Effective Algebra and Effective Dimension

By

Daniel Turetsky

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

(Mathematics)

at the

UNIVERSITY OF WISCONSIN – MADISON

2010

Abstract

Effective Dimension is a notion introduced by Lutz, which measures the density of information in an infinite sequence. Lutz asks how this concept interacts with classical topological notions. In Chapter two, I present several results concerning this.

Effective Algebra is the study of computable and relatively computable structures and the relations on them. In Chapter three, I present several results separating notions of computable categoricity. In Chapter four, I review limitwise monotonic functions and prove several new results about them. In Chapter five, I construct computable linear orders on which various natural relations are intrinsically complete.

Acknowledgements

I would like to thank my thesis advisor, Steffen Lempp, for his advice and guidance, and most of all his patience in the face of my colossal lack of organization.

I am also grateful to all the logic faculty at UW-Madison, with special mention to Joseph Miller. Thanks also to Rod Downey and Noam Greenberg for a very productive semester in New Zealand.

Thanks to Nick for being tall, and to all the graduate students and former graduate students in the math department who listened to me talk about math and made my time there enjoyable, especially Asher, Diane, Matt, Nick and Zajj.

Finally, thanks to Mom, Dad and Emma for the support through the years.

Contents

Chapter 1

Introduction

This work consists of several distinct pieces. The two main areas of my research in the general area of computability theory are algorithmic randomness, and computable algebra and model theory. Randomness is a notion which is captured in several different ways. An infinite sequence being random can be defined to mean that it contains large amounts of information, or that it is hard to predict the behavior of, or that it exhibits no atypical properties, and all are equivalent definitions. Effective dimension is a measurement of the extent to which a real is partially random. This can be defined by the real containing a smaller amount of information, or the real being only somewhat predictable, and these again yield equivalent definitions.

Computability theory seeks to understand the effective content of mathematics. Although many mathematical objects exist, computability theory asks the question of whether they can be algorithmically found. For example, although every vector space has a basis, computability theory tells us that finding a basis is not always algorithmically possible. Computable algebra is the analysis of classical mathematical structures such as rings, graphs or linear orders using the tools of computability theory, while computable model theory generalizes this to arbitrary mathematical structures.

1.1 Effective Dimension

Very broadly, Lebesgue measure separates the world into the sets of positive measure and those of measure 0. This classification is rather coarse, however. In $\mathbb{R}^2$, for example, points and lines are indistinguishable by Lebesgue measure, as they all have measure 0. Notions of dimension, such as Hausdorff dimension or packing dimension, can strengthen this classification by separating certain sets of measure 0. Points have Hausdorff dimension 0, while lines have Hausdorff dimension 1.

Similarly, randomness separates sequences into those which are random and those which are not. Effective dimension refines this classification by separating certain non-random sequences. While the classical dimension of a singleton is always 0, the effective dimension of a singleton can be nonzero, so effective dimension often studies singletons (i.e., points).

The study of effective dimension began when Lutz in [12] proved an alternate characterization of the classical notion of Hausdorff dimension. Athreya, Hitchcock, Lutz and Mayordomo extended this to packing dimension in [3]. These alternate characterizations were effectivized, giving rise to the concept of effective Hausdorff dimension and effective packing dimension. Mayordomo in [14] and Athreya, Hitchcock, Lutz and Mayordomo in [3] then showed the equivalence of Definition 1.1 with Lutz's original definition in [12].

Definition 1.1. For an infinite sequence x, define the *effective Hausdorff dimension* of x as

$$\dim_H(x) := \liminf_{s\to\infty} \frac{K(x \upharpoonright s)}{s}.$$

Notice that this value is necessarily at most 1. Effective dimension thus measures when a sequence is partially random, with random sequences having dimension 1 (although the converse fails).

Analogous to defining the effective dimension of an infinite sequence, one can define the effective Hausdorff dimension of a point in $\mathbb{R}^n$. This is no longer a value less than 1, but instead a value less than n. There are several equivalent ways of doing this, one being to simply consider the binary expansions of the coordinates.

One can then ask how frequent points of any given dimension are. Several easy facts follow:

- For $z \in \mathbb{R}^n$, $\dim_H(z) \in [0, n]$.
- For every $\alpha \in [0, n]$, there are densely many $z \in \mathbb{R}^n$ with $\dim_H(z) = \alpha$.
- The set of $z \in \mathbb{R}^n$ with $\dim_H(z) < n$ has Lebesgue measure 0.
- The set of $z \in \mathbb{R}^n$ with $\dim_H(z) > 0$ is meager.

A consequence of Lutz's work in [12] is that if X has classical Hausdorff dimension β, it contains infinitely many points of effective Hausdorff dimension greater than $\beta - \epsilon$ for any positive ϵ. In [13], Lutz and Weihrauch then asked how effective Hausdorff dimension interacts with various connectivity properties, and proved several results along these lines. In Chapter 2, I prove several further results along these lines.

1.2 Computable Categoricity

Computable categoricity is an effective version of categoricity from model theory. For a cardinal κ, a system of axioms is said to be κ-categorical if every structure of cardinality κ which satisfies those axioms is isomorphic. In making this notion effective, we restrict our attention to a computable model and computable isomorphisms between computable copies of it. Rather than considering all structures which satisfy a given axiom system, we consider only those which are isomorphic (but not necessarily computably so).

Definition 1.2. A computable structure $\mathfrak{A}$ is *computably categorical* if for any other computable structure $\mathfrak{B}$ with $\mathfrak{A} \cong \mathfrak{B}$, there exists a total computable function f with $f : \mathfrak{A} \cong \mathfrak{B}$.

For example, any linear ordering which contains no adjacencies is computably categorical; given two computable copies, one can perform a back-and-forth construction to create an isomorphism. A linear order with only finitely many adjacencies is also computably categorical, because one could begin by correctly mapping the finitely many points, then run the back-and-forth construction. In fact, this completely characterizes the computably categorical linear orders; any computable linear order with infinitely many adjacencies (e.g., the integers as a linear order) has two computable copies between which there is no computable isomorphism.

One might expect that every computably categorical structure is such because one can run a back-and-forth construction to create the isomorphism, but this turns out to correspond to a stronger notion.

Definition 1.3. A computable structure $\mathfrak{A}$ is *relatively computably categorical* if for any other structure $\mathfrak{B}$ (not necessarily computable) with $\mathfrak{A} \cong \mathfrak{B}$, there exists a total function f computable from (the open diagram of) $\mathfrak{B}$ with $f : \mathfrak{A} \cong \mathfrak{B}$.

There are analogs of both computable categoricity and relative computable categoricity for Δ_n. In Chapter 3, I introduce a new categoricity notion, relatively computable categorical above a degree, and separate it from the existing notions.

1.3 Limitwise Monotonic Functions

Limitwise monotonic functions are an important tool for studying computable structures, both because they allow classification other than via the arithmetic hierarchy, and because they often capture the fact that computable structures grow one element at a time.

In [9], the author collaborated with Kach to investigate limitwise monotonic functions on a computably ordered domain, with the hope of classifying those sets with a strong η-representation. Although the investigation proved quite fruitful, no such classification was obtained. We show here that limitwise monotonic functions are the wrong tool for such a classification.

We also prove the existence of a *totally limitwise monotonic degree*, a result later improved by the author with Downey and Kach in [4].

1.4 Degree Spectrum of Relations

For a relation R on a computable structure $\mathfrak{A}$, the *degree spectrum* of R is the collection

$$\{\deg(S) : \exists \mathfrak{B} \text{ computable}, (\mathfrak{B}, S) \cong (\mathfrak{A}, R)\}.$$

Clearly if R is definable by some Σ^c_α (Π^c_α) formula in the language of A, then $\deg(S)$ will consist entirely of Σ^0_α (Π^0_α) degrees.

One often considers the degree spectrum of certain natural relations on a class of structures. For instance:

Theorem 1.4 (Downey, Lempp, Wu [5]). *If L is a computable linear order with infinitely many adjacencies, the degree spectrum of the successivity relation on L is upwards closed in the c.e. degrees.*

In Chapter 5, we construct a collection of natural relations and study their spectrum.

Chapter 2

Connectedness of Dimension Level Sets

2.1 Introduction and Results

In [13], Lutz and Weihrauch investigate sets in $\mathbb{R}^n$ defined by the effective Hausdorff dimensions of their elements. They show the following:

Theorem 2.1. *In $\mathbb{R}^n$, the set of points of dimension strictly less than 1 is totally disconnected, as is the set of points of dimension strictly greater than $n-1$.*

Theorem 2.2. *In $\mathbb{R}^n$, the set of points of dimension less than or equal to 1 is path-connected, as is the set of points of dimension greater than or equal to $n-1$.*

Restricting these results to the simplest case of $n = 2$ suggests that the points with effective Hausdorff dimension 1 are somehow topologically numerous. We investigate the properties of the dimension one points further, proving the following results:

Theorem 2.3. *In $\mathbb{R}^n$ $(n \geq 2)$, the set of points of dimension exactly 1 is connected.*

Theorem 2.4. *In $\mathbb{R}^2$, the set of points of dimension not 1 is not path-connected.*

In Section 2.2, we review the appropriate notions. In Section 2.3, we prove the following result about the abundance of points of dimension 1, from which the above two results follow.

Theorem 2.5. *If $Z \subseteq \mathbb{R}^n$ $(n \geq 2)$ is closed, connected, and has the property that for any open set U with $Z \cap U \neq \emptyset$, $\operatorname{ind}(Z \cap U) \geq n-1$, then Z contains a point of effective Hausdorff dimension 1.*

Note that by fixing $r_0, r_1 \in \mathbb{R}$ relatively random, one can define

$$F = \{(x_0, x_1, \dots, x_n) \in \mathbb{R}^n \mid x_0 = r_0,\ x_1 = r_1\}.$$

Then F is a closed set of dimension $n-2$ with no point of effective Hausdorff dimension less than 2. So in one sense, Theorem 2.5 is optimal (i.e., $\operatorname{ind}(Z \cap U) \geq n-1$ is needed).

2.2 Semi-measures, Complexity and Dimension

Throughout the rest of the chapter, let n be a fixed positive integer greater than one.

Convention 2.6. *ε denotes the empty string in $2^{<\omega}$.*

λ denotes Lebesgue measure on $\mathbb{R}$.

$\pi_i : \mathbb{R}^n \to \mathbb{R}$ denotes projection onto the ith coordinate.

Definition 2.7. We call a function $\mu : (2^{<\omega})^n \to \mathbb{R}_{\geq 0}$ a *semi-measure* if

$$\begin{aligned} \mu(\varepsilon, \varepsilon, \dots, \varepsilon) &\leq 1 \\ \mu(\sigma_0, \dots, \sigma_i, \dots, \sigma_{n-1}) &\geq \mu(\sigma_0, \dots, \sigma_i{}^\frown 0, \dots, \sigma_{n-1}) \\ &\quad + \mu(\sigma_0, \dots, \sigma_i{}^\frown 1, \dots, \sigma_{n-1}). \end{aligned}$$

A semi-measure is *enumerable* if it is computable from below.

A semi-measure is *optimal* if it multiplicatively dominates all enumerable semi-measures.

Henceforth, μ will denote an optimal, enumerable semi-measure.

Definition 2.8. For $(\sigma_0, \ldots, \sigma_{n-1}) \in (2^{<\omega})^n$, define the *$KM$-complexity* as

$$KM(\sigma_0, \ldots, \sigma_{n-1}) := -\log \mu(\sigma_0, \ldots, \sigma_{n-1}).$$

Note that KM has the pleasing property that if $\sigma_i \subseteq \tau_i$ for all i, then

$$KM(\sigma_0, \ldots, \sigma_{n-1}) \leq KM(\tau_0, \ldots, \tau_{n-1}).$$

Definition 2.9. For $f = (f_0, \ldots, f_{n-1}) \in (2^\omega)^n$, define the *effective Hausdorff dimension* as

$$\dim_H(f) := \liminf_n \frac{KM(f_1 \restriction n, \ldots, f_{n-1} \restriction n)}{n}.$$

Identifying points in $[0, 1)$ with points in 2^ω via binary expansion, we define the effective Hausdorff dimension of points in $[0, 1)^n$. It is easily verified that the choice of binary expansion (when more than one exist) has no effect on the dimension. It is also seen that translation by a rational amount in a direction parallel to an axis has no effect on the dimension, so we extend this notion to $\mathbb{R}^n$ via such translations.

Just as we use binary expansion to identify points, we will also identify sets. Given $\sigma \in 2^\omega$, let $[\sigma] = \{f \in 2^\omega : \sigma \prec f\}$. We will identify $[\sigma]$ with the closed interval of reals whose binary expansions are contained in $[\sigma]$. That is, $[\sigma]$ is identified with $\{0.f \in \mathbb{R} : f \in [\sigma]\}$. Note that $\lambda([\sigma]) = 2^{-|\sigma|}$.

It will be convenient to partition $\mathbb{R}^n$ as:

$$\mathfrak{R}^n_m = \{x \in \mathbb{R}^n : \text{exactly } m\text{-many coordinates of } x \text{ are rational}\}$$

Our definition of effective Hausdorff dimension differs from that used in [13], but the two notions are equivalent. While we constructed dimension on $(2^\omega)^n$ and then identified this space with $\mathbb{R}^n$ in the natural way, Lutz and Weihrauch defined dimension directly upon $\mathbb{R}^n$. They also base their notion of dimension on Kolmogorov complexity, while we use KM-complexity. The reader is referred to [14] for the equivalence of martingale defined dimension and complexity defined dimension, and to [11] by Li and Vitányi for further reading on KM-complexity and its relation to Kolmogorov complexity.

We also make heavy use of (classical) inductive dimension. The necessary background can be obtained from Chapter 3 of [15] by van Mill, although we repeat the necessary results here.

For $X \subseteq \mathbb{R}^n$, let $\text{ind}(X) \in \{-1, 0, 1, \ldots, n\}$ denote the inductive dimension of a set X. The definition is such that $\text{ind}(X) = -1$ only when $X = \emptyset$.

Proposition 2.10 ([15, Proposition 3.2.10])**.** $\text{ind}(\mathfrak{R}^n_m) = 0$.

Proposition 2.11 ([15, Corollary 3.1.7])**.** *If* $\text{ind}(X) = n$*, then* X *is not contained in the union of* n*-many sets each of inductive dimension 0.*

Definition 2.12. If Y is connected, say X *separates* Y if $Y - X$ is not connected.

Proposition 2.13 ([15, Theorem 3.7.6])**.** *If* $H \subseteq \mathbb{R}^n$ *is open and connected, and* X *separates* H*, then* $\text{ind}(X) \geq n - 1$*.*

Proposition 2.14 ([15, Theorem 3.2.5])**.** *If* $X \subseteq \mathbb{R}^n$ *is closed and* $\text{ind}(X) > 0$*, then* X *is not totally disconnected.*

Proposition 2.15 ([15, Theorem 3.2.5])**.** *If* $\text{ind}(X) = 0$*, then* X *is totally disconnected.*

2.3 Proof of Results

Our main result is Theorem 2.5. The main tools to proving this are the following two lemmas. They both say, in a sense, that even if Z has small intersection with a given region, it will have large intersection with a nearby region.

Definition 2.16. Let $C, D \subset \mathbb{R}^n$ be distinct closed n-cubes. Call D *adjacent* to C if D is a translation of C, and there is some point v which is a vertex of both C and D.

Note that any given n-cube has $3^n - 1$ adjacent n-cubes.

Lemma 2.17. *Let $C \subset \mathbb{R}^n$ be a closed n-cube aligned with the axes (i.e., C is a translation of $[0, a]^n$ for some a). Let $\{D_j\}_{j<3^n-1}$ be the collection of adjacent n-cubes.*

Let $Z \subseteq \mathbb{R}^n$ be a closed, connected set. If $Z \cap C \neq \emptyset$, but $Z \not\subseteq C \cup \bigcup_j D_j$, then for some i and some D_j,

$$\lambda(\pi_i(Z \cap D_j)) \geq \frac{a}{3^{n-1}}. \tag{†}$$

Proof. Consider $\pi_i(D_j)$. Note that there is some b_i such that

$$\pi_i(D_j) \in \{[b_i, b_i + a], [b_i + a, b_i + 2a], [b_i + 2a, b_i + 3a]\}$$

for all j. Let

$$F_i^0 = \bigcup_{\pi_i(D_j)=[b_i,b_i+a]} D_j,$$

and

$$F_i^1 = \bigcup_{\pi_i(D_j)=[b_i+2a,b_i+3a]} D_j.$$

Note that 3^{n-1} many D_j participate in each F_i^*. If $\pi_i(F_i^0 \cap Z) = [b_i, b_i + a]$ or $\pi_i(F_i^1 \cap Z) = [b_i + 2a, b_i + 3a]$, then by additivity of λ, some D_j must satisfy (†).

If instead $\pi_i(F_i^0 \cap Z) \subsetneq [b_i, b_i + a]$ and $\pi_i(F_i^1 \cap Z) \subsetneq [b_i + 2a, b_i + 3a]$, then for some c_i^0, c_i^1,

$$\pi_i^{-1}(c_i^0) \cap F_i^0 \cap Z = \emptyset$$

and

$$\pi_i^{-1}(c_i^1) \cap F_i^1 \cap Z = \emptyset.$$

If these exist for every i, then

$$\bigcup_i (\pi_i^{-1}(c_i^0) \cap F_i^0) \cup (\pi_i^{-1}(c_i^1) \cap F_i^1)$$

separates Z, contradicting connectedness. □

Note: The condition that Z be closed is far more than is necessary, of course. The only place we use it in the above is to imply that $\pi_i(Z)$ is measurable. However, we will only be applying this lemma for closed Z.

Lemma 2.18. *Let $C \subset \mathbb{R}^n$ be a closed n-cube aligned with the axes (i.e., C is a translation of $[0,a]^n$ for some a). Let $\{D_j\}_{j<3^n-1}$ be the collection of adjacent n-cubes.*

Let $Z \subseteq \mathbb{R}^n$ be closed with the property that for any open set U with $Z \cap U \neq 0$, $\operatorname{ind}(Z \cap U) \geq n-1$. If $Z \cap C \neq \emptyset$, but $Z \not\subseteq C \cup \bigcup_j D_j$, then for some D_j, $Z \cap D_j$ contains a point of dimension at most 1.

Proof. Let $D = \text{interior}(\bigcup_j D_j)$. By connectedness, Z intersects D. It suffices to show:

$$Z \cap D \cap \mathfrak{R}_n^n \neq \emptyset \text{ or } Z \cap D \cap \mathfrak{R}_{n-1}^n \neq \emptyset.$$

Suppose not. Then $Z \cap D \subseteq \bigcup_{j<n-1} \mathfrak{R}_j^n$. But then by Propositions 2.10 and 2.11, this contradicts the hypothesis on Z. □

We now prove the main result.

Proof of Theorem 2.5. We build $x_0, \dots, x_{n-1} \in \mathbb{R}$ in stages by building sequences

$$\{\sigma_i^0\}_{i\in\omega}, \dots, \{\sigma_i^{n-1}\}_{i\in\omega}$$

with each $\sigma_i^m \in 2^{<\omega}$. For a fixed i, all the σ_i^m will have the same length, while for a fixed m, $\lim_i |\sigma_i^m| = \infty$. However, it will not necessarily be the case that $\sigma_i^m \subseteq \sigma_{i+1}^m$. Indeed, $\lim_i \sigma_i^m \upharpoonright s$ may not exist.

So for each σ_i^m, we shall consider a point $y_i^m \in [\sigma_i^m]$ (recalling that $[\sigma_i^m]$ is identified with a closed subset of $\mathbb{R}$) and take $x_m = \lim_i y_i^m$. Because the diameter of the $[\sigma_i^m]$ goes to zero, any choice of y_i^m will have the same limit. Our point of dimension 1 will then be $(x_0, \dots, x_{n-1})$.

At every stage, our construction employs one of two possible strategies: one strategy is for ensuring that the complexity of $(x_0, \dots, x_{n-1})$ is not too low, while the other ensures that the complexity is not too high.

Strategy 1 (not too low):

Given $\sigma_i^0, \dots, \sigma_i^{n-1}$ each of length ℓ with $D = [\sigma_i^0] \times \cdots \times [\sigma_i^{n-1}]$ satisfying (†) for some π, without loss of generality assume it satisfies it for π_0.

Suppose we wish to extend by k-many bits, for some k. We consider all possible extensions of $\sigma_i^0, \dots, \sigma_i^{n-1}$. Clearly we are not interested in extensions which take us away from Z. So consider

$$E = \{(\tau^0, \dots, \tau^{n-1}) \in (2^k)^n : [\sigma_i^0 \frown \tau^0] \times \cdots \times [\sigma_i^{n-1} \frown \tau^{n-1}] \cap Z \neq \emptyset\}.$$

By assumption, $|E| \geq |\pi_0(E)| \geq 2^k/3^{n-1}$. So there exist some $\tau^0, \dots, \tau^{n-1}$ such that

$$\frac{2^k}{3^{n-1}} \mu(\sigma_i^0 \frown \tau^0, \dots, \sigma_i^{n-1} \frown \tau^{n-1}) \leq \mu(\sigma_i^0, \dots, \sigma_i^{n-1}).$$

Thus

$$KM(\sigma_i^0 \frown \tau^0, \ldots, \sigma_i^{n-1} \frown \tau^{n-1}) \geq KM(\sigma_i^0, \ldots, \sigma_i^{n-1}) + k - (n-1)\log 3.$$

Strategy 2 (not too high):

Given $\sigma_i^0, \ldots, \sigma_i^{n-1}$ each of length ℓ with $Z \cap [\sigma_i^0] \times \cdots \times [\sigma_i^{n-1}]$ containing a point $(d_0, \ldots, d_{n-1})$ of dimension at most 1, note that $\sigma_i^k \prec d_k$.

If $(d_0, \ldots, d_{n-1})$ has dimension exactly 1, the proof is complete. If it has dimension less than one, then there exists some $m \geq i$ such that

$$KM(d_0 \upharpoonright m, \ldots, d_{n-1} \upharpoonright m) \leq m.$$

Assuming i is not such an m, choosing the least such m results in

$$KM(d_0 \upharpoonright m, \ldots, d_{n-1} \upharpoonright m) \geq m - 1,$$

because KM can only increase as m increases.

Construction:

By Lemma 2.17, choose some $\sigma_0^0, \ldots, \sigma_0^{n-1}$ all of the same length such that $D = [\sigma_0^0] \times \cdots \times [\sigma_0^{n-1}]$ satisfies (†) for some π, and such that $Z \not\subseteq D$.

At stage i, if $KM(\sigma_i^0, \ldots, \sigma_i^{n-1}) \leq |\sigma_i^0|$, use Lemma 2.17 to replace $\sigma_i^0, \ldots, \sigma_i^{n-1}$ with adjacent strings satisfying (†) for some π. Then follow strategy 1 to generate $\sigma_{i+1}^0, \ldots, \sigma_{i+1}^{n-1}$ of length $|\sigma_i^0| + i$.

Otherwise, use Lemma 2.18 to replace $\sigma_i^0, \ldots, \sigma_i^{n-1}$ with adjacent strings such that $Z \cap [\sigma_i^0] \times \cdots \times [\sigma_i^{n-1}]$ contains a point of dimension at most 1. Then follow strategy 2, either generating $\sigma_{i+1}^0, \ldots, \sigma_{i+1}^{n-1}$ or finding a point of dimension 1 and ending the construction.

Take $(x_0, \ldots, x_{n-1})$ to be the limit of $(y_i^0, \ldots, y_i^{n-1}) \in [\sigma_i^0] \times \cdots \times [\sigma_i^{n-1}]$ as previously discussed. This is our desired point.

Verification:

Clearly if we halt early via some strategy 2, the construction has succeeded. So henceforth we assume this does not happen.

There are several points to check. First, we must show that the x_k actually exist. This is an unfortunately involved proof for what is actually a fairly simple idea: for $j \geq i$, consider how $\sigma_j^0 \upharpoonright |\sigma_i^0|$ can change through the use of the two lemmas. It can be changed directly at stage $i+1$ (when we trade the cube σ_i^0 is a part of for an adjacent cube), or it can be changed indirectly at stage $j > i$ (when we trade a small cube within σ_i^0 for a small cube outside of σ_i^0). The indirect changes add up in a geometric way, and so they will only occur at one boundary of the cube of $\sigma_{i+1}^0 \upharpoonright |\sigma_i^0|$. So either $\sigma_j^0 \upharpoonright |\sigma_i^0|$ stabilizes, or it switches infinitely between two adjacent cubes which share a boundary. Either way, we see that the limit exists.

Now we make the above argument more rigorous. Without loss of generality, we consider only x_0. For a string $\sigma \in 2^\ell$, let $\mathrm{succ}(\sigma)$ denote the lexicographic successor of σ in 2^ℓ and $\mathrm{pred}(\sigma)$ denote the lexicographic predecessor of σ in 2^ℓ.

Claim 2.19. *Let $|\sigma_i^0| = \ell$. Then for any $j \geq i$, $\sigma_j^0 \upharpoonright \ell$ is one of σ_i^0, $succ(\sigma_i^0)$, $succ(succ(\sigma_i^0))$, $pred(\sigma_i^0)$, or $pred(pred(\sigma_i^0))$.*

Proof. Let $\ell_k = |\sigma_k^0|$. Because of the use of Lemma 2.17 or 2.18 in the construction, $\sigma_{k+1}^0 \upharpoonright \ell_k$ need not be σ_k^0, but if not, the two strings will be adjacent in 2^{ℓ_k}. So

$$\inf[\sigma_{k+1}^0 \upharpoonright \ell_k] = \inf[\sigma_k^0] + a_k 2^{\ell_k},$$

where $a_k \in \{-1, 0, 1\}$.

Since $[\sigma^0_{k+1}]$ has diameter $2^{-\ell_{k+1}}$, we have

$$\inf[\sigma^0_{k+1} \upharpoonright \ell_k] \leq \inf[\sigma^0_{k+1}] \leq \inf[\sigma^0_{k+1} \upharpoonright \ell_k] + 2^{-\ell_k} - 2^{-\ell_{k+1}}.$$

Thus,

$$\inf[\sigma^0_i] + \sum_{i \leq k < j} a_k 2^{-\ell_k} \leq \inf[\sigma^0_j] \leq \inf[\sigma^0_i] + 2^{-\ell_i} - 2^{-\ell_j} + \sum_{i \leq k < j} a_k 2^{-\ell_k}.$$

Taking a_k to be worst, we see

$$\inf[\sigma^0_i] - 2 \cdot 2^{-\ell_i} < \inf[\sigma^0_j] < \inf[\sigma^0_i] + 3 \cdot 2^{-\ell_i}.$$

So $\text{pred}(\text{pred}(\sigma^0_i)) \leq \sigma^0_j \upharpoonright \ell_i \leq \text{succ}(\text{succ}(\sigma^0_i))$. □

Claim 2.20. *For every i, take $y^0_i \in [\sigma^0_i]$. Then $x_0 = \lim_i y^0_i$ exists.*

Proof. Again, let $\ell_i = |\sigma^0_i|$.

For any $j \geq i$, $\sigma^0_j \upharpoonright \ell_i$ must be one of the five above values. Then consider the closed interval $J_i = [\text{pred}(\text{pred}(\sigma^0_i)] \cup [\text{pred}(\sigma^0_i)] \cup [\sigma^0_i] \cup [\text{succ}(\sigma^0_i] \cup [\text{succ}(\text{succ}(\sigma^0_i))]$. J_i has diameter $5 \cdot 2^{-\ell_i}$, and for any $j \geq i$, $y^0_j \in J_i$. Thus $\lim_i y^0_i$ converges. □

Next we must show that our point lies on Z.

Claim 2.21. $(x_0, \ldots, x_{n-1}) \in Z$.

Proof. By construction, $\big([\sigma^0_i] \times \cdots \times [\sigma^{n-1}_i]\big) \cap Z \neq 0$ for any i. Thus we can take $(y^0_i, \ldots, y^{n-1}_i) \in Z$. Since Z is closed, $(x_0, \ldots, x_{n-1}) = \lim_i (y^0_i, \ldots, y^{n-1}_i) \in Z$. □

Third, we must show that $\dim_H(x_0, \ldots, x_{n-1}) = 1$.

Claim 2.22. $\dim_H(x_0, \ldots, x_{n-1}) \geq 1$.

Proof. Our initial strings $\sigma_0^0, \ldots, \sigma_0^{n-1}$ have some complexity $KM(\sigma_0^0, \ldots, \sigma_0^{n-1}) = A$. When we follow strategy 1 at stage i, the length of our strings increase by i many bits, and the complexity increases by at least $i - (n-1)\log 3$. When we follow strategy 2, our resulting strings have length ℓ, and our resulting complexity is at least $\ell - 1$. Replacing all the σ_i^m with adjacent strings changes the complexity by at most $2\log|\sigma_i^0|$.

So let $\ell_i = |\sigma_i^0|$ and let i_0 be the last stage before stage i at which strategy 2 was followed. Then

$$\begin{aligned} KM(\sigma_i^0, \ldots, \sigma_i^{n-1}) &\geq (\ell_{i_0} - 1) + (\ell_i - \ell_{i_0}) - (i - i_0)((n-1)\log 3 + 2\log \ell_i) \\ &\geq \ell_i - i((n-1)\log 3 + 2\log \ell_i). \end{aligned}$$

If there is no such stage i_0, then

$$\begin{aligned} KM(\sigma_i^0, \ldots, \sigma_i^{n-1}) &\geq A + \ell_i - \ell_0 - i((n-1)\log 3 + 2\log \ell_i) \\ &\geq \ell_i - \ell_0 - i((n-1)\log 3 + 2\log \ell_i). \end{aligned}$$

Note that by construction, strategy 2 will never be employed at successive stages. So at stage i, strategy 1 will have been used at least every other stage. Further, since strategy 1 used at stage j always increases the length of the strings by j, $\ell_i \geq i^2/4$. Thus $-i((n-1)\log 3 + 2\log \ell_i)$ in the above is a lower order term (recalling that n is constant), and so

$$\liminf_i \frac{KM(\sigma_i^0, \ldots, \sigma_i^{n-1})}{\ell_i} \geq 1.$$

Now consider some ℓ_i. Then

$$x_0 \upharpoonright \ell_i \in \{\sigma_i^0, \mathrm{succ}(\sigma_i^0), \mathrm{succ}(\mathrm{succ}(\sigma_i^0)), \mathrm{pred}(\sigma_i^0), \mathrm{pred}(\mathrm{pred}(\sigma_i^0))\},$$

and similarly for $x_1, \ldots, x_{n-1}$. So

$$|KM(x_0 \upharpoonright \ell_i, \ldots, x_{n-1} \upharpoonright \ell_i) - KM(\sigma_i^0, \ldots, \sigma_i^{n-1})| \leq 4 \log \ell_i.$$

So

$$KM(x_0 \upharpoonright \ell_i, \ldots, x_{n-1} \upharpoonright \ell_i) \geq \ell_i - \ell_0 - i((n-1)\log 3 - 2\log \ell_i) - 4 \log \ell_i,$$

and thus

$$\liminf_i \frac{KM(x_0 \upharpoonright \ell_i, \ldots, x_{n-1} \upharpoonright \ell_i)}{\ell_i} \geq 1.$$

Finally, consider some k with $\ell_i \leq k < \ell_{i+1}$. If stage i follows strategy 1, then $k - \ell_i < i$, and thus

$$\begin{aligned} \frac{KM(x_0 \upharpoonright k, \ldots, x_{n-1} \upharpoonright k)}{k} &\geq \frac{KM(x_0 \upharpoonright \ell_i, \ldots, x_{n-1} \upharpoonright \ell_i)}{k} \\ &> \frac{KM(x_0 \upharpoonright \ell_i, \ldots, x_{n-1} \upharpoonright \ell_i)}{\ell_i + i} \\ &\geq \frac{KM(x_0 \upharpoonright \ell_i, \ldots, x_{n-1} \upharpoonright \ell_i)}{\ell_i + 2\sqrt{\ell_i}}. \end{aligned}$$

If stage i follows strategy 2, then

$$KM(\sigma_{i+1}^0 \upharpoonright k, \ldots, \sigma_{i+1}^{n-1} \upharpoonright k) > k,$$

since ℓ_{i+1} will be least such that the above does not hold. Thus

$$\begin{aligned}\frac{KM(x_0 \upharpoonright k, \ldots, x_{n-1} \upharpoonright k)}{k} &\geq \frac{KM(\sigma^0_{i+1} \upharpoonright k, \ldots, \sigma^{n-1}_{i+1} \upharpoonright k) - 4\log k}{k} \\ &> \frac{k - 4\log k}{k}.\end{aligned}$$

So

$$\dim_H(x_0, \ldots, x_{n-1}) = \liminf_k \frac{KM(x_0 \upharpoonright k, \ldots, x_{n-1} \upharpoonright k)}{k} \geq 1. \qquad \square$$

Claim 2.23. $\dim_H(x_0, \ldots, x_{n-1}) \leq 1$.

Proof. Suppose not. Then for some i_0 and all $i > i_0$,

$$KM(x_0 \upharpoonright \ell_i, \ldots, x_{n-1} \upharpoonright \ell_i) > \ell_i + 4\log \ell_i.$$

But in this case, $KM(\sigma^0_i, \ldots, \sigma^{n-1}_i) > \ell_i$, and so at stage $i+1$, strategy 2 will be invoked, resulting in $KM(\sigma^0_{i+1}, \ldots, \sigma^{n-1}_{i+1}) \leq \ell_{i+1}$, and thus

$$KM(x_0 \upharpoonright \ell_{i+1}, \ldots, x_{n-1} \upharpoonright \ell_{i+1}) \leq \ell_{i+1} + 4\log \ell_{i+1},$$

contradicting our above assumption about i_0. $\square$

Thus $\dim_H(x_0, \ldots, x_{n-1}) = 1$. This completes the proof. $\square$

Proof of Theorem 2.3. Let $X \subset \mathbb{R}^n$ be the set of points of dimension 1.

Suppose A, B are open sets in $\mathbb{R}^n$ such that $A \cap X$ and $B \cap X$ partition X. Then $X \subseteq A \cup B$ and $A \cap B \cap X = \emptyset$. But X is dense, so $A \cap B = \emptyset$.

Let $Z' =$ bd $\overline{A}$. Then Z' separates $\mathbb{R}^n$. Let Z be a non-singleton component of Z' (Propositions 2.13 and 2.14). Then for any open set U such that $U \cap Z \neq \emptyset$, $\overline{A}$ intersects

U but is not dense in U. So $Z \cap U$ separates U, and thus $\operatorname{ind}(Z \cap U) \geq n - 1$.

By the above theorem, Z contains a point of dimension 1, and since $Z \subseteq \mathbb{R}^n - (A \cup B)$, this contradicts our choice of A and B. □

Proof of Theorem 2.4. Suppose f is any non-constant path in $\mathbb{R}^2$. Its image is a connected, locally connected set. Thus in any neighborhood U with $\operatorname{im} f \cap U \neq \emptyset$, $\operatorname{ind}(\operatorname{im} f \cap U) \geq 1$ (Proposition 2.15), which in this case means at least $n - 1$. So by the theorem, it contains a point of dimension 1. □

Chapter 3

Computable Categoricity

With Greenberg, Kach and Lempp, I investigated computable categoricity of size $\aleph_1$ linear orders. We discovered a strange class of linear orders that were not relatively computable, because they contained certain d.c.e information in their order types. Relative to an oracle for this information, these linear orders became relatively computable categorical. This led to the notion of relatively computably categorical above a degree. The question arose if this notion occurs in countable structures.

We begin by illustrating a technique for constructing computably categorical structures. This technique was developed in collaboration with Downey, Kach and Lempp. In the following section, we use this technique to separate the three notions of computable categoricity.

3.1 Pushing on Isomorphisms

We first describe the general structure of the technique. We then demonstrate its use in a new proof of an existing result of Khoussainov and Shore.

Suppose we are constructing a graph $\mathfrak{A}$ through the use of various strategies, and each such strategy has four desirable properties:

1. The strategy will succeed even in the presence of finite injury.

2. At every stage s, the subgraph built by the strategy at stage s is rigid and does not

embed into the subgraph built by any other strategy at stage s (including other instances of the strategy).

3. At every stage s, the subgraph built by the strategy at stage s has a unique embedding into the subgraph built at stage $s+1$.

4. If the strategy is along the true path, the subgraph created by the full run of the strategy is computably categorical (not necessarily with any uniformity) (possibly because it is finite).

Suppose $\mathfrak{B}$ is another computable structure, and we wish to satisfy the requirement

$$\mathfrak{A} \cong \mathfrak{B} \Rightarrow (\exists f \in \Delta^0_1)[\mathfrak{A} \cong_f \mathfrak{B}].$$

Then at a certain level in the priority tree, we will have a strategy $\chi_{\mathfrak{B}}$ for meeting this requirement. $\chi_{\mathfrak{B}}$ does not construct any of $\mathfrak{A}$; however, it does construct an isomorphism f from $\mathfrak{A}$ to $\mathfrak{B}$, and it affects the construction through its choice of outcome. $\chi_{\mathfrak{B}}$ must correctly map each component in $\mathfrak{A}$ to a component in $\mathfrak{B}$, although it treats components differently depending on the strategy that constructed them. The strategy has two outcomes: "isomorphic" and "not-isomorphic".

Components created by strategies above $\chi_{\mathfrak{B}}$ in the priority tree are ignored. Since there are only finitely many such strategies, f can be extended to them non-uniformly after the construction is completed (via property (4)).

Strategies to the right of $\chi_{\mathfrak{B}}$ are reset every time $\chi_{\mathfrak{B}}$ is visited, and any components created by them will never again receive attention. $\chi_{\mathfrak{B}}$ searches $\mathfrak{B}$ for identical components and maps components appropriately. By property (2), these maps are guaranteed to be correct.

Since $\chi_{\mathfrak{B}}$ believes that it is on the true path, it believes that strategies to the left of it will never again act, and thus any components created by them will never again receive attention. It handles such components in the same fashion as the previous case.

Strategies beneath the “not-isomorphic” outcome are reset every time $\chi_{\mathfrak{B}}$ has outcome “isomorphic”. $\chi_{\mathfrak{B}}$ ignores the components created by such strategies until they have been reset, at which point it knows that those components will never again receive attention. It then handles them in the same fashion as the previous two cases.

Components created by strategies beneath the “isomorphic” outcome are only considered when $\chi_{\mathfrak{B}}$ has the “isomorphic” outcome. It only has this outcome when every such component appears identical to a component in $\mathfrak{B}$, and the identical component in $\mathfrak{B}$ is the component mapped to by f whenever f has been defined. At such a time, every such component is mapped to the corresponding component in $\mathfrak{B}$ (and the existing maps are extended, via property (3)). Again by property (2) of the strategies below the “isomorphic” outcome, if $\chi_{\mathfrak{B}}$ has the “isomorphic” outcome only finitely often, then $\mathfrak{A} \not\cong \mathfrak{B}$.

We use this technique in the following proof.

Theorem 3.1 (Khoussainov and Shore[10])**.** *There is a rigid, computably categorical structure $\mathfrak{A}$ with no formally c.e. Scott family.*

First we remind the reader what it means for a structure to have a formally c.e. Scott family.

Definition 3.2. Let A be a structure in a computable language L. A *formally Σ^0_α-Scott family* on A is a Σ^0_1 set X of Σ^0_α $L^r_{\omega_1,\omega}$-formulas satisfying the following two properties:

1. For all $\overline{a} \in A^n$, there is a $\varphi \in X$ such that $A \models \varphi(\overline{a})$.
2. For all $\overline{a}, \overline{b} \in A^n$, and any $\varphi \in X$, if $A \models \varphi(\overline{a}) \wedge \varphi(\overline{b})$, then there is an automorphism of A sending $\overline{a}$ to $\overline{b}$.

A formally Σ^0_1-Scott family is also called a *formally c.e. Scott family.*

We emphasize that the formula φ are in the language of the model, not the language of arithmetic.

Formally Σ^0_α-Scott families are of interest because of the following result, proved for $\alpha = 1$ by Goncharov [8], and for remaining computable α by Ash [1].

Theorem 3.3. *Let A be a computable structure. Then the following are equivalent:*

- *A has a formally Σ^0_α-Scott family.*
- *A is relatively Δ^0_α-categorical.*

Proof of Theorem 3.1. Construction:

Let $\{X_i\}_{i\in\omega}$ be an enumeration of all formally c.e. families. Our strategy Γ_i for defeating X_i is as follows:

1. Choose a large n. Create a vertex x_i with a loop of size 1 and a loop of size n. Choose these elements disjoint from the parameters of X_i.
2. Wait for a formula $\phi \in X_i$ to describe x_i.
3. Choose a large m. Attach a loop of size m to x_i. Choose the elements disjoint from the parameters of X_i.
4. Create a vertex y_i with a loop of size 1 and a loop of size n. Choose these elements disjoint from the parameters of X_i.

There are two possible outcomes. If the strategy waits forever at step (2), then no formula in X_i describes x_i, and thus X_i is not a Scott family for $\mathfrak{A}$. If the strategy reaches step (4), then ϕ describes both x_i and y_i, but clearly x_i and y_i are not in the same orbit, so X_i is not a Scott family for $\mathfrak{A}$.

Clearly this strategy will succeed in diagonalizing against X_i even if it is injured finitely many times. At every stage, the subgraphs are rigid and incomparable under embedding by our choice of large m and n. Since the final subgraph is finite, it is computably categorical. Note that it is essential that steps (3) and (4) occur separately and in the order listed to ensure that there is always a unique embedding from each stage to the next.

We then put these Γ_i on a tree along with $\chi_{\mathfrak{B}}$ for ensuring computable categoricity.

Verification:

We have already shown that the resulting structure $\mathfrak{A}$ has no formally c.e. Scott family. $\mathfrak{A}$ is clearly rigid. All that remains to be shown is that $\mathfrak{A}$ is computably categorical.

Claim 3.4. *If $\chi_{\mathfrak{B}}$ is along the true path and $\mathfrak{A} \cong \mathfrak{B}$, $\chi_{\mathfrak{B}}$ will have outcome "isomorphic" infinitely often.*

Proof. Suppose $\chi_{\mathfrak{B}}$ is along the true path and has outcome "isomorphic" only finitely many times. Let t_0 be a stage after the final time $\chi_{\mathfrak{B}}$ has outcome "isomorphic". Then there is some component created by some Γ_i below the "isomorphic" outcome of $\chi_{\mathfrak{B}}$ which is preventing the "isomorphic" outcome from being achieved again. There are several possibilities.

It might be that Γ_i has completed step (1), but no vertex with a loop of size 1 and a loop of size n ever appears in $\mathfrak{B}$. Then $\mathfrak{B}$ is not isomorphic to $\mathfrak{A}$.

It might be that Γ_i has completed step (3), but no loop of size m appears attached to $f(x_i)$ (recall that if we have reached step (3), then $\chi_{\mathfrak{B}}$ has defined f on x_i). In this case, $\mathfrak{B}$ contains an element with a loop of size n and no loop of size m, but $\mathfrak{A}$ contains no such element (since y_i has not yet been created). Then $\mathfrak{B}$ is not isomorphic to $\mathfrak{A}$.

It might be that Γ_i has completed step (4), but no new vertex with a loop of size 1 and a loop of size n appears in $\mathfrak{B}$. Then the element y_i has no match in $\mathfrak{B}$, and thus $\mathfrak{B}$ is not isomorphic to $\mathfrak{A}$. □

Claim 3.5. *If $\chi_{\mathfrak{B}}$ is along the true path and $\mathfrak{A} \cong \mathfrak{B}$, the map f constructed by $\chi_{\mathfrak{B}}$ is an isomorphism.*

Proof. For components built by strategies above f, f is non-uniformly defined correctly.

For components built by a strategy Γ_i which is incomparable to $\chi_{\mathfrak{B}}$ on the priority tree, if Γ_i reaches step (4), then since $\chi_{\mathfrak{B}}$ is along the true path, it reaches this step before $\chi_{\mathfrak{B}}$ attempts to extend f to these components. So when $\chi_{\mathfrak{B}}$ attempts to extend f, it searches for the two components in $\mathfrak{B}$ containing n-loops (for the appropriate n), one containing an m-loop and one not, and maps the corresponding components in $\mathfrak{A}$ to them.

For components built by a strategy Γ_i which is incomparable to $\chi_{\mathfrak{B}}$ on the priority tree, if Γ_i did not reach step (4), then there is a unique component in $\mathfrak{B}$ with an n-loop (for appropriate n). $\chi_{\mathfrak{B}}$ searches for this component and maps the corresponding component in $\mathfrak{A}$ to it.

Components built by a strategy Γ_i below the "not-isomorphic" outcome of $\chi_{\mathfrak{B}}$ are handled identically to the previous two cases.

For components built by a strategy Γ_i below the "isomorphic" outcome, the only concern is that $\chi_{\mathfrak{B}}$ might map x_i to the image of y_i. But in this case, after Γ_i reaches step (3), $\chi_{\mathfrak{B}}$ will never again have the "isomorphic" outcome, since the image of x_i will never appear identical to x_i (it will never have an m-loop). This contradicts the assumption that $\mathfrak{A} \cong \mathfrak{B}$. □

This completes the proof. □

3.2 Eventual Categoricity

Definition 3.6. For a computable structure $\mathfrak{A}$ and a degree $\mathbf{d}$, call $\mathfrak{A}$ *relatively computably categorical above* $\mathbf{d}$ (*relatively* Δ^0_α*-categorical above* $\mathbf{d}$) if for all $\mathfrak{B}, \mathfrak{C}$ with the open diagrams of $\mathfrak{B}, \mathfrak{C} \geq_T \mathbf{d}$, and $\mathfrak{C} \cong \mathfrak{B} \cong \mathfrak{A}$, there exists an isomorphism $f : \mathfrak{B} \cong \mathfrak{C}$ with f computable in $\mathfrak{B} \oplus \mathfrak{C}$ ($f \in \Delta^0_\alpha(\mathfrak{B} \oplus \mathfrak{C})$).

Lemma 3.7. *For a computable structure* $\mathfrak{A}$*, the following are equivalent:*

1. $\mathfrak{A}$ *is relatively* Δ^0_α*-categorical above* $\mathbf{d}$.
2. *For any* $\mathfrak{B}, \mathfrak{C}$ *with* $\mathfrak{C} \cong \mathfrak{B} \cong \mathfrak{A}$*, there exists an isomorphism* $f : \mathfrak{C} \cong \mathfrak{B}$ *with* $f \in \Delta^0_\alpha(\mathfrak{B} \oplus \mathfrak{C} \oplus \mathbf{d})$.

Proof. Clearly (2) implies (1). For the reverse, we use Theorem 3.2.1 in Ash and Knight [2]. Then there exist $\hat{B}, g_1 \in \deg(\mathfrak{B} \oplus \mathbf{d})$ and $\hat{C}, g_2 \in \deg(\mathfrak{C} \oplus \mathbf{d})$ such that $g_1 : \mathfrak{B} \cong \hat{B}$, $g_2 : \mathfrak{C} \cong \hat{C}$.

By relatively Δ^0_α-categoricity above $\mathbf{d}$, there exists $f : \hat{B} \cong \hat{C}$ with $f \in \Delta^0_\alpha(\mathfrak{B} \oplus \mathfrak{C} \oplus \mathbf{d})$. Then $g_2^{-1} \circ f \circ g_1 \in \Delta^0_\alpha(\mathfrak{B} \oplus \mathfrak{C} \oplus \mathbf{d})$ and $g_2^{-1} \circ f \circ g_1 : \mathfrak{B} \cong \mathfrak{C}$. □

Corollary 3.8. *If* $\mathfrak{A}$ *is relatively* Δ^0_α*-categorical above* $\mathbf{d}$*, and* $\mathbf{d} \leq \mathbf{b}^{(\beta)}$*, then* $\mathfrak{A}$ *is relatively* $\Delta^0_{\beta+\alpha}$*-categorical above* $\mathbf{b}$.

Proof. $\Delta^0_\alpha(\mathfrak{B} \oplus \mathfrak{C} \oplus \mathbf{d}) \subseteq \Delta^0_{\beta+\alpha}(\mathfrak{B} \oplus \mathfrak{C} \oplus \mathbf{b})$. □

In some cases, this notion gives us no new information.

Theorem 3.9. *A linear order is relatively computably categorical above some* $\mathbf{d}$ *iff it is relatively computably categorical.*

Proof. The proof that a computably categorical linear order must possess only finitely many adjacencies succeeds in the presence of a $\mathbf{d}$ oracle. □

The following three theorems, however, separate this notion from the other categoricity notions.

Theorem 3.10. *For any nonzero c.e. degree* $\mathbf{y}$*, there exists a structure* $\mathfrak{A}$ *which is relatively computably categorical above* $\mathbf{y}$ *(and thus relatively* Δ^0_2*-categorical), but* $\mathfrak{A}$ *is not computably categorical.*

Proof. Choose $Y \in \mathbf{y}$ a c.e. set. Our structure is a graph.

Construction:

Begin by constructing an "ω-spine"—a component of type ω. To each element of the spine, attach a single path of length 1.

When n enters Y, attach a new path of length 2 to the nth element of the spine.

Verification:

Given $\mathfrak{B} \cong \mathfrak{A}$, we show how $\mathfrak{B} \oplus \mathbf{y}$ computes an isomorphism f.

We non-uniformly know the initial elements of the spines in $\mathfrak{A}$ and $\mathfrak{B}$. f maps the ω-spines in the obvious way. For the nth element of the spine, if $n \in Y$, f waits until both a path of length 2 and a path of length 1 appear in both $\mathfrak{A}$ and $\mathfrak{B}$. Then it maps them as appropriate. If $n \notin Y$, f only waits for paths of length 1 to map.

We build a computable copy $\mathfrak{A}'$ isomorphic to $\mathfrak{A}$, but not by any computable isomorphism. Begin by simply copying $\mathfrak{A}$. Since Y is properly c.e., if ϕ_e is a total computable function from $\mathfrak{A}$ to $\mathfrak{A}'$, there will be infinitely many n which enter Y after ϕ_e has converged on the 1-path attached to the nth element. Add the 2-path in $\mathfrak{A}'$ to defeat ϕ_e (i.e., if ϕ_e maps the 1-path in $\mathfrak{A}$ to the 1-path in $\mathfrak{A}'$, extend the 1-path in $\mathfrak{A}'$ to a 2-path and add a new 1-path). □

Theorem 3.11. *There exists a structure* $\mathfrak{A}$ *which is computably categorical, relatively computably categorical above* $\mathbf{0}''$ *(and thus relatively* Δ^0_2*-categorical above* $\mathbf{0}'$*), but* $\mathfrak{A}$ *is* *not relatively Δ^0_2-categorical.*

Proof. Again our structure is a graph.

Construction:

Again begin with an ω-spine. Coming off each vertex in the spine, attach two cliques, one larger than the other. Since the formulae in our Scott families are Σ^0_2, a formula may appear, in a Σ^0_2 fashion, to hold of an element. When a formula from Scott family X_n appears to hold of an element from each clique attached to the nth element, we stop growing the cliques. When it ceases to appear to hold of both elements, we resume growing both cliques, always maintaining one larger than the other. We push on the isomorphisms to ensure computable categoricity.

Verification:

Computable categoricity is by the standard isomorphism pushing.

$\mathbf{0}''$ can tell the sizes of the cliques attached to the nth element, including possibly infinite. If infinite, either can map to either. If finite, simply wait until the correct number of elements have appeared and then map.

It is not relatively Δ^0_2-categorical because every formally Σ^0_2 Scott family is defeated. If some sentence forever describes elements of both cliques, then it fails, since the cliques have different finite sizes. If no sentence describes them both, then since they are both infinite, they are in the same orbit, and thus the family has failed. □

Theorem 3.12. *There exists a structure $\mathfrak{A}$ which is computably categorical, relatively Δ^0_2-categorical, and not relatively computably categorical above any degree* $\mathbf{d}$.

Proof. Again our structure is a directed graph.

Construction:

Let $\langle i, j \rangle$ be the standard pairing function. The basic strategy is to create a vertex x_i with loops of size $\langle 2i, j \rangle$, and simultaneously to create elements $y_{i,j}$, for $j \in \omega$. $y_{i,j}$ will have loops of size $\langle 2i, n \rangle$ for each $n \leq j$, and also a loop of size $\langle 2i+1, j \rangle$.

Thus the basic strategy takes the form:

1. Choose a unique i and set $j = 0$.
2. Create the element x_i with a loop of size $\langle 2i, 0 \rangle$
3. Attach a loop of size $\langle 2i, j+1 \rangle$ to x_i.
4. Create the element $y_{i,j}$ with all appropriate loops.
5. Increment j, return to step (3).

We place these strategies on a tree along with standard isomorphism pushing strategies.

Verification:

The structure is computably categorical because of standard pushing.

Consider the formulae $\phi_{i,j}(z) =$ "there exists a loop of size $\langle 2i+1, j \rangle$ attached to z" and $\psi_i(z) =$ "there exists a loop of size $\langle 2i, 0 \rangle$ attached to z, and for all $j \in \omega$, there does not exist a loop of size $\langle 2i+1, j \rangle$ attached to z." These are formally Σ_2 formulae which isolate $y_{i,j}$ and x_i, respectively, and they extend to a formally Σ_2 Scott family for $\mathfrak{A}$ in the natural fashion. Thus $\mathfrak{A}$ is relatively Δ^0_2-categorical.

On the other hand, any degree $\mathbf{d} \geq_T \mathbf{0}''$ is capable of determining the true path of the construction. Thus such a degree is capable of building $\mathfrak{B} \cong \mathfrak{A}$, with $\mathfrak{B}$ not isomorphic to $\mathfrak{A}$ via any $\mathbf{d}$-computable isomorphism: for a given $\mathbf{d}$-computable partial function ϕ, choose an $x_i \in \mathfrak{A}$ which does grow to be infinite. Wait until ϕ converges on x_i, and then arrange that $\phi(x_i)$ is not part of an infinite component by making it isomorphic to a $y_{i,j}$ for large j. Thus the structure is not relatively computably categorical above $\mathbf{d}$.

This suffices because if a structure is relatively computably categorical above some degree $\mathbf{a}$, then it immediately follows that it is relatively computably categorical above any $\mathbf{d} \geq_T \mathbf{a}$. In particular, consider $\mathbf{d} = \mathbf{a}'' \geq_T \mathbf{0}''$. $\square$

Chapter 4

Limitwise Monotonic Functions

4.1 Basics

Definition 4.1. A function F is *limitwise monotonic* if there is a computable approximation function $f(\cdot,\cdot)$ such that, for all x,

(i) $F(x) = \lim_s f(x,s)$.

(ii) For all s, $f(x,s) \leq f(x,s+1)$.

A set S is *limitwise montonic* if it is the range of a limitwise monotonic function.

Definition 4.2 (Kach and Turetsky [9])**.** A function $F : \mathbb{Q} \to \omega$ is *support (strictly) increasing* if $F(q_1) \leq F(q_2)$ ($F(q_1) < F(q_2)$) whenever $q_1 < q_2$ and $F(q_1), F(q_2) > 0$, the range of F is unbounded, and the support of F has order type ω.

A function $F : \mathbb{Q} \to \omega$ is *support (strictly) increasing limitwise monotonic on* $\mathbb{Q}$ if it is support (strictly) increasing and there is a computable approximation function $f : \mathbb{Q} \times \omega \to \omega$ such that $F(q) = \lim_s f(q,s)$ and $f(q,s) \leq f(q,s+1)$.

The intuition here is that most $F(q)$ will be zero, but once we see $F(q) > 0$ at some stage (when $f(q,s) > 0$), then we "know" its relationship with all those q' with $F(q') > 0$.

We obtain the relativized notion *support (strictly) increasing* $\mathbf{0}'$*-limitwise monotonic on* $\mathbb{Q}$ by allowing the approximation function to be $\mathbf{0}'$-computable instead of merely computable.

The following useful lemma is easily proved:

Lemma 4.3. *There is a computable enumeration $\{f_i(\cdot,\cdot)\}_{i\in\omega}$ of total computable functions f satisfying $f(x,s) \leq f(x,s+1)$ for all x, s, and such that every limitwise monotonic function F is the limit of some f_i.*

Similar results hold for support increasing and support strictly increasing.

Definition 4.4. The *strong η-representation* of a set $S = \{n_0 < n_1 < n_2 < \dots\}$ is the linear order

$$\eta + n_0 + \eta + n_1 + \eta + n_2 + \dots .$$

A set is said to have a *computable strong η-representation* if its strong η-representation has a computable presentation.

Definition 4.5. A degree $\mathbf{a}$ is *totally limitwise monotonic* if every set $B \leq_T \mathbf{a}$ is a limitwise monotonic set.

4.2 A Separation Result

Kach and Turetsky introduced the notions of support (strictly) increasing $\mathbf{0}'$-limitwise monotonic on $\mathbb{Q}$ in the hope of classifying those sets with a computable strong η-representation.

It is easily seen that every support strictly increasing $\mathbf{0}'$-limitwise monotonic on $\mathbb{Q}$ set has a computable strong η-representation, while every set with a computable strong η-representation is support increasing $\mathbf{0}'$-limitwise monotonic on $\mathbb{Q}$. However, Frolov and Zubkov ([7]) and Kach and Turetsky ([9]) have shown that the second implication does not reverse, while we show here that the first does not.

Theorem 4.6 (Turetsky)**.** *There is a set S with a computable strong η-representation that is not support strictly increasing $\mathbf{0}'$-limitwise monotonic on $\mathbb{Q}$.*

Proof. Let $\{f_i(x,s)\}_{i\in\omega}$ be an enumeration of candidate total $\mathbf{0}'$-computable monotonic approximations on $\mathbb{Q}$ (as in Lemma 4.3). By the Limit Lemma, let $\{\hat{f}_i(x,s,t)\}_{i\in\omega}$ be an enumeration of computable approximations to f_i so that $f_i(x,s) = \lim_t \hat{f}_i(x,s,t)$. Note that since the f_i are total, the limit $\lim_t \hat{f}_i(x,s,t)$ will always converge to a finite limit.

We construct a computable presentation of a strong η-representation and let S be the set represented. We meet the following requirements:

$$\mathcal{R}_i : \text{The set } S \text{ is not the range of } F_i.$$

The strategy to assure $\mathcal{R}_i$ hinges on the fact that support strictly increasing limitwise monotonic functions cannot cope with two blocks in a strong η-representation merging. This fact is exploited to force a column to infinity.

Strategy for $\mathcal{R}_i$: Let $<_{\mathbb{Q}}$ be the natural ordering on $\mathbb{Q}$. The current stage will be denoted by t.

1. Choose a large number n_0 and create blocks B_0 and B of sizes $n_0 - 1$ and n_0 in $\mathcal{L}$ at an appropriate location. Restrain other strategies from changing these blocks.

2. Wait for a (least) pair $\langle x, u_0\rangle$ to appear with $\hat{f}_i(x, u_0, t) = n_0$.

3. Wait for a (least) pair $\langle x_0, s_0\rangle$ to appear with $\hat{f}_i(x_0, s_0, t) = n_0 - 1$ and $x_0 <_{\mathbb{Q}} x$.

4. Merge B_0 and B and any existing larger blocks into a single block of some size m_0 and release any restraint on this block. Restrain any blocks from forming of sizes between $n_0 - 1$ and m_0.

5. Wait for an $s_0' > s_0$ with $\hat{f}_i(x_0, s_0', t) = m_0'$ for some $m_0' \geq m_0$. If more than one such s_0' exist, choose the least.

6. Release the restraint created at Step 4.

7. Wait for a $u_1 > u_0$ with $\hat{f}_i(x, u_1, t) = n_1$ for some $n_1 > m_0$ with n_1 the size of a block in $\mathcal{L}$.

8. Create a block B_1 of size $n_1 - 1$ and restrain other strategies from changing this block or the block found in the previous step. Return to Step 3 with n_1 instead of n_0.

Note that our actions in Step 4 and Step 8 can be undone — we can resume densifying the interval between B_0 and B to separate the blocks, and we can densify the block B_1 to destroy it. Indeed, this capacity is essential, since there will be times we will need to roll back the construction to an earlier point. If, on some pair we chose, $\hat{f}_i$ changes its value, we return to the step at which we chose it, undoing all work done in the interim.

Thus, if at some stage t, $\hat{f}_i(x, u_0, t) \neq n_0$, we roll back the construction to Step 2. If at some stage t, $\hat{f}_i(x_j, s_j, t) \neq n_j - 1$, we roll back the construction to Step 3 in the jth loop. If at some stage t, $\hat{f}_i(x_j, s_j', t) \neq m_j'$, we roll back the construction to Step 5 in the jth loop, reestablishing the appropriate restraint. If at some stage t, $\hat{f}_i(x, u_j, t) \neq n_j$ (for $j > 0$), we roll back the construction to Step 7 in the jth loop.

Outcomes for $\mathcal{R}_i$: There are several possible outcomes for the strategy:

2: The strategy is infinitely often at Step 2, either because it waits at this step forever, or because it is infinitely often rolled back to this step. In either case, n_0 does not appear in the range of F_i but does appear as a block size in $\mathcal{L}$, and thus F_i does not enumerate S.

$\langle \mathbf{3}, \mathbf{j} \rangle$: The strategy is infinitely often at Step 3 in the jth loop, either because it waits at this step forever, or because it is infinitely often rolled back to this step. Further, none of outcomes **2**, $\langle \mathbf{3}, \mathbf{j'} \rangle$, $\langle \mathbf{5}, \mathbf{j'} \rangle$ or $\langle \mathbf{7}, \mathbf{j'} \rangle$ with $j' < j$ apply. In this case, $n_j - 1$ does not appear in the range of F_i but does appear as a block size in $\mathcal{L}$, and thus F_i does not enumerate S.

$\langle \mathbf{5}, \mathbf{j} \rangle$: The strategy is infinitely often at Step 5 in the jth loop, either because it waits at this step forever, or because it is infinitely often rolled back to this step. Further, none of outcomes **2**, $\langle \mathbf{3}, \mathbf{j'} \rangle$ with $j' \leq j$, or $\langle \mathbf{5}, \mathbf{j'} \rangle$ or $\langle \mathbf{7}, \mathbf{j'} \rangle$ with $j' < j$ apply. In this case, if $F_i(x_j)$ converges, then $F_i(x_j)$ is between $n_j - 1$ and m_j. However, S will have no element between $n_j - 1$ and m_j, and thus F_i does not enumerate S.

$\langle \mathbf{7}, \mathbf{j} \rangle$: The strategy is infinitely often at Step 7 in the jth loop, either because it waits at this step forever, or because it is infinitely often rolled back to this step. Further, none of outcomes **2**, $\langle \mathbf{3}, \mathbf{j'} \rangle$ or $\langle \mathbf{5}, \mathbf{j'} \rangle$ with $j' \leq j$, or $\langle \mathbf{7}, \mathbf{j'} \rangle$ with $j' < j$ apply. Then if $F_i(x)$ converges, it does so to a value not contained in S. Thus F_i does not enumerate S.

∞: The strategy spends only finitely many stages at every step in every loop. Since $F_i(x) \geq n_j$ for all j, and $n_j < m_j < n_{j+1}$, $F_i(x)$ diverges.

The Tree: We order the outcomes of a strategy by:

$$\mathbf{2} < \langle \mathbf{3}, \mathbf{0} \rangle < \langle \mathbf{5}, \mathbf{0} \rangle < \langle \mathbf{7}, \mathbf{0} \rangle < \langle \mathbf{3}, \mathbf{1} \rangle < \langle \mathbf{5}, \mathbf{1} \rangle < \langle \mathbf{7}, \mathbf{1} \rangle < \cdots < \infty$$

As usual for infinite injury arguments, the *true outcome* of a strategy is the limit infimum of the outcomes.

We arrange the strategies on a tree in the usual fashion. When a strategy τ is rolled back, we also roll back the work done by any strategies ρ directly below τ.

If strategy ρ is below some non-∞ outcome of strategy τ, the strategy ρ chooses a large n_0 and works with values larger than those used by τ. It is possible that ρ will be injured by a later merge step of τ. However, if we return to ρ, it will mean we have rolled back τ to before the merger, thus healing the injury to ρ.

If strategy ρ is below the ∞ outcome of strategy τ, the strategy ρ waits for the restraint of τ to move to a sufficiently late interval that there is sufficient room for ρ to work with values beneath the restraint. It chooses its n_0 smaller than the restraint of τ, but larger than the current size of any blocks which existed when ρ was initialized. When ρ wishes to perform a merger, it waits until τ reaches a Step 6. It then performs the merger as described, including merging larger blocks that τ previously used. If at some later point τ is rolled back, the strategy ρ is rolled back with it.

If ρ is below the infinite outcome of τ, it is possible that τ will violate the restraint of ρ (if τ's n_j is ρ's m_k). In this case, ρ waits until τ performs a merger, and then reassigns m_k to the value of this new block (so ρ's m_k is τ's m_j). Barring roll back, τ will never again violate this restraint.

In this fashion, strategies respect the restraints imposed by strategies directly above them in the tree. Strategies pay no attention to restraints of any other strategies.

Verification: Define the true path inductively using the limit infimum of the temporary outcomes.

Claim 4.7. *If τ is along the true path, and τ is active at stage t and has a restraint at stage t, then that restraint is not currently violated by some ρ directly below τ.*

Proof. If ρ is below some finite outcome of τ, it creates blocks of size larger than the restraint of τ. If ρ is below the infinite outcome of τ, it respects the restraint of τ as discussed above. □

Claim 4.8. *If τ is along the true path, and τ is active at stage t and has a restraint at stage t, then that restraint is not currently violated by some ρ off the true path.*

Proof. Note that the restraint is not violated at the stage it is originally imposed.

Assume ρ is not directly below τ, as that case is handled above.

If the true path follows a finite outcome at the first place it and ρ differ, and ρ is to the right of the true path, then any activity by ρ between the stage at which the restraint is imposed and the current stage has been rolled back.

If the true path follows a finite outcome at the first place it and ρ differ, and ρ is to the left of the True path, then ρ cannot act between the stage at which the restraint is imposed and the current stage (as in order for it to act, τ would have to be rolled back, removing the restraint).

If the true path follows an infinite outcome at the first place it and ρ differ, then let σ be the meet of τ and ρ. Then ρ created blocks above the restraint of σ, while τ imposes its restraint beneath that of σ. □

Claim 4.9. *If τ is along the true path, and τ imposes a restraint, there will come a stage t when either τ will be rolled back to before it imposed this restraint, τ will release* *this restraint and this release will never be rolled back, or the restraint will never be violated after stage t.*

Proof. Suppose that the restraint is neither rolled back nor released by τ. Then τ will wait until the σ above it stop violating the restraint. The strategy σ can only violate the restraint of τ if τ extends the infinite outcome of σ, and if σ has infinite final outcome, it can only be rolled back to any given step finitely many times. Thus, eventually, σ will never again violate the restraint of τ. Since no other strategies are capable of violating the restraint of τ, the restraint is never again violated. □

Claim 4.10. *For any block created in $\mathcal{L}$, the limit infimum of its size is finite.*

Proof. Let B be some block created by some strategy τ.

Suppose ρ is some other strategy. Let σ be ρ meet τ. In order for ρ to affect B, either ρ is σ or ρ is below the infinite outcome of σ, and either τ is σ or τ is below the finite outcome of σ. But by our construction of how strategies below an infinite outcome behave, ρ must have been initialized before B was created.

Thus there are only finitely many ρ that can affect B. Further, barring roll back, each strategy will only affect a given block finitely many times. Thus either one of these strategies is infinitely often rolled back, in which case B is constantly returned to a given finite size, or the size of B stabilizes. □

Claim 4.11. *There are blocks of arbitrarily large size in $\mathcal{L}$.*

Proof. Let τ be a strategy along the true path being initialized at stage t such that this initialization will never be rolled back. During initialization, τ creates a large block. Since τ will never have its initialization rolled back, this block will never be destroyed. It may be grown into a larger block, but by the above, some large block will result. Thus $\mathcal{L}$ has arbitrarily large blocks. □

Claim 4.12. *Each strategy along the true path meets its requirement.*

Proof. Immediate from construction. □

This completes the proof. □

4.3 A Totally Limitwise Monotonic Degree

We prove the following theorem:

Theorem 4.13. *There exists a noncomputable c.e. degree* $\mathbf{a}$ *such that for every set* $B \leq_T \mathbf{a}$*,* B *is limitwise monotonic.*

This theorem was later strengthened in by the author with Downey and Kach [4]:

Theorem 4.14 (Downey, Kach, Turetsky)**.** *A computably enumerable degree* $\mathbf{a}$ *is totally limitwise monotonic if and only if* $\mathbf{a}$ *is non-high.*

Proof of Theorem 4.13. We construct a c.e. set A and computable functions $f_j(\cdot,\cdot)$. The f_j will be nondecreasing in the second coordinate and will witness that A is totally limitwise monotonic as follows: for Turing functional Φ_j, if Φ_j^A is an infinite set, then $F_j(\cdot) = \lim_s f_j(\cdot, s)$ will be total and range $F_j = \Phi_j^A$.

We thus must meet the following three sorts of requirements:

$$\begin{array}{rc} \mathcal{P}_i : & A \neq \overline{W_i} \\ \mathcal{R}_j : & \Phi_j^A \subseteq \text{range } F_j \\ \mathcal{N}_{k,x} : & |\Phi_j^A| = \infty \Rightarrow \lim_t f_k(x,t) < \infty \ \&\ \lim_t f_k(x,t) \in \Phi_j^A. \end{array}$$

Strategy for $\mathcal{P}_i$*:*

Our strategy here is standard: choose a large element y and keep y out of A. When y enters W_i, enumerate y into A.

Strategy for $\mathcal{R}_j$:

Whenever an element z appears in $\Phi_{j,s}^{A_s}$ that is not in the range of $f(\cdot, s)$, we choose a large x and define $f_j(x, s+1) = z$.

Strategy for $\mathcal{N}_{k,x}$:

Let s_0 be the stage at which this strategy was initialized, and let $z_0 = f_k(x, s_0)$. If $|\Phi_j^A| = \infty$, then there must eventually be a $z_1 \in \Phi_{j,s}^{A_s}$ with $z_1 \geq z_0$. When this occurs, restrain $A \upharpoonright \varphi(z_1)$ and define $f_k(x, s+1) = z_1$.

Construction:

We arrange the $\mathcal{P}_i$ and $\mathcal{N}_{k,x}$ requirements on a priority tree in the usual fashion. The $\mathcal{R}_j$ requirements do not go on the tree.

We define $f_j(x, 0) = 0$ for every j, x. At the end of stage s, if $f_j(x, s+1)$ has not been defined by some $\mathcal{R}_j$ or $\mathcal{N}_{j,x}$ strategy, define $f_j(x, s+1) = f_j(x, s)$.

Verification:

Claim 4.15. *f_j is a total computable function which is nondecreasing in the second coordinate.*

Proof. Immediate from the construction. □

Claim 4.16. *$\Phi_j^A \subseteq$ range F_j.*

Proof. Suppose $z \in \Phi_j^A$ with use $\varphi(z)$. Choose a stage s such that $A_s \upharpoonright \varphi(z)$ has converged, and $\Phi_{j,s}^A(z) \downarrow$. Then at this stage, if there is not already an x such that $f_j(x, s) = z$, a new x will be chosen for this purpose. For all $s' > s$, $f_j(x, s') = z$, and thus $F_j(x) = z$. □

Claim 4.17. *Every strategy is injured only finitely many times.*

Proof. By induction on the priority of the strategy. □

Claim 4.18. *Every $\mathcal{P}_j$ and $\mathcal{N}_{k,x}$ strategy meets its requirement.*

Proof. Immediate from the construction. □

This completes the proof. □

Chapter 5

Intrinsically Complete Relations

This chapter is motivated by the following result.

Theorem 5.1 (Downey, Moses [6])**.** *There is a computable linear order L such that the successivity relation on L is intrinsically Δ^0_2-complete. That is, the degree spectrum of the successivity relation on L is precisely $\{\mathbf{0}'\}$.*

We extend the above result by introducing higher complexity natural relations and constructing linear orders on which they are intrinsically Δ^0_α-complete for the natural α. The proof involves a metatheorem of Ash, which we review.

5.1 Relations

We recall that for a linear order L, the *condensation of* L is defined to be $L/\sim$, where

$$x \sim y \leftrightharpoons |\{z : x <_L z <_L y\}| < \omega.$$

The *α-condensation of* L, denoted $L^{(\alpha)}$, is then defined inductively.

Definition 5.2. Given a linear order L, denote by $\mathrm{Succ}(L)$ the set

$$\{\{a, b\} : a, b \in L \text{ and } \{a, b\} \text{ is a successivity in } L\}.$$

Definition 5.3. Given a linear order L, define the binary relation S_α by

$$S_\alpha(x,y) \leftrightharpoons \{[x]_{L^{(\alpha)}}, [y]_{L^{(\alpha)}}\} \in \mathrm{Succ}(L^{(\alpha)}).$$

Definition 5.4. Given a linear order L, define the binary relation I_α by

$$I_\alpha(x,y) \leftrightharpoons [x]_{L^{(\alpha)}} = [y]_{L^{(\alpha)}}.$$

Definition 5.5. Given a linear order L, define the binary relation D_α by

$$D_\alpha(x,y) \leftrightharpoons ([x]_{L^{(\alpha)}}, [y]_{L^{(\alpha)}})^{(L^{(\alpha)})} \text{ is infinite dense without endpoints.}$$

Note that if L is computable, then S_α is $\Delta^0_{2\alpha+2}$, while I_α is $\Delta^0_{2\alpha+1}$ and D_α is $\Delta^0_{2\alpha+3}$.

5.2 Metatheorem

We introduce the necessary terminology and state without proof the metatheorem. For a proof, see Ash and Knight [2].

Definition 5.6. Let L and U be sets. An *alternating tree* on L and U is a tree P consisting of non-empty finite alternating sequences $\ell_0 u_1 \ell_1 u_2 \ell_2 \ldots$, where $\ell_i \in L$ and $u_i \in U$.

Definition 5.7. For P an alternating tree on L and U, an *instruction function* for P is a function q from the set

$$\{\tau \in P : |\tau| = 2n+1\}$$

to U, such that if $q(\tau) = u$, then $\tau u \in P$.

Definition 5.8. For P an alternating tree on L and U, and q an instruction function for P, a *run* of (P, q) is a path

$$\pi = \ell_0 u_1 \ell_1 u_2 \ell_2 \dots$$

such that $\pi \upharpoonright m \in P$ for all m, and $\pi(2m+1) = q(\pi \upharpoonright 2m+1)$.

Definition 5.9. An α-system is a structure of the form

$$(L, U, \ell_0, P, E, (\leq_\beta)_{\beta<\alpha}),$$

where L and U are c.e. sets, $\ell_0 \in L$, P is a c.e. alternating tree on L and U, all sequences in P begin with the element ℓ_0, E is a computable function $E : L \to [\omega]^{<\omega}$ (where $E(\ell)$ is a canonical index for a finite set), and the $\leq_\beta$ are uniformly c.e. binary relations on L satisfying the following:

1. $\leq_\beta$ is reflexive and transitive.
2. For $\gamma < \beta$, $\ell \leq_\beta \ell' \Rightarrow \ell \leq_\gamma \ell'$.
3. $\ell \leq_0 \ell' \Rightarrow E(\ell) \subseteq E(\ell')$.
4. If $\tau\ell^0 u \in P$, and
$$\ell^0 \leq_{\gamma_0} \ell^1 \leq_{\gamma_1} \cdots \leq_{\gamma_{k-1}} \ell^k,$$
for $\alpha > \gamma_0 > \gamma_1 > \cdots > \gamma_k$, then there exists ℓ^* such that $\tau\ell^0 u\ell^* \in P$, and $\ell^i \leq_{\gamma_i} \ell^*$, for all $i \leq k$.

We extend E to paths through P by taking unions, i.e, defining

$$E(\pi) = \bigcup_{i\in\omega} E(\pi(2i+1)).$$

Theorem 5.10 (Ash). *Let $(L, U, \ell_0, P, E, (\leq_\beta)_{\beta<\alpha})$ be an α-system. Then for any Δ^0_α-computable instruction function q, there is a run π of (P, q) such that $E(\pi)$ is c.e., and a c.e. index can be found uniformly for indices for q and the α-system.*

5.3 Back-and-Forth Relations

We introduce a further concept that will be necessary for the main result. Although the notions are quite general, applying to any collection of computable structures (and can be found in Ash and Knight [2]), we describe them for the special case of linear orders.

Definition 5.11. We define the *standard back-and-forth relations* by recursion. Let A and B be linear orders, and $\overline{a} \in A$, $\overline{b} \in B$:

We define $(A, \overline{a}) \leq_0 (B, \overline{b})$ if and only if $|\overline{a}| \leq |\overline{b}|$, and the map $a_i \mapsto b_i$ is an embedding of $\overline{a}$ into $\overline{b}$ (as finite suborders of A and B).

For $\alpha > 0$, we define $(A, \overline{a}) \leq_\alpha (B, \overline{b})$ if and only if $|\overline{a}| \leq |\overline{b}|$, and for each $\overline{d} \in B$ and each $\beta < \alpha$, there exists $\overline{c} \in A$ such that $(B, \overline{bd}) \leq_\beta (A, \overline{ac})$.

We will use the following proposition in the next section. For a proof, see [2].

Proposition 5.12. *For any computable ordinal α, there exists a uniformly computable sequence $\{C_n, \overline{c}_n\}_{n\in\omega}$ of linear orders where each C_n has order type $\omega^\alpha \cdot n$, and $\overline{c}_n$ is the n-tuple of "first elements" from copies of ω^α in C_n. Further, for $\beta < \alpha$, the standard back-and-forth relations $\leq_\beta$ on pairs $(C_n, \overline{a})$ are uniformly c.e. in β.*

5.4 Result

Theorem 5.13. *For any computable ordinal α, there is a computable linear order L such that I_α on L is intrinsically $\Delta^0_{2\alpha+1}$-Turing complete, S_α is intrinsically $\Delta^0_{2\alpha+2}$-Turing* *complete and D_α is intrinsically $\Delta^0_{2\alpha+3}$-Turing complete.*

It suffices to prove the following three lemmas:

Lemma 5.14. *For any computable ordinal α, there is a computable linear order L such that I_α on L is intrinsically $\Delta^0_{2\alpha+1}$-Turing complete.*

Lemma 5.15. *For any computable ordinal α, there is a computable linear order L such that S_α on L is intrinsically $\Delta^0_{2\alpha+2}$-Turing complete.*

Lemma 5.16. *For any computable ordinal α, there is a computable linear order L such that D_α on L is intrinsically $\Delta^0_{2\alpha+3}$-Turing complete.*

Proof of Lemma 5.15. Let $\{C_i, \overline{c}_i\}_{i\in\omega}$ be the sequence from the previous section.

Let $\{B_i\}_{i\in\omega}$ be an effective listing of all computable linear orders. We build L to have the form

$$w^\alpha \cdot (\eta + 3 + A_0 + 4 + A_1 + 5 + A_2 + \dots),$$

where each A_j contains no block of size greater than two. Thus if $B_i \cong L$, the image of $\omega^\alpha \cdot ((3+i) + A_i + (4+i))$ is uniquely defined in B_i, and can be identified by the unique blocks of size $\omega^\alpha \cdot (3+i)$ and $\omega^\alpha \cdot (4+i)$.

We build each A_i to ensure that if $B_i \cong L$, then $S_\alpha(B_i)$ restricted to the image of $\omega^\alpha \cdot A_i$ is $\Delta^0_{2\alpha+2}$-Turing complete.

Definition of U:

Let

$$\begin{aligned} U = \{\langle \{w_i\}_{i<n}, \{X_i\}_{i<n}, &\{R_i\}_{i<n}, \{S_i\}_{i<n}, \sigma\rangle : \\ & w_i \text{ is a finite linear order,} \\ & X_i \subseteq [w_i]^2, \\ & \{a,b\},\{c,d\} \in X_i \Rightarrow a,b \leq_{w_i} c,d \text{ or } a,b \geq_{w_i} c,d, \\ & R_i, S_i \subseteq w_i, \\ & |R_i| = 3+i \text{ and } |S_i| = 3, \text{ or } R_i = S_i = \emptyset, \\ & \text{every element of } R_i \text{ is to the left of every element of } S_i, \\ & \{a,b\} \text{ is an adjacency in } R_i \Rightarrow \{a,b\} \in X_i, \\ & \{a,b\} \text{ is an adjacency in } S_i \Rightarrow \{a,b\} \in X_i, \\ & \sigma \in 2^{<\omega}.\} \end{aligned}$$

Let us pause a moment to give some intuition for what these tuples are. We wish to diagonalize against all B_i, and so each w_i will be B_i at some stage s.

R_i will be our guess for the left separator in B_i (the $\omega^\alpha \cdot (3+i)$ block). It will not be the full block, but rather a single point from each copy of ω^α. Similarly, S_i will be our guess for the right separator in B_i—actually just the leftmost three elements of it (a point each from the leftmost three ω^α copies in the $\omega^\alpha \cdot (4+i)$ block of B_i). We describe these as guesses, because they will not necessarily be correct at first—our instruction function is one jump too weak to be able to compute these blocks. For example, when searching for the $(\omega^\alpha \cdot 7)$-block, we may mistake the $(\omega^\alpha \cdot 9)$-block for it. However, we will eventually realize this mistake and move left in search of the correct block. This will eventually settle on the correct block (assuming B_i has the correct form).

σ will be an initial segment of the set we wish to code into $S_\alpha(B_i)$—in this case, $\mathbf{0}^{(2\alpha+2)}$.

X_i will be pairs in $S_\alpha(B_i)$; however, we cannot make X_i all of $S_\alpha(B_i)$, for a reason we now explain. Suppose $\{a,b\} \in S_\alpha(B_i)$. Then for any $\hat{a} \in [a]_{L(\alpha)}$, $\hat{b} \in [b]_{L(\alpha)}$, it is the case that $\{\hat{a},\hat{b}\} \in S_\alpha(B_i)$, and there are infinitely many such pairs $\hat{a},\hat{b}$. Suppose every computation we define in our reduction were to use only these pairs. Then the set computed could be computed from knowledge of a, b and $B_i^{(\alpha)}$, which would require only $\mathbf{0}^{(2\alpha+1)}$. For this reason, it is important that the pairs used in our computations span infinitely many α-condensation classes.

We achieve this with the third requirement above: no two pairs in X_i can partake in precisely the same α-condensation classes. If we think of $S_0(B_i^{(\alpha)})$ as a collection of equivalence classes on $S_\alpha(B_i)$, the third requirement requires that X_i contain no more than a single element from each equivalence class. In fact, X_i will contain exactly one element from each class.

Definition of L:

Let

$$
\begin{aligned}
L = \{\langle v, p, Y, \{T_i\}_{i<n+1}, &\{w_i\}_{i<n}, \{X_i\}_{i<n}, \{R_i\}_{i<n}, \{S_i\}_{i<n}, \{\Gamma_i\}_{i<n}, \{z_i\}_{i<n}, \sigma\rangle : \\
&v \text{ is a finite linear order}, \\
&p \text{ is a finite partial injection from } \omega \text{ to } C_{|v|}, \\
&Y \subseteq [v]^2, \\
&\{a,b\} \in Y \Rightarrow \{a,b\} \text{ is an adjacency in } v, \\
&T_i \subseteq v, \\
&|T_i| = 3 + i, \\
&\{a,b\} \text{ is an adjacency in } T_i \ \Rightarrow \{a,b\} \in Y, \\
&w_i \text{ is a finite linear order}, \\
&X_i \ \subseteq [w_i]^2, \\
&|X_i| < \omega, \\
&\{a,b\}, \{c,d\} \in X_i \Rightarrow a, b \leq_{w_i} c, d \text{ or } a, b \geq_{w_i} c, d, \\
&R_i, S_i \subseteq w_i, \\
&|R_i| = 3 + i \text{ and } |S_i| = 3, \text{ or } R_i = S_i = \emptyset, \\
&\text{every element of } R_i \text{ is to the left of every element of } S_i, \\
&\{a,b\} \text{ is an adjacency in } R_i \ \Rightarrow \{a,b\} \in X_i, \\
&\{a,b\} \text{ is an adjacency in } S_i \ \Rightarrow \{a,b\} \in X_i, \\
&\sigma \in 2^{<\omega}, \\
&|(Y)_i| = |\sigma| = n, \\
&|X_i^*| \leq n, \\
&\{a,b\}, \{b,c\} \in X_i^* \Rightarrow a = c, \\
&z_i \in \omega, \\
&\Gamma_i \text{ is a finite set of consistent computations}, \\
&\Gamma_i^{X_i^*} \text{ has a computation for every } m < |X_i^*|, \\
&\gamma_i(m) \text{ contains at least } m+1 \text{ many ones}, \\
&\text{the computations in } \Gamma_i \text{ check only positive information}, \\
&(\forall m > z_i)\ \Gamma_i^{X_i^*}(m) \neq \sigma(m) \Rightarrow \text{ Order Property 1 for } m, \\
&(\forall m > z_i)\ \Gamma_i^{X_i^*}(m) = \sigma(m) \Rightarrow \text{ Order Property 2 for } m.\}
\end{aligned}
$$

We explain some of the terminology in the above. Let

$$(Y)_i := \{\{a,b\} \in Y : T_i <_v a \text{ and } b <_v T_{i+1}\}.$$

Let

$$X_i^* := \{\{a,b\} \in X_i : R_i <_{w_i} a \text{ and } b <_{w_i} S_i\}.$$

If $R_i = S_i = \emptyset$, we let $X_i^* = \emptyset$.

Order Property 1 for m is the following: Consider x the finite linear order consisting of the first $(m+1)$-many elements of X_i^* (by Gödel number), with the ordering inherited from w_i, and y the finite linear order consisting of the first $(m+1)$-many elements of $(Y)_i$, with the ordering inherited from v. Our requirement on the ordering of the pairs in X_i, along with the fact that every element of $(Y)_i$ is an adjacency of v, justifies these orderings. As finite linear orderings of the same size, there is a unique isomorphism between them. Order Property 1 states that the first element (by Gödel number) of x maps to the $(m+1)$st element (by Gödel number) of y.

Order Property 2 for m is the following: Consider y the finite linear order consisting of the first $(m+1)$-many elements of $(Y)_i$ (by Gödel number), with the ordering inherited from v. Order Property 2 states that in y, the $(m+1)$st element (by Gödel number) is the immediate successor of the mth element.

As we did before, we pause a moment to give an intuition for what these tuples represent. Every element of the tuples in U recurs in the tuples of L, with the same meaning.

v will be a linear order we are building which will have the form $\eta + 3 + A_0 + 4 + A_1 + 5 + A_2 + \dots$. In other words, v will be the α-condensation of L. We make no claim that v will be a computable order; in fact, it will necessarily have jump $\mathbf{0}^{(2\alpha+2)}$.

p is used in constructing a computable $L \cong \omega^\alpha \cdot v$. For more details, see Ash & Knight Chapter 18, §4 [2].

T_i will be the ith separator in v (i.e., the $3+i$-block).

Γ_i will be the reduction we build witnessing $S_\alpha(B_i) \geq_T \mathbf{0}^{(2\alpha+2)}$. In fact, it will not compute $\mathbf{0}^{(2\alpha+2)}$, but rather some set which differs in only a finite number of elements.

As discussed before, R_i and S_i will not necessarily be correct at first. Any computations created while they are wrong cannot be trusted, nor can they necessarily be corrected. z_i tracks this fact by denoting the level below which our computations cannot be trusted. It will not increase while R_i and S_i remain constant.

Definition of the System:

Given $\ell \in V$, we define $E(\ell) = \mathrm{Diag}_{\mathrm{at}}(\mathrm{dom}(p)) \sqcup \Gamma_0 \sqcup \Gamma_1 \sqcup \ldots$. Here we think of $\mathrm{dom}(p)$ as having the ordering induced by $p^{-1}(C_{|v|})$.

Given $\ell, \ell' \in V$ and $\beta < 2\alpha + 1$, we define

$$\begin{aligned} \ell \leq_\beta \ell' \risingdotseq\ & E(\ell) \subseteq E(\ell'), \\ & \& \ v \subseteq v', \\ & \& \ (\omega^\alpha + C_{|v|}, \mathrm{range}(p)) \leq_\beta (\omega^\alpha + C_{|v'|}, \mathrm{range}(p')). \end{aligned}$$

We define

$$\begin{aligned}
\ell \leq_{2\alpha+1} \ell' \coloneqq\; & \ell \leq_{2\alpha} \ell', \\
& \& \; n \leq n', \\
& \& \; T_i = T_i', \\
& \& \; w_i \subseteq w_i', \\
& \& \; R_i \neq \emptyset \Rightarrow R_i' \neq \emptyset, \\
& \& \; R_i = R_i' \text{ or } R_i' \text{ is left of } R_i, \\
& \& \; S_i = S_i' \text{ or } S_i' \text{ is left of } S_i, \\
& \& \; [R_i = R_i' \neq \emptyset \text{ and } S_i = S_i' \neq \emptyset] \Rightarrow z_i = z_i', \\
& \& \; \text{The first } |X_i^*|\text{-many elements of } (Y)_i \text{ (by Gödel number) are in } (Y')_i, \\
& \& \; X_i \subseteq X_i', \\
& \& \; |X_i^*| = |X_i'^*| \Rightarrow (Y)_i \subseteq (Y')_i, \\
& \& \; \sigma \subseteq \sigma', \\
& \& \; (Y' \backslash Y) \cap [v]^2 = \emptyset.
\end{aligned}$$

We let $\ell_0 = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$, and let P consist of those finite alternating sequences $\ell_0 u_1 \ell_1 u_2 \ldots$ such that $\ell_k \in L$, $u_k \in U$, and if

$$\begin{aligned}
u_k &= \langle \{w'_{k,i}\}_{i<n'_k}, \{X'_{k,i}\}_{i<n'_k}, \{R'_{k,i}\}_{i<n'_k}, \{S'_{k,i}\}_{i<n'_k}, \sigma'_k \rangle \\
\ell_k &= \langle v_k, p_k, Y_k, \{T_{k,i}\}_{i<n_k+1}, \{w_{k,i}\}_{i<n_k}, \{X_{k,i}\}_{i<n_k}, \\
&\qquad \{R_{k,i}\}_{i<n_k}, \{S_{k,i}\}_{i<n_k}, \{\Gamma_{k,i}\}_{i<n_k}, \{z_{k,i}\}_{i<n_k}, \sigma_k \rangle
\end{aligned}$$

then the following hold:

1. $n_k = n'_k = k$, $w_{k,i} = w'_{k,i}$, $X_{k,i} = X'_{k,i}$, $R_{k,i} = R'_{k,i}$, $S_{k,i} = S'_{k,i}$ and $\sigma_k = \sigma'_k$.
2. $w_{k,i} \subseteq w_{k+1,i}$, $X_{k,i} \subseteq X_{k+1,i}$, $R_{k,i} \subseteq R_{k+1,i}$, $S_{k,i} \subseteq S_{k+1,i}$ and $\sigma_k \subseteq \sigma_{k+1}$.
3. $\ell_k \leq_{2\alpha+1} \ell_{k+1}$.
4. $w_{k,i}$ is an ordering of the first k many constants from the universe of B_i.
5. v_k is an ordering containing the first k many constants from A.
6. $|X^*_{k+1,i}| - |X^*_{k,i}| < 2$.
7. There exists $d_i, e_i \in v_{k+1} \backslash v_k$ with d_i the immediate successor of T_i and e_i the immediate predecessor of T_{i+1}.

8. If $a, b \in v_k$ but $\{a, b\} \notin Y_k$, then there exists $c \in v_{k+1}$ between a and b.
9. $k \subseteq \mathrm{dom}(p_k)$.
10. $\overline{c}_{|v_k|} \subseteq \mathrm{range}(p_k)$.
11. $f_{k+1}^{-1} \circ p_{k+1} \circ p_k^{-1} \circ f_k = i$. Here f_m is the (unique) isomorphism from v_m to $\overline{c}_{|v_m|}$ (since they are finite linear orders of the same cardinality), while i is the inclusion map from v_k to v_{k+1}. In particular, we require that $p_{k+1} \circ p_k^{-1}(\overline{c}_{|v_k|}) \subseteq \overline{c}_{|v_{k+1}|}$.

Claim 5.17. *$(L, U, \ell_0, P, E, (\leq_\beta)_{\beta<2\alpha+2})$ is a $(2\alpha+2)$-system.*

Proof. The only non-trivial condition is the final partial order condition.

Suppose $\tau\ell^0 u \in P$, and

$$\ell^0 \leq_{\beta_0} \ell^1 \leq_{\beta_1} \cdots \leq_{\beta_{m-1}} \ell^m,$$

for $2\alpha + 2 > \beta_0 > \cdots > \beta_m$.

The relations $\leq_\beta$ for $\beta < 2\alpha + 1$ concern only the maps p, and constructing a map p^* for a chain of such proceeds exactly as in Chapter 18, §4 of [2]. This allows us to restrict to the case $\ell^0 \leq_{2\alpha+1} \ell^1$.

Let

$$\begin{aligned} u &= \langle \{w_{u,i}\}_{i<n_u}, \{X_{u,i}\}_{i<n_u}, \{R_{u,i}\}_{i<n_u}, \{S_{u,i}\}_{i<n_u}, \sigma_u \rangle, \\ \ell^0 &= \langle v_0, Y_0, \{T_{0,i}\}_{i<n_0+1}, \{w_{0,i}\}_{i<n_0}, \{X_{0,i}\}_{i<n_0}, \\ &\qquad \{R_{0,i}\}_{i<n_0}, \{S_{0,i}\}_{i<n_0}, \{\Gamma_{0,i}\}_{i<n_0}, \{z_{0,i}\}_{i\in n_0}, \sigma_0 \rangle, \\ \ell^1 &= \langle v_1, Y_1, \{T_{1,i}\}_{i<n_1+1}, \{w_{1,i}\}_{i<n_1}, \{X_{1,i}\}_{i<n_1}, \\ &\qquad \{R_{1,i}\}_{i<n_1}, \{S_{1,i}\}_{i<n_1}, \{\Gamma_{1,i}\}_{i<n_1}, \{z_{1,i}\}_{i\in n_1}, \sigma_1 \rangle. \end{aligned}$$

We construct ℓ^* as follows:

Let $n_* = n_u = n_0 + 1$, $\{w_{*,i}\}_{i<n_*} = \{w_{u,i}\}_{i<n_u}$, $\{X_{*,i}\}_{i<n_*} = \{X_{u,i}\}_{i<n_u}$, $\{R_{*,i}\}_{i<n_*} = \{R_{u,i}\}_{i<n_u}$, $\{S_{*,i}\}_{i<n_*} = \{S_{u,i}\}_{i<n_u}$, $\sigma_* = \sigma_u$. For $i < n_0$, let $T_{*,i} = T_{0,i}$.

We create v_* by adding elements to v_1. We define Y_* by defining $(Y_*)_i$. We add n_0+3 additional elements to the far right of v_1 and let T_{*,n_0} consist of these new elements. For any two elements $a, b \in v_0$, if $\{a, b\} \notin Y_0$, we add an element between them if there is not already such an element.

We then consider each $i < n_*$ separately. There are three cases.

(Case 1.) Suppose $R_{u,i} \neq R_{0,i}$ or $R_{u,i} = \emptyset$. (The commonality of this case is that we may define $z_{*,i}$ as we please.)

Let f, g be new elements. We let $(Y_*)_i = \{\{f, g\}\} \cup (Y_0)_i$.

We add f and g to v_1 such that $(Y_*)_i$ satisfies Order Property 2. We add new least and greatest elements to the interval between T_i and T_{i+1}.

We let $\Gamma_{*,i} = \Gamma_{1,i}$. We choose $z_{*,i}$ larger than any m for which $\Gamma_{*,i}$ contains a computation.

(Case 2.) $R_{u,i} = R_{0,i} \neq \emptyset$ and $X_{0,i} = X_{u,i}$.

Let $z_{*,i} = z_{0,i}$.

Let f, g be new elements. We let $(Y_*)_i = \{\{f, g\}\} \cup (Y_0)_i$.

We add f and g to v_1 such that $(Y_*)_i$ satisfies Order Property 2. We add new least and greatest elements to the interval between T_i and T_{i+1}.

We let $\Gamma_* = \Gamma_1$.

(Case 3.) $R_{u,i} = R_{0,i} \neq \emptyset$ and $X_{0,i} \neq X_{u,i}$. Let $|X^*_{0,i}| = p$.

Let $z_{*,i} = z_{0,i}$.

Let $f_p, g_p, \ldots, f_{n_0}, g_{n_0}$ be new elements. We define $(Y_*)_i$ to contain the first p many elements of $(Y_0)_i$, along with $\{f_{p+1}, g_{p+1}\}, \ldots, \{f_n, g_n\}$.

If $\Gamma_1^{X_{u,i}}$ does not contain a computation for p, we let Γ_* be Γ_1 along with an additional computation correctly computing $\sigma_u(p)$ from $X_{u,i}$. Otherwise, we let $\Gamma_* = \Gamma_1$.

If $\Gamma_*^{X_{u,i}} = \sigma_u(p)$, we add f_p and g_p such that $(Y_*)_i$ satisfies Order Property 2. Otherwise, we add them such that $(Y_*)_i$ satisfies Order Property 1. We add f_m and g_m satisfying Order Property 2 for $p < m \leq n_0$. We add new least and greatest elements to the interval between T_i and T_{i+1}.

We let Y_* and v_* be as constructed in this fashion. We define

$$\ell^* = \langle v_*, p_*, Y_*, \{T_{*,i}\}_{i<n_*+1}, \{w_{*,i}\}_{i<n_*}, \{X_{*,i}\}_{i<n_*}, \\ \{R_{*,i}\}_{i<n_*}, \{S_{*,i}\}_{i<n_*}, \{\Gamma_{*,i}\}_{i<n_*}, \{z_{*,i}\}_{i\in n_*}, \sigma_*\rangle.$$

By construction, ℓ^* is precisely as required. □

Definition of the Instruction Function:

For $\tau \in P$ of length $2n-1$, let $q(\tau) = \langle \{w_i\}_{i<n}, \{X_i\}_{i<n}, \{R_i\}_{i<n}, \{S_i\}_{i<n}, \sigma\rangle$ where $\sigma = \emptyset^{(2\alpha+2)} \restriction n$ ($\emptyset^{(2\alpha+1)}$ for finite α), $w_i = B_i \restriction n$, X_i is the Gödel least choice set for $S_\alpha(B_i)$ restricted to w_i, R_i is the Gödel least choice set for the leftmost $(i+3)$-block in the α-condensation of B_i restricted to w_i, and S_i is the Gödel least choice set for the leftmost 3-block in the α-condensation of B_i restricted to w_i to the right of R_i.

Verification:

By the metatheorem, there is a run π of (P, q) such that $E(\pi)$ is c.e., and an index for $E(\pi)$ can be effectively found. Let L_π be the linear order whose atomic diagram is enumerated by $E(\pi)$, and let $\{\Gamma_{\pi,i}\}_{i\in\omega}$ be the Turing functionals enumerated by $E(\pi)$.

Claim 5.18. *Define $A_\pi = \bigcup v_k$ for the v_k along π. Then $L_\pi = \omega^\alpha \cdot A_\pi$.*

Proof. As in Chapter 18, §4 of [2]. □

Claim 5.19. *L_π has the form*

$$\omega^\alpha \cdot (\eta + 3 + A_0 + 4 + A_1 + 5 + A_2 + \dots),$$

where each A_i contains no block of size greater than two.

Proof. Immediate by construction. $\square$

Claim 5.20. *If B_i has the form*

$$\omega^\alpha \cdot (\eta + 3 + B_{i,0} + 4 + B_{i,1} + 5 + B_{i,2} + \dots),$$

then A_i has infinitely many successivities.

Proof. Clearly a pair $\{a, b\}$ is a successivity in A_i iff $\{a, b\} \in (Y_k)_i$ for cofinitely many k.

Because of the form of B_i, there exists a k_0 such that $R_{k,i} = R_{k_0,i}$ and $S_{k,i} = S_{k_0,i}$ for all $k > k_0$. We restrict our attention to $k > k_0$.

There are two possibilities. If $B_{i,i}$ has only finitely many successivities, then there is a k_1 with $X^*_{k_1,i}$ a full choice set for $S_\alpha(B_i)$. Then $X^*_{k,i} = X^*_{k_1,i}$ for all $k > k_1$. The $(Y_k)_i$ thus form a strictly increasing chain for $k > k_1$, and thus A_i has infinitely many successivities.

If $B_{i,i}$ has infinitely many successivities, then for any m, there is a k_1 such that $|X^*_{k_1,i}| > m$. Then the first m many elements of $(Y_{k_1})_i$ will be elements of $(Y_k)_i$ for any $k > k_1$, and thus A_i will have at least m many successivities. Thus A_i has infinitely many successivities. $\square$

Claim 5.21. *If B_i has the form*

$$\omega^\alpha \cdot (\eta + 3 + B_{i,0} + 4 + B_{i,1} + 5 + B_{i,2} + \dots),$$

then the successivities of A_i are ordered with type $\omega + n$, $n + \omega^*$ or $\omega + \omega^*$ for some finite (possibly empty) n.

Proof. Because of the form of B_i, there exists a k_0 such that $R_{k,i} = R_{k_0,i}$ and $S_{k,i} = S_{k_0,i}$ for all $k > k_0$. We restrict our attention to $k > k_0$.

There are two cases. If $B_{i,i}$ has only finitely many successivities, let k_1 be such that $X^*_{k_1,i} = \mathrm{Succ}(B_{i,i})$. Let n be the number of successivities in $(Y_k)_i$ to the right of the newest successivity. Then every additional successivity will be added to A_i to satisfy Order Property 2, so the type of the successivities in A_i will be $\omega + n$.

If $B_{i,i}$ has infinitely many successivities, call a *true stage* a stage $k+1$ at which $X^*_{k+1,i} \neq X^*_{k,i}$. Let $|X^*_{k+1,i}| = m + 1$. Then let C_k consist of the mth element of $(Y_{k+1})_i$ and those elements of $(Y_k)_i$ to the left of the mth element of $(Y_{k+1})_i$, and let D_k consist of those elements of $(Y_k)_i$ to the right of the mth element of $(Y_{k+1})_i$. Note that at the next true stage, all successivities not in C_k or D_k will be removed, and new successivities will only be added between C_k and D_k. Thus $C = \bigcup_k C_k$ and $D = \bigcup_k D_k$ partition the successivities of A_i, and C is finite or ω, while D is finite or ω^*. □

Claim 5.22. *If B_i has the form*

$$\omega^\alpha \cdot (\eta + 3 + B_{i,0} + 4 + B_{i,1} + 5 + B_{i,2} + \dots),$$

and $B_{i,i}$ has infinitely many successivities, then $\Gamma^{S_\alpha(B)}_{\pi,i}$ is total.

Proof. Immediate from construction. □

Claim 5.23. *If $B_i \cong L_\pi$, then $\Gamma^{S_\alpha(B)}_{\pi,i} \equiv_{fin} \emptyset^{(2\alpha+2)}$ (or $\emptyset^{(2\alpha+1)}$ for finite α).*

Proof. Let $\{a, b\}$ be the first successivity of $B_{i,i}$ (by Gödel number). In $B_{i,i}$, there are either only n-many successivities to the left of $\{a, b\}$ or only n-many to the right, for some finite n. Without loss of generality, we consider the first case. Let $\{c, d\}$ be the successivity in A_i corresponding to $\{a, b\}$. Note that this is well-defined because $\omega + n$, $n + \omega^*$ and $\omega + \omega^*$ are all rigid.

Let k be a stage such $R_{k,i}$ and $S_{k,i}$ have converged and such that these n-many successivities have all appeared in $X^*_{k,i}$, and $|X^*_{k,i}| > z_{k,i}$, and $\{c, d\}$ has appeared in $(Y_k)^*$, and the n-many successivities to the left of $\{c, d\}$ have all appeared in $(Y_k)^*$. At a true stage $k'+1 > k$, let $m+1 = |X^*_{k'+1,i}|$. Then $\Gamma^{S_\alpha(B)}_{\pi,i}(m) = \Gamma^{X^*_{k'+1,i}}_{\pi,i}(m)$ must be correct, else an $(n+1)$st successivity would be placed immediately to the left of $\{c, d\}$, contradicting our choice of $\{c, d\}$. □

This completes the proof. □

Lemmas 5.14 and 5.16 use a similar construction. For Lemma 5.14, the A_i each have the form $\zeta \cdot \tau$, where ζ is the order-type of the integers, and τ is either $\omega + \omega^*$, $\omega + n$ or $n + \omega$.

For Lemma 5.16, the A_i resemble the A_i of Lemma 5.15, but the adjacencies are each replaced with a copy of ζ.

In both cases, the separators are of the form $\omega^* + \zeta \cdot (3 + i) + \omega$.

Proof of Theorem 5.13. Let L_0, L_1 and L_2 be the linear orders from Lemmas 5.14, 5.15 and 5.16. Let $L = L_0 + 1 + L_1 + 1 + L_2$. □

Bibliography

[1] C. J. Ash. Categoricity in hyperarithmetical degrees. *Ann. Pure Appl. Logic*, 34(1):1–14, 1987.

[2] C. J. Ash and J. Knight. *Computable structures and the hyperarithmetical hierarchy*, volume 144 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 2000.

[3] Krishna B. Athreya, John M. Hitchcock, Jack H. Lutz, and Elvira Mayordomo. Effective strong dimension in algorithmic information and computational complexity. In *STACS 2004*, volume 2996 of *Lecture Notes in Comput. Sci.*, pages 632–643. Springer, Berlin, 2004.

[4] Rodney G. Downey, Asher M. Kach, and Daniel Turetsky. Limitwise monotonic functions and applications. Submitted.

[5] Rodney G. Downey, Steffen Lempp, and Guohua Wu. On the complexity of the successivity relation in computable linear orderings. Submitted.

[6] Rodney G. Downey and Michael F. Moses. Recursive linear orders with incomplete successivities. *Trans. Amer. Math. Soc.*, 326(2):653–668, 1991.

[7] Andrey N. Frolov and Maxim V. Zubkov. Increasing η-representable degrees. Submitted.

[8] Sergei S. Goncharov. Autostability and computable families of constructivizations. *Algebra i Logika*, 14:392–408, 1975 (English translation).

[9] Asher M. Kach and Daniel Turetsky. Limitwise monotonic functions, sets, and degrees on computable domains. *J. Symbolic Logic*, 75(1):131–154, 2010.

[10] Bakhadyr Khoussainov and Richard A. Shore. Computable isomorphisms, degree spectra of relations, and Scott families. *Ann. Pure Appl. Logic*, 93(1-3):153–193, 1998. Computability theory.

[11] Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications*. Texts in Computer Science. Springer, New York, third edition, 2008.

[12] Jack H. Lutz. Dimension in complexity classes. In *15th Annual IEEE Conference on Computational Complexity (Florence, 2000)*, pages 158–169. IEEE Computer Soc., Los Alamitos, CA, 2000.

[13] Jack H. Lutz and Klaus Weihrauch. Connectivity properties of dimension level sets. *MLQ Math. Log. Q.*, 54(5):483–491, 2008.

[14] Elvira Mayordomo. A Kolmogorov complexity characterization of constructive Hausdorff dimension. *Inform. Process. Lett.*, 84(1):1–3, 2002.

[15] Jan van Mill. *The infinite-dimensional topology of function spaces*, volume 64 of *North-Holland Mathematical Library*. North-Holland Publishing Co., Amsterdam, 2001.