

THE RADO PATH DECOMPOSITION THEOREM

PETER A. CHOLAK, GREGORY IGUSA, LUDOVIC PATEY,
MARIYA I. SOSKOVA, AND DAN TURETSKY

1. INTRODUCTION

Fix $c : [\mathbb{N}]^2 \rightarrow r$, an r -coloring of the pairs of natural numbers. An ordered list of distinct integers, $a_0, a_1, a_2 \dots a_{i-1}, a_i, a_{i+1} \dots$ is a *monochromatic path for color k* , if, for all $i \geq 1$, $c(\{a_{i-1}, a_i\}) = k$. The empty list is considered a path of any color k . Similarly the list of one element, a_0 , is also considered a path of any color k . For any monochromatic path of length two or more the color is uniquely determined. Paths can be finite or infinite. Since all paths considered in this article are monochromatic we will drop the word monochromatic.

Definition 1.1. Let c be an r -coloring of $[\mathbb{N}]^2$ ($[n]^2$). A *path decomposition for c* is a collection of r paths P_0, P_1, \dots, P_{r-1} such that P_j is a path of color j and every integer (less than n) appears on exactly one path.

Improving on an unpublished result of Erdős, Rado [10] published a theorem which implies:

Theorem 1.2 (Rado Path Decomposition, RPD, or RPD_r). *Every r -coloring of the pairs of natural numbers has a path decomposition.*

In Section 2, we provide three different proofs of this result. The first proof makes use of an ultrafilter on the natural numbers. This ultrafilter proof is clearly known but, as far as we are aware, unpublished. The remaining proofs are interesting new modifications of the ultrafilter proof.

All of the proofs presented are highly non-computable. In Section 3, we show that a non-computable proof is necessary. A coloring $c : [\mathbb{N}]^2 \rightarrow r$ is *stable* if and only if $\lim_y c(\{x, y\})$ exists for every x . We

Date: October 11, 2016.

2010 *Mathematics Subject Classification.* 05C55 05C70 03D80 03B30 03F35.

This work was partially supported by a grant from the Simons Foundation (#315283 to Peter Cholak). Igusa was partially supported by EMSW21-RTG-0838506. Soskova's research was supported by Sofia University Science Fund Grant 54/12.04.2016 and by the L'Oréal-UNESCO program "For women in science".

show there is a computable stable 2-coloring c of $[\mathbb{N}]^2$ such that any path decomposition for c computes the halting set. In Section 4, we give a non-uniform proof of the fact that the halting set can compute a path decomposition for any computable 2-coloring.

In Section 5 we show that if our primary Δ_2^0 construction from Section 4 fails then it is possible to find a path decomposition which is as simple as possible: one path is finite and the other computable. But even with this extra knowledge, we show, in Theorem 5.6, that there is no uniform proof of the fact that the halting set can compute a path decomposition for any computable 2-coloring. In Theorem 5.7, we improve this to show no finite set of Δ_2^0 indices works.

In Section 7 we show that the halting set can also compute a path decomposition for stable colorings with any number of colors. The rest of Section 7 discusses Rado Path Decomposition within the context of mathematical logic and, in particular, from the viewpoint of computability theory and reverse mathematics. In Section 6, we discuss two differences between Rado Path Decomposition and Ramsey Theorem for pairs.

Most of the sections can be read in any order, although Section 5 relies on Section 4, and Section 7 relies on Section 2.

Our notation is standard. Outside of Section 7, and possibly Section 5, our use of computability theory and mathematic logic is minimal and very compartmentalized. One needs to be aware of the halting set and the first few levels of the arithmetic hierarchy. A great reference for this material is Weber [13]. For more background in reverse mathematics, including all notions discussed in Section 7, we suggest Hirschfeldt [5].

Our interest in the RPD was sparked by Soukup [11]. Thanks!

2. SOME PROOFS OF RPD

In this section we will provide several proofs of RPD. We need to start with some notation and definitions. The union of pairwise disjoint sets is written as $X_0 \sqcup X_1 \dots \sqcup X_i$. Two sets are equal modulo finite, $X =^* Y$, if and only if their symmetric difference $X \Delta Y$ is finite. If $X_0 \sqcup X_1 \dots \sqcup X_i =^* Z$ then every possible intersection of the X_j 's is empty and their union is equivalent modulo finite to Z . If $Z = \mathbb{N}$ then $X_0 \sqcup X_1 \dots \sqcup X_i$ is a partition of \mathbb{N} .

Definition 2.1. A collection \mathcal{U} of subsets of \mathbb{N} is an *ultrafilter* (on \mathbb{N}) if and only if $\emptyset \notin \mathcal{U}$, \mathcal{U} is closed under superset, \mathcal{U} is closed under intersection, and, for all $X \subseteq \mathbb{N}$, either $X \in \mathcal{U}$ or its complement

$\overline{X} \in \mathcal{U}$. An ultrafilter is *non-principal* if and only if, for all $a \in \mathbb{N}$, $\{a\} \notin \mathcal{U}$.

We will call a subset X of \mathbb{N} *large* if and only if $X \in \mathcal{U}$.

Remark 2.2. The two key facts that we will need about a non-principal ultrafilter \mathcal{U} are as follows.

- (1) \mathcal{U} does not have finite members. (This statement follows by an easy induction on the size of the finite set.)
- (2) If

$$X_0 \sqcup X_1 \dots \sqcup X_i =^* \mathbb{N}$$

then *exactly one* of the X_j is large. (No more than one of these sets of can be large, because if X_{j_0} and X_{j_1} are distinct then they have an empty intersection. Assume that for all j , $\overline{X}_j \in \mathcal{U}$. It follows that $\bigcap_{j \leq i} \overline{X}_j =^* \emptyset \in \mathcal{U}$. But no finite set can be a member of a non-principal ultrafilter, giving us the desired contradiction.)

For the rest of this section a coloring $c : [\mathbb{N}]^2 \rightarrow r$ will be fixed.

2.1. Ultrafilter Proof. The existence of a non-principal ultrafilter on the natural numbers is a strong assumption that unfortunately cannot be shown in Zermelo Fraenkel set theory, see Feferman [2]; the axiom of choice is sufficient see Jech [6]. Nevertheless, we give a proof of RPD that uses this assumption, because we believe that it provides insight into the combinatorics of this statement. Later in this section we will give alternative proofs of RPD that do not use a non-principal ultrafilter.

Let \mathcal{U} be a non-principal ultrafilter. We will denote the set of neighbors of m with color i by

$$N(m, i) = \{n : c(\{m, n\}) = i\}.$$

Note that $N(m, i)$ is computable in our coloring c . Furthermore, if we fix m then the sets $N(m, i)$ where $i < r$ form a partition of \mathbb{N} . By Remark 2.2 for every m there is a unique $j < r$ such that $N(m, j)$ is large. Let $A_j = \{m : N(m, j) \text{ is large}\}$. The sets A_j where $j < r$ also partition \mathbb{N} . If $m \in A_j$ then we will say that m has color j . It follows that every natural number is assigned in this way a unique color.

For any pair of points $m < n$ in A_j , $N(m, j) \cap N(n, j)$ is large. So there are infinitely many $v \in N(m, j) \cap N(n, j)$. For all such v , $c(m, v) = c(v, n) = j$. Note that any such v is likely much larger than m and n and not necessary in A_j .

Construction. We will construct our path decomposition P_0, P_1, \dots, P_{r-1} in stages. Let $P_{j,0} = \emptyset$ for all $j < r$. The path $P_{j,0}$ is the empty path of color j . Assume that for each $j < r$, $P_{j,s}$ is a finite path of color j such that if $P_{j,s}$ is nonempty then its last member is of color j (i.e. in A_j). Assume also that every $t < s$ appears in one of the $P_{j,s}$. If s already appears in one of the r paths, then let $P_{j,s+1} = P_{j,s}$ for all $j < r$. Otherwise, s has some color k . For $j \neq k$, let $P_{j,s+1} = P_{j,s}$. If $P_{k,s}$ is empty, then let $P_{k,s+1} = \{s\}$. Otherwise, let e be the end of the path $P_{k,s}$. There is a v not appearing in any of the finite paths $P_{j,s}$ such that $v \in N(e, k) \cap N(s, k)$. Add v and s to the end of $P_{k,s}$ in that order to get $P_{k,s+1}$. To complete the construction we set $P_j = \lim_s P_{j,s}$ for every $j < r$. The desired path decomposition is given by P_0, P_1, \dots, P_{r-1} . \square

The proof described above is very close to the well known ultrafilter proof of Ramsey's theorem for pairs. To illustrate this we include this proof below. An infinite set H is *homogenous* for c if and only if $c([H]^2)$ is constant. Ramsey's theorem for pairs is the statement that every r -coloring of the pairs of natural numbers has an infinite homogeneous set.

Remark 2.3 (Proof of the existence of a homogenous set for c). Recall that A_0, A_1, \dots, A_{r-1} gives a partition of \mathbb{N} . Fix the unique j such that A_j is large. We can *thin* A_j to get an infinite homogenous set H of color j as follows: we build an infinite sequence $\{h_n\}_{n \in \mathbb{N}}$ of elements in A_j by induction so that $H = \{h_0, h_1, \dots\}$ is as desired. Let h_0 be the least element of A_j . Suppose that we have constructed a homogeneous set $\{h_0, \dots, h_i\} \subseteq A_j$. Since $A_j \cap \bigcap_{k \leq i} N(h_k, j)$ is the finite intersection of large sets, it is also large and hence infinite. We define h_{i+1} to be the least member of $A_j \cap \bigcap_{k \leq i} N(h_k, j)$ that is larger than h_i .

2.2. Cohesive Proof. As noted, above we would like to remove the use of the non-principal ultrafilter from the proof of RPD. For this we will extract the specific relationship that \mathcal{U} had with the sets $N(m, j)$.

Remark 2.4. Reflecting on the above construction, we see that the important things about largeness were that

- (1) for every m there is a unique $j < r$ such that $N(m, j)$ is large,
- (2) large sets are not finite, and
- (3) the intersection of two large sets is large.

Definition 2.5. An infinite set C is *cohesive* with respect to the sequence of sets $\{X_n\}_{n \in \mathbb{N}}$ if and only if for every n either $C \subseteq^* X_n$ or $C \subseteq^* \overline{X_n}$.

Lemma 2.6. *There is a set C that is cohesive with respect to the sequence $\{N(m, j)\}_{j < r, m \in \mathbb{N}}$.*

Proof. Once again we will use a stagewise construction. We will construct two sequences of sets: $\{C_s\}_{s \in \mathbb{N}}$ and $\{R_s\}_{s \in \mathbb{N}}$. The first sequence will be increasing and the second decreasing with respect to the subset relation. Start with $C_0 = \emptyset$ and $R_0 = \mathbb{N}$. Fix some indexing of all pairs $\langle m, i \rangle$. Inductively assume that, for all $\langle m, i \rangle < s$, either $R_s \subseteq N(m, i)$ or $R_s \subseteq \overline{N(m, i)}$, C_s is finite, R_s is infinite, and C_s and R_s are disjoint. At stage $s + 1$, let c_s be the least element of R_s . Let $C_{s+1} = C_s \cup \{c_s\}$. Assume that $s = \langle m, i \rangle$. Since R_s is an infinite set, at least one of $R_s \cap N(m, i)$ or $R_s \cap \overline{N(m, i)}$ is infinite. If $R_s \cap N(m, i)$ is infinite let $R_{s+1} = (R_s \cap N(m, i)) - \{c_s\}$. Otherwise let $R_{s+1} = (R_s \cap \overline{N(m, i)}) - \{c_s\}$. $C = \lim_s C_s = \{c_0, c_1, \dots\}$ is the desired cohesive set. \square

Fix such a set C . We can now redefine largeness by using C instead of an ultrafilter. Call a set X *large* if and only if $C \subseteq^* X$. This new notion of largeness has the three key properties outlined above with respect to the sets $N(m, i)$: for every m there is a unique $j < r$ such that $N(m, j)$ is large, because C cannot be a subset of two disjoint sets, even if we allow a finite error; large sets are not finite, because C is infinite; and the intersection of two large sets is large, because if $C \subseteq^* X$ and $C \subseteq^* Y$ then $C \subseteq^* X \cap Y$. We can now repeat the original construction using this notion of largeness to produce a path decomposition.

2.3. Stable Colorings. Recall that a coloring c is *stable* if and only if for every m the limit $\lim_n c(\{m, n\})$ exists. Rephrasing this property in terms of sets of neighbors, we get that there is a unique $j < r$ such that $N(m, j)$ is cofinite. So to construct a path decomposition for stable colorings we do not even need a cohesive set. We can redefine large to mean cofinite and use once again the original construction.

2.4. Generic Path Decompositions. In this section we will provide a forcing-style construction of a path decomposition. To avoid confusion with our ultrafilter proof, our construction will use sequences of conditions rather than poset filters.

Conditions are tuples $(P_0, P_1 \dots P_{r-1}, X)$ such that

- (1) $X \subseteq \mathbb{N}$ is infinite,
- (2) P_j is a finite path of color j for every $j < r$,
- (3) no integer appears on more than one of the paths, and

- (4) if P_j is nonempty and e_j is its last element then $X \subseteq^* N(e_j, j)$
 (so e_j has color j with respect to X).

It follows that $(\emptyset, \emptyset, \dots, \emptyset, \mathbb{N})$ is a condition, because it trivially satisfies the third requirement. A condition $(\hat{P}_0, \hat{P}_1, \dots, \hat{P}_{r-1}, \hat{X})$ extends $(P_0, P_1, \dots, P_{r-1}, X)$ if and only if, for all j , P_j is an initial subpath of \hat{P}_j , and $\hat{X} \subseteq X$.

Given a sequence of conditions $\langle C_i \rangle_{i \in \mathbb{N}}$ such that for every i , C_{i+1} extends C_i , we think of this sequence as approximating a tuple of paths as follows.

If $C_i = (P_0^i, P_1^i, \dots, P_{r-1}^i, X^i)$, then the sequence $\langle C_i \rangle$ approximates the tuple of paths $(\tilde{P}_0, \tilde{P}_1, \dots, \tilde{P}_{r-1})$ where $\tilde{P}_j = \lim_i P_j^i$.

Such a tuple of paths need not be a path decomposition, since it might happen that some integer does not appear on any of the limit paths. The purpose of the X values in the conditions will be to ensure that the approximated paths do form a path decomposition if the sequence $\langle C_i \rangle$ is *generic* (defined below).

A set of conditions \mathcal{D} is *dense* if every condition is extended by a condition in \mathcal{D} . A sequence $\langle C_i \rangle$ *meets* \mathcal{D} if there is some i such that $C_i \in \mathcal{D}$.

Given any collection of dense sets, a sequence is *generic* for that collection if it meets every \mathcal{D} in that collection. Note that if we have a countable collection of dense sets \mathcal{D}_i then it is straightforward to build a generic sequence for that collection, by inductively choosing each C_{i+1} to extend C_i and be in \mathcal{D}_{i+1} .

Let \mathcal{D}_i be the set of conditions $(P_0, P_1, \dots, P_{r-1}, X)$ such that i is on some path P_j . The lemma below shows that \mathcal{D}_i is dense. Any generic for $\{\mathcal{D}_i\}$ gives a path decomposition for c .

Lemma 2.7. *For every i the set \mathcal{D}_i is dense.*

Proof. Fix i and a condition $(P_0, P_1, \dots, P_{r-1}, X)$. If i is on one of the paths P_j then we are done. Otherwise, X is an infinite set, so there must be a j such that $N(i, j) \cap X$ is infinite. If $k \neq j$ then let $\tilde{P}_k = P_k$. Let $\tilde{X} = X \cap N(i, j)$. If P_j is empty let \tilde{P}_j be i . Otherwise let e be the end of P_k . Since $(P_0, P_1, \dots, P_{r-1}, X)$ is a condition, there is a v such that $v \in N(e, j) \cap N(i, j)$. Let \tilde{P}_j be P_j with v and i added to the end in that order. It follows that $(\tilde{P}_0, \tilde{P}_1, \dots, \tilde{P}_{r-1}, \tilde{X})$ is a condition in \mathcal{D}_i extending $(P_0, P_1, \dots, P_{r-1}, X)$. \square

The generic construction is very much in the style of Rado's original proof.

3. PATH DECOMPOSITIONS WHICH COMPUTE THE HALTING SET

Recall the halting set $K = \{e \mid (\exists s)\varphi_{e,s}(e) \downarrow\}$ is the set of codes e for programs which, when started with input e , halt after finitely many steps. The halting set is one of the first examples of a set that is not computable. The goal of this section is to show the following theorem.

Theorem 3.1. *There is a computable stable 2-coloring c such that any path decomposition of c computes the halting set.*

We devote the rest of the section to the proof of this theorem. For colors we will use RED and BLUE. Once again a coloring c is *stable* if, for all m , $\lim_n c(m, n)$ exists.

We will give a computable stagewise construction for c . The goal will be to construct c so that:

- (1) The BLUE path in any path decomposition is infinite;
- (2) Any path decomposition can compute the elements of K via the following algorithm: If e is a natural number then the construction will associate a marker m_e to e in a way that is computable from any path decomposition for c . We enumerate the BLUE and RED paths until all numbers $x \leq m_e$ have appeared on one of the two paths. Let t be the next element on the BLUE path. Then $e \in K$ if and only if $\varphi_{e,t}(e)$ is defined (i.e. $\varphi_e(e)$ halts after $\leq t$ many steps).

Each $x \in \mathbb{N}$ will have a default color. Initially it will be BLUE. The default color of a number might be changed once during the construction to RED. At stage s , we will define $c(\{x, s\})$ for every $x < s$ and we will always set this value to be the current default color for x . So our construction will produce a stable coloring. To achieve our first goal, it will be sufficient to ensure that for infinitely many intervals $[k, 2k + 1]$ the default color BLUE is never changed. This is because if all elements in the interval $[k, 2k + 1]$ are colored BLUE with every greater number, then, in any path decomposition, the BLUE path must contain a node in this interval: if m is in this interval and on a RED path then the next and previous nodes on this RED path must be a number less than k , so the RED path can only contain at most k of the nodes in this interval. The length of this interval is $k + 1$, so at least one of the nodes in this interval must be BLUE. This idea is reflected in the way we associate markers m_e to elements e .

We will say that a number k is *fresh* at stage s if and only if k is larger than any number mentioned/used at any stage t where $t \leq s$. All markers m_e are initially undefined, i.e. $m_{e,0} \uparrow$. At each stage s before we proceed with the definition of $c(x, s)$ for $x < s$ we first update the

markers: for the least e where $m_{e,s-1}$ is not defined, we will select a fresh number k and define $m_{e,s} = 2k + 2$. (Note that this means that if n is fresh after stage s then $n > 2k + 2$.) Unless we say otherwise (see below) at all later stages t we will keep $m_{e,t} = m_{e,s}$. It will follow that $\lim_s m_{e,s} = m_e$ exists.

We also update the default colors as follows. For every $e < s$ we check if $\varphi_{e,s-1}(e)\uparrow$, $\varphi_{e,s}(e)\downarrow$, and $m_{e,s}$ is defined. If so we change the default color of all $x \in [m_{e,s}, s + 1]$ to RED and make all $m_{i,s}$ undefined for all $i > e$. If we can show that this construction satisfies our first goal, then we can easily argue that it also satisfies the second: Fix any path decomposition and assume that t is the first element on the BLUE path after all numbers $x \leq m_e$ have shown up on one of the two paths. Suppose further that $\varphi_e(e)$ halts in s many steps. We must show that $t > s$. If at stage s we have that $m_{e,s}$ is not defined then $t > m_e > s$. If $m_{e,s}$ is defined and we assume that $t < s + 1$ then t has color RED with every number that has not yet appeared on either path and so the BLUE path cannot be extended. It follows that the BLUE path is finite, contradicting our assumption.

For every e the value of the marker $m_{e,s}$ can be cancelled at most e many times and then stays constant, so $\lim_s m_{e,s} = m_e$ does exist. It is furthermore computable from any path decomposition by the following procedure. The marker for 0 is never cancelled, so $m_0 = m_{0,1}$. If we know the value of m_e then we run the construction until we see the first stage t_0 such that $m_e = m_{e,t_0}$. It follows that after stage t_0 we can cancel m_{e+1} only for the sake of e . We can also figure out if $e \in K$ by looking for the first t_1 on the BLUE path after all numbers $x \leq m_e$ have shown up on one of the two paths and checking whether or not $\varphi_e(e)$ halts in t_1 steps. Let $t = \max(t_0, t_1) + 1$. We claim that $m_{e+1,t} = m_{e+1}$. If $e \notin K$ then m_{e+1} is not cancelled at any stage greater than t_0 and is defined by stage t . If $e \in K$ then m_{e+1} can possibly be cancelled after stage t_0 but no later than at stage t_1 and so once again its final value will be defined by stage t .

Finally, by induction on e , we will show that there are e intervals $[k, 2k + 1]$ where the default color BLUE for all x in the interval is never changed and $2k + 1 \leq m_e$. Assume inductively this is true for all $e' \leq e$ and let s be the stage when $m_{e+1,s} = m_{e+1}$ is defined. By construction $m_{e+1,s}$ is defined as $2k_{e+1} + 2$ for some fresh $k_{e+1} > m_e$. The default color for all x in the interval $[k_{e+1}, 2k_{e+1} + 1]$ is never changed from BLUE. \square

4. 2-COLORINGS

As we will see in this section 2-colorings are very special. For this section we will use BLUE and RED as our colors. Perhaps the starting point for all theorems about path decompositions is the following theorem.

Theorem 4.1 (Gerencsér and Gyárfás [3]). *Every 2-coloring c of $[n]^2$ has a path decomposition.*

Proof. We prove this statement by induction on n . Clearly the statement is true for $[2]^2$. Assume c is a 2-coloring of $[n+1]^2$. In particular c induces a 2-coloring on the subgraph $[n]^2$.

By induction there is a path decomposition of $[n]^2$. So there is a RED path, P_r and a BLUE path, P_b , such that, if $i < n$, then i is on exactly one of P_r or P_b .

If P_r is empty then P_b and $\{n\}$ is a path decomposition for c . Similarly if P_b is empty.

Let x_r be the end of the RED path and let x_b be the end of the BLUE path. Look at the color of the edge between x_r and n . If it is RED then add n to the end of P_r to get a path decomposition for c . Similarly, if the color of the edge between x_b and n is BLUE then add n to the end of P_b .

Otherwise look at the color of the edge between x_r and x_b . If this is RED add x_b, n (in that order) to the end of the RED path and remove x_b from the end of the BLUE path. We will say that x_b *switches to RED*. If $c(\{x_r, x_b\})$ is BLUE then add x_r, n (in that order) to the end of the BLUE path and remove x_r from the end of the RED path. In this case x_r *switches to BLUE*. In all cases we have obtained a path decomposition of $[n+1]^2$, thereby completing the inductive step. \square

We are going to extend this theorem to the following:

Theorem 4.2. *If $c : [\mathbb{N}]^2 \rightarrow 2$ then there is a Δ_2^c Path Decomposition. In particular, if c is computable then it has a path decomposition that is computable from the halting set K .*

The rest of this section is devoted to the proof of this theorem. This proof will be nonuniform. We will also discuss other issues along the way. Our first goal is to understand why we cannot simply iterate Theorem 4.1 infinitely often to get such a proof. We need to examine the path constructed in Theorem 4.1 very closely.

Definition 4.3. Suppose \tilde{P}_b is a BLUE path and \tilde{P}_r a RED path. Then the pair $(\tilde{P}_b, \tilde{P}_r)$ is a *one-step path extension* of (P_b, P_r) if exactly one of the following holds:

- (1) \tilde{P}_b is P_b with one additional element at the end and $\tilde{P}_r = P_r$, or
- (2) \tilde{P}_r is P_r with one additional element at the end and $\tilde{P}_b = P_b$, or
- (3) \tilde{P}_b is the path P_b with the last element x_b removed and \tilde{P}_r is P_r with x_b and some integer x added in that order at the end, (in this case x_b switches to RED), or
- (4) \tilde{P}_r is the path P_r with the last element x_r removed and \tilde{P}_b is P_b with x_r and some integer x added in that order at the end, (in this case x_r switches to BLUE).

$(\tilde{P}_b, \tilde{P}_r)$ is a *path extension* of (P_b, P_r) if it can be obtained from (P_b, P_r) by a sequence of one-step path extensions. Also, if $(\tilde{P}_b, \tilde{P}_r)$ is a path extension of (P_b, P_r) , $\tilde{P}_r = P_r$, and x is the last element of \tilde{P}_b , then we say that $(\tilde{P}_b, \tilde{P}_r)$ is a *BLUE path extension* of (P_b, P_r) to x . We similarly define a *RED path extension* of (P_b, P_r) to x .

Note that if x is on one of P_b or P_r and $(\tilde{P}_b, \tilde{P}_r)$ is a path extension, then x is on one of \tilde{P}_b or \tilde{P}_r .

The proof of Theorem 4.1 shows:

Lemma 4.4. *Given any two finite disjoint paths P_b and P_r and an integer n not on either path, we can computably in c find a one-step path extension $(\tilde{P}_b, \tilde{P}_r)$ such that n appears on exactly one of these paths.*

We cannot generalize this lemma for more than 2 colors. In fact, Theorem 4.1 fails for more than 2 colors.

Theorem 4.5 (Pokrovskiy [9]). *Given any $r > 2$ there are infinitely many m such that there is a r -coloring c of $[m]^2$ without a path decomposition.*

As our first attempt to prove Theorem 4.2, we will iterate Lemma 4.4 infinitely often to build paths $P_{b,s}$ and $P_{r,s}$ by stages. Start with $P_{b,0} = P_{r,0} = \emptyset$. At stage $s + 1$, apply Lemma 4.4 to $P_{b,s}$ and $P_{r,s}$ and the least integer not on either path n to get $P_{b,s+1}$ and $P_{r,s+1}$. Once we have constructed these sequences, we need a way to extract from them two paths P_b and P_r and then try to argue that they form a path decomposition. We can do this if the position of every number eventually stabilizes. This idea is captured by the following definition.

Definition 4.6. Suppose that, for every natural number s , $P_s = x_0^s, \dots, x_{k_s}^s$ is a finite BLUE (RED) path. We define the BLUE (RED) path $\lim_s P_s = x_0, x_1, \dots, x_n, \dots$ by $x_n = \lim_s x_n^s$ as long as $\lim_s x_i^s$ exists for all $i \leq n$. If there is an $i \leq n$ for which $\lim_s x_i^s$ does not exist, then we leave x_n undefined.

Given the sequences $\{P_{b,s}\}_{s \in \mathbb{N}}$ and $\{P_{r,s}\}_{s \in \mathbb{N}}$, we know that every n eventually appears on one of $P_{b,s}$ and $P_{r,s}$ at some stage s and remains on either $P_{b,t}$ and $P_{r,t}$ at all stages $t > s$. So if every n only switches between the two sides finitely often then the pair P_b and P_r is a path decomposition. The limit will exist, although we might not have an explicit way to compute the limit.

However, it is possible for an n to switch infinitely often. It is, in fact, even possible to build a c such that every number n switches sides infinitely often. For such a c , it would be the case that the $\limsup_s |P_{b,s}| = \infty$ but the $\lim_s P_{b,s}$ is empty, and likewise for P_r .

We have to alter our approach. We will still build our path decomposition as the stagewise limit of path extensions, although they will no longer be one-step path extensions. At each stage s we will have disjoint finite paths $P_{b,s}$ and $P_{r,s}$. The pair $(P_{b,s+1}, P_{r,s+1})$ will be a path extension of $(P_{b,s}, P_{r,s})$. The integers $x_{r,s}$ and $x_{b,s}$ will be the ends of these paths at stage s . When the stage is clear, we will abuse notation and drop the s in $x_{r,s}$ and $x_{b,s}$. We will need the following:

Definition 4.7. Suppose $(\tilde{P}_b, \tilde{P}_r)$ is a one-step path extension of (P_b, P_r) obtained by Case (3) (i.e. x_b switches to RED and is followed by x on \tilde{P}_r .) We say that x_b *strongly switches* to RED if there is no BLUE path extension to x . We similarly define what it means for x_r to *strongly switch* to BLUE.

We say $(\tilde{P}_b, \tilde{P}_r)$ forms a *strong one-step path extension* of (P_b, P_r) if the pair $(\tilde{P}_b, \tilde{P}_r)$ is a one-step path extension of (P_b, P_r) via either Cases (1) or (2), or via Cases (3) or (4) with a strong switch.

We say that $(\tilde{P}_b, \tilde{P}_r)$ forms a *strong path extension* of (P_b, P_r) if it can be obtained from (P_b, P_r) by a sequence of strong one-step path extensions.

The following lemma is the key combinatorial property that will provide stability to constructions that are performed using strong path extensions.

Lemma 4.8. *If n strongly switches to RED then n can never switch back to BLUE by a path extension. The RED path up to n is stable.*

Before we prove Lemma 4.8, we require a more basic order-preservation lemma concerning path extensions.

Lemma 4.9. *Assume $(\tilde{P}_b, \tilde{P}_r)$ is a path extension of (P_b, P_r) . Assume that n and m are two numbers such that one of the following holds.*

- n appears before m in P_r .
- m appears before n in P_b .

- n appears in P_r and m appears in P_b .

Then one of the following holds.

- n appears before m in \tilde{P}_r .
- m appears before n in \tilde{P}_b .
- n appears in \tilde{P}_r and m appears in \tilde{P}_b .

Proof. The proof is an easy induction argument using the fact that only the last element of a path can ever switch to the other path. Any time the elements attempt to switch which path they are on, the latter element must switch before the earlier element. \square

We now proceed to prove Lemma 4.8.

Proof (Lemma 4.8). Assume not. Let $(P_{b,0}, P_{r,0}), (P_{b,1}, P_{r,1}), (P_{b,2}, P_{r,2}), (P_{b,3}, P_{r,3})$ be finite paths such that

- (1) $(P_{b,1}, P_{r,1})$ is a one-step path extension of $(P_{b,0}, P_{r,0})$ in which n strongly switches to RED,
- (2) $(P_{b,2}, P_{r,2})$ is a path extension of $(P_{b,1}, P_{r,1})$ in which n never switches,
- (3) $(P_{b,3}, P_{r,3})$ is a one-step path extension of $(P_{b,2}, P_{r,2})$ in which n switches to BLUE.

Let m be the element following n in $P_{r,1}$.

By definition of a strong switch, we have that there is no BLUE path extension of $(P_{b,0}, P_{r,0})$ to m . In particular, there is no BLUE path from n to m that does not involve any integers (besides n) from $P_{b,0}$ or $P_{r,0}$.

By hypothesis, n is in $P_{b,3}$, so by Lemma 4.9, m must appear before n in $P_{b,3}$.

But then $P_{b,3}$ has both n and m in it, and thus there is a BLUE path from n to m . This provides a contradiction provided that we can prove that this path P does not involve any integers besides n from $P_{b,0}$ or $P_{r,0}$.

To show this, note that n and m are the last two elements of $P_{r,1}$. In particular, every element of $P_{r,0}$ appears before n in $P_{r,1}$, and so by Lemma 4.9, if it is in $P_{b,3}$, it must appear after n . (This does not happen, although we do not need this fact for this proof.) Likewise, every element, besides n , of $P_{b,0}$ is in $P_{b,1}$. Therefore, by Lemma 4.9, it must appear before m in $P_{b,3}$.

Note that we now have that if n strongly switches to RED, then the RED path up to n is stable: n can never switch back to BLUE, and so nothing can be added to or removed from the RED path before n . \square

Below we will modify the initial construction and require that all switches are strong. This avoids the problem of instability discussed above. We will from now on use only strong path extensions.

We note here that if our goal was only to provide another proof of Theorem 1.2, then we would be done. The analogue of Lemma 4.4 is not true with strong one-step extensions, but it is true with strong extensions, so we could simply use the initial construction with strong extensions to provide a path decomposition for c . However, this process might not produce a Δ_2^c path decomposition for the following reason.

If there are infinitely many strong RED switches, but only finitely many BLUE switches, then the RED path is stabilized in a way that allows it to be computed in a Δ_2^c manner, but the BLUE path can only be computed in a Δ_3^c manner. It is Δ_2^c to recognize a strong switch, but it is Δ_3^c to recognize that an element will never strongly switch in the future. We will discuss this in more detail in Section 4.4. For now, the key point is that we must sacrifice some of the simplicity of the construction in order to provide a construction that can be carried out by a computationally weaker oracle.

We now describe our construction explicitly. As suggested in the above paragraph, the construction will depend on whether the number of strong BLUE switches is finite or infinite and similarly for RED. This leads us to a case-by-case analysis of our path decomposition. In Section 5 we will show that there is no uniform way to produce a Δ_2^c path decomposition, which implies that there is no way to prove Theorem 4.2 without some sort of case-by-case analysis.

The following allows us to define our cases nicely.

Definition 4.10. For a coloring c , we will say that we can *always strongly RED switch* if for every pair (P_b, P_r) of disjoint finite BLUE and RED paths, there is a strong path extension $(\tilde{P}_b, \tilde{P}_r)$ of (P_b, P_r) such that there was a strong RED switch at some point during the path extension between (P_b, P_r) and $(\tilde{P}_b, \tilde{P}_r)$.

We define being able to *always strongly BLUE switch* similarly.

Lemma 4.11. *If the pair (P_b, P_r) witnesses that we cannot always strongly RED switch and $(\tilde{P}_b, \tilde{P}_r)$ is a strong path extension of (P_b, P_r) , then $(\tilde{P}_b, \tilde{P}_r)$ also witnesses that we cannot always strongly RED switch.*

Proof. Strong path extension is transitive. If there is a path extension of $(\tilde{P}_b, \tilde{P}_r)$ that includes a strong RED switch, then that same path extension is also a path extension of (P_b, P_r) that includes a strong RED switch. \square

Our construction of a path decomposition breaks down into three different procedures depending on whether or not we can always strongly BLUE and RED switch.

4.1. We can always strongly BLUE and RED switch. We will inductively define $(P_{b,s}, P_{r,s})$ by multiple stages at once. For each s , $(P_{b,s+1}, P_{r,s+1})$ will be a strong path extension of $(P_{b,s}, P_{r,s})$.

Start with $P_{b,0} = P_{r,0} = \emptyset$.

Let k the least stage where $(P_{b,k}, P_{r,k})$ has yet to be defined. Let x be the least integer not in the paths $P_{b,k-1}$ and $P_{r,k-1}$. If there is a BLUE path extension to x , let $(P_{b,k}, P_{r,k})$ by that path extension. If this fails, try the same for RED. If both fail, switch either x_b or x_r as in Lemma 4.4 to get $P_{b,k}$ and $P_{r,k}$. It follows that this switch is a strong switch. Next we stabilize some initial segment of our paths: Let $(P_{b,k+1}, P_{r,k+1})$ be a strong path extension of $(P_{b,k}, P_{r,k})$ that includes a RED switch, and let $(P_{b,k+2}, P_{r,k+2})$ be a strong path extension of $(P_{b,k+1}, P_{r,k+1})$ that includes a BLUE switch.

We then repeat for the next integer not yet on either path.

All switches are strong switches, and by Lemma 4.8, the limits of these paths exist. Since every integer is placed on our paths at some stage, and since every integer can be moved at most once, we have that every integer is on exactly one of the two limiting paths.

Therefore this construction gives a path decomposition.

4.2. We cannot always strongly RED switch. Let $(P_{b,0}, P_{r,0})$ witness that we cannot always RED switch, and furthermore assume that among all such witnesses the length of $P_{r,0}$ is minimal.

Now consider $(P_{b,s}, P_{r,s})$ and x the least number not on these two finite paths. If there is a BLUE path extension, to x , use that extension for $(P_{b,k+1}, P_{r,k+1})$. If this fails try to do the same for RED.

We claim that one of the two options listed above will always work. Towards a contradiction, suppose that both fail. To add x we must switch (like in Theorem 4.1). Call the resulting pair $(\tilde{P}_b, \tilde{P}_r)$. Then $(\tilde{P}_b, \tilde{P}_r)$ is a strong extension of $(P_{b,s}, P_{r,s})$. So if x_b switches to RED then it strongly RED switches. This is not possible, by our choice of $(P_{b,0}, P_{r,0})$.

It follows that $c(\{x_b, x_r\}) = BLUE$ and x_r strongly switches to BLUE. By Lemma 4.11, since $(\tilde{P}_b, \tilde{P}_r)$ is a strong extension of $(P_{b,s}, P_{r,s})$, it must also witness that we cannot always RED switch. Note that \tilde{P}_r is shorter than $P_{r,s}$, so by the minimality of $P_{r,0}$, we know that $P_{r,s} \neq P_{r,0}$. So x_r was added to the RED path at some stage $t \leq s$.

Hence there is no BLUE path from $x_{b,t}$ to x_r which is otherwise disjoint from $P_{b,t}$ and $P_{r,t}$, as otherwise x_r would have been added to $P_{b,t}$.

On the other hand there is a BLUE path from $x_{b,t}$ to x_b , witnessed by the fact that $x_{b,t}$ and x_b are both on the BLUE path at stage s . (By induction, our construction has no switches up to this point, so $x_{b,t}$ is still on the blue path at stage s .) We also have that the pair x_b, x_r is colored BLUE, so there is a BLUE path from $x_{b,t}$ to x_r , a contradiction.

4.3. We cannot always strongly BLUE switch. This case is dealt with in the same way as the previous one.

4.4. The use of the oracle c' . The existence of a path from x to n is existential in the coloring. The lack of a path from x to n is universal in the coloring. So deciding if “ \tilde{P}_b and \tilde{P}_r is a one-step path extension of P_b and P_r and x_b strongly switches to RED” is universal in the coloring and so computable in c' .

As a result, c' can be used as an oracle to implement both of the above constructions. In the case where we can always strongly RED and BLUE switch, we can then use c' to compute both of the paths because both paths are stabilized by strong switches, and c' can recognize the strong switches. In the case where we cannot strongly RED (BLUE) switch, we can also use c' to compute both of the paths because both paths are already stable: no numbers ever switch from one path to the other.

Note that Definition 4.10 is Π_3 in the coloring. Our division of cases depends on the truth of this statement and the witness to its failure. This is finite information but as a result the proof is not uniform in c' . In Theorem 5.6, we will show that this nonuniformity cannot be removed.

The more naive construction, always greedily adding the next element by a strong extension with no case-by-case breakdown, can be implemented uniformly by c' , but the construction could potentially result in infinitely many RED switches and finitely many BLUE switches (or vice-versa). In this case, the RED path would be computable from c' because the strong RED switches would stabilize it. On the other hand, the BLUE path would not necessarily be computable from c' : the statement that an initial segment of the BLUE path has stabilized is universal in the construction (“for all future steps of the construction, none of these elements ever RED switch”) and so is Π_2 in the coloring.

Thus, the naive construction could potentially produce a path decomposition in which one of the two paths was computable from c'' , but not c' .

In our proof of Theorem 4.2, if we can always switch then both the BLUE and RED paths are infinite. But if we cannot always switch one of the paths might be finite. In that case the constructed paths are both computable from the coloring. In Theorem 5.1, we will see by a more delicate case breakdown that this actually always happens. Thus, although our strongly switching proof does not work for all colorings, it does work for all “difficult” colorings: colorings for which there is no computable solution.

5. UNIFORMITY

In the above section, we have provided a nonuniform Δ_2^0 construction and a uniform Δ_3^0 construction of a path decomposition. Furthermore, Theorem 4.2 showed that, in general, we cannot hope for a construction that is simpler than Δ_2^0 , so the complexity of the construction cannot be reduced. Here we address the question of whether the nonuniformity of the Δ_2^0 construction can be reduced.

Theorem 5.1 shows that if our primary construction for Theorem 4.2 fails, then there must be a path decomposition for c in which one of the two paths is finite and the other is computable from c .

Theorem 5.6 shows that there is no uniform Δ_2^0 path decomposition. Theorem 5.7 improves this to show we cannot even get by with a finite set of possible Δ_2^0 indices for our path composition. Thus, nonuniformity is unavoidable.

In light of this, the result in Theorem 5.1 is the closest possible thing to a reduction of nonuniformity: All of the noncomputable cases are handled by a single uniform Δ_2^0 construction, which cannot have its complexity reduced due to Theorem 4.2, and all of the nonuniform cases (unavoidable, by Theorem 5.7) are handled by computable constructions that are as simple as possible, with one path finite and the other computable.

5.1. When we cannot always strongly BLUE and RED switch.

Theorem 5.1. *There is a computable function f with the property that for any e , if e is an index for a computable coloring c , then either $f(e)$ is an index for a Δ_2^0 path decomposition for c , or there is a computable path decomposition for c in which one of the two paths is finite.*

Note that this theorem relativizes: there is a uniform way to take a coloring c , and attempt to produce a Δ_2^c path decomposition so that

either the attempt succeeds, or there is a c -computable path decomposition for c in which one of the two paths is finite.

Proof. The proof hinges on the case analysis from the proof of Theorem 4.2.

In the case where we can always strongly BLUE and RED switch, the proof is uniform: we alternate between adding the next element, adding a RED switch, and adding a BLUE switch, and our Δ_2^c path decomposition is simply the path decomposition stabilized by the switches. Our function f will be the function corresponding to attempting to do that construction.

It remains to show that if we cannot always RED switch, then there is always a c -computable path decomposition in which one of the two paths is finite. (The case where we cannot always BLUE switch will follow by symmetry.)

As in the proof of Theorem 4.2, let $(P_{b,0}, P_{r,0})$ witness that we cannot always RED switch, and furthermore assume that among all such witnesses the length of $P_{r,0}$ is minimal. Let x_b and x_r be the endpoints of the paths $P_{b,0}, P_{r,0}$. We split our proof into two cases.

Case 1: Assume that for every $n \in \mathbb{N}$, and for every BLUE extension $(P_b, P_{r,0})$ of $(P_{b,0}, P_{r,0})$, if n is not on either P_b or $P_{r,0}$, then there is a BLUE path extension of $(P_b, P_{r,0})$ to n .

In this case, we will use $P_{r,0}$ as our RED path, and grow our BLUE path to cover the rest of \mathbb{N} . We use a basic greedy algorithm.

At stage s , let $(P_{b,s}, P_{r,0})$ be the pair of paths that we have, and let n_s be the smallest number not on either path. We search for a BLUE path extension of $(P_{b,s}, P_{r,0})$ to n_s , and when we find such an extension, we let $(P_{b,s+1}, P_{r,0})$ be the first such extension that we find.

By hypothesis, we will always find such an extension, and this algorithm clearly covers all of \mathbb{N} in the limit.

Case 2: Assume Case 1 does not hold. Let $(P_{b,1}, P_{r,0})$ be a BLUE path extension of $(P_{b,0}, P_{r,0})$, and $n_0 \in \mathbb{N}$ so that there is no BLUE path extension of $(P_{b,1}, P_{r,0})$ to n_0 .

We claim that in this case, we actually have that for every $n \in \mathbb{N}$, and for every RED extension (P_{b_1}, P_r) of $(P_{b,1}, P_{r,0})$, if n is not on either P_{b_1} or P_r , then there is a RED path extension of (P_{b_1}, P_r) to n . Thus we may use the RED analogue of the previous algorithm. The proof of this claim will be somewhat circuitous.

First we show that there is no n such that $(P_{b,1}, P_{r,0})$ has both a RED extension to n and a BLUE extension to n . After this we will show that actually either there is no n such that $(P_{b,1}, P_{r,0})$ has a BLUE extension

to n or there is no n such that $(P_{b,1}, P_{r,0})$ has a RED extension to n . We will then show that the first case is true. Finally, from there, we will show that for every $n \in \mathbb{N}$, and for every RED extension (P_{b_1}, P_r) of $(P_{b,1}, P_{r,0})$ there is a RED path extension of (P_{b_1}, P_r) to n .

During these proofs, we will repeatedly use the following facts.

- (1) $(P_{b,1}, P_{r,0})$ is a witness to the fact that we cannot always RED switch.
- (2) $P_{r,0}$ has minimal length among the RED paths of such witnesses.
- (3) There is no BLUE path extension of $(P_{b,1}, P_{r,0})$ to n_0 .

The first two facts follow from the fact that $(P_{b,1}, P_{r,0})$ is a BLUE path extension of $(P_{b,0}, P_{r,0})$. The third is from our hypothesis in Case 2.

Claim 5.2. *There is no n such that $(P_{b,1}, P_{r,0})$ has both a RED extension to n and a BLUE extension to n :*

Proof. Assume not, and let n_1 be such an n . Replacing n_1 if necessary, we may assume that the BLUE path extension to n_1 and the RED path extension to n_1 do not intersect before n_1 . Consider the edge from n_1 to n_0 .

If this edge is BLUE, then we can add n_0 to the end of the BLUE extension to n_1 , creating a BLUE path to n_0 . This contradicts fact (3).

If the edge is RED, then consider the path extension $(\tilde{P}_b, \tilde{P}_r)$ of $(P_{b,1}, P_{r,0})$ in which \tilde{P}_b is created by the BLUE path extension to n_1 , and \tilde{P}_r is created by the RED path extension to n_1 , but with n_1 removed from the end. Let x be the last element of \tilde{P}_r . Now we have that the edge from x to n_1 is RED, the edge from n_1 to n_0 is RED, and there is no BLUE path extension from $(\tilde{P}_b, \tilde{P}_r)$ to n_0 . So we can strongly switch n_1 to RED. This contradicts fact (1). \square

Claim 5.3. *Either there is no n such that $(P_{b,1}, P_{r,0})$ has a nontrivial BLUE extension to n or there is no n such that $(P_{b,1}, P_{r,0})$ has a nontrivial RED extension to n .*

A path extension $(\tilde{P}_b, \tilde{P}_r)$ of (P_b, P_r) is *nontrivial* if $(\tilde{P}_b, \tilde{P}_r) \neq (P_b, P_r)$.

Proof. Assume not, and let n_1, n_2 be such that $(P_{b,1}, P_{r,0})$ has a nontrivial BLUE extension to n_1 and a nontrivial RED extension to n_2 . Consider the edge between n_1 and n_2 . If the edge is RED, then we can use it with the RED path to n_2 to create a RED path to n_1 . But then $(P_{b,1}, P_{r,0})$ has both a RED extension to n_1 and a BLUE extension to n_1 , contradicting Claim 1. If the edge is BLUE, we can similarly conclude that $(P_{b,1}, P_{r,0})$ has both a RED and a BLUE extension to n_2 , again contradicting Claim 5.2. \square

Claim 5.4. *There is no n such that $(P_{b,1}, P_{r,0})$ has a nontrivial BLUE extension to n .*

Proof. Again, assume not. Then by Claim 5.3, there is no n such that $(P_{b,1}, P_{r,0})$ has a nontrivial RED extension to n . In particular, $(P_{b,1}, P_{r,0})$ does not have a RED extension to n_0 . By fact (3), $(P_{b,1}, P_{r,0})$ also does not have a BLUE extension to n_0 . It follows, by Lemma 4.4, that we can add n_0 by a switch, and the switch must be a strong switch.

The switch cannot be a strong RED switch by fact (1). Also, the switch cannot be a strong BLUE switch because if we performed a strong BLUE switch on $(P_{b,1}, P_{r,0})$, it would create a strong path extension in which we decreased the length of the RED path. This contradicts fact (2), recalling that by Lemma 4.11, any strong path extension of $(P_{b,1}, P_{r,0})$ must also witness that we cannot always RED switch. \square

Claim 5.5. *For every $n \in \mathbb{N}$, and for every RED extension $(P_{b,1}, P_r)$ of $(P_{b,1}, P_{r,0})$, if n is not on either $P_{b,1}$ or P_r , then there is a RED path extension of $(P_{b,1}, P_r)$ to n .*

Proof. Let $(P_{b,1}, P_r)$ be a RED extension of $(P_{b,1}, P_{r,0})$, and let n_1 be an element of \mathbb{N} that is on neither $P_{b,1}$ nor P_r . Assume there is no RED extension of $(P_{b,1}, P_r)$ to n . By Claim 5.4, there is also no BLUE extension of $(P_{b,1}, P_r)$ to n , because extending the RED path cannot make it any easier to find a BLUE extension. Then, by Lemma 4.4, to add n_1 to $(P_{b,1}, P_r)$, we must do a switch, and the switch must be strong.

We split our proof into two cases:

If $(P_{b,1}, P_r) = (P_{b,1}, P_{r,0})$, then the proof proceeds exactly as the proof of Claim 5.4. The switch cannot be a strong RED switch by fact (1), and the switch cannot be strong BLUE switch by fact (2).

If $(P_{b,1}, P_r) \neq (P_{b,1}, P_{r,0})$, then the switch still cannot be a strong RED switch by fact (1), recalling again that by Lemma 4.11, any strong path extension of $(P_{b,1}, P_{r,0})$ must also witness that we cannot always RED switch. The switch also cannot be a BLUE switch, because for the switch to be a BLUE switch, the edge from the end of the $P_{b,1}$ to the end of P_r must be BLUE, contradicting Claim 5.4. \square

This completes of Theorem 5.1. \square

Note that if we are in Case 2 of the above construction, we have no way of knowing whether we will find the x we are looking for, so we cannot know when to switch to the construction for Case 1. This does not concern us. We are only proving that there is a computable path decomposition with one path finite, not that the path decomposition can be found uniformly.

5.2. No Uniform Δ_2^0 index.

Theorem 5.6. *There is no partial computable function f such that if e is an index for a computable coloring c , then $f(e)$ is an index for a Δ_2^0 path decomposition for c .*

Proof. Let f be a partial computable function. We create a computable coloring with index e such that if $f(e)$ is defined, then $f(e)$ is not an index for a Δ_2^0 path decomposition for c .

Our construction is in stages. During stage s of the construction, for every $t \leq s$, we color the pair $\{t, s+1\}$.

By the recursion theorem, we may know the index, e , for the coloring that we are constructing. We begin computing $f(e)$, and while we wait for it to halt, we color everything BLUE with everything else.

If $f(e)$ halts, then by the limit lemma, we begin approximating the Δ_2^0 paths P_r and P_b computed by the $f(e)$ th Δ_2^0 index. We let $P_{r,s}$ and $P_{b,s}$ be the stage s approximations to P_r and P_b .

The only facts we will need about the approximations is that if P_b is defined and empty, then there is some s such that for all $t \geq s$, $P_{b,t}$ is empty and likewise that if P_b is defined and nonempty, then there is some s such that for all $t \geq s$, $P_{b,t}$ is nonempty and the first element of $P_{b,t}$ is the first element of P_b .

We define x_s to be the first element of $P_{b,s}$ (with x_s undefined if $P_{b,s}$ is empty). We may assume that $x_s \leq s$.

At stage s if $P_{b,s}$ is empty, we color everything BLUE with $s+1$. On the other hand, if $P_{b,s}$ is nonempty, then let s_0 be minimal such that for all $t \in [s_0, s]$, $x_t = x_s$. Then, if $t \leq s_0$, we color $\{t, s+1\}$ RED, and if $s_0 < t < s+1$, we color $\{t, s+1\}$ BLUE.

This completes the construction. We now verify that if $f(e)$ is defined and if the Δ_2^0 set with index $f(e)$ is defined, then it is not a path decomposition for c .

Assume $f(e)$ and the Δ_2^0 set with index $f(e)$ are defined. We have two cases. Either P_b is empty or it is not empty.

If P_b is empty, then there is some s_0 such that for all $s \geq s_0$, $P_{b,s}$ is empty. But then, after stage s_0 , we only create BLUE edges, so at most finitely many edges are RED. Thus P_r can only be a finite length, and P_b is empty, so P_r and P_b do not form a path decomposition for c .

If P_b is not empty, then it has some smallest element x , and there is some smallest s_0 such that for all $s \geq s_0$, $x_s = x$. Note that $s_0 \geq x_{s_0} = x$. But then no number $\leq s_0$ is ever colored BLUE with any number greater than s_0 , so in particular if x is in the BLUE path, then the BLUE path cannot go through any numbers larger than s_0 , and must be finite. However, every number larger than s_0 is colored

BLUE with every other number larger than s_0 , so for every $t > s_0$ on the RED path, the next element on the RED path must be $\leq s_0$. Thus, the RED path can have length at most $2s_0$, and is also finite. Again, P_r and P_b do not form a path decomposition for c . \square

We remark now that the technique of Theorem 5.6 only allows us to build a coloring c that can defeat any single uniform manner of attempting to produce a Δ_2^0 path decomposition from c . For the coloring c that we create, it is not at all difficult to produce a path decomposition; it is just the case that the $f(e)$ th Δ_2^0 set fails to do so.

5.3. No finite set of Δ_2^0 indices. We show now, by a strengthening of the argument from Theorem 5.6 that it is impossible to reduce the nonuniformity to a finite collection of Δ_2^0 indices.

Theorem 5.7. *There is no partial computable function f such that if e is an index for a computable coloring c , then $f(e)$ is an index for a finite c.e. set $W_{f(e)}$ one of whose elements is an index for a Δ_2^0 path decomposition for c .*

Proof. Let f be a partial computable function. We create a computable coloring with index e such that if $f(e)$ is defined, and if $W_{f(e)}$ is finite, then no element of $W_{f(e)}$ is an index for a Δ_2^0 path decomposition for c .

Our construction is in stages. During stage s of the construction, for every $t \leq s$, we color the pair $\{t, s + 1\}$.

By the recursion theorem, we may know the index, e , for the coloring that we are constructing. We begin computing $f(e)$, and while we wait for it to halt, we color everything BLUE with everything else.

If $f(e)$ halts, then we begin enumerating $W_{f(e)}$. Let $W_{f(e),s}$ be the stage s approximation to $W_{f(e)}$.

For each i in $W_{f(e)}$, let $P_{b,i,s}$ and $P_{r,i,s}$ be the stage s approximations to the BLUE and RED paths of the i th Δ_2^0 approximation to a path decomposition. We may assume $P_{b,i,s}$ and $P_{r,i,s}$ have no elements larger than s . Let $P_{b,i}$ and $P_{r,i}$ be the limits, if they exist, of $P_{b,i,s}$ and $P_{r,i,s}$.

We will use x as a variable for a color (BLUE or RED), as well as for a letter for a color (b or r). We will write $1 - x$ to refer to the other color.

For each $i \in W_{f(e)}$, we will have two strategies $S_{b,i}$ and $S_{r,i}$, which will work to ensure that if $(P_{b,i}, P_{r,i})$ forms a path decomposition for c , then $P_{b,i}$ (respectively $P_{r,i}$) is finite. Note that if both $S_{b,i}$ and $S_{r,i}$ achieve their goals, then $(P_{b,i}, P_{r,i})$ cannot form a path decomposition for c .

At stage s , if every $P_{x,i,s}$ with $i \in W_{f(e)}$ is empty, then we continue to color everything BLUE.

If at least one $P_{x,i,s}$ is nonempty, and if there are no currently active strategies, then we will *activate* one $S_{x,i}$ such that $P_{x,i}$ has been stably nonempty for the longest time. More precisely, for each (x,i) such that $i \in W_{f(e),s}$, and $P_{x,i,s}$ is nonempty, let $t_{x,i}$ be minimal such that $\forall t \in [t_{x,i}, s]$, $P_{x,i,t}$ is nonempty, with first element equal to the first element of $P_{x,i,s}$. Let $t_0 = \min_{(x,i)}(t_{x,i})$, and let (x_0, i_0) be minimal among all (x_i) such that $t_{x,i} = t_0$. We activate strategy S_{x_0, i_0} .

When this happens, S_{x_0, i_0} *claims* the set $[0, s]$, declaring that $\forall t \in [0, s] \forall s' > s$, the color of $\{t, s'\}$ will be $1 - x_0$. This restraint will remain in effect either until the end of the construction, or until some later stage when the smallest element of P_{x_0, i_0} changes.

If at some point the smallest element of P_{x_0, i_0} changes, we *injure* every active strategy in the construction: Every active strategy forgets its claim set and deactivates. After that, we will probably be back in the case where there are no active strategies, and we are choosing a new strategy to activate.

Note that if S_{x_0, i_0} is never injured, then it will have achieved its goal: P_{x_0, i_0} has an element in $[0, s]$, and there are no numbers in $[0, s]$ that are colored x_0 with any numbers $> s$, so P_{x_0, i_0} will never be able to exit the interval $[0, s]$.

Now, assume that at stage s , there are finitely many active strategies $S_{x_0, i_0}, S_{x_1, i_1}, \dots, S_{x_k, i_k}$ with claim sets $[0, s_0], (s_0, s_1], \dots, (s_{k-1}, s_k]$.

We say that an inactive strategy $S_{x,i}$ is eligible to act if

- $i \in W_{f(e),s}$,
- The paths $P_{b,i,s}$ and $P_{r,i,s}$ include every element of $[0, s_k]$, and
- The last element of $P_{x,i,s}$ is larger than s_k .

If any strategy is eligible to act, then we will choose one to activate in the following manner: For each (x,i) such that $S_{x,i}$ is eligible to act, let $\hat{P}_{x,i,s}$ be the shortest initial segment of $P_{x,i,s}$ that includes every element of $[0, s_k]$ that is in $P_{x,i,s}$, and that ends with an element not in $[0, s_k]$. Let $\hat{\hat{P}}_{1-x,i,s}$ be the shortest initial segment of $P_{1-x,i,s}$ that includes every element of $[0, s_k]$ that is in $P_{1-x,i,s}$. (The double hat on $\hat{\hat{P}}_{1-x,i,s}$ is used because $\hat{P}_{1-x,i,s}$ may already be defined if $(1-x, i)$ also satisfies the properties above. Note that if both are defined, then $\hat{P}_{1-x,i,s}$ is $\hat{\hat{P}}_{1-x,i,s}$ with one extra element at the end.)

Let $t_{x,i}$ be minimal such that $\forall t \in [t_{x,i}, s]$, $\hat{P}_{x,i,s}$ is an initial segment of $P_{x,i,t}$ and $\hat{\hat{P}}_{1-x,i,s}$ is an initial segment of $P_{1-x,i,t}$. We think of $t_{x,i}$ as measuring how long $S_{x,i}$'s reason for acting has been valid.

Let $t_{k+1} = \min_{(x,i)}(t_{x,i})$, and let (y, j) be minimal among all (x, i) such that $t_{x,i} = t_{k+1}$. We activate strategy $S_{y,j}$. So the strategy that is activated is the one whose reason for acting has been valid the longest.

When this happens, $S_{y,j}$ claims the set $(s_k, s]$, declaring that $\forall t \in (s_k, s] \forall s' > s$, the color of $\{t, s'\}$ will be $1 - y$. This restraint will remain in effect either until the end of the construction, or until some later stage when $S_{y,j}$ becomes injured.

If at some stage $s' > s$, we see that $\hat{P}_{y,j,s}$ is not an initial segment of $P_{y,j,s'}$ or $\hat{P}_{1-y,j,s}$ is not an initial segment of $P_{1-y,j,s'}$, then at that stage, injure $S_{y,j}$ as well as every strategy that became active after stage s . The injured strategies all forget their claim sets and deactivate. If this happens, we say that the reason $S_{y,j}$ activated is no longer valid.

Note again that if $S_{y,j}$ is never injured, then it will have achieved its goal as follows. The last element of $\hat{P}_{y,j,s}$ is in $(s_k, s]$. Every element of $(s_k, s]$ is colored $1 - y$ with every element larger than s , so $P_{y,j}$ cannot exit $(s_k, s]$ by moving to a number larger than s . Finally, every number $\leq s_k$ is already in either $\hat{P}_{y,j,s}$ or $\hat{P}_{1-y,j,s}$. These are initial segments of $P_{y,j}, P_{1-y,j}$ (because otherwise $S_{y,j}$ would be injured) and so $P_{y,j}$ cannot exit $(s_k, s]$ by moving to a number $\leq s_k$, because all of those numbers are already on one of the two paths. Thus $P_{y,j}$ is unable to have any elements $> s$.

At the end of stage s , after determining which, if any, strategy becomes activated at stage s , for every $t \leq s$, we color $\{t, s + 1\}$ by the following algorithm. If t is in any active strategy's claim set, then we color $\{t, s + 1\}$ according to that strategy's restriction. (Note that the claim sets of our active strategies are always disjoint.) If t is not claimed by any strategy, then color $\{t, s + 1\}$ BLUE.

This completes the construction of c . We now verify that if $W_{f(e)}$ is finite, then for every $i \in W_{f(e)}$, $(P_{b,i}, P_{r,i})$ is not a path decomposition for c . The verification will consist of two claims. Claim 5.8 was already verified during the construction.

Claim 5.8. *For every (x, i) , if at some stage s , $S_{x,i}$ becomes active, and is never injured after stage s , and if $(P_{b,i}, P_{r,i})$ forms a path decomposition for c , then $P_{x,i}$ is finite.*

Claim 5.9. *If $W_{f(e)}$ is finite, then for every (x, i) , if $i \in W_{f(e)}$, $(P_{b,i}, P_{r,i})$ forms a path decomposition for c , and $P_{x,i}$ is infinite, then there is an s such that $S_{x,i}$ becomes active at stage s , and is never injured after stage s .*

Proof. Fix some pair (y, j) such that $j \in W_{f(e)}$, $(P_{b,j}, P_{r,j})$ forms a path decomposition for c , and $P_{y,j}$ is infinite.

Let $S_{x_0, i_0}, S_{x_1, i_1}, \dots, S_{x_k, i_k}$ be the strategies which at some point activate and never get injured again. If $S_{y, j}$ is one of these strategies, then we are done, so assume it is not.

For each S_{x_n, i_n} , let s_n be the stage when S_{x_n, i_n} is activated for the last time, never to be injured again. Reordering the S_{x_n, i_n} so that the s_n are in increasing order, we may assume that at stage s_k , the strategies $S_{x_0, i_0}, S_{x_1, i_1}, \dots, S_{x_k, i_k}$ are all active, with claim sets $[0, s_0], (s_0, s_1], \dots, (s_{k-1}, s_k]$, that none of the S_{x_n, i_n} will ever be injured after stage s_k , and that no other strategies are active at stage s_k .

(They are all active and will never be injured again because s_k is the max of the s_n . There are no other strategies active because S_{x_k, i_k} just became activated at stage s_k , and, by definition of the S_{x_n, i_n} , if any other strategy was active, it would get injured again later, but then this would injure S_{x_k, i_k} , contradicting the choice of s_k .)

For each x , let $\hat{P}_{x, j}$ be the shortest initial segment of $P_{x, j}$ that includes every element of $[0, s]$ that is on $P_{x, j}$. By assumption, $(P_{b, j}, P_{r, j})$ forms a path decomposition for c . Therefore, in particular, $\hat{P}_{b, j}, \hat{P}_{r, j}$ together cover all the elements of $[0, s_k]$. Also by assumption, $P_{y, j}$ is infinite, so let $\hat{P}_{y, j}$ be the initial segment of $P_{y, j}$ whose length is one greater than the length of $\hat{P}_{y, j}$.

Let t_0 be minimal such that $t_0 > s_k$, $j \in W_{f(e), t_0}$, and for all $t > t_0$, $\hat{P}_{y, j}$ is an initial segment of $P_{y, j, t}$, and $\hat{P}_{1-y, j}$ is an initial segment of $P_{1-y, j, t}$. We claim that there will be some stage $s \geq t_0$ so that $S_{x_0, i_0}, S_{x_1, i_1}, \dots, S_{x_k, i_k}$ are the only strategies active at the beginning of stage s , and $S_{y, j}$ becomes active at stage s . We also claim that after that stage s , $S_{y, j}$ will never be injured.

Claim 5.10. *There is some $s \geq t_0$ so that $S_{x_0, i_0}, S_{x_1, i_1}, \dots, S_{x_k, i_k}$ are the only strategies active at the beginning of stage s , and $S_{y, j}$ becomes active at stage s .*

Proof. The proof of Claim 5.10 hinges on the following.

Claim 5.11. *For every pair (x, i) , if $(x, i) \neq (y, j)$, then there can be at most one stage $s \geq t_0$ such that $S_{x_0, i_0}, S_{x_1, i_1}, \dots, S_{x_k, i_k}$ are the only strategies active at the beginning of stage s , and $S_{x, i}$ becomes active at stage s .*

Proof. To see this, assume that $s \geq t_0$, $S_{x_0, i_0}, S_{x_1, i_1}, \dots, S_{x_k, i_k}$ are the only strategies active at the beginning of stage s , and $S_{x, i}$ becomes active at stage s . Assume further that $S_{x, i}$ is injured at stage $s' > s$. Let $s'' \geq s'$ be a stage at which $S_{x_0, i_0}, S_{x_1, i_1}, \dots, S_{x_k, i_k}$ are the only active strategies.

Then, at stage s'' , even if $S_{x,i}$ is eligible to act, $t_{x,i}$ must be $\geq s'$. This is because t_0 is larger than the stage at which any of the $S_{x_0,i_0}, S_{x_1,i_1}, \dots, S_{x_k,i_k}$ is ever injured, and so, at stage s' , when $S_{x,i}$ was injured, it must have been injured because its reason for acting was no longer valid. Thus, $t_{x,i} \geq s'$, but, by choice of t_0 , we know that $S_{y,j}$ is eligible to act and $t_{y,j} \leq t_0$. Thus, it is not possible for $S_{x,i}$ to become active at stage s'' , because $S_{y,j}$ is ahead of it in queue. \square

Claim 5.10 now follows directly, because, by definition of $S_{x_0,i_0}, S_{x_1,i_1}, \dots, S_{x_k,i_k}$, we know that every $S_{x,i}$ that becomes active after stage s_k must become injured at a later stage. We have that $W_{f(e)}$ is finite, and so by Claim 5.11 there can be at most finitely many stages after t_0 when $S_{x_0,i_0}, S_{x_1,i_1}, \dots, S_{x_k,i_k}$ are the only active strategies, and some strategy other than $S_{y,j}$ becomes active. Each time such a strategy becomes active at such a time, and later becomes injured, we have another stage at which $S_{x_0,i_0}, S_{x_1,i_1}, \dots, S_{x_k,i_k}$ are the only active strategies. By choice of t_0 , at each such stage, $S_{y,j}$ is eligible to act, so once no other strategy is able to become active, we must have that $S_{y,j}$ becomes active at a stage at which $S_{x_0,i_0}, S_{x_1,i_1}, \dots, S_{x_k,i_k}$ are the only active strategies. \square

We conclude the proof of Theorem 5.7 with a proof of the following.

Claim 5.12. *Assume $S_{y,j}$ becomes active at a stage $s > t_0$ such that $S_{x_0,i_0}, S_{x_1,i_1}, \dots, S_{x_k,i_k}$ are the only strategies active at the beginning of stage s . Then $S_{y,j}$ is never injured after stage s .*

Proof. None of the $S_{x_0,i_0}, S_{x_1,i_1}, \dots, S_{x_k,i_k}$ are ever injured after stage s_k , and t_0 was chosen to be large enough that if $S_{y,j}$ becomes active at stage $s \geq t_0$, and if the previously claimed elements at stage s are $[0, s_k]$, then $S_{y,j}$'s reason for acting will remain valid forever, and so $S_{y,j}$ will never injure itself.

$S_{y,j}$ can only be injured by itself or by a strategy that was active when $S_{y,j}$ became active. It follows that $S_{y,j}$ will not be injured after stage s . \square

This completes the proof of Claim 5.9, as we have shown that there must be a stage so that $S_{y,j}$ becomes active at that stage and is never injured after that stage. \square

Theorem 5.7 now follows from Claims 5.8 and 5.9, because \mathbb{N} is infinite, so any path decomposition of \mathbb{N} must have an infinite path. Considered together, Claims 5.8 and 5.9 say that if $W_{f(e)}$ is finite, and if $i \in W_{f(e)}$, then either $(P_{b,i}, P_{r,i})$ do not form a path decomposition for c , or both $P_{b,i}$ and $P_{r,i}$ are finite (in which case they cannot form a path decomposition for c). \square

6. RPD COMPARED TO RAMSEY'S THEOREM FOR PAIRS

One fact about (infinite) Ramsey's Theorem that is regularly used is that for every coloring $c : [\mathbb{N}]^2 \rightarrow r$ and every infinite set X , there is an infinite homogeneous set $H \subseteq X$. However, a path decomposition of a set $X \subseteq \mathbb{N}$ for the restricted coloring $c : [X]^2 \rightarrow r$ does not help us to find a path decomposition for the unrestricted coloring $c : [\mathbb{N}]^2 \rightarrow r$.

There is a proof which uses compactness to show the infinite version of Ramsey theorem implies the finite version. For example, see Graham et al. [4]. By Theorem 4.5, we know this compactness argument fails for the Rado Path Decomposition Theorem. A compactness argument breaks down since the paths linking numbers below m might also involve numbers larger than m .

7. COROLLARIES IN MATHEMATICAL LOGIC

The existence of a non-principal ultrafilter on the natural numbers is a strong assumption that unfortunately cannot be shown in Zermelo Fraenkel set theory, see Feferman [2]; the axiom of choice is sufficient see Jech [6]. By independent results of Towsner [12], Enayat [1], and Kreuzer [8], the ultrafilter proof of Rado Path Decomposition implies that for every r -coloring c of $[\mathbb{N}]^2$ there is a path decomposition arithmetical in c , and as a statement of second order arithmetic the Rado Path Decomposition Theorem holds in ACA_0 .

A careful analysis of the cohesive proof shows that a path decomposition can always be found in the jump of the cohesive set C' . The key issue is that exactly one $N(m, j)$ is large (with respect to our cohesive set C). It is Δ_2^C (but not computable in C) to determine which one.

Jockusch and Stephan [7] have showed that \mathbf{d} is PA over $0'$ if and only if there is a cohesive set C such that $C' \leq_T \mathbf{d}$. This implies that there is always a path decomposition computable in c'' , equivalently there is a Δ_3^c path decomposition. It also implies there is a path decomposition whose jump is c'' .

For 2-colorings, Theorem 4.2 shows this bound can be improved to Δ_2^c . For stable colorings the bound can also be improved to Δ_2^c . (Use the stable proof of RPD and note that determining m 's color is Δ_2^c .)

Theorem 3.1 shows that we cannot expect to do better than Δ_2^c . So for stable and 2-coloring the bound of Δ_2^c is tight.

For more than two colors, we do not have an exact calibration of the effectivity of path decomposition.

Question 7.1. *Does every 3-coloring c have a Δ_2^c path decomposition?*

Question 7.2. *Is there an unstable 3-coloring c such that every path decomposition is PA over \mathcal{O}' ?*

Question 7.3. *Does increasing the number of colors past 3 have any effect on the above two questions?*

Theorem 3.1 shows as a statement of second order arithmetic the Rado Path Decomposition Theorem implies ACA_0 .

One might wonder why we cannot use the generic construction to answer Question 7.2 by building a path decomposition that avoids the cone of degrees above \mathcal{O}' . The problem is that forcing Σ_1^G statements (like does $\Phi^G(w)\downarrow$) is Σ_2^X not Σ_1^X . The ends of finite paths P_j must have color j and determining this is not Σ_1^X .

REFERENCES

- [1] Ali Enayat. From bounded arithmetic to second order arithmetic via automorphisms. In *Logic in Tehran*, volume 26 of *Lect. Notes Log.*, pages 87–113. Assoc. Symbol. Logic, La Jolla, CA, 2006.
- [2] S. Feferman. Some applications of the notions of forcing and generic sets. *Fund. Math.*, 56:325–345, 1964/1965. ISSN 0016-2736.
- [3] L. Gerencsér and A. Gyárfás. On Ramsey-type problems. *Ann. Univ. Sci. Budapest. Eötvös Sect. Math.*, 10:167–170, 1967. ISSN 0524-9007.
- [4] Ronald L. Graham, Bruce L. Rothschild, and Joel H. Spencer. *Ramsey theory*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., New York, second edition, 1990. ISBN 0-471-50046-1. A Wiley-Interscience Publication.
- [5] Denis R. Hirschfeldt. *Slicing the truth*, volume 28 of *Lecture Notes Series. Institute for Mathematical Sciences. National University of Singapore*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2015. ISBN 978-981-4612-61-6. On the computable and reverse mathematics of combinatorial principles, Edited and with a foreword by Chitao Chong, Qi Feng, Theodore A. Slaman, W. Hugh Woodin and Yue Yang.
- [6] Thomas Jech. *Set theory*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2003. ISBN 3-540-44085-2. The third millennium edition, revised and expanded.
- [7] Carl G. Jockusch, Jr. and Frank Stephan. A cohesive set which is not high. *Math. Log. Quart.*, 39:515–530, 1993.
- [8] Alexander P. Kreuzer. Non-principal ultrafilters, program extraction and higher-order reverse mathematics. *J. Math.*

- Log.*, 12(1):1250002, 16, 2012. ISSN 0219-0613. doi: 10.1142/S021906131250002X. URL <http://dx.doi.org/10.1142/S021906131250002X>.
- [9] Alexey Pokrovskiy. Partitioning edge-coloured complete graphs into monochromatic cycles and paths. *J. Combin. Theory Ser. B*, 106:70–97, 2014. ISSN 0095-8956. doi: 10.1016/j.jctb.2014.01.003. URL <http://dx.doi.org/10.1016/j.jctb.2014.01.003>.
- [10] Richard Rado. Monochromatic paths in graphs. *Ann. Discrete Math.*, 3:191–194, 1978. Advances in graph theory (Cambridge Combinatorial Conf., Trinity College, Cambridge, 1977).
- [11] Daniel T Soukup. Decompositions of edge-coloured infinite complete graphs into monochromatic paths II. *arXiv.org*, July 2015.
- [12] Henry Towsner. Ultrafilters in reverse mathematics. *J. Math. Log.*, 14(1):1450001, 11, 2014. ISSN 0219-0613. doi: 10.1142/S0219061314500019. URL <http://dx.doi.org/10.1142/S0219061314500019>.
- [13] Rebecca Weber. *Computability theory*, volume 62 of *Student Mathematical Library*. American Mathematical Society, Providence, RI, 2012. ISBN 978-0-8218-7392-2. doi: 10.1090/stml/062. URL <http://dx.doi.org/10.1090/stml/062>.

(Cholak) DEPARTMENT OF MATHEMATICS, UNIVERSITY OF NOTRE DAME, NOTRE DAME, IN 46556-5683

E-mail address, Cholak: Peter.Cholak.1@nd.edu

URL, Cholak: <http://www.nd.edu/~cholak>

(Igusa) SCHOOL OF MATHEMATICS AND STATISTICS, VICTORIA UNIVERSITY OF WELLINGTON, WELLINGTON 6140, NEW ZEALAND

E-mail address, Igusa: greg.igusa@vuw.ac.nz

URL, Igusa: <http://homepages.ecs.vuw.ac.nz/~igusagr/index.shtml>

(Patey) DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA, BERKELEY, CA 94720-3840

E-mail address, Patey: ludovic.patey@computability.fr

URL, Patey: <http://www.ludovicpatey.com/>

(Soskova) DEPARTMENT OF MATHEMATICS AND INFORMATICS, SOFIA UNIVERSITY, 5 JAMES BOURCHIER BLVD, SOFIA 1164

E-mail address, Soskova: msoskova@fmi.uni-sofia.bg

URL, Soskova: <https://www.fmi.uni-sofia.bg/fmi/logic/msoskova/>

(Turetsky) SCHOOL OF MATHEMATICS AND STATISTICS, VICTORIA UNIVERSITY OF WELLINGTON, WELLINGTON, NEW ZEALAND

E-mail address, Turetsky: dan.turetsky@vuw.ac.nz

URL, Turetsky: <http://homepages.ecs.vuw.ac.nz/~dan/>