. Elsevier, 1978.
- [200] Richard A. SHORE et Theodore A. SLAMAN : Defining the Turing Jump. *Mathematical Research Letters*, 6(6):711–722, 1999.
- [201] Stephen G. SIMPSON : First-Order Theory of the Degrees of Recursive Unsolvability. *Annals of Mathematics*, pages 121–139, 1977.
- [202] Stephen G. SIMPSON : Partial Realizations of Hilbert’s Program. *J. Symbolic Logic*, 53(2):349–363, 1988.
- [203] Stephen G. SIMPSON : *Subsystems of Second Order Arithmetic*, volume 1. Cambridge University Press, 2009.
- [204] Theodore A. SLAMAN : Σ_n -Bounding and Δ_n -Induction. 132:2449–2449, 2004.
- [205] Theodore A SLAMAN et John R STEEL : Definable Functions on Degrees. In *Cabal Seminar 81–85*, pages 37–55. Springer, 1988.
- [206] Theodore A. SLAMAN et William H. WOODIN : Definability in the Turing Degrees. *Illinois Journal of Mathematics*, 30(2):320–334, 1986.
- [207] Theodore A. SLAMAN et William H. WOODIN : Definability in Degree Structures. *Preprint*, 2005.

- [208] Robert I. SOARE : Turing Computability. *Theory and Applications of Computability*. Springer, 2016.
- [209] Ray J. SOLOMONOFF : A Preliminary Report on a General Theory of Inductive Inference. United States Air Force, Office of Scientific Research, 1960.
- [210] Ray J. SOLOMONOFF : A Formal Theory of Inductive Inference. Part I. *Information and Control*, 7(1):1–22, 1964. Publisher : Elsevier.
- [211] Robert M SOLOVAY : On the Cardinality of Σ_2^1 Sets of Reals. *In Foundations of Mathematics*, pages 58–73. Springer, 1969.
- [212] Robert M. SOLOVAY : Draft of Paper (Or Series of Papers) on Chaitin's Work. mai 1975.
- [213] Ernst SPECKER : Ramsey's Theorem does not hold in Recursive Set Theory. *In Studies in Logic and the Foundations of Mathematics*, volume 61, pages 439–442. Elsevier, 1971.
- [214] Clifford SPECTOR : Recursive Well-Orderings. *The Journal of Symbolic Logic*, 20(2):151–163, 1955. Publisher : JSTOR.
- [215] Clifford SPECTOR : On Degrees of Recursive Unsolvability. *Ann. of Math. (2)*, 64:581–592, 1956.
- [216] John R. STEEL : Forcing with Tagged Trees. *Annals of Mathematical Logic*, 15(1):55–74, 1978. Publisher : Elsevier.
- [217] John R STEEL : A Classification of Jump Operator. *The Journal of Symbolic Logic*, 47(2):347–358, 1982.
- [218] Frank STEPHAN : Martin-Löf Random and PA-complete Sets. *In Logic Colloquium*, volume 2, pages 342–348. Association for Symbolic Logic and AK Peters, Ltd, 2006.
- [219] Mikhail SUSLIN : Sur une définition des ensembles mesurables B sans nombres transfinis. *CR Acad. Sci. Paris*, 164(2):88–91, 1917.
- [220] William W TAIT : Finitism. *The Journal of Philosophy*, pages 524–546, 1981.
- [221] Hisao TANAKA *et al.* : A Basis Result for Π_1^1 -Sets of Positive Measure. *Rikkyo Daigaku sugaku zasshi*, 16(2):115–127, 1967.
- [222] S. A. TERWIJN et D. ZAMBELLA : Algorithmic Randomness and Lowness. *ILLC Technical Report*, ML-1997-07, 1997.
- [223] Steven K. THOMASON : Sublattices of the Recursively Enumerable Degrees. *Mathematical Logic Quarterly*, 17(1):273–280, 1971.
- [224] Henry TOWNSNER : A Simple Proof and some Difficult Examples for Hindman's Theorem. *Notre Dame J. Form. Log.*, 53(1):53–65, 2012.
- [225] John TROMP et Gunnar FARNEBÄCK : Combinatorics of Go. *In International Conference on Computers and Games*, pages 84–99. Springer, 2006.

- [226] Alan Mathison TURING : Systems of Logic Based on Ordinals. *Proceedings of the London Mathematical Society*, 2(1):161–228, 1939. Publisher : Wiley Online Library.
- [227] Michiel van LAMBALGEN : *Random Sequences*. Ph.D. Dissertation, University of Amsterdam, 1987.
- [228] Jean VILLE : Étude critique de la notion de collectif. *Bull. Amer. Math. Soc*, 45(11):824, 1939.
- [229] Hao WANG : A Variant to Turing’s Theory of Computing Machines. *Journal of the ACM (JACM)*, 4(1):63–92, 1957.
- [230] Hao WANG : *Popular Lectures on Mathematical Logic*. Dover Publications, Inc., New York, seconde édition, 1993.
- [231] Wei WANG : Some Logically Weak Ramseyan Theorems. *Advances in Mathematics*, 261:1–25, 2014.
- [232] Wei WANG : The Definability Strength of Combinatorial Principles. *J. Symb. Log.*, 81(4):1531–1554, 2016.
- [233] K. WEIHRAUCH : *Computable Analysis : An Introduction*. Texts in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2000.
- [234] Mike C. E. YATES : A Minimal Pair of Recursively Enumerable Degrees. *Journal of Symbolic Logic*, 31(2):159–168, juin 1966.
- [235] Mike C. E. YATES : Initial Segments of the Degrees of Unsolvability Part II : Minimal Degrees. *The Journal of Symbolic Logic*, 35(2):243–266, 1970.
- [236] Liang YU : Descriptive Set Theoretical Complexity of Randomness Notions. *Fundamenta Mathematicae*, 215(3):219–231, 2011.
- [237] A. K. ZVONKIN et L. A. LEVIN : The complexity of finite objects and the basing of the concepts of information and randomness on the theory of algorithms. *Uspehi Mat. Nauk*, 25(6(156)):85–127, 1970.

Notations

$ E = F $	Les ensembles E et F sont équipotents	15
$ E \leq F $	L'ensemble E est subpotent à l'ensemble F	16
$\langle n, m \rangle$	Le résultat de la bijection de Cantor sur n, m	20
$\langle x_1, \dots, x_k \rangle$	Le résultat de la bijection de Cantor sur $x_1, \dots, x_k$	20
$\mathcal{P}(A)$	L'ensemble des sous-ensembles de A	23
$2^{\mathbb{N}}$	L'espace de Cantor	25
$X(n)$	Le n -ième élément de X (en commençant à 0)	26
$f(x) = g(x)$	Les fonctions calculables f et g s'arrêtent sur x et retournent la même valeur, ou bien ne s'arrêtent pas sur x .	34
Φ_e	La fonction calculable de code e	35
$\Phi_e(x) \downarrow$	L'exécution de Φ_e s'arrête sur l'entrée x	35
$\Phi_e(x) \uparrow$	L'exécution de Φ_e ne s'arrête pas sur l'entrée x	35
$\Phi_e(x) \downarrow = y$	L'exécution de Φ_e s'arrête sur l'entrée x et retourne la valeur y	35
$\Phi_e(x) \uparrow \neq y$	L'exécution de Φ_e ne s'arrête pas sur l'entrée x ou bien s'arrête et retourne une valeur différente de y	35
$\Phi_e(x)[t] \downarrow$	L'exécution de Φ_e s'arrête sur l'entrée x en moins de t étapes de calcul	36
$\Phi_e(x)[t] \uparrow$	L'exécution de Φ_e ne s'arrête pas sur l'entrée x en moins de t étapes de calcul	36
π_0, π_1	Les fonctions inverses de $(n, m) \mapsto \langle n, m \rangle$	37
$A[s]$	L'approximation de A à l'étape s	47
$\emptyset'$	Le problème de l'arrêt	47
$2^{<\mathbb{N}}$	L'ensemble des chaînes binaires	51
ϵ	La chaîne vide	51
$\sigma\tau$	La concaténation de σ et τ	51
$\sigma \preceq \tau$	La chaîne σ est un préfixe de la chaîne τ	51
$ \sigma $	La taille de σ	51
$\sigma(n)$	Le bit numéro n de σ (en commençant à 0)	51
σX	La concaténation de σ et X	52
$\sigma \prec X$	La chaîne σ est un préfixe de X	52
$X \upharpoonright_n$	Le préfixe de X de taille n	52
$\Phi_e(A, n)$	Le résultat du calcul de la fonctionnelle Φ_e avec l'oracle A et sur l'entrée n	53
$\Phi_e^A(n)$	Autre notation pour $\Phi_e(A, n)$	53
$\Phi_e(A, n)[t] \downarrow$	La fonctionnelle Φ_e avec l'oracle A , s'arrête en moins de t étapes sur l'entrée n	53
$\Phi_e(A, n)[t] \uparrow$	La fonctionnelle Φ_e avec l'oracle A , ne s'arrête pas en moins de t étapes sur l'entrée n	53

$\Phi_e(\sigma, n) \downarrow$	La fonctionnelle Φ_e avec le morceau d'oracle σ , s'arrête en moins de t étapes sur l'entrée n	56
$\Phi_e^\sigma(n) \downarrow$	La fonctionnelle Φ_e avec le morceau d'oracle σ , ne s'arrête pas en moins de t étapes sur l'entrée n	56
use_Φ^X	La fonction de l'usage de X sur la fonctionnelle Φ	56
$\leq_T$	Réduction Turing	57
$\text{deg}_T(X)$	Le degré Turing de X	57
$\equiv_T$	Équivalence Turing	57
$(\mathcal{D}, \leq)$	La structure des degrés Turing	57
$A \oplus B$	La jointure de A et B	59
$\Phi_e(X) = A$	$\forall n \Phi_e(X, n) \downarrow = A(n)$	59
X'	Le saut Turing de X	60
$[\sigma]$	L'ensemble $\{X : \sigma \prec X\}$	69
$\Sigma_n^0(\text{ens.})$	Ensemble Σ_n^0	82
$\Pi_n^0(\text{ens.})$	Ensemble Π_n^0	82
$\Delta_n^0(\text{ens.})$	Ensemble Δ_n^0	83
$\Sigma_1^0(\text{ens.})$	Ensemble c. e.	87
$\Delta_1^0(\text{ens.})$	Ensemble calculable	87
$\Sigma_n^0(X)(\text{ens.})$	Ensemble Σ_n^0 relativement à X	88
$\Pi_n^0(X)(\text{ens.})$	Ensemble Π_n^0 relativement à X	88
$\Delta_n^0(X)(\text{ens.})$	Ensemble Δ_n^0 relativement à X	89
$\leq_m$	Réduction many-one	89
$\equiv_m$	Équivalence many-one	89
$X^{(n)}$	Le n -ième saut Turing de X	91
W_e	L'ensemble c. e. de code e	95
W_e^X	L'ensemble X -c. e. de code e	95
$g(\bar{x})$	Un raccourci pour $g(x_1, \dots, x_n)$ (avec $g : \mathbb{N}^n \rightarrow \mathbb{N}$)	100
$\text{dom } f$	Le domaine de définition de la fonction f	111
$\text{Im } f$	L'image de f	111
p_i^n	Fonction primitive récursive de projection	117
c_i^n	Fonction primitive récursive constante	117
succ	Fonction primitive récursive successeur	117
$\leq_{tt}$	Réduction truth-table	144
$\equiv_{tt}$	Équivalence truth-table	144
$\forall^\infty x$	Pour tout x sauf un nombre fini	147
$\exists^\infty x$	Pour une infinité de x	147
$[T]$	L'ensemble des chemins de T	154
i^∞	La suite infinie qui répète le bit $i \in \{0, 1\}$	158
$[W]$	L'ensemble $\bigcup_{\sigma \in W} [\sigma]$	158
$\Sigma_1^0(\text{classe})$	Un ouvert effectif	165
$\Pi_1^0(\text{classe})$	Un fermé effectif	165
$f^{<\mathbb{N}}$	L'ensemble des chaînes $\sigma \in \mathbb{N}^{<\mathbb{N}}$ telles que pour tout $n < \sigma $, $\sigma(n) < f(n)$	182

$\mathbf{Q}$	L'arithmétique de Robinson	203
$T \vdash F$	F est démontrable dans T	204
$T \nVdash F$	F n'est pas démontrable dans T	204
$T \vdash \perp$	T est incohérente	204
$\mathcal{M} \models F$	$\mathcal{M}$ est modèle de F	206
Δ_0 (form.)	Formule Δ_0	213
Σ_n (form.)	Formule Σ_n^0	213
Π_n (form.)	Formule Π_n^0	213
$\text{Coh}(T)$	Formule qui exprime la cohérence de T	220
Σ_n^0 (form.)	Formule Σ_n^0	233
Π_n^0 (form.)	Formule Π_n^0	233
$\sigma \Vdash^* \mathcal{R}$	σ force le contrat $\mathcal{R}$ (cas Σ_1^0/Π_1^0)	235
$\sigma \perp W$	Aucune extension de σ n'est dans W	249
$W^\perp$	L'ensemble $\{\sigma \in 2^{<\mathbb{N}} : \sigma \perp W\}$	249
$\bigoplus_{n \in \mathbb{N}} A_n$	La jointure effective de $(A_n)_{n \in \mathbb{N}}$	253
$\sigma \Vdash \mathcal{R}$	σ force sémantiquement le contrat $\mathcal{R}$	260
$\sigma \Vdash^* \mathcal{R}$	σ force le contrat $\mathcal{R}$ (cas général)	260
$\mathcal{B}\Delta\mathcal{U}$	La classe $(\mathcal{B} \setminus \mathcal{U}) \cup (\mathcal{U} \setminus \mathcal{B})$	263
$[c]_\infty$	L'ensemble des filtres maximaux contenant c	274
$\dot{F}$	L'élément correspondant au filtre maximal F dans un forcing de Cantor	275
$[c]$	L'ensemble $\{\dot{F} : F \in [c]_\infty\}$	276
$? \vdash$	Question de forcing (J.-S. ou S., cas Σ_1^0/Π_1^0)	288
$? \vdash$	Question de forcing (J.-S. ou S., cas général)	290
$? \vdash$	Question de forcing	292
$\leq_w$	Réduction de Muchnik	306
$\leq_s$	Réduction de Medvedev	306
$\mathbf{0}_w$	Le degré Muchnik de $\{\emptyset\}$	306
$\mathbf{a} \cup \mathbf{b}$	La borne supérieure de $\mathbf{a}$ et $\mathbf{b}$	343
$C_M(\sigma)$	La complexité de Kolmogorov de σ pour la machine M	370
$C(\sigma)$	La complexité de Kolmogorov de σ	371
$C(n)$	La complexité de Kolmogorov de n	374
$\leq^+$	Inférieur ou égal à constante additive près	374
$=^+$	Égal à constante additive près	374
$K_M(\sigma)$	La complexité sans préfixe de σ pour la machine M	380
$K(\sigma)$	La complexité sans préfixe de σ	380
Ω	Le nombre Ω de Chaitin	382
$\text{poids}(A)$	Le poids d'un ensemble borné de requêtes A	385
Π_2^0 (classe)	Une intersection effective d'ouverts effectifs	400
Σ_n^0 (classe)	Classe borélienne Σ_n^0	402
Π_n^0 (classe)	Classe borélienne Π_n^0	402
Δ_n^0 (classe)	Classe borélienne Δ_n^0	402

Σ_n^0 (classe)	Classe borélienne effective Σ_n^0	403
Π_n^0 (classe)	Classe borélienne effective Π_n^0	403
Δ_n^0 (classe)	Classe borélienne effective Δ_n^0	403
λ	La mesure de Lebesgue	407
$\lambda(\mathcal{B} \mid [\sigma])$	La mesure de $\mathcal{B} \cap [\sigma]$ à l'intérieur de $[\sigma]$	419
$K_M^X(\sigma)$	La complexité sans préfixe de σ pour la machine M , relativisée à X	433
$\rho(\mathcal{B} \mid Z)$	La densité inférieure de Z dans $\mathcal{B}$	442
$\bar{\rho}(\mathcal{B} \mid Z)$	La densité supérieure de Z dans $\mathcal{B}$	442
$\mathcal{L}_{Z_2}$	Le langage de l'arithmétique du second ordre	482
Z_2	La théorie de l'arithmétique du second ordre	485
ω	Les entiers standard	488
RCA_0	La théorie RCA_0	497
ACA_0	La théorie ACA_0	501
WKL_0	La théorie WKL_0	505
WWKL_0	La théorie WWKL_0	509
ACA'_0	La théorie ACA'_0	513
ACA_0^+	La théorie ACA_0^+	514
ATR_0	La théorie ATR_0	516
$\Pi_1^1\text{-CA}_0$	La théorie $\Pi_1^1\text{-CA}_0$	516
I_{ouvert}	Le schéma d'axiomes d'induction pour les formules sans quantificateur	523
$\text{I}\Delta_0^0$	La théorie $\text{I}\Delta_0^0$	526
$x \mid y$	x divise y	526
$x \dot{-} y$	$x - y$ si $x > y$ et 0 sinon.	527
$\text{I}\Sigma_n^0$	Le schéma d'axiomes d'induction pour les formules Σ_n^0	528
$\text{I}\Pi_n^0$	Le schéma d'axiomes d'induction pour les formules Π_n^0	528
$\text{B}\Sigma_n^0$	Le schéma d'axiomes de collection bornée pour les formules Σ_n^0	528
$\text{B}\Pi_n^0$	Le schéma d'axiomes de collection bornée pour les formules Π_n^0	528
$\text{L}\Sigma_n^0$	Le schéma d'axiomes de minimum pour les formules Σ_n^0	531
$\text{L}\Pi_n^0$	Le schéma d'axiomes de minimum pour les formules Π_n^0	531
$\text{I}\Delta_n^0$	Le schéma d'axiomes d'induction pour les formules Δ_n^0	533
$x \in y$	x appartient à l'ensemble codé par y	539
$\text{BC}\Sigma_n^0$	Le schéma d'axiomes de compréhension borné pour les formules Σ_n^0	541
$\text{BC}\Pi_n^0$	Le schéma d'axiomes de compréhension borné pour les formules Π_n^0	541
$\text{BC}\Delta_n^0$	Le schéma d'axiomes de compréhension borné pour les formules Δ_n^0	541
$\Sigma_1\text{-PA}$	La théorie $\Sigma_1\text{-PA}$	546
$\mathcal{M}[G]$	L' ω -extension de $\mathcal{M}$ constitué des ensembles définissables par des formules Δ_1^0 à paramètres dans $\mathcal{M} \cup \{G\}$	548

$\mathcal{PR}$	La classe des fonctions primitives récursives	554
PRA	La théorie PRA	554
Π_2^1 (prob.)	Un problème de la forme $\forall X(F(X) \rightarrow \exists Y G(X, Y))$	559
ADS	L'énoncé « Ascending-Descending Sequence »	560
$\leq_\omega$	ω -réduction	560
KL	Le problème associé au lemme de König	561
J	Le problème du saut Turing	561
$\leq_c$	Réduction calculatoire	565
$\leq_W$	Réduction Weihrauch	567
LPO	Le principe limité d'omniscience	569
LLPO	Le principe très limité d'omniscience	569
$\hat{P}$	La parallélisation de problème P	570
$G(Q \rightarrow P)$	Jeu de réduction entre les problèmes P et Q	571
$P \leq_\omega^n Q$	Le joueur 2 possède une stratégie gagnante pour le jeu $G(Q \rightarrow P)$ en au plus $n + 1$ tours	573
$\leq_{sc}$	Réduction calculatoire forte	573
$\leq_{sW}$	Réduction forte de Weihrauch	574
$[X]^n$	L'ensemble des sous-ensembles de X de taille n	578
RT_k^n	Le théorème de Ramsey pour les n -uplets et k couleurs	579
cRT_k^n	Variante du théorème de Ramsey pour la réduction Weihrauch	581
$\sigma \cup \tau$	L'union des chaînes σ et τ vues comme des ensembles	597
$\tau - \sigma$	La différence des chaînes σ et τ vues comme des ensembles	597
$? \vdash$	Question de forcing pour le forcing de Dzhabarov-Jockusch	600
$? \not\vdash$	Négation de la question de forcing	600
COH	Le principe d'existence d'ensemble cohésif	612
$\mathcal{O}$ (Kleene)	L'ensemble $\mathcal{O}$ de Kleene	636
$<_o$	L'ordre partiel sur les éléments de $\mathcal{O}$	636
H_a	L'itération du saut sur le code d'ordinal a	637
$\mathfrak{n}$	L'ordinal fini $\mathfrak{n}$	639
$\alpha < \beta$	L'ordinal α est plus petit que l'ordinal β	640
$\text{succ}(\alpha)$	Le successeur de l'ordinal α	643
$\sup_{i \in I} \alpha_i$	Le supremum des ordinaux $(\alpha_i)_{i \in I}$	643
Ord	La classe des ordinaux	643
$ < $	L'ordinal isomorphe au bon ordre $<$	646
Σ_α^0 (classe)	Classe borélienne Σ_α^0	647
Π_α^0 (classe)	Classe borélienne Π_α^0	647
ω_1	Le plus petit ordinal indénombrable	654
ω_n	Le plus petit ordinal non subpotent à ω_{n-1}	657
$\mathcal{O}_{<\alpha}$	L'ensemble des codes d'ordinaux inférieurs à α	658
$\mathcal{O}_{=\alpha}$	L'ensemble des codes de l'ordinal α	658
ω_1^{ck}	Le plus petit ordinal non-calculable	658
$T \upharpoonright_\sigma$	L'arbre des nœuds de T comparables avec σ	662
$T \upharpoonright_\sigma$	L'arbre $T \upharpoonright_\sigma$ mais en « prenant σ comme racine »	662

$ T $	L'ordinal codé par l'arbre bien fondé T	662
$\mathcal{T}$	L'ensemble des codes c. e. d'arbres bien fondés	663
$\mathcal{T}_{<\alpha}$	L'ensemble des codes c. e. d'arbres bien fondés tels que $ T < \alpha$	663
$\mathcal{T}_{=\alpha}$	L'ensemble des codes c. e. d'arbres bien fondés tels que $ T = \alpha$	663
$<_{\text{KB}}$	L'ordre de Kleene-Brouwer	663
$\mathcal{O}^X$	L'ensemble $\mathcal{O}$ de Kleene relativisé à X	664
$\mathcal{T}^X$	L'ensemble $\mathcal{T}$ relativisé à X	665
ω_1^X	Le plus petit ordinal non X -calculable	665
Σ_α^0 (ens.)	Ensemble Σ_α^0	667
Π_α^0 (ens.)	Ensemble Π_α^0	667
Δ_α^0 (ens.)	Ensemble Δ_α^0	667
$\emptyset^{(\alpha)}$	L'itération α du saut Turing	675
H_a^X	L'itération du saut de X sur le code d'ordinal a	677
Σ_α^0 (classe)	Classe borélienne effective Σ_α^0	678
Π_α^0 (classe)	Classe borélienne effective Π_α^0	678
Δ_α^0 (classe)	Classe borélienne effective Δ_α^0	678
Σ_1^1 (ens.)	Ensemble Σ_1^1	688
Π_1^1 (ens.)	Ensemble Π_1^1	688
Δ_1^1 (ens.)	Ensemble Δ_1^1	688
Σ_1^1 (classe)	Classe analytique effective	688
Π_1^1 (classe)	Classe co-analytique effective	688
Δ_1^1 (classe)	Classe Δ_1^1 effective	688
Σ_1^1 (classe)	Classe analytique	692
Π_1^1 (classe)	Classe co-analytique	692
$\vee$	Fonction « ou » sur les arbres	704
$\wedge$	Fonction « et » sur les arbres	704
$\leq_h$	Réduction hyperarithmétique	709
HYP	La classe des ensembles hyperarithmétiques	720
$\mathcal{S}$	La classe des ensembles qui calculent ω_1^{ck}	721
$X \leq_T \emptyset^{(\alpha)}$	$\forall Y (\alpha < \omega_1^Y \rightarrow X \leq_T Y^{(\alpha)})$	722
$\mathcal{C}$	La plus grande classe Π_1^1 qui ne contient pas de fermé parfait	722
J_a	L'hypersaut itéré sur le code a	728
ω_α^{ck}	Le α -ième plus petit ordinal non X -calculable pour un certain X	729
$\text{WO}(<, A)$	L'ordre $<$ est bien fondé sur A	733
$L_{\mathcal{O}}(a, X)$	a est un code d'ordre linéaire avec oracle X	736
$W_{\mathcal{O}}(a, X)$	a est un code d'ordre bien fondé avec oracle X	736
PBO^X	L'ensemble des X -pseudos bons ordres	738
SUPP_X	L'ensemble des supports de saut	739

Index

- 1-aléatoire (ensemble), 436
- 1-générique (ensemble), 249
- 2-aléatoire (ensemble), 436
- Π_1^1 -CA₀, 516
- β -modèle, 745
- λ -calcul, 102
- λ -fonction, 102
- ω -extension, 544, 547
- ω -réduction, 558
- ω -sous-structure, 544, 547
- ω -structure, 490
- n -aléatoire (ensemble), 436
- n -générique (ensemble), 253

- ACA'₀, 513
- ACA⁺₀, 514
- ACA₀, 501
- Ackermann
 - fonction, 121
- addition
 - ordinaire, 647
- aléatoire
 - Chaitin/Levin, 382
 - différence, 439
 - Martin-Löf, 416
 - Martin-Löf relativisé, 432
 - Oberwolfach, 464
- analytique
 - classe, 686, 690
 - ensemble, 686
 - hiérarchie, 516
- anti-chaîne
 - degré Turing, 349
- approchable
 - par la gauche, 383
 - par le dessus, 372
- approximation
 - Δ_2^0 , 63
 - c. e., 46
 - par la gauche, 383
 - par le dessus, 372

- arbre, 154, 179
 - bien fondé, 660
 - binaire, 154
 - branchement fini, 180
 - c. e., 339
 - calculable, 156
 - calculatoirement borné, 182
 - chemin, 154, 179
 - f-arbre, 140
 - feuille, 154
 - hauteur, 660
 - mesure positive, 508
 - morphisme, 701
 - nœud, 154
 - parfait, 164
 - stratégies, 329
- arithmétique
 - formule, 197, 483
 - hiérarchie, 81
 - langage, 195
 - Peano, 203
 - primitive récursive, 554
 - Robinson, 203, 522
 - second ordre, 223, 232, 481
 - terme, 196
- arrêt, 47
- Arslanov
 - critère de complétude, 136
- astuce de Posner, 324
- atomique
 - formule, 197
- ATR₀, 516
- axiome
 - choix, 228, 229, 652, 710, 724
 - choix dénombrable, 653
 - collection, 528
 - compréhension, 225, 484, 496
 - compréhension bornée, 540
 - extensionnalité, 224
 - fondation, 227
 - induction, 485, 497, 523, 526, 528, 533

- induction ordonnée, 532
 - infini, 225
 - minimum, 531
 - remplacement, 226
- Baire
- catégorie, 238, 240
 - espace, 179
 - fermé, 180
 - ouvert, 180
 - propriété, 263
- base
- base calculatoirement dominée, 170
 - base d'évitement de cône, 171
 - base d'évitement de cône hyperarithmétique, 714
 - base hyperlow, 711
 - base low, 169
 - classes Π_1^0 parfaites, 172
 - classe Π_1^0 , 168
 - classe Σ_1^1 , 711
 - pour l'aléatoire, 446
- bien fondé
- arbre, 660
 - ordre, 642
- bijection, 15
- Cantor, 119
- blessure
- finie, 312
 - infinie, 323
- bon ordre, 515, 642
- pseudo, 736
- borne
- inférieure, 336
 - supérieure, 336
- Borélien
- classe, 645
 - hiérarchie, 403, 645
 - hiérarchie effective, 403, 676
- borélien
- classe, 402
 - hiérarchie, 402
- branchement fini (arbre), 180
- c. e.
- approximation, 46
 - arbre, 339, 661
 - degré, 136, 310
 - ensemble, 45
 - opérateur, 302
- calculable
- à la limite, 61
 - ensemble, 36
 - fonction, 32, 36
 - ordinal, 657
 - par programme goto, 112
 - par programme structuré, 112
 - relativement à X , 53
 - uniformément, 63
- calculatoirement
- borné (arbre), 182
 - dominé, 140
 - énumérable, 45, 136
 - indépendant, 347
 - réductible, 563
 - réductible (fortement), 571
 - vrai, 565
- Cantor
- bijection, 19, 119
 - compact, 160
 - cylindre, 158
 - ensemble triadique, 26
 - espace, 25, 158
 - fermé, 158
 - fonction continue, 162
 - forcing, 276
 - ouvert, 158
- Cantor-Bernstein, 17
- cappable
- degré, 332
- capture
- test de différence, 439
 - test de Martin-Löf, 416
 - test de Solovay, 417
- castor affairé, 373
- catégorie
- Baire, 238, 240
- Cauchy
- suite, 493

- Chaitin
 - Ω (nombre), 382
 - théorème de Chaitin, 377
 - théorème KC, 386
- chaîne
 - binaire, 51
 - compatible, 51
 - degré Turing, 349
 - extension, 51
 - incompatible, 51
- chemin
 - arbre (espace de Baire), 179
 - arbre binaire, 154
 - f-arbre, 141
- choix
 - axiome, 228, 229, 652, 710, 724
 - dénombrable, 653
- Church-Turing (thèse), 104
- classe, 153
 - $\mathcal{C}$, 720
 - Δ_n^0 , 403
 - Δ_1^1 , 686
 - Π_1^0 , 165
 - Π_2^0 , 400
 - Π_n^0 , 403
 - Π_1^1 , 686
 - $\mathcal{S}$, 719
 - Σ_1^0 , 165
 - Σ_n^0 , 403
 - Σ_1^1 , 686
 - $\underline{\Delta}_n^0$, 402
 - $\underline{\Pi}_n^0$, 402
 - $\underline{\Pi}_1^1$, 690
 - $\underline{\Sigma}_n^0$, 402
 - $\underline{\Sigma}_1^1$, 690
 - HYP, 718
 - à oracle, 433
 - analytique, 690
 - analytique effective, 686
 - Borélienne, 645
 - co-analytique, 690
 - co-analytique effective, 686
 - co-maigre, 240
 - compacte, 160
 - dense, 239
 - fermée, 158
 - intérieur, 239
 - large, 602
 - maigre, 240
 - mesurable, 408, 695
 - ouverte, 158
 - parfaite, 164, 241
 - théorie des ensembles, 642
 - universelle, 177
- clos
 - formule, 197
 - terme, 196
- Club des cinq, 478, 517
- clôture
 - formule, 483
 - universelle, 484
- co-analytique
 - classe, 690
 - ensemble, 686
- co-maigre (classe), 240
- codage
 - acceptable, 39
 - listes, 123, 215, 536
 - programmes, 35
- code
 - Δ_1^0 (ensemble), 95
 - Δ_α^0 (classe), 676
 - Δ_α^0 (ensemble), 665
 - Δ_n^0 (ensemble), 96
 - Π_α^0 (classe), 676
 - Π_α^0 (ensemble), 665
 - Σ_1^0 (ensemble), 95
 - Σ_α^0 (classe), 676
 - Σ_α^0 (ensemble), 665
 - Σ_n^0 (ensemble), 96
 - canonique d'un ensemble, 97, 131, 539
 - classe Π_1^0 , 165
 - classe Σ_1^0 , 165
 - de lowness, 96
 - de programme, 35
- cohérence
 - $\text{Coh}(T)$, 220
 - théorie, 174, 204
- cohésif
 - ensemble, 610
 - principe, 610

- collection
 - schéma, 528
- coloriage (fonction), 576
- compacité, 160
- compact
 - classe, 160
 - question de forcing, 294
- compatibilité
 - chaîne, 51
 - condition, 274
- complet
 - théorie, 174, 209
 - Turing, 61
- complexité
 - Kolmogorov, 370, 371
 - pleine, 371, 380
 - sans préfixe, 380
- complétion
 - arithmétique de Peano, 174
- complétude
 - théorème de Gödel, 209
- compréhension
 - Δ_1^0 , 496
 - bornée, 540
 - schéma, 484, 732
- condition
 - compatible, 274
 - Dzhafarov-Jockusch, 596
 - forcing, 273
 - incompatible, 274
 - Jockusch-Soare, 274
 - Mathias, 595
 - Sacks, 274
- conjecture
 - Martin, 304
- conservation
 - extension, 543
 - théorème, 543
- continu
 - fonction, 162, 494
 - hypothèse, 229, 654, 716, 726
- contrat, 66, 234
- coupure, 531
- couverture minimale, 341
- couverture minimale forte, 342
- coût
 - fonction, 461
 - fonction adaptative, 461
- cylindre, 158
 - $[W]$, 158
 - $[\sigma]$, 69, 158
 - problème, 572
- cône (degré Turing), 171, 302
- degré
 - c. e., 136, 310
 - calculatoirement dominé, 140
 - calculatoirement énumérable, 136
 - cappable, 332
 - complet/incomplet, 61
 - couverture minimale, 341
 - couverture minimale forte, 342
 - DNC, 132
 - DNC₂, 135, 175
 - DNC_f, 135
 - high, 74
 - hyperimmune, 138
 - low, 73
 - many-one, 89
 - Medvedev, 306
 - minimal, 336
 - Muchnik, 306
 - noncappable, 332
 - PA, 174
 - spectre, 176
 - truth-table, 144
 - Turing, 57
- demi-treillis, 343
 - inférieur, 343
 - supérieur, 343
- dense
 - classe, 239
 - ensemble, 236, 275
 - sous σ , 259
 - sous c , 276
- densité
 - inférieure, 442
 - inférieure positive, 443
 - supérieure, 442
 - supérieure positive, 443
- diagonalement non calculable, 132

- diagramme de programmation, 109
- différence
 - aléatoire, 439
 - test, 439
- distributif
 - treillis, 361
- DNC
 - degré, 132
 - fonction, 132
- DNC_2 , 135, 175
- DNC_f , 135
- domination, 138
- définissable (dans une structure), 545
- dénombrable
 - ensemble, 18
 - ordinal, 651
- échappante
 - fonction, 246
- effectivement immune, 131
- énoncé, 197
 - ADS, 558
 - COH, 610
 - cRT_k^n , 579
 - J, 559
 - KL, 559
 - LLPO, 567
 - LPO, 567
 - RT_k^n , 577
 - WKL, 505
 - WWKL, 509
- ensemble
 - 1-aléatoire, 436
 - 1-générique, 249
 - 2-aléatoire, 436
 - K -trivial, 389
 - X -c. e., 53
 - X -calculable, 53
 - X -calculatoirement énumérable, 53
 - Δ_1^0 , 87
 - Δ_n^0 , 83
 - $\Delta_n^0(X)$, 89
 - Δ_1^1 , 686
 - Π_n^0 , 82
 - Π_n^0 -complet, 91
 - $\Pi_n^0(X)$, 88
 - $\Pi_n^0(X)$ -complet, 91
 - Π_1^1 , 686
 - Σ_1^0 , 87
 - Σ_n^0 , 82
 - Σ_n^0 -complet, 91
 - $\Sigma_n^0(X)$, 88
 - $\Sigma_n^0(X)$ -complet, 91
 - Σ_1^1 , 686
 - n -aléatoire, 436
 - n -générique, 253
 - aléatoire de Chaitin/Levin, 382
 - analytique effectif, 686
 - bien ordonné, 515, 642
 - borné de requêtes, 385
 - c. e., 45
 - c. e. maximal, 49, 150
 - calculable, 36
 - calculatoirement dominé, 140
 - calculatoirement indépendant, 347
 - calculatoirement inséparable, 49
 - calculatoirement énumérable, 45
 - co-analytique effectif, 686
 - codé, 539
 - cohésif, 610
 - complet/incomplet, 61
 - d'entiers, 25, 26
 - dense, 236, 275
 - dense sous σ , 259
 - densité positive, 443
 - DNC_2 , 175
 - définissable, 545
 - dénombrable, 18
 - effectivement immune, 131
 - équipotent, 15
 - faiblement 1-générique, 243, 245
 - faiblement n -générique, 245
 - fortement MLR, 427
 - générique, 236
 - high, 74
 - homogène, 576
 - hyperarithmétique, 665
 - hyperimmune, 131
 - immune, 130
 - indénombrable, 18
 - infini, 539

- low, 73
- low généralisé, 251
- low_n généralisé, 271
- Martin-Löf aléatoire, 416
- minimal, 410
- MLR, 416, 419
- promptement simple, 332
- pré-homogène, 584
- r.e., 45
- récuratif, 37
- récurivement énumérable, 45
- sans préfixe, 379
- sans préfixe minimal, 410
- simple, 389
- sous-jacent, 205
- subpotent, 16
- sémantique de codes, 93
- transitif, 638
- triadique de Cantor, 26
- ensemble complet
 - Π^0_α , 671
 - Π^0_n , 91
 - $\Pi^0_n(X)$, 91
 - Π^1_1 , 693
 - Σ^0_α , 669, 671, 672
 - Σ^0_n , 91
 - $\Sigma^0_n(X)$, 91
 - Σ^1_1 , 743
- entier
 - non standard, 222, 488
 - standard, 488
- Entscheidungsproblem, 99
- équipotence, 15
- équivalence
 - many-one, 89
 - truth-table, 144
 - Turing, 57
- espace
 - Baire, 179
 - Cantor, 25, 158
- étape de calcul, 36
- évitement (chaînes), 249
- exponentielle
 - ordinaire, 647
- extensible
 - nœud, 156
- extension
 - ω -extension, 544, 547
 - chaîne, 51
 - compatible, 274
 - conservative, 543
 - consistante, 349
 - finie (méthode), 66
 - forcing, 273
- f-arbre, 140, 281
 - Γ -scindé, 337
 - Γ -uni, 337
 - chemin, 141
 - sous-f-arbre, 141
 - uniforme, 337
- faiblement
 - 1-générique, 243
 - 1-générique relativisé, 245
 - n -générique, 245
- fermé
 - Baire, 180
 - classe, 158
 - effectif, 165
- feuille, 154
- filtre, 274
 - générique, 275
 - maximal, 274
 - suffisamment générique, 276
- fonction
 - β de Gödel, 215
 - succ (entiers), 101, 117
 - c^n_i , 101, 117
 - p^n_i , 101, 117
 - Ackermann, 121
 - bijective, 15
 - calculable, 32, 36
 - continue, 162, 494
 - de calcul, 65
 - de coût, 461
 - de coût adaptative, 461
 - diagonalement non calculable, 132
 - DNC, 132
 - DNC₂, 135, 175
 - DNC_f, 135
 - domination, 138

- échappante, 246
- générale récursive, 100
- hyperimmune, 138
- injective, 15
- libre de point fixe, 133
- modulus, 675
- primitive récursive, 100, 554
- primitive récursive relativisée, 522
- prouvablement totale, 520
- représentée, 215
- récursive, 100, 107
- Solovay, 470
- stable, 61
- surjective, 15
- fonctionnelle, 52
 - formule, 520
 - Turing, 52
- force
 - syntactiquement, 260, 278
 - sémantiquement, 260, 264, 277
- forcing
 - Cohen, 231
 - condition, 273
 - de Cantor, 276
 - Dzhafarov-Jockusch, 596
 - extension, 273
 - f-arbre, 140
 - Jockusch-Soare, 279
 - Mathias, 595
 - question, 288, 290, 292
 - relation, 260, 277, 278
 - relation $\Vdash$, 260, 264
 - relation $\Vdash^*$, 235, 260
 - Sacks, 281
 - syntactique, 278
 - sémantique, 260, 264, 277
- forme
 - normale, 689
 - normale relativisée, 690
 - prénexe, 202
- formule
 - Δ_0 , 213
 - $\mathcal{L}_{Z_2}$, 483
 - Π_n^0 , 233
 - Π_n , 213
 - Σ_n^0 , 233
 - Σ_n , 213
 - à paramètres, 490
 - arithmétique, 197, 483
 - arithmétique du second ordre, 483
 - atomique, 197
 - close, 197, 483
 - fonctionnelle, 520
 - forme prénexe, 202
 - littéral, 197
 - paramétrée, 206
 - prouvablement totale, 520
 - second ordre, 232
 - vraie, 206
- fortement MLR, 427
- fusionnable (II-fusionnable), 295
- Gödel
 - théorème d'incomplétude, 218, 220
- goto (programme), 110
- général récursif
 - fonction, 100
 - prédicat, 118
- générique
 - ensemble, 236
 - filtre, 275
 - suffisamment, 237
- Gödel
 - fonction β , 215
 - théorème de complétude, 209
- hauteur
 - arbre, 660
- Henkin
 - structure, 486
 - témoin, 210, 211
- high, 74
- Hilbert
 - programme, 553
- hiérarchie
 - ω_α^{ck} (ordinaux), 727
 - analytique, 516
 - arithmétique, 81
 - arithmétique relativisée, 88

- Borélienne, 645
- borélienne, 402
- Borélienne effective, 403, 676
- hyperarithmétique, 665
- hypersaut, 726
- Kleene, 665
- homogène
 - ensemble, 576
- hyperarithmétique
 - ensemble, 665
 - hiérarchie, 665
 - réduction, 707
- hyperimmune
 - degré, 138
 - ensemble, 131, 138
 - fonction, 138
- hyperlow, 711
- hypersaut, 708, 726
 - hiérarchie, 726
- hypothèse
 - du continu, 229, 654, 716, 726
- idéal
 - de saut, 503
 - de Scott, 505
 - Turing, 498
- immune, 130
- impredicativité, 192, 516
- incompatibilité
 - chaîne, 51
- induction
 - Δ_n^0 , 533
 - Π_n^0 , 528
 - Σ_1^0 , 497
 - Σ_n^0 , 528
 - ordonnée, 532
 - schéma, 485
 - transfinie, 649
- indénombrable
 - ensemble, 18
 - ordinal, 652
- infini
 - ensemble, 539
- injection, 15
- instance
 - problème, 557
 - universelle, 614
- intersection décroissante, 159
- intérieur (classe), 239
- invariant
 - opérateur, 302
 - uniformément, 302
- isomorphisme
 - ordre, 643
- itération transfinie
 - ATR_0 , 515, 732
 - hypersaut, 726
 - saut Turing, 635
 - saut Turing relativisé, 675
- jeu
 - Borélien, 226
 - de réduction, 569
 - détermination de Martin, 226
- jointure
 - effective, 59, 253
 - treillis, 343
- K-trivial, 389
- König
 - lemme, 180
 - lemme faible, 155
- KC (théorème), 386
- Kleene
 - $\mathcal{O}$, 634
 - $\mathcal{O}$ relativisé, 663
 - hiérarchie, 665
 - ordinal, 656
 - point fixe, 41
- Kleene-Brouwer
 - ordre, 662
- Kolmogorov
 - complexité, 370, 371
- langage
 - $\mathcal{L}_{Z_2}$, 482
 - arithmétique, 195
 - arithmétique du second ordre, 482
 - premier ordre, 196
- large (classe), 602
- Lebesgue
 - lemme de densité, 420
 - mesure, 399, 407
 - théorème de densité, 442

- lemme
 - densité de Lebesgue, 420
 - déduction, 200, 210
 - Gauss, 535
 - König, 180
 - König faible, 155
 - limite de Shoenfield, 61
 - remplissage, 40
- libre
 - variable, 197
- libre de point fixe
 - fonction, 133
- littéral
 - formule, 197
- lié
 - variable, 197
- low
 - ensemble, 73
 - généralisé, 251
 - hyperlow, 711
 - pour l'aléatoire de Martin-Löf, 435
 - pour-K, 445
- low_n généralisé
 - ensemble, 271
- low-pour-K, 445
- lowness, 445
- machine, 370
 - à oracle, 54, 433
 - à registre, 107
 - de Turing, 103
 - sans préfixe, 379
 - universelle, 370
- machine de Turing
 - universelle, 38, 127
- maigre (classe), 240
- majorant, 336
- many-one
 - degré, 89
 - réduction, 89
- Martin
 - conjecture, 304
 - théorème de détermination, 226
- Martin-Löf
 - aléatoire, 416
 - aléatoire relativisé, 432
 - test, 416
 - test relativisé, 432
- masse (problème), 306
- Mathias
 - condition, 595
 - forcing, 595
- maximalité
 - filtre, 274
- Medvedev
 - degré, 306
 - réduction, 306
- mesurable
 - classe, 408, 695
- mesure
 - arbre, 508
 - Lebesgue, 399, 407
 - positive, 508
 - relative, 419
- minimal
 - degré, 336
 - paire, 323
- minimum
 - schéma, 531
- minorant, 336
- MLR, 419
 - approchable par la gauche, 441
 - fortement, 427
- modulus, 65, 675
- modèle
 - β -modèle, 745
 - non standard, 222, 488
 - standard, 488
 - structure, 207
- morphisme
 - arbre, 701
- Muchnik
 - degré, 306
 - réduction, 306
- multiplication
 - ordinaire, 647
- méthode
 - de priorité, 312
 - des extensions finies, 66
 - permission, 311

- non standard
 - entier, 222, 488
 - modèle, 222, 488
 - ordinal, 739
- noncappable
 - degré, 332
- normal
 - forme, 689
- nœud, 154
 - branchant, 154, 179
 - extensible, 156
 - feuille, 154
 - successeur, 154
- Oberwolfach
 - aléatoire, 464
 - test, 464
- opérateur
 - c. e., 302
 - invariant, 302
 - uniformément invariant, 302
- oracle, 52
- ordinal
 - calculable, 657
 - calculable relativisé, 663
 - Church-Kleene, 656
 - constructif, 656
 - constructif relativisé, 663
 - de von Neumann, 638
 - dénombrable, 651
 - fini, 638
 - induction transfinie, 649
 - indénombrable, 652
 - Kleene, 656
 - limite, 637
 - non standard, 739
 - réursion transfinie, 650
 - récursivement inaccessible, 729
 - successeur, 637
- ordre
 - bien fondé, 642
 - Kleene-Brouwer, 662
- ordre partiel
 - plongement, 347
- ouvert
 - Baire, 180
 - classe, 158
 - effectif, 165
- PA
 - degré, 174
 - théorie, 203
- paire
 - exacte, 345
 - minimale, 323
- parallélisation
 - problème, 568
- paramètre, 483
 - formule, 206, 490
- parfait
 - arbre, 164
 - classe, 164
- partie du premier ordre, 544
- passé
 - test de différence, 439
 - test de Martin-Löf, 416
 - test de Solovay, 417
- Peano
 - arithmétique, 203
- permission
 - méthode, 311
- plongement
 - ordre partiel, 347
- point isolé, 164
- Posner
 - astuce, 324
- Post
 - problème, 297
 - problème de correspondance, 298
- premier ordre
 - partie, 544
 - terme, 482
- primitif récursif
 - fonction, 100
 - fonction relativisée, 522
 - prédicat, 118
- principe
 - de cohésion, 610
 - des tiroirs infini, 577
 - limité d'omniscience, 567
 - très limité d'omniscience, 567
- problème, 557
 - Π_2^1 , 557
 - calculatoirement vrai, 565
 - correspondance de Post, 298

- cylindre, 572
- de l'arrêt, 47
- de masse, 306
- Hilbert, 299
- parallélisation, 568
- pavage du plan, 298
- Post, 297
- produit, 572
- satisfaction, 558
- uniformément vrai, 566
- produit
 - problème, 572
- programme
 - goto, 110
 - Hilbert, 553
 - propre, 115
 - structuré, 108
 - universel, 38, 127
- promptement simple, 332
- propriété
 - Borel-Lebesgue, 160
 - de Baire, 263
 - de faiblesse, 66, 560
 - de force, 66
 - de l'usage, 56
 - Heine-Borel, 160
 - syntactique, 93
 - sémantique, 93
- prouvablement
 - calculable, 520
 - total, 520
- pré-ordre, 57
- pré-homogène
 - ensemble, 584
- prédicat, 37
 - Σ_n^0 , 82
 - général récursif, 118
 - primitif récursif, 118
- prénexe (forme), 202
- préservation
 - k cônes, 562
 - forte, 616
 - forte d'un cône, 619
 - hiérarchie arithmétique, 292
 - propriété, 561
- pseudo
 - bon ordre, 736
- puissance du continu, 24
- Q, 484
- question
 - II-fusionnable, 295
 - compacte, 294
 - forcing, 288, 290, 292
 - Sacks, 302
- r.e.
 - ensemble, 45
- RAM (machine), 107
- RCA_0 , 497
- relation
 - forcing, 260, 277, 278
- relativisation, 54
- rencontre (chaîne), 236
- réunion croissante, 159
- Robinson
 - arithmétique, 203
- récursion, 37
 - générale, 100
 - primitive, 100
 - transfinie, 515, 650, 732
- récursivement
 - énumérable, 45
 - inaccessible, 729
- réduction
 - ω -réduction, 558
 - calculatoire, 563
 - calculatoire forte, 571
 - forte de Weihrauch, 572
 - hyperarithmétique, 707
 - jeu, 569
 - many-one, 89
 - Medvedev, 306
 - Muchnik, 306
 - Solovay, 384
 - truth-table, 144
 - Turing, 57
 - Weihrauch, 565
- Sacks
 - forcing, 281
 - stratégie de préservation, 323
 - théorème de densité, 360
- sans préfixe
 - ensemble, 379
 - machine, 379
 - minimal, 410

- satisfaction
 - approximation Δ_2^0 , 461
 - problème, 558
- saut
 - hypersaut, 726
 - idéal, 503
 - itérations transfinies, 635
 - itéré relativisé, 675
 - support, 737
 - Turing, 60
- schéma
 - $\text{BC}\Sigma_n^0/\text{BC}\Pi_n^0$, 541
 - $\text{B}\Sigma_n^0/\text{B}\Pi_n^0$, 528
 - $\text{I}\Delta_0^0$, 526
 - $\text{I}\Delta_n^0$, 533
 - $\text{I}\Sigma_n^0/\text{I}\Pi_n^0$, 528
 - I_{ouvert} , 523
 - $\text{L}\Sigma_n^0/\text{L}\Pi_n^0$, 531
 - collection, 528
 - compréhension, 484, 732
 - compréhension bornée, 540
 - compréhension Δ_1^0 , 496
 - induction, 485
 - induction ordonnée, 532
 - induction Σ_1^0 , 497
 - minimum, 531
 - réursion transfinie, 515, 732
- scindé (f-arbre), 337
- scission (Γ -scission), 337
- Scott
 - idéal, 505
- second ordre
 - arithmétique, 223, 232, 481
 - formule, 232, 483
 - terme, 482
- segment
 - final, 350
 - initial, 350
- simple
 - ensemble, 389
- singleton
 - Π_1^0 , 626
 - Π_2^0 , 626
 - Π_1^1 , 726
- SMN (théorème), 39
- Solovay
 - fonction, 470
 - réduction, 384
 - test, 417
- solution
 - problème, 557
- sous-f-arbre, 141
- spectre (degré), 176
- stable (fonction), 61
- standard
 - ω , 488, 519
 - entier, 488
 - modèle, 488
 - sémantique, 487
- stratégie
 - arbre, 329
 - préservation de Sacks, 323
- structure
 - ω -structure, 490
 - arithmétique, 205
 - ensemble sous-jacent, 205
 - Henkin, 486
 - modèle, 207
 - pleine, 487
- subpotence, 16
- successeur
 - arbre, 154
 - fonction, 101, 117
- suffisamment générique, 237, 276
- suite
 - Catalan, 620
 - Cauchy, 493
 - finie de bits, 51
 - infinie de bits, 25
 - uniformément calculable, 46
- support
 - de saut, 737
- sur un cône, 302
- surjection, 15
- système
 - axiomatique (voir théorie), 202
 - de déduction, 198
- sémantique
 - modèles pleins, 487
 - standard, 487

- tableau c. e., 131
- temps de calcul, 36
- terme
 - arithmétique, 196
 - clos, 196
 - premier ordre, 482
 - second ordre, 482
- test
 - différence, 439
 - Martin-Löf, 416
 - Martin-Löf relativisé, 432
 - Oberwolfach, 464
 - Solovay, 417
- thèse
 - Church-Turing, 104
- théorie, 202
 - ACA'_0 , 513
 - ACA^+_0 , 514
 - ACA_0 , 501
 - ATR_0 , 516, 732
 - $BC\Sigma^0_n/BC\Pi^0_n$, 541
 - $B\Sigma^0_n/B\Pi^0_n$, 528
 - $I\Delta^0_0$, 526
 - $I\Delta^0_n$, 533
 - $I\Sigma^0_n/I\Pi^0_n$, 528
 - I_{ouvert} , 523
 - $L\Sigma^0_n/L\Pi^0_n$, 531
 - $\Pi^1_1\text{-}CA_0$, 516, 733
 - PRA, 554
 - Π^1_1 Comprehension Axiom, 516
 - RCA_0 , 497
 - Q , 203, 484, 522
 - $\Sigma_1\text{-}PA$, 546
 - WKL_0 , 505
 - $WWKL_0$, 509
 - Z_2 , 485
 - Arithmetical Comprehension Axiom, 501
 - Arithmetical Transfinite Recur-
sion, 516
 - arithmétique du second ordre, 485
 - catégorique, 487
 - cohérente, 174, 204
 - complète, 174, 209
 - PA, 203
 - Recursive Comprehension Axiom, 497
 - Weak König's Lemma, 505
 - Weak Weak König's Lemma, 509
 - Zermelo, 224
 - ZF, 229
 - ZFC, 223, 229
- théorème
 - Arslanov, 136
 - base, 168
 - base calculatoirement dominée, 170
 - base d'évitement de cône, 171
 - base de Gandy, 711
 - base de Kreisel, 714
 - base low, 169
 - Bolzano-Weierstrass, 503
 - Borel-Lebesgue, 507
 - Cantor-Bernstein, 17
 - Chaitin, 377
 - cohérence, 208
 - complétude de Gödel, 209
 - de codage, 388
 - densité de Lebesgue, 442
 - densité de Sacks, 360
 - domination de Martin, 147
 - détermination de Martin, 226
 - Friedberg-Muchnik, 312
 - Hindman, 514
 - incomplétude de Gödel, 218, 220
 - incomplétude de Gödel-Rosser, 219
 - inversion de saut de Friedberg, 256
 - KC, 386
 - Kleene, 41
 - Kučera/Gács, 422
 - Löwenheim-Skolem, 543
 - Liu, 601
 - point fixe, 41
 - Posner-Robinson, 256
 - Post, 92
 - Ramsey, 576
 - Ramsey fini, 578
 - restes chinois, 215
 - Rice, 44, 94

- SMN, 39
 - Solovay, 390
 - valeurs intermédiaires, 500
- transitif
 - ensemble, 638
- treillis, 343
 - distributif, 361
 - inférieur, 343
 - jointure, 343
 - supérieur, 343
- truth-table
 - degré, 144
 - réduction, 144
- Turing
 - complet, 61
 - degré, 57
 - équivalence, 57
 - idéal, 498
 - machine, 103
 - réduction, 57
 - saut, 60
 - thèse, 104
- témoin
 - Henkin, 210, 211
 - réduction Weihrauch, 565
- uniformité, 46
- uniformément
 - Σ_n^0 , 85
 - calculable, 63
 - vrai, 566
- universel
 - classe, 177
 - instance, 614
 - machine, 370
- usage, 56
- variable
 - libre, 197
 - liée, 197
- vérité
 - formule, 206
- Weihrauch
 - réduction, 565
 - réduction forte, 572
- WKL₀, 505
- WWKL₀, 509
- Z₂, 485
- ZF, 229
- ZFC, 223, 229

Chez le même éditeur,
la collection *Im-et-Ker*

- 101. — Bernard Randé & Franck Taïeb. *Les clefs pour l'X*
- 102. — Roger Mansuy & Bernard Randé. *Les clefs pour l'X (2)*
- 103. — Françoise Fontanez & Bernard Randé. *Les clefs pour les Mines*
- 104. — Hervé Gianella, Romain Krust, Franck Taïeb & Nicolas Tosel. *Problèmes clefs pour mathématiques supérieures*
- 105. — Roger Mansuy & Bernard Randé. *Les clefs pour la PSI et la PSI**
- 106. — Éric Kouris. *Une année de colles en Math Sup MPSI*
- 107. — Philippe Gallic & Jean-Louis Grappin. *Les clefs pour les Hautes Études Commerciales*
- 108. — Jean-Denis Eiden. *Le jardin d'Eiden. Une année de colles en MP**
- 109. — Maxime Zavidovique. *Un Max de Maths*
- 110. — Jérôme Gärtner. *Mathématiques pour la voie économique et commerciale*
- 111. — Thierry Meyre. *Probabilités. Cours et exercices corrigés (1)*
- 112. — Bernard Randé, Alix Deleporte-Dumont, Quentin Guignard. *Les clefs pour l'écrit MP de mathématiques (session 2015)*
- 113. — Quentin Guignard et Bernard Randé. *Les clefs pour l'oral MP de mathématiques, ENS-X (session 2015)*
- 114. — Clément de Seguin Pazzis. *Les clefs pour l'écrit de mathématiques des concours 2016, filière MP*
- 115. — Ismael Belghiti, Roger Mansuy et Jill-Jënn Vie. *Les clefs pour l'Info*
- 116. — Bernard Randé. *Les nouvelles clefs pour les Mines-CCP (tome I). Oral MP, 2015-16*
- 117. — Georges Skandalis. *Agrégation interne. Algèbre générale, algèbre linéaire et un peu de géométrie*. Nouvelle édition revue et corrigée
- 118. — Jean-Louis Roque. *Florilège d'exercices de l'oral de HEC*
- 119. — Clément de Seguin Pazzis. *Les clefs pour l'écrit MP 2017 – Mathématiques*
- 120. — L. Cozar, N. Jousse, B. Randé, L. Sartre. *Les clefs pour l'écrit de mathématiques et d'informatique. Filière PSI 2015-2016*

- 121. — Georges Skandalis. *Agrégation interne. Analyse*
- 122. — Thomas Blomme, Louise Gassot, Quentin Guignard, Bernard Randé.
Les clefs pour l'oral MP de mathématiques, ENS-X (2016-2017)
- 123.** — Éric Kouris. *Une année de colles en MPSI (nouvelle édition)*
- 124. — Thierry Meyre. *Probabilités. Cours et exercices corrigés (2)*
- 125. — Philippe Caldero et Marie Peronnier. *Carnet de voyage en Algérie*
- 126. — Bernard Randé. *Les clefs pour l'oral MP, mathématiques, ENS-X (session 2018)*
- 127. — Mohamed Amine Ben Boubakeur. *L'indispensable en analyse pour les Spé MP et MP*. Concours Mines-Centrale-X*

Chez le même éditeur,
la collection *Mathématiques en devenir*

- 101. — Jacques Faraut. *Analyse sur les groupes de Lie – Une introduction.* Nouvelle édition revue et augmentée
- 102. — Patrice Tauvel. *Corps commutatifs et théorie de Galois.* Troisième édition revue et bonifiée
- 103. — Jean Saint Raymond. *Topologie, calcul diff. et variable complexe*
- 104. — Clément de Seguin Pazzis. *Invitation aux formes quadratiques*
- 105. — Bruno Ingrao. *Coniques projectives, affines et métriques*
- 106. — Wolfgang Bertram. *Calcul différentiel topologique élémentaire*
- 107. — Henri Lombardi & Claude Quitté. *Algèbre commutative. Méthodes constructives.* Nouvelle édition revue et augmentée
- 108. — Frédéric Testard. *Analyse mathématique. La maîtrise de l'implicite*
- 109. — Grégory Berhuy. *Modules : théorie, pratique... et un peu d'arithmétique.* Nouvelle édition
- 110. — Bernard Candelpergher. *Théorie des probabilités. Une introduction élémentaire (Nouveau tirage bonifié)*
- 111. — Philippe Caldero et Jérôme Germoni. *Histoires hédonistes de groupes et de géométries. Tome premier*
- 112. — Gema-Maria Díaz-Toca, Henri Lombardi & Claude Quitté. *Modules sur les anneaux commutatifs*
- 113. — Philippe Caldero et Jérôme Germoni. *Histoires hédonistes de groupes et de géométries. Tome second – encores*
- 114. — Alain Debreil. *Groupes finis et treillis de leurs sous-groupes*
- 115. — François Rouvière. *Initiation à la géométrie de Riemann*
- 116. — Nikolaï Nikolski. *Matrices et opérateurs de Toeplitz*
- 117. — Philippe Caldero et Jérôme Germoni. *Nouvelles histoires hédonistes de groupes et de géométries. Tome premier (Nouveau tirage)*
- 118. — Martine et Hervé Queffélec. *Analyse complexe et applications.* Nouveau tirage, bonifié et corrigé
- 119. — Alain Debreil, Jean-Denis Eiden, Rached Mneimné et Tuong-Huy NGuyen. *Formes quadratiques et géométrie*
- 120. — Christian Leruste. *Topologie algébrique – Une introduction, et au-delà*

121. — Grégory Berhuy. *Algèbre : le grand combat*. Nouvelle édition
122. — Philippe Caldero et Jérôme Germoni. *Nouvelles histoires hédonistes de groupes et de géométries. Tome second* (Nouveau tirage)
123. — Charles-Michel Marle. *Géométrie symplectique et géométrie de Poisson*
124. — Pascal Boyer. *Petit compagnon des nombres et de leurs applications*
- 125 et 126. — Laurent Le Floch, Frédéric Testard. *Probabilités 1 et 2 – Le hasard est la nécessité*.
127. — David Chiron. *Chemins d'analyse (1) - Espace de Schwartz, distributions Tempérées et transformation de Fourier*.
128. — Gentiana Danila, Jean-Denis Eiden et Rached Mneimné. *Algèbre éclectique*
129. — Yves Coudène. *La géométrie élémentaire d'Euclide à aujourd'hui*

Chez le même éditeur,
la collection *Nano*

- 101. — Benoît Kloeckner. *Un bref aperçu de la géométrie projective*
- 102. — Michel Balazard. *Le théorème des nombres premiers*
- 103. — Bruno Kahn. *Fonctions zêta et L de variétés et de motifs*
- 104. — Patrick Dehornoy. *Le calcul des tresses*
- 105. — Alain Debreil et Rached Mneimné. *Le groupe symétrique $\mathfrak{S}_4$ et ses métamorphoses*
- 106. — Roger Mansuy. *Introduction aux graphes aléatoires (et à la méthode probabiliste)*
- 107. — François Berteloot. *Les familles normales*
- 108. — Antoine Chambert-Loir. *Théorie de l'information — Trois théorèmes de Claude Shannon*
- 109. — Anna Cadoret. *Catégories et représentations*

La collection *La perle et le harnais*

- 101. — Mauricion Garay. *Mathématiques pédestres, le monde pythagorique*
- 102. — Mauricio Garay. *Le monde selon Einstein. Années 1900-1914*
- 103. — Joanne Brueton, Antoine Houlou-Garcia et Bernard Randé. *Le compas et la lyre*

Chez le même éditeur,
la collection *Orizzonti*

- 101. — Michèle Audin. *Souvenirs de Sofia Kowalevskaya*
- 102. — Maurice Kleman. *Chronologie d'un physicien*
- 103. — Rémi Goblots. *L'infini en mathématiques*
- 104. — Michel Garcia. *Mathématiques à travers les siècles (I). Géométrie et arithmétique de l'Antiquité à nos jours*
- 105. — Michel Garcia. *Mathématiques à travers les siècles (II). Essor des mathématiques du XVII^e au XIX^e siècles. Algèbre, analyse, topologie, probabilités*

Imprimé en France et achevé sur les presses de Laballery
Dépôt légal mars 2022