Lowness 4

Prerequisites: Chapters 2 and 3

Recall that a set X is *low* if $X' \leq_T \emptyset'$. Constructing sets of low degree given a notion of forcing with a Σ^0_1-preserving forcing question is not a huge conceptual step from cone avoidance. It simply consists in effectivizing[1] the construction of a generic set with an appropriate representation of forcing conditions and a refined analysis of the properties of the forcing question.

Effectivization of a forcing construction first requires to fix a coding of forcing conditions. Whenever a condition is a finite object, any reasonable coding, such as a Gödel numbering, is sufficient. For any such numbering, one can switch from one representation to the other computably, and this does not affect the complexity of the overall construction. In most cases however, forcing conditions are naturally defined as infinitary mathematical objects, and one must use an appropriate finitary representation of their effective version.

1: Effectiveness is a concept more general than computability. Any construction requiring some amount of computability, such as being c.e., or arithmetic, or even involving some higher computational models, is considered as effective. On the other hand, a forcing construction is not considered as effective, even if its forcing conditions are computable, as the construction of the generic filter does not have any computability restriction.

4.1 Motivation

One of the main motivation of the development of a framework of iterated jump control is reverse mathematics. To prove the existence of an ω-model of a problem P which is not a model of Q, one needs to find an invariant property preserved by P but not by Q. These invariant properties can be divided into two big families: genericity properties, and effectiveness properties.

- A *genericity property* is a property which may locally involve some computability-theoretic features, but does not require the overall construction to be effective. Such properties can be satisfied by every sufficiently generic set for the appropriate notion of forcing. Cone avoidance, preservation of hyperimmunity, or preservation of 1 non-Σ^0_1 definition are examples of such properties.
- An *effectiveness property* is a property which requires the overall construction to satisfy some amount of computability. Being c.e., arithmetic, or of low degree, are examples of such effectiveness properties. Usually, only countably many sets satisfy these properties.

Effectiveness properties are arguably more complex to satisfy than genericity properties, as one usually needs to resort to coding to represent forcing conditions, and the proofs of density require to satisfy some amount of uniformity. This is why genericity properties are preferably used when one only cares about proving a separation from a problem to another in reverse mathematics. On the other hand, effectiveness properties are closer to the original motivation of computability-theory in general, and of reverse mathematics in particular: identifying the right amount of computability needed to find a solution to a problem. From this perspective, the existence of a low solution is very informative.

Definition 4.1.1. A problem P admits a *low basis* if for every set Z and every Z-computable instance X of P, there is a solution Y to X such that $(Y \oplus Z)' \leq_T Z'$. ◊

Besides the intrinsic interest of proving that a problem admits a low basis, such a notion has two technical applications. First, lowness is a natural class of Δ^0_2 sets which is closed under relativization:

Exercise 4.1.2. A set X is *low over* Y if $(X \oplus Y)' \leq_T Y$. Show that if X is low over Y and Y is low, then X is low. ★

It follows that if a problem admits a low basis, then it admits a model with only sets of low degree, and therefore a model with only Δ^0_2 sets.[2]

2: A problem P admits a Δ^0_2 basis if for every set Z and every Z-computable instance X of P, there is a $\Delta^0_2(Z)$ solution Y to X. The Turing jump problem, which to any instance X associates a unique solution X', admits a Δ^0_2 basis, but one easily sees that any ω-model of it contains all the arithmetic sets.

Proposition 4.1.3. Let P be a Π^1_2 problem which admits a low basis. There exists an ω-model of RCA_0 + P with only low sets. ★

Proof. Recall that an ω-model is fully characterized by its second-order part, and that it satisfies RCA_0 iff its second-order part is a Turing ideal. Also recall that $\langle \cdot, \cdot \rangle : \mathbb{N}^2 \to \mathbb{N}$ is Cantor's pairing function.

We are going to define a sequence of sets $Z_0 \leq_T Z_1 \leq_T \dots$ such that for all $n \in \mathbb{N}$,

(1) if $n = \langle e, s \rangle$ and $\Phi_e^{Z_s}$ is a P-instance X, then Z_{n+1} computes a solution to X;
(2) Z_n is of low degree.

$Z_0 = \emptyset$. Suppose we have defined Z_n and say $n = \langle e, s \rangle$. If $\Phi_e^{Z_s}$ is not a P-instance, then let $Z_{n+1} = Z_n$. Otherwise, since P admits a low basis, there is a solution Y to $\Phi_e^{Z_s}$ such that $(Y \oplus Z_n)' \leq_T Z_n' \leq_T \emptyset'$. Let $Z_{n+1} = Z_n \oplus Y$.

Let $\mathcal{I} = \{X \in 2^{\mathbb{N}} : \exists n\ X \leq_T Z_n\}$. By construction, the class $\mathcal{I}$ is a Turing ideal. Moreover, by (1), every P-instance $X \in \mathcal{I}$ admits a solution in $\mathcal{I}$. Last, by (2), every set in $\mathcal{I}$ is of low degree. ■

As an immediate consequence, if a Π^1_2 problem admits a low basis, then it does not imply ACA_0 over RCA_0. Indeed, every ω-model of ACA_0 contains all arithmetic sets by the arithmetic comprehension axiom, thus the model of Proposition 4.1.3 does not satisfy ACA_0. However, as mentioned above, effectiveness properties are harder to satisfy than genericity properties, so since cone avoidance is enough to prove a separation from ACA_0, one usually prefers to prove the latter.

Some other problems, such as Ramsey's theorem for pairs, admit cone avoidance, but not a low basis.[3]

3: The Chain-AntiChain principle (CAC) is the problem whose instances are infinite partial orders, and whose solutions are either infinite chains, or infinite antichains. By Herrmann [21], there is a computable linear order with no Δ^0_2 infinite chains or antichains. Thus, CAC does not admit a Δ^0_2 basis.

The Ascending Descending Sequence principle (ADS) is the problem whose instances are infinite linear orders, and whose solutions are either infinite ascending or descending sequences. By Manaster (see Downey [22]), ADS admits a Δ^0_2 basis, but by Hirschfeldt and Shore [23], there is a computable infinite linear ordering with no low infinite ascending or descending sequence.

It follows that if a Π^1_2 problem admits a low basis, then it implies neither CAC, nor ADS over RCA_0.

Exercise 4.1.4 (Jockusch [16]). Construct a computable coloring $f : [\mathbb{N}]^2 \to 2$ with no Δ^0_2 infinite homogeneous set. ★

Thus, proving that a Π^1_2 problem admits a low basis is a way to separating it from Ramsey's theorem for pairs.

The second technical advantage of the low basis theorem concerns iterated jump control. As we shall see in Chapter 9, iterated jump is much more difficult to control than first jump. On the other hand, if a set G is of low degree, then by Post's theorem, every $\Sigma^0_2(G)$ property is $\Sigma^0_1(G')$, so by lowness is $\Sigma^0_1(\emptyset')$, and again by Post's theorem is Σ^0_2. Thus, if a problem admits a low basis, it satisfies every weakness property at the second jump and higher jump levels.

Exercise 4.1.5. Suppose that a problem P admits a low basis. Let C be a non-Δ^0_2 set, and X be a computable instance of P. Show that there is a solution Y to X such that C is not $\Delta^0_2(Y)$. ⋆

One will therefore rather prove the existence of a low basis than control higher jump if possible.

4.2 Indices

Consider a finite set $F \subseteq \mathbb{N}$. There exists multiple unequivalent ways to represent it by an integer, depending on whether it is considered as finite, computable, c.e., among others. Depending on the representation, some functions such as the cardinality, or the maximum, are not uniformly computable. We explore some natural representations and their limitations.

Definition 4.2.1. The *canonical index* of a finite set $F \subseteq \mathbb{N}$ is the integer $\sum_{x \in F} 2^x$. ◊

The canonical index of a finite set keeps the full information about it. One can list all its elements, compute the size of the set, and decide whether an element belongs to it or not.

Definition 4.2.2. A Δ^0_1-*index*[4] of a computable set $X \subseteq \mathbb{N}$ is an integer $e \in \mathbb{N}$ such that Φ_e is the characteristic function of X. ◊

4: One could as well have considered to code computable sets X by pairs $\langle e, i \rangle$ such that e and i are Σ^0_1-indices of X and $\overline{X}$, respectively. However, one can switch from one representation to the other computably.

Given a Δ^0_1-index e of a computable set $X \subseteq \mathbb{N}$, one can decide uniformly whether an element belongs to it or not. However, one cannot uniformly find a canonical index of a finite set from a Δ^0_1-index:

Lemma 4.2.3 (Soare [3]). There is no partial computable function Φ_e such that for every $n \in \mathbb{N}$, if Φ_n is the characteristic function of a finite set F, then $\Phi_e(n)\downarrow$ and equals the canonical index of F. ⋆

Proof. Suppose Φ_e exists. Using Kleene's fixpoint theorem, define the following total computable function Φ_n, knowing n in advance. $\Phi_n(x)\downarrow= 1$ if x is the least stage such that $\Phi_e(n)[x]\downarrow$, and $\Phi_n(x)\downarrow= 0$ otherwise. By construction, Φ_n is the characteristic function of either the empty set, or a singleton x, thus $\Phi_e(n)\downarrow$ and x is defined. By convention, if $\Phi_e(n)[x]\downarrow$, then $\Phi_e(n)[x] < x$, so $\Phi_e(n)$ is not the canonical index of $\{x\}$. ■

Using a Δ^0_1-index of a finite set F and its cardinality, one can compute the canonical index of F. Therefore, the cardinality function is not uniformly computable from a Δ^0_1-index.

Definition 4.2.4. A Σ^0_1-*index* of a c.e. set $X \subseteq \mathbb{N}$ is an integer $e \in \mathbb{N}$ such that $W_e = X$. ◊

From a Σ^0_1-index of a c.e. set X, one can list exhaustively all its elements over time, but not in order. Furthermore, if X is computable, one cannot uniformly compute a Δ^0_1-index of X.

Lemma 4.2.5 (Soare [3]). There is no partial computable function Φ_e such that for every $n \in \mathbb{N}$, if W_n is computable, then $\Phi_e(n)\downarrow$ and equals a Δ^0_1-index of W_n. ⋆

Proof. Suppose Φ_e exists. Using Kleene's fixpoint theorem, define the following partial computable function Φ_n, knowing n in advance. Let $\Phi_n(0)\downarrow$ if $\Phi_e(n)\downarrow = y$ and $\Phi_y(0)\downarrow = 0$. For every $x > 0$, $\Phi_n(x)\uparrow$. Thus, W_n is either empty, or the singleton 0, so $\Phi_e(n)\downarrow = y$ for some $y \in \mathbb{N}$ such that Φ_y is total. By construction of Φ_n, $\Phi_y(0)\downarrow = 0$, iff $0 \in W_n$, so Φ_y is not the characteristic function of W_n. ■

One can generalize the previous definitions to every level of the arithmetic hierarchy, either using the representation of sets by formulas, or using Post's theorem, by iterations of the Turing jump. Both representations are equivalent, as one can switch from one to another computably.

As we have seen, when using a representation of a mathematical object as part of a larger family of objects, one might loose some information. It is therefore important to choose the most precise representation as possible, given the provided information. For instance, consider a low set X. It is in particular Δ^0_2, so one could use a Δ^0_2-index, that is, an integer e such that $\Phi^{\emptyset'}_e$ is the characteristic function of X. However, this would loose the lowness information of X. It is therefore preferable to represent it by a Δ^0_2-index of X', that is, an integer e such that $\Phi^{\emptyset'}_e$ is the characteristic function of X'.

Definition 4.2.6. A *lowness index* of a low set $X \subseteq \mathbb{N}$ is an integer $e \in \mathbb{N}$ such that $\Phi^{\emptyset'}_e$ is the characteristic function of X'. ◇

Exercise 4.2.7. Show that is no partial computable function Φ_e such that for every $n \in \mathbb{N}$, if $\Phi^{\emptyset'}_n$ is the characteristic function of a low set X, then $\Phi_e(n)\downarrow$ and is a lowness index of X. ⋆

4.3 Coding ideals

Recall that a Turing ideal is a class of sets $\mathcal{M} \subseteq 2^{\mathbb{N}}$ closed under the effective join, and downward-closed under the Turing reduction. Turing ideals are exactly the second-order parts of ω-models of RCA_0.[5]

5: The class of all the computable sets, and the class of all the arithmetic sets are two basic examples of Turing ideals. More generally, given a set X, the class of all X-computable sets is a Turing ideal. On the other hand, the class of all low sets is downward-closed under the Turing reduction, but not closed under the effective join: There exist two low c.e. sets A and B such that $A \cup B = \emptyset'$.

Coding Turing ideals plays an important role in effectivization of forcing constructions, as some combinatorial notions of forcing such as Mathias forcing can be effectivized by restricting their conditions to ω-models of some appropriate theory. For example, solutions to COH can be produced using Mathias forcing over ω-models of RCA_0, in other words, over Turing ideals. Solutions to arbitrary instances of RT^1_2 or computable instances of RT^2_2 can be obtained using a variant of Mathias forcing over ω-models of WKL_0. The second-order part of ω-models of WKL_0 are precisely Scott ideals, that is, Turing ideals which are closed under the existence of PA degrees.

There exist multiple natural ways to code members of countable Turing ideals. The infinite effective join of an infinite sequence $Z_0, Z_1, \dots$ is the set $\bigoplus_i Z_i = \{\langle i, x\rangle : x \in Z_i\}$.

Definition 4.3.1. A set M *codes* a family $\mathcal{M} = \{Z_0, Z_1, \dots\}$ if $M = \bigoplus_i Z_i$. An M-*index* of a set $X \in \mathcal{M}$ is an integer $i \in \mathbb{N}$ such that $X = Z_i$. ◊

By an immediate diagonalization argument, no Turing ideal contains its own code. Therefore, it requires more computational power to compute the code of a Turing ideal than to compute its members. On the other hand, Scott ideals are particularly interesting, as any PA degree computes the code of a Scott ideal. In other words, it does not require more computational power to compute the code of a Scott ideal than to compute its members. Fix an enumeration of all the primitive recursive functionals $T_0, T_1, \dots$ such that for every $X \in 2^{\mathbb{N}}$, T_e^X is an infinite binary tree.[6]

6: Such an enumeration exists, as given a primitive recursive tree functional S_e, one can define a primitive recursive tree functional T_e which, if at some level, sees all the nodes of S_e die, keeps in T_e the last node alive. Thus, given $X \in 2^{\mathbb{N}}$, if S_e^X is infinite, then $T_e^X = S_e^X$, and otherwise, T_e^X is any infinite binary tree.

Theorem 4.3.2 (Scott [24])
The following class is Π^0_1 and non-empty:

$$\mathscr{C} = \left\{ \bigoplus_i Z_i : \forall a \forall b \forall c\ Z_{\langle a,b,c \rangle} \in [T_c^{Z_a \oplus Z_b}] \right\}$$

Moreover, every member of $\mathscr{C}$ codes a Scott ideal.[7]

7: Note that with an appropriate numbering of the listing $T_0, T_1, \dots$, the resulting code M admits some stronger properties: one can computably obtain M-indices of sets witnessing downward-closure, effective join and PA closure. For example, there exists a total computable function which, given an M-index a and a Turing index e such that $\Phi_e^{Z_a}$ is total, outputs an M-index b such that $Z_b = \Phi_e^{Z_a}$.

Proof. The class $\mathscr{C}$ is clearly Π^0_1 and non-empty by choice of $T_0, T_1, \dots$ Let $\bigoplus_i Z_i \in \mathscr{C}$ and say $\mathcal{M} = \{Z_0, Z_1, \dots\}$. We claim that $\mathcal{M}$ is a Scott ideal.

- Downward-closure: Suppose that $Z_a \in \mathcal{M}$ and $Y \leq_T Z_a$. Say $\Phi_e^{Z_a} = Y$ for some $e \in \mathbb{N}$. Then, the primitive recursive tree functional T_b defined by[8]
$$T_c^{A \oplus B} = \{\sigma \in 2^{<\mathbb{N}} : \sigma \text{ and } \Phi_e^A[|\sigma|] \text{ are compatible } \}$$
is such that $[T_c^{Z_a \oplus Z_b}] = \{Y\}$, so $Z_{\langle a,b,c \rangle} = Y \in \mathcal{M}$.
- Effective join: Suppose that $Z_a, Z_b \in \mathcal{M}$. Then the primitive recursive tree functional T_c defined by
$$T_c^A = \{\sigma \in 2^{<\mathbb{N}} : \sigma \prec A\}$$
is such that $[T_c^{Z_a \oplus Z_b}] = \{Z_a \oplus Z_b\}$, so $Z_{\langle a,b,c \rangle} = Z_a \oplus Z_b \in \mathcal{M}$.
- PA closure: Suppose that $Z_a \in \mathcal{M}$. Then the primitive recursive tree functional T_c defined by
$$T_c^{A \oplus B} = \{\sigma \in 2^{<\mathbb{N}} : \forall e < |\sigma|\ \Phi_e^A(e)[|\sigma|]\uparrow \vee \downarrow \neq \sigma(e)\}$$
is such that $[T_c^{Z_a \oplus Z_b}]$ is the class of all $\{0,1\}$-valued DNC functions relative to Z_a. Thus $Z_{\langle a,b,c \rangle}$ is PA over Z_a and in $\mathcal{M}$. ■

8: By "compatible", we mean that for every $x < |\sigma|$, if $\Phi_e^A(x)[|\sigma|]\downarrow$, then the value equals $\sigma(x)$.

In particular, there exists a computable infinite binary tree such that every path codes a Scott ideal.[9]

9: By an immediate relativization, for every set X, there exists an X-computable infinite binary tree such that every path codes a Scott ideal containing X.

Exercise 4.3.3. Let T be a computable tree functional such that for every $X \in 2^{\mathbb{N}}$, $[T^X]$ is the class of all $\{0,1\}$-valued DNC functions relative to X.

1. Show that the class $\{X \oplus Y : X \in T^{\emptyset} \wedge Y \in T^X\}$ is Π^0_1 and non-empty.
2. Deduce that for every PA degree $\mathbf{a}$, there is a PA degree $\mathbf{b} < \mathbf{a}$ such that $\mathbf{a}$ is PA over $\mathbf{b}$. ⋆

Given a Turing ideal $\mathcal{M}$, a set A $\mathcal{M}$-*computes* B if there is some $X \in \mathcal{M}$ such that $B \leq_T A \oplus X$. A Turing ideal $\mathcal{M}$ is *topped by* X if $\mathcal{M} = \{Z \in 2^{\mathbb{N}} : Z \leq_T X\}$.

Computation over Turing ideals can be seen as a generalization of regular computation. Indeed, computation over a topped Turing ideal is nothing but relativized computation. Interesting behaviors happen when working with non-topped Turing ideals, such as Scott ideals. By definition, when a Turing ideal is not topped, it cannot be represented as the collection of sets computable by a single set X. However, Spector [25] proved that every countable Turing ideal can be represented by two sets A and B.

Definition 4.3.4. A pair of sets A, B forms an *exact pair* for a countable Turing ideal $\mathcal{M}$ if $\mathcal{M} = \{Z \in 2^{\mathbb{N}} : Z \leq_T A \wedge Z \leq_T B\}$. ◊

Theorem 4.3.5 (Spector [25])
Every countable Turing ideal $\mathcal{M}$ admits an exact pair.

Proof. Say $\mathcal{M} = \{Z_0, Z_1, \dots\}$. The idea is to construct two sets $G_0 = \bigoplus_n X^0_n$ and $G_1 = \bigoplus_n X^1_n$ such that each column X^i_n for $i \in \{0,1\}$ is equal to the set Z_n, except for a finite number of bits. It is then clear that every set in $\mathcal{M}$ is computable both by G_0 and G_1. However, one must build the sets G_0 and G_1 so that they satisfy the following requirements:[10]

$$\mathcal{R}_{e_0,e_1} : \Phi^{G_0}_{e_0} = \Phi^{G_1}_{e_1} \to \Phi^{G_0}_{e_0} \in \mathcal{M}$$

Consider the notion of forcing whose conditions are 3-tuples (σ_0, σ_1, n) where $\sigma_0, \sigma_1 \in 2^{<\mathbb{N}}$ and $n \in \mathbb{N}$. The parameter n is used to "lock" the n first columns of G_0 and G_1, meaning that from now on, these columns will coincide with the n first sets of $\mathcal{M}$. [11] The *interpretation* of a condition (σ_0, σ_1, n) is the class of all pairs of finite or infinite sequences[12] (G_0, G_1) such that

- $\sigma_i \preceq G_i$;
- for every $k < n$ and every $\langle k, a\rangle$ such that $|\sigma_i| \leq \langle k, a\rangle < |G_i|$, $G_i(\langle k, a\rangle) = Z_k(a)$.

A condition (τ_0, τ_1, m) *extends* (σ_0, σ_1, n) if $n \leq m$ and $(\tau_0, \tau_1) \in [\sigma_0, \sigma_1, n]$. Any filter $\mathcal{F}$ induces two sets $G_{\mathcal{F},0}$ and $G_{\mathcal{F},1}$, defined by $G_{\mathcal{F},i} = \bigcup\{\sigma_i : (\sigma_0, \sigma_1, n) \in \mathcal{F}\}$. Note that $(G_{\mathcal{F},0}, G_{\mathcal{F},1}) \in \bigcap\{[\sigma_0, \sigma_1, n] : (\sigma_0, \sigma_1, n) \in \mathcal{F}\}$. We now prove the core lemma:

Lemma 4.3.6. Let $p = (\sigma_0, \sigma_1, n)$ be a condition and $e_0, e_1 \in \mathbb{N}$. There is an extension (τ_0, τ_1, n) of p forcing $\mathcal{R}_{e_0,e_1}$. ★

Proof. There are three cases:

- Case 1: there is some $x \in \mathbb{N}$ and some finite pair $(\tau_0, \tau_1) \in [\sigma_0, \sigma_1, n]$ such that $\Phi^{\tau_0}_{e_0}(x)\downarrow \neq \Phi^{\tau_1}_{e_1}(x)\downarrow$. Then (τ_0, τ_1, n) is an extension of p forcing $\mathcal{R}_{e_0,e_1}$.
- Case 2: there is some $x \in \mathbb{N}$ and some $i < 2$ such that for every finite pair $(\tau_0, \tau_1) \in [\sigma_0, \sigma_1, n]$, $\Phi^{\tau_i}_{e_i}(x)\uparrow$. Then the condition p already forces $\mathcal{R}_{e_0,e_1}$.
- Case 3: none of Case 1 and Case 2 holds. We claim that p forces $\Phi^{G_0}_{e_0}$ to be either partial, or $Z_0 \oplus \dots \oplus Z_{n-1}$-computable, hence to be in $\mathcal{M}$. Indeed, define the partial $Z_0 \oplus \dots \oplus Z_{n-1}$-computable function h by searching on every input $x \in \mathbb{N}$ for some finite pair $(\tau_0, \tau_1) \in [\sigma_0, \sigma_1, n]$ such that $\Phi^{\tau_1}_{e_1}(x)\downarrow$, and return the output. By negation of Case 2, the function h is total. Moreover, by negation of Case 1, p forces $\Phi^{G_0}_{e_0}$ to be either partial, or equal to h. ■

10: There are three ways to satisfy this requirement: either force partiality of $\Phi^{G_i}_{e_i}$ for some $i < 2$, or force $\Phi^{G_0}_{e_0}$ and $\Phi^{G_1}_{e_1}$ to both halt on a same value and disagree, or force $\Phi^{G_0}_{e_0} \in \mathcal{M}$.

11: This notion of forcing has a similar flavor as the one used in Theorem 3.2.4. In particular, both have a lock playing the same role.

12: More formally, $G_i \in 2^{\leq\mathbb{N}}$, and we let $|G_i| \in \mathbb{N} \cup \{\mathbb{N}\}$ be the length of this sequence.

We are now ready to prove Theorem 4.3.5. Let $\mathcal{F}$ be a sufficiently generic filter for this notion for forcing. For each $i < 2$, let $G_i = G_{\mathcal{F},i}$. For every $k \in \mathbb{N}$, the set of conditions (σ_0, σ_1, n) such that $\min(|\sigma_0|, |\sigma_1|, n) \geq k$ is dense, so if $\mathcal{F}$ is sufficiently generic, then $(G_{\mathcal{F},0}, G_{\mathcal{F},1})$ is a pair of infinite sequences and the set $\{n \in \mathbb{N} : (\sigma_0, \sigma_1, n) \in \mathcal{F}\}$ is infinite. It follows that eventually, the kth column of $G_{\mathcal{F},0}$ will be equal to Z_k, except for a finite number of bits. Thus, every set in $\mathcal{M}$ is both G_0 and G_1-computable. Moreover, by Lemma 4.3.6, if $G_0 \geq_T X$ and $G_1 \geq_T X$, then $X \in \mathcal{M}$. Thus, G_0, G_1 is an exact pair for $\mathcal{M}$. This completes the proof of Theorem 4.3.5. ■

This notion was introduced by Spector to give an alternative proof that the Turing degrees do not form a lattice.

Exercise 4.3.7 (Kleene and Post [26]). Show that for every ascending sequence of sets $X_0 <_T X_1 <_T \dots$, the family $\mathcal{M} = \{Z \in 2^{\mathbb{N}} : \exists n\ Z \leq_T X_n\}$ is a countable Turing ideal. Deduce from Theorem 4.3.5 that there exists two Turing degrees with no greatest lower bound. ⋆

4.4 Basic constructions

As mentioned, low sets are typically obtained by effectivizing the construction of a generic set for a notion of forcing with a Σ^0_1-preserving forcing question. For any reasonable notion of forcing, and any fixed set A, the set of conditions forcing $G \neq A$ is dense. Hence, for any sufficiently generic filter $\mathcal{F}$, the set $G_{\mathcal{F}}$ will not belong to the arithmetic hierarchy or more generally to any fixed countable collection of sets. Thus, effectivizing the construction of a filter restricts its amount of genericity. In particular, for the construction of low sets, 1-genericity is the appropriate amount of genericity.

Definition 4.4.1. A condition p *decides* a formula $\varphi(G)$ if p forces $\varphi(G)$ or its negation. A filter $\mathcal{F}$ *decides* a formula if it contains a condition deciding it. A filter $\mathcal{F}$ is n-*generic*[13] if it decides every Σ^0_n formula. ◇

13: The definition is slightly different for Cohen forcing, but they coincide if one considers an appropriate forcing relation.

When effectivizing forcing constructions, we shall work with infinite decreasing sequences of conditions rather than with actual filters. Recall that any decreasing sequence of conditions $p_0 \geq p_1 \geq \dots$ induces a filter $\mathcal{F} = \{q \in \mathbb{P} : \exists n\ p_n \leq q\}$. By extension, we call such a decreasing sequence n-*generic* if its induced filter is n-generic. In many situations, the partial order will not be computable, and therefore the induced filter will be less computable than the decreasing sequence.

The most basic example of effectivization of a forcing construction is the proof of the existence of a non-computable set of low degree using Cohen forcing.

Theorem 4.4.2
There exists a non-computable set of low degree.

Proof. We shall construct a 1-generic decreasing sequence of Cohen conditions[14] computably in $\emptyset'$. As a byproduct of our decision procedure for 1-genericity, the resulting set G will not be computable. However, for the sake of simplicity, we shall explicitly satisfy the non-computability requirements. We therefore prove two lemmas which will ensure 1-genericity and non-computability, respectively.

14: Cohen conditions are finite objects, and therefore don't need any specific coding.

Lemma 4.4.3. For every condition $\sigma \in 2^{<\mathbb{N}}$ and every Turing index $e \in \mathbb{N}$, there is an extension $\tau \geq \sigma$ deciding $\Phi_e^G(e)\downarrow$. Furthermore, the extension τ and the decision can be obtained $\emptyset'$-computably uniformly in σ and e. ⋆

Proof. The oracle $\emptyset'$ can decide whether there is some $\tau \geq \sigma$ such that $\Phi_e^\tau(e)\downarrow$.[15] In the former case, such a τ can be found computably in σ and e while in the latter case, σ already forces $\Phi_e^G(e)\uparrow$. ■

15: Recall that for a Σ_1^0 formula $\varphi(G)$, $\sigma ?\vdash \varphi(G)$ is defined as $\exists \tau \geq \sigma\ \varphi(\tau)$. Since this is a Σ_1^0-preserving forcing question, $\emptyset'$ can decide whether it holds or not. Furthermore, in either case, the extension witnessing it can be found $\emptyset'$-computably.

Lemma 4.4.4. For every condition $\sigma \in 2^{<\mathbb{N}}$ and every Turing index $e \in \mathbb{N}$, there is an extension $\tau \geq \sigma$ forcing $G \neq \Phi_e$.[16] Furthermore, the extension τ can be obtained $\emptyset'$-computably uniformly in σ and e. ⋆

16: Here, $G \neq \Phi_e$ is a notation for

$$\exists x \Phi_e(x)\uparrow \vee \exists x \Phi_e(x)\downarrow \neq G(x)$$

Proof. Letting $x = |\sigma|$, the oracle $\emptyset'$ can decide whether $\Phi_e(x)\downarrow$ or not. In the former case, let $\tau = \sigma \cdot (1 - \Phi_e(x))$, so that τ forces $G \neq \Phi_e$. In the latter case, σ already forces $G \neq \Phi_e$, so let $\tau = \sigma$. In either case, τ can be found $\emptyset'$-computably uniformly in σ and e. ■

We are now ready to prove Theorem 4.4.2. Thanks to Lemma 4.4.3 and Lemma 4.4.4, define a $\emptyset'$-computable infinite decreasing sequence of Cohen conditions $\sigma_0 \prec \sigma_1 \prec \dots$ such that for every $e \in \mathbb{N}$, σ_{2e+1} decides $\Phi_e^G(e)\downarrow$ and σ_{2e+2} forces $G \neq \Phi_e$. Moreover, for every e, we can ensure that $|\sigma_e| \geq e$, so that $\bigcap_e [\sigma_e]$ is a singleton G. Note that $G = G_{\mathcal{F}}$ where $\mathcal{F}$ is the induced filter for this sequence. By construction, $G' \leq_T \emptyset'$ and G is not computable. This completes the proof of Theorem 4.4.2. ■

Exercise 4.4.5. Every non-computable set of low degree is of hyperimmune degree, so Theorem 4.4.2 implies the existence of a hyperimmune set of low degree. Adapt the proof of Theorem 4.4.2 to directly construct such a set. ⋆

The next example is known as the low basis theorem, and is arguably one of the most useful theorems of computability theory.

Theorem 4.4.6 (Jockusch and Soare [9])
Fix a non-empty Π_1^0 class $\mathcal{P} \subseteq 2^{\mathbb{N}}$. There exists a member $G \in \mathcal{P}$ of low degree.

Proof. Consider the Jockusch-Soare forcing defined in Theorem 3.2.6, that is, the notion of forcing whose conditions are computable infinite binary trees, partially ordered by the inclusion relation. A condition $T \subseteq 2^{<\mathbb{N}}$ can be coded by a Δ_1^0-index, that is, some Turing index b such that $\Phi_b = T$. We shall construct an infinite $\emptyset'$-computable sequence of Δ_1^0-indices $b_0, b_1, \dots$ of a 1-generic decreasing sequence of conditions $T_0 \supseteq T_1 \supseteq \dots$ The following lemma ensures that 1-genericity can be obtained $\emptyset'$-uniformly.

Lemma 4.4.7. For every condition $T \subseteq 2^{<\mathbb{N}}$ and every Turing index $e \in \mathbb{N}$, there is an extension $S \subseteq T$ deciding $\Phi_e^G(e)\downarrow$. Furthermore, a Δ_1^0-index of S and the decision can be obtained $\emptyset'$-computably uniformly in e and a Δ_1^0-index of T. ⋆

Proof. The oracle $\emptyset'$ can decide whether there exists a level $\ell \in \mathbb{N}$ in the tree such that for every $\sigma \in T$ of length ℓ, $\Phi_e^\sigma(e)\downarrow$.[17] In the former case, T already forces $\Phi_e^G(e)\downarrow$. In the latter case, the tree $S = \{\sigma \in T : \Phi_e^\sigma(e)\uparrow\}$ is an extension of T forcing $\Phi_e^G(e)\uparrow$. In both cases, the witness can be found $\emptyset'$-computably. ■

17: Here again, recall that for a Σ_1^0 formula $\varphi(G)$, $T ?\vdash \varphi(G)$ is defined as $\forall P \in [T]\ \varphi(P)$, or equivalently by compactness $(\exists \ell)(\forall \sigma \in T \cap 2^\ell)\varphi(\sigma)$. Since this is a Σ_1^0-preserving forcing question, $\emptyset'$ can decide whether it holds or not. This lemma shows that in either case, the witnessing extension can be found $\emptyset'$-computably.

We are now ready to prove Theorem 4.4.6. Thanks to Lemma 4.4.7, define a $\emptyset'$-computable infinite sequence of Δ^0_1-indices $b_0, b_1, \dots$ of a decreasing sequence of conditions $T_0 \supseteq T_1 \supseteq \dots$ starting with $[T_0] = \mathcal{P}$ and such that for every $e \in \mathbb{N}$, T_{e+1} decides $\Phi^G_e(e)\downarrow$. Note that $\bigcap_e [T_e]$ is a singleton G, as for every $n \in \mathbb{N}$, there is a Turing functional Φ_e such that $\Phi^G_e(e)\downarrow$ iff $G(n) = 1$. Note again that $G = G_{\mathcal{F}}$ where $\mathcal{F}$ is the induced filter for this sequence. By definition of a condition, $G \in [T_0] = \mathcal{P}$, and by construction $G' \leq_T \emptyset'$. This completes the proof of Theorem 4.4.6. ■

In summary, both constructions were obtained by constructing an infinite $\emptyset'$-computable sequence of codes of a 1-generic decreasing sequence of conditions. For Cohen forcing, the situation was slightly simpler as conditions were identified with their own code. In any case, such a sequence was obtained by proving the existence of a Σ^0_1-preserving forcing question such that the codes of their witnessing extensions were obtained $\emptyset'$-computably uniformly in codes of the conditions.

4.5 Weak preservation

Contrary to cone avoidance, it is not necessary to have a Σ^0_1-preserving forcing question to produce a set of low degree. It is sufficient to have a Δ^0_2 forcing question for Σ^0_1 formulas[18], uniformly in its parameters (including the condition, under the appropriate coding). This is in particular the case of the following theorem, stating the existence of an infinite subset of low degree.

What is a sufficient largeness condition for a Σ^0_2 set to have an infinite subset of low degree? Being infinite is not sufficient, as there exists infinite Δ^0_2 sets such that every infinite subset computes $\emptyset'$: consider the set of all initial segments of the halting set $A = \{\sigma \in 2^{<\mathbb{N}} : \sigma \prec \emptyset'\}$. Recall that an *array* is a sequence of pairwise disjoint finite sets $\{F_n\}_{n\in\mathbb{N}}$. An array $\{F_n\}_{n\in\mathbb{N}}$ is c.e. if there is a total computable function $f : \mathbb{N} \to \mathbb{N}$ such that $f(n)$ is the canonical code of F_n. Last, an infinite set A is *hyperimmune* if for every c.e. array $\{F_n\}_{n\in\mathbb{N}}$, there is some $n \in \mathbb{N}$ such that $A \cap F_n = \emptyset$.

18: As mentioned in Section 3.5, Σ^0_n sets are arguably more natural than Δ^0_n sets, as the former class is syntactic, while the latter is semantic. As a consequence, when proving a theorem with a purely combinatorial hypothesis through forcing, the forcing question for Σ^0_1 formulas will naturally be either Σ^0_1-preserving, or not even Δ^0_2. In other words, all constructions in this section will exploit some computational distorsion of the combinatorics. In Theorem 4.5.2, the co-hyperimmunity hypothesis is computability-theoretic and is responsible of this distorsion.

Exercise 4.5.1. Recall that a function $f : \mathbb{N} \to \mathbb{N}$ is hyperimmune if it is not dominated by any computable function. The *principal function* of an infinite set $A = \{x_0 < x_1 < \dots\}$ is the function $p_A : \mathbb{N} \to \mathbb{N}$ defined by $p_A(n) = x_n$. Show that an infinite set A is hyperimmune iff its principal function is hyperimmune. ★

Informally, if A is hyperimmune, then $\overline{A}$ contains a lot of elements. Therefore, co-hyperimmunity is a notion of largeness.

Theorem 4.5.2
For every Σ^0_2 co-hyperimmune set A, there is an infinite set $H \subseteq A$ of low degree.

Proof. Consider a variant of Cohen forcing where conditions $\sigma \in 2^{<\mathbb{N}}$ are subsets of A, that is, $\forall x < |\sigma|\ \sigma(x) = 1 \to x \in A$. To avoid confusion, we shall write $\tau \leq \sigma$ for condition extension and keep $\preceq$ for the usual strings extension. Therefore, $\tau \leq \sigma$ iff $\sigma \preceq \tau$ and $\tau \subseteq A$. The interpretation[19] of a condition σ is $[\sigma] = \{Z \in 2^{\mathbb{N}} : \sigma \prec Z\}$. We shall construct a 1-generic

19: One could have defined $[\sigma]$ as

$$\{Z \in 2^{\mathbb{N}} : \sigma \prec Z \wedge Z \subseteq A\}$$

decreasing sequence of conditions computably in $\emptyset'$. The core of the argument lies in the following lemma.

Lemma 4.5.3. For every condition $\sigma \in 2^{<\mathbb{N}}$ and every Turing index $e \in \mathbb{N}$, there is an extension $\tau > \sigma$ deciding $\Phi_e^G(e)\downarrow$. Furthermore, the extension τ and the decision can be obtained $\emptyset'$-computably uniformly in σ and e. ⋆

Proof. Let 0^n denote the string of length n with only 0's. Given a condition σ, we claim that at least one of the following two Σ_2^0 statements is true:

(1) There is some $\tau \geq \sigma$ with $\tau \subseteq A$ such that $\Phi_e^\tau(e)\downarrow$.
(2) There is some $n \in \mathbb{N}$ such that, letting $\tau = \sigma \cdot 0^n$, for every $\mu \geq \tau$, $\Phi_e^\mu(e)\uparrow$.

Suppose not. Then, by negation of (2) for every $n \in \mathbb{N}$, there is some $\mu_n \geq \sigma \cdot 0^n$ such that $\Phi_e^{\mu_n}(e)\downarrow$. For every $n \in \mathbb{N}$, let $F_n = \{x > |\sigma| + n : \mu_n(x) = 1\}$. By negation of (1), $F_n \cap \overline{A} \neq \emptyset$ for every n. By considering a pairwise disjoint computable sub-collection of sets to obtain a c.e. array, we contradict hypermmunity of $\overline{A}$.

Thus, since both statements are Σ_2^0, search $\emptyset'$-computably for some τ witnessing either case.[20] ■

20: Because of the combinatorial distorsion induced by the co-hyperimmunity assumption, the statement of the forcing question is not natural: Given a Σ_1^0 formula $\varphi(G)$, let $\sigma \mathrel{?\vdash} \varphi(G)$ hold if the first witness found in the $\emptyset'$-computable search belongs to the first case.

We are now ready to prove Theorem 4.5.2. Thanks to Lemma 4.5.3, define a $\emptyset'$-computable infinite decreasing sequence of conditions $\sigma_0 \geq \sigma_1 \geq \dots$ such that for every $e \in \mathbb{N}$, σ_{e+1} decides $\Phi_e^G(e)\downarrow$. Moreover, since A is co-hyperimmune, it is infinite, so for every e, we can ensure that $\operatorname{card} \sigma_e = \{n : \sigma_e(n) = 1\} \geq e$ by waiting $\emptyset'$-computably for some new elements of A to be enumerated. As a consequence, $\bigcap_e [\sigma_e]$ is a singleton G. Note that $G = G_{\mathcal{F}}$ where $\mathcal{F}$ is the induced filter for this sequence. By construction, $G' \leq_T \emptyset'$ and G is an infinite subset of A. This completes the proof of Theorem 4.5.2. ■

Theorem 4.5.2 has some interesting consequences for the computable analysis of partial and linear orders. Let ω be the order type of $(\mathbb{N}, <)$. Given two order types α, β, let α^* be the reverse order, and $\alpha + \beta$ be the order type such that every element of α is smaller than every element of β. A linear order $\mathcal{L} = (\mathbb{N}, <_{\mathcal{L}})$ is ***stable*** if it is of order type $\omega + \omega^*$, that is, for every element $x \in \mathbb{N}$, either $\forall^\infty y(x <_{\mathcal{L}} y)$ or $\forall^\infty y(x >_{\mathcal{L}} y)$. Here, the notation $\forall^\infty$ means "for all but finitely many".

Exercise 4.5.4 (Hirschfeldt and Shore [23]). Let $\mathcal{L} = (\mathbb{N}, <_{\mathcal{L}})$ be a computable stable linear order. Let $A = \{x : \forall^\infty y\ (x <_{\mathcal{L}} y\}$ and $A^* = \{x : \forall^\infty y\ (y <_{\mathcal{L}} x\}$.

1. Show that $A \sqcup A^* = \mathbb{N}$ and A is Δ_2^0.
2. Show that A and A^* are immune iff they are hyperimmune.[21]
3. Use Theorem 4.5.2 to prove that $\mathcal{L}$ admits an infinite ascending or descending sequence of low degree. ⋆

21: An infinite set A is *immune* if it has no infinite computable subset, or equivalently no infinite c.e. subset.

4.6 Beyond $\emptyset'$

Some problems do not admit a low basis, but always have a solution which is close to being low, in the sense that every PA degree over $\emptyset'$ computes the jump of a solution. The various basis theorems for Π^0_1 classes show that PA degrees share many features of the $\mathbf{0}$ degree: the computably dominated and the cone avoidance basis theorems say that the existence of a PA degree does not help computing fast-growing functions[22], or computing fixed non-computable sets. By relativization over $\emptyset'$, having the jump of a solution computed by any PA degree over 0' is close to having a the jump of a solution computed by $\emptyset'$, in other words to having a solution of low degree.

22: In the sense that a non-decreasing hyperimmune function is growing so fast that no computable function dominates it.

Definition 4.6.1. A problem P admits a *weakly low basis* if for every set Z and every PA degree P over Z', every Z-computable instance X of P admits a solution Y such that $(Y \oplus Z)' \leq_T P$. ◇

At first sight, Definition 4.6.1 does not yield an invariant property, as one would require P to be PA over $(Y \oplus Z)'$ instead of only computing $(Y \oplus Z)'$. However, based on the density properties of PA degrees, Definition 4.6.1 is actually equivalent to the stronger statement.

Exercise 4.6.2. Use Exercise 4.3.3 to prove that if a problem P admits a weakly low basis, then for every set Z and every PA degree P over Z', every Z-computable instance X of P admits a solution Y such that P is of PA degree over $(Y \oplus Z)'$. ★

A set X is of *low*$_2$ degree if $X'' \leq_T \emptyset''$. If a problem admits a weakly low basis, then it always admits solutions of low$_2$ degree, by choosing an appropriate PA degree.

Exercise 4.6.3. A problem P admits a *low*$_2$ *basis* if for every set Z and every Z-computable instance X of P, there is a solution Y to X such that $(Y \oplus Z)'' \leq_T Z''$. Use the low basis theorem for Π^0_1 classes (Theorem 4.4.6) to show that if P admits a weakly low basis, then it admits a low$_2$ basis. ★

As for sets of low degree, if a set G is of low$_2$ degree, then by Post's theorem, every $\Sigma^0_3(G)$ property is Σ^0_3. Thus, if a problem admits a low$_2$ basis, then it satisfies every weakness property at the third and higher jump levels. Some weakness properties at the second jump level are also preserved, depending on the existence of the appropriate basis theorem for Π^0_1 classes.

Exercise 4.6.4. Suppose that a problem P admits a weakly low basis. Let C be a non-Δ^0_2 set, and X be a computable instance of P. Use the cone avoidance basis theorem for Π^0_1 classes (Theorem 3.2.6) to show that there is a solution Y to X such that C is not $\Delta^0_2(Y)$. ★

There is a well-known correspondence between computability and definability. By Post's theorem, Δ^0_n sets are exactly the $\emptyset^{(n-1)}$-computable ones. Historically, the Turing jump of a set X is defined as $X' = \{e : \Phi^X_e(e)\downarrow\}$, but it could be equivalently defined as the set of codes of true $\Sigma^0_1(X)$ formulas. PA degrees also admit a characterization in terms of decidability of formulas:

Exercise 4.6.5. Let $\varphi_0, \varphi_1, \dots$ be an effective enumeration of all $\Pi^0_1(X)$ sentences. Show that any PA degree over X computes a total function $f : \mathbb{N}^2 \to 2$ such that for every $(a, b) \in \mathbb{N}^2$ for which at least one of φ_a, φ_b is true, if $f(a, b) = 0$ then φ_a is true, and if $A(n) = 1$ then φ_b is true.[23] ★

23: If φ_a and φ_b have the same truth value, then $f(a, b)$ can be either 0 or 1 but must output a value anyway. The careful reader will have recognized the behavior of $\{0, 1\}$-valued DNC functions.

By Post's theorem, any PA degree over $\emptyset'$ is able to choose, given a sequence of pairs of Π^0_2 formulas such that for every pair at least one is true, a sequence of true formulas. Among the natural Π^0_2 formulas, we shall be particularly interested in infinity of a computable set.

Exercise 4.6.6. Let $X_0, X_1, \dots$ a uniformly computable sequence of sets. Use Exercise 4.6.5 to show that any PA degree over $\emptyset'$ computes a sequence $A \in 2^{\mathbb{N}}$ such that for every n, if $A(n) = 0$ then X_n is infinite, and if $A(n) = 1$, then $\overline{X}_n$ is infinite. ⋆

4.7 Ramsey's theorem for pairs

The main application of the previous section will be the proof by Cholak, Jockusch and Slaman [27] that Ramsey's theorem for pairs admits a weakly low basis. The *jump*[24] of a problem P is the problem P′ whose instances are Δ^0_2 approximations of an instance X of P, in other words, stable functions $f : \mathbb{N}^2 \to 2$ whose limit is X, and whose solutions are P-solutions to X. Following Theorem 3.4.1, RT^2_2 can be obtained by applying the cohesiveness principle (COH), and then the pigeonhole principle for Δ^0_2 instances ($\mathsf{RT}^{1\prime}_2$).[25] Thanks to Exercise 4.6.2, it suffices to independently prove that COH and $\mathsf{RT}^{1\prime}_2$ admit a weakly low basis to obtain the same conclusion for RT^2_2.

24: The notion of jump of a problem comes from Weihrauch complexity.

25: The problem $\mathsf{RT}^{1\prime}_2$ is also known as D^2_2 in the literature. More generally, D^n_k is the statement "For every Δ^0_n k-partition $A_0 \sqcup \dots \sqcup A_{k-1} = \mathbb{N}$, there is some $i < k$ and an infinite set $H \subseteq A_i$". The practice shows that it is more convenient to think of it as the jump of the pigeonhole principle.

Recall that by Exercise 3.4.3, for every uniformly computable sequence of sets $\vec{R} = R_0, R_1, \dots$, there is a non-empty $\Pi^0_1(\emptyset')$ class $\mathcal{P} \subseteq 2^{\mathbb{N}}$ such that the degrees computing an $\vec{R}$-cohesive set are exactly those whose jump compute a member of $\mathcal{P}$.

Exercise 4.7.1. Use Exercise 3.4.3 to prove that COH admits a weakly low basis, but does not admit a low basis. ⋆

We will now give an alternative direct proof that COH admits a weakly low basis using an effectivization of computable Mathias genericity. This will serve as a warm-up to the proof that $\mathsf{RT}^{1\prime}_2$ admit a weakly low basis.[26]

26: This proof, due to Cholak, Jockusch and Slaman [27], is actually very close to the original proof of Jockusch and Stephan [13], except we decide the jump of an $\vec{R}$-cohesive set C in a set P of PA degree over $\emptyset'$, while the original proof used a Δ^0_2 approximation of P to construct C. In both proofs, there is a "delay" in the satisfaction of cohesiveness: in our case, this is due to the genericity requirements, while in the original proof, the Δ^0_2 approximation of P may take some time to converge to a right answer.

Theorem 4.7.2 (Jockusch and Stephan [13])
Let $\vec{R} = R_0, R_1, \dots$ be an infinite uniformly computable sequence of sets and let P be of PA degree over $\emptyset'$. There exists an infinite $\vec{R}$-cohesive set C such that $C' \leq_T P$.

Proof. Recall that a computable Mathias condition is a Mathias condition (σ, X) whose reservoir X is computable. Any computable Mathias condition (σ, X) can therefore be coded by a pair $\langle \sigma, b \rangle$ such that b is a Δ^0_1-index of X. We shall construct an infinite P-computable sequence of codes $\langle \sigma_0, b_0 \rangle, \langle \sigma_1, b_1 \rangle, \dots$ representing a 1-generic decreasing sequence of computable Mathias conditions $(\sigma_0, X_0) \geq (\sigma_1, X_1) \geq \dots$. The following lemma shows that such a sequence can be obtained $\emptyset'$-computably:

Lemma 4.7.3. For every condition (σ, X) and every Turing index $e \in \mathbb{N}$, there is an extension (τ, Y) deciding $\Phi^G_e(e)\downarrow$. Furthermore, a code for (τ, Y) and the decision can be obtained $\emptyset'$-computably uniformly in a code for (σ, X) and e. ⋆

Proof. The oracle $\emptyset'$ can decide whether there exists a finite string $\rho \subseteq X$ such that $\Phi_e^{\sigma \cup \rho}(e)\downarrow$. If so, then $(\sigma \cup \rho, X \setminus \{0, \dots, |\rho|\})$ is an extension forcing $\Phi_e^G(e)\downarrow$. Otherwise, (σ, X) already forces $\Phi_e^G(e)\uparrow$. Note that a Δ_1^0-index of $X \setminus \{0, \dots, |\rho|\}$ can be computably found in a Δ_1^0-index of X and ρ. Therefore, a code for the extension can be obtained $\emptyset'$-computably uniformly in a code for (σ, X) and e. ∎

Lemma 4.7.3 only requires $\emptyset'$ instead of a PA degree over $\emptyset'$. Therefore, one can obtain a $\emptyset'$-computable 1-generic decreasing sequence of computable Mathias conditions. However, the resulting set will not be $\vec{R}$-cohesive. We need to interleave steps to satisfy cohesiveness for more and more sets. This is the purpose of the following lemma:

Lemma 4.7.4. For every condition (σ, X) and every computable set R, there is an extension (σ, Y) such that $Y \subseteq R$ or $Y \subseteq \overline{R}$. Furthermore, a code for (σ, Y) and the decision can be obtained P-computably uniformly in a code for (σ, X) and a Δ_1^0-index of R. ⋆

Proof. Fix an effective enumeration of all Π_2^0 sentences $\varphi_0, \varphi_1, \dots$ Let $f : \mathbb{N}^2 \to 2$ be the P-computable function satisfying Exercise 4.6.5. From Δ_1^0-indices of X and R, one can compute codes $a, b \in \mathbb{N}$ such that $\varphi_a \equiv \forall x \exists y (y > x \wedge y \in X \cap R)$ and $\varphi_b \equiv \forall x \exists y (y > x \wedge y \in X \cap \overline{R})$. Note that at least one of φ_a and φ_b is true. Thus, if $f(a, b) = 0$, $(\sigma, X \cap R)$ is a valid extension, and if $f(a, b) = 1$, $(\sigma, X \cap \overline{R})$ is a valid extension. In both cases, Δ_1^0-indices of $X \cap R$ and $X \cap \overline{R}$ can be obtained computably from Δ_1^0-indices of X and R, so a code for the extension can be obtained P-computably in a code for (σ, X) and a Δ_1^0-index of R. ∎

We are now ready to prove Theorem 4.7.2. Thanks to Lemma 4.7.3 and Lemma 4.7.4, define a P-computable infinite sequence of codes

$$\langle \sigma_0, b_0 \rangle, \langle \sigma_1, b_1 \rangle, \dots$$

representing a decreasing sequence of computable Mathias conditions

$$(\sigma_0, X_0) \geq (\sigma_1, X_1) \geq \dots$$

such that for every $e \in \mathbb{N}$, $(\sigma_{2e+1}, X_{2e+1})$ decides $\Phi_e^G(e)\downarrow$ and either $X_{2e+2} \subseteq R_e$, or $X_{2e+2} \subseteq \overline{R}_e$. Moreover, for every e, we can ensure that $\operatorname{card} \sigma_e \geq e$, so that $G = \bigcup_e \sigma_e$ is an infinite set. By construction, $G' \leq_T P$ and G is $\vec{R}$-cohesive. This completes the proof of Theorem 4.7.2. ∎

The previous example involved a Σ_1^0-preserving forcing question with the appropriate uniformity properties to build a set of low degree, but the additional requirements to produce a cohesive set used a PA degree over $\emptyset'$. In the following example, the Σ_1^0-preserving forcing question itself will require a PA degree over $\emptyset'$ to produce a code of an extension.

Theorem 4.7.5 (Cholak, Jockusch and Slaman [27])
Let A be a Δ_2^0 set and let P be of PA degree over $\emptyset'$. There exists an infinite set $G \subseteq A$ or $G \subseteq \overline{A}$ such that $G' \leq_T P$.

Proof. By the low basis theorem for Π^0_1 classes (Theorem 4.4.6) and Theorem 4.3.2, there exists a set $M = \bigoplus_n Z_n$ of low degree coding for a Scott ideal $\mathcal{M} = \{Z_0, Z_1, \dots\}$. For simplicity, let $A_0 = A$ and $A_1 = \overline{A}$.

As in the proof of Theorem 3.4.6, consider a variant of Mathias forcing, whose *conditions* are triples (σ_0, σ_1, X) where

1. (σ_i, X) is a Mathias condition for each $i < 2$;
2. $\sigma_i \subseteq A_i$;
3. $X \in \mathcal{M}$.

A condition (τ_0, τ_1, Y) *extends* (σ_0, σ_1, X) if (τ_i, Y) Mathias extends (σ_i, X). Recall that an M-code of a set $X \in \mathcal{M}$ is an integer $a \in \mathbb{N}$ such that $X = Z_a$. A code for a condition (σ_0, σ_1, X) is therefore a 3-tuple $\langle \sigma_0, \sigma_1, a \rangle$ where a is an M-code for X.

Following the proof of Theorem 3.4.6, we shall make the following assumption to ensure that both sets G_0 and G_1 will be infinite:

There is no infinite set $H \subseteq A$ or $H \subseteq \overline{A}$ such that $H \in \mathcal{M}$. (H1)

Since $\mathcal{M}$ contains only sets of low degree, if the assumption is false, then the statement of the theorem holds, so suppose it is true.

Lemma 4.7.6. Suppose (H1). Let $p = (\sigma_0, \sigma_1, X)$ be a condition and $i < 2$. There is an extension (τ_0, τ_1, Y) of p and some $n > |\sigma_i|$ such that $n \in \tau_i$. Furthermore, a code for (τ_0, τ_1, Y) can be found $\emptyset'$-computably uniformly in a code for p and i. ⋆

Proof. If $X \cap A^i$ is empty, then $X \subseteq A^{1-i}$, but $X \in \mathcal{M}$, which contradicts (H1). Thus, there is some $n \in X \cap A^i$. Let $\tau_i = \sigma_i \cup \{n\}$, and $\tau_{1-i} = \sigma_{1-i}$. Then, $(\tau_0, \tau_1, X \setminus \{0, \dots, n\})$ is an extension of p such that $n \in \tau_i$. Moreover, since A is Δ^0_2, and $M' \leq_T \emptyset'$, the oracle $\emptyset'$ can find such an n from an M-code of X and $i < 2$. An M-code of $X \setminus \{0, \dots, n\}$ can be found computably from an M-code of X and n, so a code for (τ_0, τ_1, Y) can be found $\emptyset'$-computably uniformly in a code for p and i. ■

Due to the disjunctive nature of the notion of forcing, we need to redefine what it means for a filter to be 1-generic. Recall that the interpretation of a Mathias condition (σ, X) is the class $[\sigma, X]$ of all sets G such that $\sigma \subseteq G \subseteq \sigma \cup X$. Each condition (σ_0, σ_1, X) has two interpretations, namely, $[\sigma_0, X]$ and $[\sigma_1, X]$, depending on the side.[27] A condition (σ_0, σ_1, X) *decides* $(\varphi_0(G_0), \varphi_1(G_1))$ if there is some $i < 2$ such that (σ_i, X) decides $\varphi_i(G)$. A filter $\mathcal{F}$ *decides* $(\varphi_0(G_0), \varphi_1(G_1))$ if there is a condition $p \in \mathcal{F}$ deciding $(\varphi_0(G_0), \varphi_1(G_1))$. A filter $\mathcal{F}$ is *1-generic* if it decides every pair of Σ^0_1 formulas.

27: This interpretation of a condition is different from the one in the proof of Theorem 3.4.6, where we considered a class of pairs of sets.

Lemma 4.7.7. For every condition $p = (\sigma_0, \sigma_1, X)$ and every pair of Turing indices $e_0, e_1 \in \mathbb{N}$, there is an extension $q = (\tau_0, \tau_1, Y)$ deciding $(\Phi^{G_0}_{e_0}(e_0)\downarrow, \Phi^{G_1}_{e_1}(e_1)\downarrow)$. Furthermore, a code for q and the decision can be obtained P-computably uniformly in a code for p and e_0, e_1. ⋆

Proof. Let $\mathcal{P}$ be the $\Pi^0_1(X)$ class of all $B \in 2^{\mathbb{N}}$ such that, letting $B_0 = B$ and $B_1 = \overline{B}$, for every $i < 2$ and every $\rho \subseteq X \cap B_i$, $\Phi^{\sigma_i \cup \rho}_{e_i}(e_i)\uparrow$. The oracle $\emptyset'$ can decide whether $\mathcal{P}$ is empty or not from an M-code of X, since M is of low degree.[28]

28: The careful reader will have recognized the disjunctive forcing question of Exercise 3.4.10.

- Suppose $\mathcal{P} = \emptyset$. Then, by compactness, there is a level $\ell \in \mathbb{N}$ such that for every set $\beta \in 2^\ell$, letting $\beta_0 = \beta$ and β_1 be the bitwise negation of β, there is some $i < 2$ and some $\rho \subseteq X \cap \beta_i$ such that $\Phi_{e_i}^{\sigma_i \cup \rho}(e_i)\downarrow$. Such an $\ell \in \mathbb{N}$ can be found M-computably from an M-code of X and e_0, e_1. Since A is Δ^0_2, the oracle $\emptyset'$ can find $\beta = A \upharpoonright_\ell$, and the associated $i < 2$ and ρ. Let $\tau_i = \sigma_i \cup \rho$ and $\tau_{1-i} = \sigma_{1-i}$. Then $q = (\tau_0, \tau_1, X \setminus \{0, \dots, |\rho|\})$ is an extension of p such that $(\tau_i, X \setminus \{0, \dots, |\rho|\})$ forces $\Phi_{e_i}^G(e_i)\downarrow$, hence q decides $(\Phi_{e_0}^{G_0}(e_0)\downarrow, \Phi_{e_1}^{G_1}(e_1)\downarrow)$. Moreover, an M-code for $X \setminus \{0, \dots, |\rho|\}$ can be computed from an M-code for X and ρ, so a code for q can be obtained $\emptyset'$-computably from a code for p.
- Suppose $\mathcal{P} \neq \emptyset$. Then one can obtain an M-code for some $B \in \mathcal{P} \cap \mathcal{M}$ computably from an M-code for X. Using Exercise 4.6.5, since P is of PA degree over M', P can find some $i < 2$ such that $X \cap B_i$ is infinite, and an M-code of $X \cap B_i$. The condition $q = (\sigma_0, \sigma_1, X \cap B_i)$ is an extension of p such that $(\sigma_i, X \cap B_i)$ forces $\Phi_{e_i}^G(e_i)\uparrow$, hence q decides $(\Phi_{e_0}^{G_0}(e_0)\downarrow, \Phi_{e_1}^{G_1}(e_1)\downarrow)$. Moreover, a code for q can be obtained P-computably from a code for p.[29] ■

29: Note that in this lemma, a PA degree over $\emptyset'$ is only used in the second case, to find a side of B whose intersection with X is infinite.

We are now ready to prove Theorem 4.7.5. As usual, thanks to Lemma 4.7.6 and Lemma 4.7.7 and we shall construct an infinite P-computable sequence of codes

$$\langle \sigma_{0,0}, \sigma_{1,0}, b_0 \rangle, \langle \sigma_{0,1}, \sigma_{1,1}, b_1 \rangle, \dots, \langle \sigma_{0,s}, \sigma_{1,s}, b_s \rangle, \dots$$

for a 1-generic decreasing sequence of conditions

$$(\sigma_{0,0}, \sigma_{1,0}, X_0) \geq (\sigma_{0,1}, \sigma_{1,1}, X_1) \geq \dots \geq (\sigma_{0,s}, \sigma_{1,s}, X_s) \geq \dots$$

such that for every $s \in \mathbb{N}$, letting $s = \langle e_0, e_1 \rangle$, $(\sigma_{0,s}, \sigma_{1,s}, X_s)$ decides $(\Phi_{e_0}^{G_0}(e_0)\downarrow, \Phi_{e_1}^{G_1}(e_1)\downarrow)$, and there is some $n_0, n_1 > s$ such that $n_i \in \sigma_{i,s}$. Moreover, P computes the side deciding each formula, and the decision. More precisely, P computes two functions $f, g : \mathbb{N}^2 \to 2$ such that for every $e_0, e_1 \in \mathbb{N}$, letting $s = \langle e_0, e_1 \rangle$ and $i = f(e_0, e_1)$, if $g(e_0, e_1) = 0$ then $(\sigma_{i,s}, X_s)$ forces $\Phi_{e_i}^G(e_i)\uparrow$, and if $g(e_0, e_1) = 1$, then $(\sigma_{i,s}, X_s)$ forces $\Phi_{e_i}^G(e_i)\downarrow$.

By the pigeonhole principle, there is a side $i < 2$ such that for every $e_i \in \mathbb{N}$, there is some $e_{1-i} \in \mathbb{N}$ such that $f(e_0, e_1) = i$. Let $G_i = \bigcup_s \sigma_{i,s}$. By definition of a condition, $G_i \subseteq A_i$, and by construction, G_i is infinite. Last, given $e_i \in \mathbb{N}$, to decide $e_i \in G_i'$, search P-computably for some $e_{1-i} \in \mathbb{N}$ such that $f(e_0, e_1) = i$, and output $g(e_0, e_1)$. Thus, $G_i' \leq_T P$. This completes the proof of Theorem 4.7.5. ■

By Exercise 4.7.1, COH admits a weakly low basis, but not low basis. Actually, every computable instance of COH with no computable solution admits no low solution. What about $\mathsf{RT}^{1\prime}_2$? Downey, Hirschfeldt, Lempp and Solomon [28] proved that $\mathsf{RT}^{1\prime}_2$ admits no low basis.

Theorem 4.7.8 (Downey et al [28])
There exists a Δ^0_2 set A with no low infinite subset $H \subseteq A$ or $H \subseteq \overline{A}$.

First, notice that by Theorem 4.5.2, such an A can be neither hyperimmune or co-hyperimmune, as every Σ^0_2 co-hyperimmune set admits an infinite subset of low degree. The proof of Theorem 4.7.8 involves an infinite injury priority construction and is outside the scope of this book.[30]

30: Note that the proof of Theorem 4.7.8 is intrinsically complicated, as Chong, Slaman and Yang [29] constructed a non-standard model of $\mathsf{WKL}_0 + \mathsf{RT}^{1\prime}_2$ with only low sets. They exploited a failure of Σ^0_2-induction.

One can put together Theorem 4.7.2 and Theorem 4.7.5 to prove that Ramsey's theorem for pairs admits a weakly low basis.

Theorem 4.7.9 (Cholak, Jockusch and Slaman [27])
Let $f : [\mathbb{N}]^2 \to 2$ be a computable coloring and let P be of PA degree over $\emptyset'$. There exists an infinite f-homogeneous set G such that $G' \leq_T P$.

Proof. The proof follows the one of Theorem 3.4.1. Fix f and P. Let $\vec{R} = R_0, R_1, \dots$ be the computable sequence of sets defined for every $x \in \mathbb{N}$ by $R_x = \{y \in \mathbb{N} : f(x, y) = 1\}$. By Theorem 4.7.2 and Exercise 4.6.2, there is an infinite $\vec{R}$-cohesive set $X \subseteq \mathbb{N}$ such that P is PA over X'. In particular, for every $x \in X$, $\lim_{y \in X} f(x, y)$ exists. Let $\hat{f} : X \to 2$ be the limit coloring of f, that is, $\hat{f}(x) = \lim_{y \in X} f(x, y)$. By Theorem 4.7.5, there is an infinite $\hat{f}$-homogeneous set $Y \subseteq X$ for some color $i < 2$ such that $(Y \oplus X)' \leq_T P$. Since for every $x \in Y$, $\lim_{y \in Y} f(x, y) = i$, one can Y-computably thin out the set Y to obtain an infinite f-homogeneous subset $H \subseteq Y$. Since $H \leq_T Y$, $H' \leq_T P$. ■

Recall that Seetapun's theorem states that Ramsey's theorem for pairs admits cone avoidance. The modern proof goes through the decomposition into cohesiveness and the pigeonhole principle, but the original proof was direct and left as an exercise (Exercise 3.4.12).

Exercise 4.7.10. Adapt Exercise 3.4.12 to give a direct proof that Ramsey's theorem for pairs admits a weakly low basis. ★