

A relation on $(\omega, <)$ of intermediate degree spectrum on a cone

Jad Damaj and Matthew Harrison-Trainor*

April 12, 2025

Abstract

We examine the degree spectra of relations on $(\omega, <)$. Given an additional relation R on $(\omega, <)$, such as the successor relation, the degree spectrum of R is the set of Turing degrees of R in computable copies of $(\omega, <)$. It is known that all degree spectra of relations on $(\omega, <)$ fall into one of four categories: the computable degree, all of the c.e. degrees, all of the Δ_2^0 degrees, or intermediate between the c.e. degrees and the Δ_2^0 degrees. Examples of the first three degree spectra are easy to construct and well-known, but until recently it was open whether there is a relation with intermediate degree spectrum on a cone. Bazhenov, Kalociński, and Wroclawski constructed an example of an intermediate degree spectrum, but their example is unnatural in the sense that it is constructed by diagonalization and thus not canonical, that is, which relation you obtain from their construction depends on which Gödel encoding (and hence order of enumeration) of the partial computable functions / programs you choose. In this paper, we use the “on-a-cone” paradigm to restrict our attention to “natural” relations R . Our main result is a construction of a natural relation on $(\omega, <)$ which has intermediate degree spectrum. This relation has intermediate degree spectrum because of structural reasons.

1 Introduction

Let \mathcal{A} be a mathematical structure such as a group, graph, or linear order. For this paper, we will be solely interested in the case where \mathcal{A} is the linear order $(\omega, <)$ though the definitions can be made in general. Let R be an additional relation on \mathcal{A} not in the signature of \mathcal{A} . Typical examples are the relation of linear independence on a vector space or the successor relation on a linear order.

What is the intrinsic complexity of R ? One way to measure this is to look at how the complexity of R behaves under isomorphisms. In particular, we consider all computable copies \mathcal{B} of \mathcal{A} (isomorphic presentations of \mathcal{A} where all of the functions and relations are computable) and look at the Turing degree of R in \mathcal{B} . The collection of all such Turing degrees is the degree spectrum of R . In other words, the degree spectrum measures the possible complexity of R while fixing the complexity of the presentation of the underlying structure.

*Harrison-Trainor was partially supported by the National Science Foundation under Grant DMS-2153823 / DMS-2419591. This work began while Damaj was an REU student at the University of Michigan funded under this grant.

Definition 1.1. Given a computable structure \mathcal{A} and a relation R on \mathcal{A} , we define the degree spectrum of R , $\text{DgSp}_{\mathcal{A}}(R)$, to be the set of Turing degrees

$$\{\text{deg}_T(\varphi(R)) : \mathcal{B} \text{ is a computable copy of } \mathcal{A} \text{ and } \varphi : \mathcal{A} \cong \mathcal{B}\}$$

i.e., the images of R in all computable copies of \mathcal{A} under all isomorphisms.

In this paper we follow a series of papers [DKMY09, Kno09, Wri18, HT18, BKW22, BK23, BKW24] studying the degree spectra of computable relations on the structure $(\omega, <)$. The successor relation S plays a particularly important role in this structure. There is one copy of $(\omega, <)$, the standard copy, where S is computable. In any other computable copy $\mathcal{L} = (L, \prec)$ we have the successor relation $S^{\mathcal{L}} \subseteq L^2$ on \mathcal{L} . $S^{\mathcal{L}}$ is always a co-c.e. set, and hence of c.e. degree, and in fact the degree spectrum of S is exactly the c.e. degrees.

In any computable copy \mathcal{L} there is a unique isomorphism $f_{\mathcal{L}} : \mathcal{L} \rightarrow (\omega, <)$, and the Turing degree of this isomorphism is exactly the Turing degree of $S^{\mathcal{L}}$. Given any other computable relation R on $(\omega, <)$, we obtain its image in \mathcal{L} by $R^{\mathcal{L}} = f(R)$, and so $\mathcal{R}^{\mathcal{L}} \leq_T f_{\mathcal{L}} =_T S^{\mathcal{L}} \leq_T \emptyset'$. For many relations R , we also always have $R^{\mathcal{L}} \geq_T S^{\mathcal{L}}$, so that the degree spectrum of R is the same as the degree spectrum of S , that is, all of the c.e. degrees. This is the case, for example, for the double-successor relation. Given $x < y$, we have that y is the successor of x if and only if there is $z > y$ such that z is the double-successor of x . Thus the degree spectrum of the double-successor relation—a d.c.e. relation—is only the c.e. degrees, rather than the d.c.e. degrees as one might expect. (A more general version of this argument shows that the degree spectrum of any intrinsically n -c.e. relation on $(\omega, <)$ will be the c.e. degrees; see Section 4 of [HT18].¹)

There are also examples of computable relations on $(\omega, <)$ whose degree spectra are only the computable degree (such as an empty relation, or the identity function) and all Δ_2^0 degrees (such as an infinite and co-infinite unary relation). Because $(\omega, <)$ is \mathcal{O}' -categorical, every degree spectrum of a computable relation on $(\omega, <)$ is contained within the Δ_2^0 degrees.

Wright showed that for any computable relation R on $(\omega, <)$, the degree spectrum is either just the computable degree, or must contain all of the c.e. degrees.² Thus no degree spectrum could be intermediate between the computable degree and the c.e. degrees. Wright left open the question of which degree spectra intermediate between the c.e. degrees and the Δ_2^0 degrees were possible, and in particular, he left open the question of the existence of a degree spectrum strictly intermediate between the c.e. degrees and the Δ_2^0 degrees.

In [BKW22], Bazhenov, Kalociński, and Wroclawski showed that there is a unary function whose degree spectrum is intermediate. However this relation is unnatural in the sense that it is built via a complicated priority argument and diagonalization. It is not canonical because what relation one obtains by the construction depends on, e.g., the particular choice of Gödel coding for c.e. sets. While computability theory is full of such unnatural counterexamples, such examples are unlikely to show up in the normal course of mathematics.

In this paper we consider degree spectra of *natural* relations on $(\omega, <)$. Of course what it means for a relation to be natural is not well-defined and so we use the “on a cone”

¹It was incorrectly stated in [Wri18] that there are intrinsically n -c.e. relations on $(\omega, <)$ whose degree spectra are exactly the n -c.e. degrees.

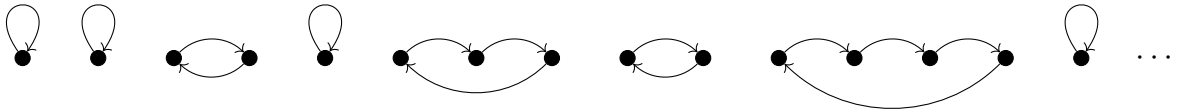
²Downey, Khousainov, Miller, and Yu [DKMY09] had previously shown this for unary relations.

formalism to capture this notion. This formalism originated with Martin’s conjecture (see [Mon19]) and there has been a recent program, first suggested by Montalbán, of studying computable structure theory on a cone (e.g., [Mon13, CHT17, HT18, AHTS21]). Degree spectra on a cone were first studied in the second author’s monograph [HT18], where relations on $(\omega, <)$ were specifically considered.³ There the second author asked the on-a-cone version of Wright’s question: Is there a relation on $(\omega, <)$ whose degree spectrum is intermediate on a cone? At the time an answer to Wright’s original question, not on a cone, was still not known. While [BKW22] resolved Wright’s question, it did not resolve the on-a-cone version: the degree spectrum of their relation is the c.e. degrees on a cone.

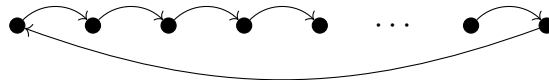
In this paper we resolve the on-a-cone version of Wright’s question:

Theorem 3.9. *There are relations on $(\omega, <)$ whose degree spectrum is strictly between the c.e. degrees and the Δ_2^0 degrees on a cone.*

In particular, there are computable examples where the cone is the trivial cone, that is, computable relations R on $(\omega, <)$ such that relative to any degree \mathbf{d} , the degree spectrum of R relative to \mathbf{d} is strictly between the \mathbf{d} -c.e. degrees and the degrees Δ_2^0 relative to \mathbf{d} . Such examples are natural relations of intermediate degree spectrum. To illustrate how natural these examples are, we describe our example. As was the example in [BKW22], our example will be a unary function f . An initial segment of f is as follows:



One can see that the domain of f is divided up into blocks, and in each block f is a loop. We write L_n for the loop of length n , i.e., for the block



with n nodes. Then f consists of the following blocks:

$$L_1 L_1 L_2 L_1 L_3 L_2 L_4 L_1 L_5 L_2 L_6 L_3 L_7 L_1 L_8 \dots$$

The pattern here is that the blocks in odd positions follow the pattern $L_1 L_2 L_3 L_4 \dots$ enumerating the natural numbers in increasing order, while the blocks in even positions $L_1 L_1 L_2 L_1 L_2 L_3 \dots$ are an enumeration of all of the natural numbers such that each number occurs infinitely many times. Thus every block appears infinitely many times, but any pair of blocks appears adjacent to each other at most once.

While we can describe our example simply, the example of [BKW22] does not have a simple description but is actually the result of a complicated priority construction. Moreover, what relation one gets from the priority construction depends on certain non-canonical choices that one makes, such as the choice of Gödel encoding of the partial computable functions. Finally, their example does not relativize, i.e., the computable relation R produced does not have intermediate degree spectrum relative to $0'$.

³Though this monograph appeared in publication slightly before Wright’s paper [Wri18], a preprint of Wright’s paper was available before the second author started working on [HT18].

When Montalbán first suggested studying computable structure theory on a cone, the hope was that one might find more structure in an area of mathematics that generally involved non-structure theorems. Though this is sometimes true (e.g., in [CHT17]) it seems to not always be the case; e.g., we know that there are incomparable degree spectra on a cone [HT18]. The second half of this paper is dedicated to showing that even for relations on $(\omega, <)$, one of the simplest non-trivial examples, it seems hard to find structure in the degree spectra on a cone. One of the main results here is the following theorem:

Theorem 4.16. *Fix $\alpha \geq 6$ even. There is a unary function f on $(\omega, <)$ whose degree spectrum on a cone contains all of the β -c.e. degrees for $\beta < \alpha$ and does not contain all of the α -c.e. degrees.*

Thus there are uncountably many different degree spectra on a cone. We also prove in Theorem 4.3 that the degree spectra on a cone of these functions are not contained in the β -c.e. degrees for any β ; that is, for any β , the degree spectrum on a cone contains a non- β -c.e. degree.

2 Preliminaries

2.1 Cones and Martin's measure

Given a set $A \subseteq 2^\omega$, we say that A is *degree-invariant* if whenever $X \in A$ and $Y \equiv_T X$, $Y \in A$. If A is degree invariant, we can identify it with the corresponding set of Turing degrees $\{\deg_T(X) \mid X \in A\}$.

Definition 2.1. Given $X \subseteq \omega$, the *cone above X* is

$$C_X = \{Y \mid Y \geq_T X\}.$$

As a consequence of Martin's proof of Borel determinacy [Mar75], one gets the following theorem.

Theorem 2.2 (Martin, [Mar68]). *Every degree-invariant Borel subset of 2^ω either contains a cone or is disjoint from a cone.*

With more determinacy (e.g., analytic determinacy) this can be extended to more complicated degree-invariant sets. For this paper, Borel determinacy will be sufficient.

Thinking of cones as large sets, one can define the $\{0, 1\}$ -valued Martin's measure on Borel degree-invariant sets by setting $\mu(A) = 1$ if A contains a cone, and $\mu(A) = 0$ if A is disjoint from a cone. Note that the intersection of countably many cones contains a cone, which makes Martin's measure is countably additive.

2.2 Degree spectra on a cone

Definition 2.3. If \mathcal{A} is X -computable, we define the *degree spectrum of R relative to X* , $\text{DgSp}_\mathcal{A}^X(R)$, to be the set of degrees

$$\text{DgSp}_\mathcal{A}^X(R) = \{\deg_T(\varphi(R) \oplus X) : \mathcal{B} \text{ is an } X\text{-computable copy of } \mathcal{A} \text{ and } \varphi : \mathcal{A} \cong \mathcal{B}\}.$$

Let R and S be relations on \mathcal{A} and \mathcal{B} respectively. The set of Y such that $\text{DgSp}_{\mathcal{A}}^Y(R) = \text{DgSp}_{\mathcal{B}}^Y(S)$ is a degree-invariant Borel set,⁴ and so by Borel determinacy, either contains a cone or is disjoint from a cone. If it contains a cone, then we think of the degree spectra of R and S being equal for most degrees Y . Otherwise, there is a cone on which $\text{DgSp}_{\mathcal{A}}^Y(R) \neq \text{DgSp}_{\mathcal{B}}^Y(S)$ and we think of their degree spectra as being different for most degrees Y .

Definition 2.4. Let R and S be relations on \mathcal{A} and \mathcal{B} respectively. We say that the degree spectrum of R is equal to the degree spectrum of S on a cone if there is some X such that for all $Y \geq_T X$, $\text{DgSp}_{\mathcal{A}}^Y(R) = \text{DgSp}_{\mathcal{B}}^Y(S)$. The *degree spectrum of R on a cone* is the equivalence class for R modulo this equivalence relation.

We can also define what it means for the degree spectrum of R to be contained in the degree spectrum of S on a cone: if there is some X such that for all $Y \geq_T X$, $\text{DgSp}_{\mathcal{A}}^Y(R) \subseteq \text{DgSp}_{\mathcal{B}}^Y(S)$. Sometimes we also want to talk about the contents of the degree spectrum of R on a cone without reference to some other relation, as in the following definition.

Definition 2.5. Let R be a relation on \mathcal{A} . We say that the degree spectrum of R is equal to the c.e. degrees on a cone if there is some X such that for all $Y \geq_T X$, $\text{DgSp}_{\mathcal{A}}^Y(R)$ is the set of Y -c.e. degrees above Y . Similarly, we can define what it means for a degree spectrum to be equal to the computable degree or the Δ_2^0 degrees on a cone.

Note that throughout this paper, the relations and structures need not be computable. This is because for any structure and relation there is a cone on which that structure and relation are computable. Thus we may essentially behave as if all structures and relations are computable.

2.3 Notation and Definitions

Recall that Wright [Wri18] showed that given a computable relation R on $(\omega, <)$, either the degree spectrum of R is the computable degree $\{\mathbf{0}\}$ or the degree spectrum of R contains all of the c.e. degrees. For specific types of relations, more is known. For example, Wright showed that if R is unary, then the degree spectrum is either the computable degree $\{\mathbf{0}\}$ or all the Δ_2^0 degrees.

In the process of showing that there is a relation of intermediate degree spectrum, Bazhenov, Kalociński, and Wroclawski [BKW22] introduced the following class of relatively simple unary functions on $(\omega, <)$.

Definition 2.6. We say that a function $f : (\omega, <) \rightarrow (\omega, <)$ is a *block function* if for each n there is some interval $[a, b]$, containing n , that is closed under f and f^{-1} . We call the minimal such interval the f -block, or simply the block if f is understood, containing n . Distinct f -blocks cannot have any overlap, and so the domain of f can be divided into blocks.

The example they constructed, as well as the examples that we construct in this paper, will be block functions.

⁴To see this, one must use Scott's theorem [Sco65] that for a fixed structure \mathcal{A} , determining whether $\mathcal{B} \cong \mathcal{A}$ is Borel.

Each block is finite, so we can list out all of the possible isomorphism types of blocks. We call these isomorphism types the *block types* or simply *types*. We say that f *realises* or *has* a block type if that block type is the type of an f -block. We denote the n th block type in this enumeration by I_n and will use this to provide an alternate way to represent an arbitrary block function.

Definition 2.7. Given a block function f , we define a sequence α_f of natural numbers, called the string corresponding to f , by $\alpha_f(i) = n$ where I_n is the type of the i th block that occurs in f .

The string α_f completely determines f .

We note that the example of Bazhenov, Kalociński, and Wroclawski of an intermediate degree spectrum relies on not only the string α_f corresponding to f being non-computable, but the *counting function* $c_f(n) = \#\{i : \alpha_f(i) = n\}$ being non-computable. In our examples, both α_f and c_f will be computable together with any other reasonable information that one might ask for.

Given two f -blocks, we say that one embeds into the other if there is an order-preserving and f -preserving map from the one block to the other. Given block types I and J , we say that I embeds into J if there is an order-preserving and f -preserving map from any block of type I to any block of type J .

Frequently in this paper we will build a computable linear order $(\mathcal{A}, <_{\mathcal{A}}) \cong (\omega, <)$ (possibly relative to some oracle) by stages. At each stage, we will define a finite linear order \mathcal{A}_s whose domain is a subset of ω . At each stage $s + 1$, we may add new elements to \mathcal{A}_s to obtain \mathcal{A}_{s+1} , saying where they are in relation to all of the elements of \mathcal{A}_s , so that $\mathcal{A}_0 \subseteq \mathcal{A}_1 \subseteq \mathcal{A}_2 \subseteq \dots$. We build $\mathcal{A} = \bigcup_s \mathcal{A}_s$.

For clarity, we will always denote by $A = \{a_0, a_1, a_2, \dots\}$ the domain of \mathcal{A} to distinguish it from the particular isomorphism type $(\omega, <)$. At each stage s , we have a guess at an isomorphism $\mathcal{A} \rightarrow (\omega, <)$: 0 corresponds to the first element of \mathcal{A}_s , 1 to the second element, and so on. We write $\pi_s: \mathcal{A}_s \rightarrow (\omega, <)$ for the guess at stage s at this partial isomorphism: Given $a \in \mathcal{A}_s$, we write $\pi_s(a)$ for the corresponding element of $(\omega, <)$, that is, $\pi_s(a) = n$ where n is the number of elements less than a in \mathcal{A}_s .⁵ For a tuple $\bar{a} = (a_1, \dots, a_n)$, we write $\pi_s(\bar{a})$ for the corresponding tuple $(\pi_s(a_1), \dots, \pi_s(a_n))$. We also write π for the isomorphism $\pi: \mathcal{A} \rightarrow (\omega, <)$, and so write $\pi(a) = n$ for the corresponding element of $(\omega, <)$ at the end of the construction. Think of $\pi_s(a)$ as a stage-by-stage approximation of the final value $\pi(a)$. At each stage $s + 1$, new elements may be added to \mathcal{A} . If they are added above $a \in \mathcal{A}_s$, then $\pi_{s+1}(a) = \pi_s(a)$. But if they are added below $a \in \mathcal{A}_s$, then $\pi_{s+1}(a) > \pi_s(a)$.

While in some constructions we will build a copy \mathcal{A} of $(\omega, <)$ as described above, at other times we will be diagonalizing against all computable copies of $(\omega, <)$. Let $(\mathcal{A}_e)_{e \in \omega}$ be a list of the computable orderings, or more precisely the (possibly partial) $<$ -structures \mathcal{A}_e computed by the e th partial computable function. Generally if \mathcal{A}_e is not a linear order we recognize this at some finite stage and so can ignore it, and since we are only concerned with linear orders isomorphic to $(\omega, <)$ we can effectively assume that \mathcal{A}_e is infinite. So, morally speaking, we can think of each \mathcal{A}_e as being an infinite computable linear order. At each stage s , we have computed finite linear order $\mathcal{A}_{e,s}$. We can make definitions,

⁵Distinguishing between elements a of \mathcal{A} and the elements $\pi_s(a)$ they correspond to in $(\omega, <)$ at each stage s , a correspondence which will change between stages, will be the primary notational difficulty in this paper and we apologize to the reader for how hard it is to write down any notation for this in an easily readable form.

like $\pi_{e,s}(a)$, similar to the definitions above. We will usually omit the e , writing $\pi_s(a)$ when the structure \mathcal{A}_e is understood. Note that if \mathcal{A}_e is not actually isomorphic to $(\omega, <)$ then it is possible that $\lim_{s \rightarrow \infty} \pi_s(a) = \infty$ as infinitely many elements are added to \mathcal{A}_s below a ; but if \mathcal{A}_e really is isomorphic to $(\omega, <)$ then $\lim_{s \rightarrow \infty} \pi_s(a)$ will exist and be the isomorphic image of a in $(\omega, <)$.

Given $\bar{a} = (a_1, \dots, a_\ell)$ and $\bar{b} = (b_1, \dots, b_\ell)$ in $(\omega, <)$, we say that \bar{a} *extends to* \bar{b} if $b_1 \geq a_1$ and, for each i , $b_{i+1} - b_i \geq a_{i+1} - a_i$. Then \bar{a} extends to \bar{b} if and only if when $\bar{u} \in \mathcal{A}_s$, and $\pi_s(\bar{u}) = \bar{a}$, it is possible to add elements to \mathcal{A}_s to produce \mathcal{A}_{s+1} with $\pi_{s+1}(\bar{u}) = \bar{b}$. Also, given a tuple $\bar{a} = (a_1, \dots, a_\ell) \in (\omega, <)$ and $n \in \mathbb{Z}$ we write $\bar{a} + n$ for $(a_1 + n, \dots, a_\ell + n)$.

If f is a unary function on $(\omega, <)$, then we write $f^{\mathcal{A}_s}$ for the guess at stage s at the values of f on the elements of \mathcal{A}_s . We define $f^{\mathcal{A}_s}(a) = b$ if and only if $f(\pi_s(a)) = \pi_s(b)$. Note that $f^{\mathcal{A}_s}$ might not be a total function, as there might not be enough elements in \mathcal{A}_s . However if f is a block function, then there are initial segments of $(\omega, <)$ which are closed under f , and so by making some small adjustments we can assume that $f^{\mathcal{A}_s} : \mathcal{A}_s \rightarrow \mathcal{A}_s$ is total.

3 Intermediate Degree Spectrum

In this section we will construct an example of a unary function whose degree spectrum on a cone is intermediate between the c.e. degrees and the Δ_2^0 degrees. First, we will give a general characterization for when the degree spectrum on a cone of a block function strictly contains the c.e. degrees. This is the easiest part, and is essentially adapting Section 3 of [HT18] to the setting of block functions. Second, we will give a general condition for when the degree spectrum of a block function is all Δ_2^0 degrees. This is where most of the work is done. Finally, we give an example and prove that it meets the first condition but not the second condition and thus strictly contains the c.e. degrees and is strictly contained within the Δ_2^0 degrees.

3.1 Containing a non-c.e. degree

We take from [HT18] a sufficient condition for the degree spectrum of a relation R on an arbitrary structure to strictly contain the c.e. degrees on a cone. Properly re-interpreted in the specific case of a block function on $(\omega, <)$, this will give us a condition which is both necessary and sufficient.

Definition 3.1. Let R be an additional relation on a structure \mathcal{A} . We say that \bar{a} is difference-free (or d-free) over \bar{c} if for any tuple \bar{b} and quantifier-free formula $\varphi(\bar{c}, \bar{u}, \bar{v})$ true of \bar{a}, \bar{b} there are \bar{a}', \bar{b}' satisfying $\varphi(\bar{c}, \bar{u}, \bar{v})$ such that (1) R restricted to \bar{c}, \bar{a}' is not the same as R restricted to \bar{c}, \bar{a} and (2) for any existential formula $\psi(\bar{c}, \bar{u}, \bar{v})$ true of \bar{a}', \bar{b}' , there are \bar{a}'' and \bar{b}'' satisfying $\psi(\bar{c}, \bar{u}, \bar{v})$ and such that R restricted to $\bar{c}, \bar{a}'', \bar{b}''$ is the same as R restricted to $\bar{c}, \bar{a}, \bar{b}$.⁶

The sufficient condition from [HT18] is as follows.

⁶Note that in [HT18], the formula φ in the definition was allowed to be existential. The definition we give is equivalent and simpler.

Theorem 3.2 (Proposition 3.4 of [HT18]). *Let \mathcal{A} be a structure and R a relation on \mathcal{A} . Suppose that for each tuple \bar{c} there is \bar{a} which is d -free over \bar{c} . Then the degree spectrum of R strictly contains the c.e. degrees on a cone.*

We do not know whether this condition is also necessary. However if we interpret this for a block function on $(\omega, <)$, we get the following condition which is simpler as well as both necessary and sufficient.

Theorem 3.3. *Let f be a block function which is not intrinsically computable on a cone. Then f is intrinsically of c.e. degree on a cone if and only if there are infinitely many blocks that do not embed into a later block.*

Proof. First, suppose that there are only finitely many blocks that do not embed into a later block. After some finite initial segment, all blocks that occur embed into some later block. By non-uniformly fixing this initial segment, we can assume that *every* block embeds into another block. We show that for any tuple \bar{c} , there is some tuple \bar{a} which is d -free over \bar{c} . Given \bar{c} , let \bar{a} be some f -block of size greater than one such that all its elements are greater than those of \bar{c} . Since we assumed that f was not intrinsically computable, it cannot be the identity almost everywhere and so there must be infinitely many blocks of size greater than one. We claim that \bar{a} is d -free over \bar{c} .

Now, with \bar{c} , \bar{a} as above, suppose there is some quantifier-free formula $\varphi(\bar{x}, \bar{u}, \bar{v})$ and tuple \bar{b} such that $\varphi(\bar{c}, \bar{a}, \bar{b})$ is true. We make some simplifying assumptions. First, we may assume that there is no repetition amongst the elements of all of the tuples. Second, by including in \bar{c} any elements of \bar{b} which are less than the elements of \bar{a} , we may assume that all of the elements of \bar{b} are greater than all of the elements of \bar{a} . And third, we can assume that \bar{b} consists of n distinct adjacent blocks $\bar{b} = \bar{b}_1 \bar{b}_2 \cdots \bar{b}_n$ with \bar{b}_1 the first block after \bar{a} .

Now define $\bar{a}' = \bar{a} + 1$ and $\bar{b}' = \bar{b} + 1$. Then $\bar{c} \bar{a}' \bar{b}'$ has the same order type as $\bar{c}, \bar{a}, \bar{b}$ and so still satisfies $\varphi(\bar{x}, \bar{u}, \bar{v})$. However the values of f on \bar{a} are not the same as f on \bar{a}' as \bar{a} is a block and \bar{a}' is not. Furthermore, suppose $\psi(\bar{x}, \bar{u}, \bar{v})$ is some existential formula satisfied by $\bar{c}, \bar{a}', \bar{b}'$. We can again replace this by a quantifier free formula $\chi(\bar{x}, \bar{u}, \bar{v}, \bar{w})$ which is satisfied by $\bar{c}, \bar{a}', \bar{b}', \bar{e}'$ for some tuple \bar{e}' . Noting that there are no gaps between the elements of \bar{a}', \bar{b}' , we may split \bar{e}' into $\bar{e}'_1 \bar{e}'_2$ where the elements of \bar{e}'_1 are less than the elements of \bar{a}' and the elements of \bar{e}'_2 are greater than the elements of $\bar{a}' \bar{b}'$.

Now let \bar{a}'' be the image of \bar{a} in some block into which it embeds, and similarly let $\bar{b}''_1, \dots, \bar{b}''_n$ be the images of $\bar{b}_1, \dots, \bar{b}_n$ in blocks into which they embed, choosing these images sufficiently large that $\bar{c} < \bar{e}'_1 < \bar{a}'' < \bar{b}''_1 < \bar{b}''_2 < \cdots < \bar{b}''_n$. Finally, choose \bar{e}''_2 larger than all of these. Let $\bar{e}'' = \bar{e}'_1 \bar{e}''_2$.

Our construction has ensured that $\bar{c} \bar{a}'' \bar{b}'' \bar{e}''$ has the same order type as $\bar{c} \bar{a}' \bar{b}' \bar{e}'$ and so $\bar{c} \bar{a}'' \bar{b}''$ satisfies the formula $\psi(\bar{x}, \bar{u}, \bar{v})$. Moreover, the values of f on $\bar{c} \bar{a}'' \bar{b}''$ are the same as the values on $\bar{c} \bar{a} \bar{b}$. Thus, \bar{a} is d -free over \bar{c} , as desired. By Theorem 3.2, the degree spectrum of f strictly contains the c.e. degrees on a cone.

In the other direction, suppose that there are infinitely many f -blocks which embed into no larger f -blocks. We work on the cone on which we can compute α_f and whether a given block embeds into a later block. We will show that, for any X on this cone, if f^A is the copy of f in an X -computable copy $(\mathcal{A}, <_{\mathcal{A}})$ of $(\omega, <)$ then f^A is of X -c.e. degree. This implies that the degree spectrum of R on a cone is contained in the c.e. degrees. To simplify notation, we will suppress X , assuming that α_f is computable and that we can compute whether a given block embeds into any later block.

To show that $f^{\mathcal{A}}$ is of c.e. degree, we will show that for any k it computes the k th element of \mathcal{A} . This implies that it computes the successor relation on \mathcal{A} . Recall that the successor relation on \mathcal{A} is always of c.e. degree and computes $f^{\mathcal{A}}$, so this will imply that $f^{\mathcal{A}}$ is of c.e. degree.

Given k , look in $(\omega, <)$ for a block, all of whose elements are greater than k , which does not embed into any greater block. Suppose that this block is the elements $(\ell, \dots, \ell+m-1)$ of ω , so that there are ℓ elements less than this block. Let the block type be I . In \mathcal{A} , look for a tuple of elements $a_0 < a_1 < \dots < a_\ell < \dots < a_{\ell+m-1}$ of \mathcal{A} such that $f^{\mathcal{A}}$ on $(a_\ell, \dots, a_{\ell+m-1})$ has block type I . (This means that f on $(\ell, \dots, \ell+m-1)$ is the same as $f^{\mathcal{A}}$ on $(a_\ell, \dots, a_{\ell+m-1})$.) Once we find such elements, we know that a_k is the k th element of \mathcal{A} . This is because block type I does not embed into any later $f^{\mathcal{A}}$ -block (and noting that a block cannot embed into a sequence of blocks without embedding into one of them). \square

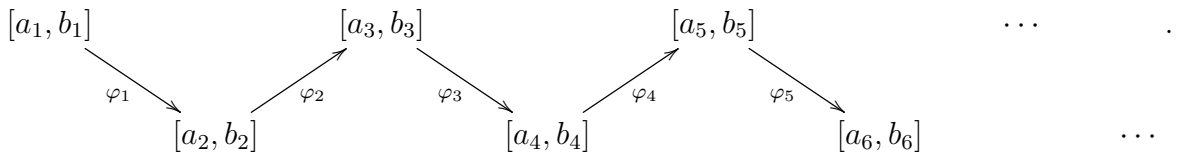
3.2 Not all Δ_2^0 degrees

In this section we give conditions to determine whether the degree spectrum of a block function on a cone consists of all Δ_2^0 degrees or whether the degree spectrum is properly contained in the Δ_2^0 degrees. The key is whether or not an infinite f -coding sequence, defined as follows, exists.

Definition 3.4. Given a block function f on $(\omega, <)$, we say that a sequence of intervals $[a_1, b_1], [a_2, b_2], \dots$ and a collection of maps $\varphi_i: [a_i, b_i] \rightarrow [a_{i+1}, b_{i+1}]$ form an f -coding sequence if they satisfy the following conditions

- (1) Each interval completely contains all f -blocks it intersects, and hence is closed under f and f^{-1} .
- (2) $\varphi_i: [a_i, b_i] \rightarrow [a_{i+1}, b_{i+1}]$ are non-decreasing embeddings which preserve the ordering.
- (3) $\varphi_{i+1} \circ \varphi_i: [a_i, b_i] \rightarrow [a_{i+2}, b_{i+2}]$ preserves f .
- (4) $\varphi_1: [a_1, b_1] \rightarrow [a_2, b_2]$ does not preserve f , i.e., there is some $x \in [a_1, b_1]$ such that $\varphi_1(f(x)) \neq f(\varphi_1(x))$. Together with the previous property this implies that none of the φ_i preserves f .
- (5) $a_{i+1} > b_i$ so that all of the elements of each interval are greater than the elements of the previous interval.

For $i < j$ we write $\varphi_{i \rightarrow j}$ for $\varphi_{j-1} \circ \dots \circ \varphi_i: [a_i, b_i] \rightarrow [a_j, b_j]$. If i and j have the same parity, then $\varphi_{i \rightarrow j}$ preserves f , and otherwise it does not. Also, the $\varphi_{i \rightarrow j}$ compose nicely, i.e., $\varphi_{j \rightarrow k} \circ \varphi_{i \rightarrow j} = \varphi_{i \rightarrow k}$.



This definition can be motivated as follows. Given a Δ_2^0 set X , we want to produce a computable copy \mathcal{A} of $(\omega, <)$ such that $f^{\mathcal{A}} \equiv_T X$. For each e we want to introduce

some elements of \bar{u}_e of \mathcal{A} coding whether $e \in X$. We will build \mathcal{A} by stages, with a finite linear order \mathcal{A}_s at each stage, and $\mathcal{A}_0 \subseteq \mathcal{A}_1 \subseteq \mathcal{A}_2 \subseteq \dots$ with \mathcal{A} as the union. At each stage s , we guess that n th element of \mathcal{A}_s is really the n th element of \mathcal{A} (hence, the isomorphic image of $n - 1 \in (\omega, <)$). Thus at stage s we have a guess at the values of $f^{\mathcal{A}}$ on these elements. When e enters or leaves X at a stage, we must insert elements into \mathcal{A} below and/or between the elements of \bar{u}_e to change the guess at the values of $f^{\mathcal{A}}$ on these elements. If, say, e left X , so we added new elements to \mathcal{A} to change the guess at the values of $f^{\mathcal{A}}$ on \bar{u}_e , and then later e re-enters X , we must again add new elements to \mathcal{A} so that the values of $f^{\mathcal{A}}$ on \bar{u}_e look the same as they did when previously we had $e \in X$. It is important that we cannot just make the values of $f^{\mathcal{A}}$ on \bar{u}_e different every time e enters or leaves X , because then we might be able to compute information about how many times e entered or left X .

An f -coding sequence is set up exactly to perform such a construction. Given a single e , say $e \notin X$ at this stage, we begin with \bar{u}_e being $[a_1, b_1]$. Then, when e enters X , we add elements to \mathcal{A} so that \bar{u}_e becomes $\varphi_1([a_1, b_1]) \subseteq [a_2, b_2]$. When e later leaves X we add elements to \mathcal{A} so that \bar{u}_e becomes $\varphi_2(\varphi_1([a_1, b_1])) = \varphi_{1 \rightarrow 3}([a_1, b_1]) \subseteq [a_3, b_3]$. We continue in this way until e settles. Of course there is much more to deal with, such as all of the other values of e , and we have not exactly explained why $X \geq_T f^{\mathcal{A}}$ as there are many other elements of \mathcal{A} . But this is the basic idea.

Only one infinite coding sequence is necessary to code a Δ_2^0 set as the coding tuples for all values of e can be moved along this same sequence; for this, property (5) is important. Further, we will show that the absence of such a sequence is enough to miss a Δ_2^0 set. This is because, if all coding sequences are finite, then we can produce a Δ_2^0 set which changes the value of each element more times than the length of the coding sequence which attempts to determine its value. This idea will be made more formal in the proofs to follow.

When constructing an f -coding sequence, it will be more convenient to omit (5). We call such a sequence of intervals and maps a *weak f -coding sequence*. We may occasionally refer to an f -coding sequence as a *strong f -coding sequence* to emphasize that it is not just a weak f -coding sequence. When the function f is clear, we may simply say a coding sequence. In this section, we will be concerned solely with infinite f -coding sequences, but in the following section we will be concerned with finite f -coding sequences. In those sections we may not always write “infinite” or “finite” respectively.

By general arguments, one can always turn an infinite weak f -coding sequence into an infinite f -coding sequence.

Lemma 3.5. *Let f be a unary function on $(\omega, <)$, and suppose that f has an infinite weak f -coding sequence. Then f has an infinite (strong) f -coding sequence.*

Proof. Suppose that we have a weak f -coding sequence with intervals $[a_1, b_1], [a_2, b_2], \dots$ and maps φ_i . Recall that all of the maps are non-decreasing.

We say that an element $x \in [a_1, b_1]$ *eventually increases to infinity* if, for each y , there is some composition of the φ which, when applied to x , increase it above y . That is, if for every y there is some n such that $\varphi_n(\dots(\varphi_1(x))\dots) > y$. If x does not eventually increase to infinity, then there must be some y such that for all sufficiently large n , $\varphi_n(\dots(\varphi_1(x))\dots) = y$. Note that b_1 eventually increases to infinity, as otherwise there would be n such that for all $m \geq n$ the map φ_m would be the identity, contradicting (4). Also, we note that if x eventually increases to infinity, then so does every $y \geq x$, and also every y in the same block as x .

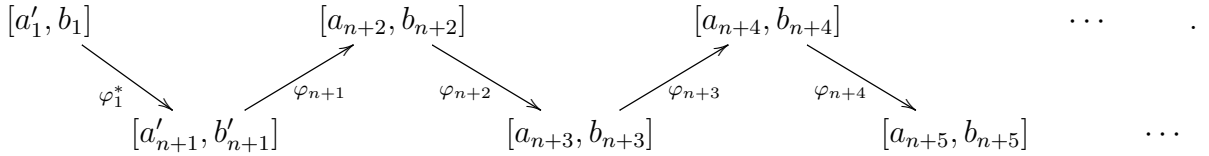
Let $a'_1 \in [a_1, b_1]$ be such that every element of $[a_1, a'_1)$ does not eventually increase to infinity, and every element of $[a'_1, b_1]$ does eventually increase to infinity. By the above remarks, $[a'_1, b_1]$ is closed under blocks. First, we argue that φ_1 does not preserve f when restricted to $[a'_1, b_1]$. This is because it does preserve f when restricted to $[a_1, a'_1)$, as we now argue. There is n even such that for all $x \in [a_1, a'_1)$

$$\varphi_n(\cdots(\varphi_1(x))\cdots) = \varphi_{n+1}(\varphi_n(\cdots(\varphi_1(x))\cdots)).$$

But $\varphi_{n+1} \circ \cdots \circ \varphi_2 = \varphi_{2 \rightarrow n+1}$ preserves f since n is even. So φ_1 preserves f on $[a_1, a'_1)$, and so does not preserve f when restricted to $[a'_1, b_1]$.

Now since a'_1 eventually increases to infinity, there must be some odd n such that $\varphi_n(\cdots(\varphi_1(a'_1))) > b_1$. (Since b_1 is the largest element of its block, this also implies that all elements of any block containing the image of an element $[a'_1, b_1]$ under $\varphi_n \circ \cdots \circ \varphi_1$ is also greater than b_1 .) We now replace $[a_1, b_1]$ by $[a'_1, b_1]$ and delete all of the blocks $[a_2, b_2], \dots, [a_n, b_n]$ so that the next block after $[a'_1, b_1]$ is $[a_{n+1}, b_{n+1}]$. We still have a map $\varphi_1^* = \varphi_{1 \rightarrow n+1}$ from $[a'_1, b_1]$ to $[a_{n+1}, b_{n+1}]$. Now we replace $[a_{n+1}, b_{n+1}]$ with the smallest interval $[b'_{n+1}, a'_{n+1}]$ which contains the image of $[a'_1, b_1]$ under φ_1^* and which is closed under blocks. By choice of n , we get that $a'_{n+1} > b_1$.

Thus we have now obtained



This is a weak f -coding sequence with $a'_{n+1} > b_1$. By continuing this process, now with the second interval $[a'_{n+1}, b'_{n+1}]$, and so on, we eventually obtain an f -coding sequence. \square

Theorem 3.6. *Let f be a block function such that every block, except for finitely many, embeds into some later block. Then the degree spectrum of f , on a cone, is all Δ_2^0 degrees if and only if there is an infinite f -coding sequence*

Proof. First, suppose there is an infinite f -coding sequence $[a_1, b_1], [a_2, b_2], [a_3, b_3], \dots$ with functions φ_i . We show that, on the cone above α_f and this coding sequence, the degree spectrum of f is all Δ_2^0 degrees. As usual, we will suppress the base of the cone and assume that α_f and this coding sequence are computable. We will also assume, by fixing a finite initial segment, that every block embeds into a later block. Then we must show that the degree spectrum of f is all Δ_2^0 degrees. Since the degree spectrum of f is contained in the Δ_2^0 degrees, we must show that every Δ_2^0 is in the degree spectrum of f .

Fix a Δ_2^0 set X . We will build a computable copy \mathcal{A} of $(\omega, <)$ such that $f^{\mathcal{A}} \equiv_T X$. During this construction, the elements we place into the linear order will fall into one of two categories: *coding elements* and *padding elements*. Padding elements are elements that are added to move coding elements, and play no other role in the coding. We will have to ensure that the approximation $f^{\mathcal{A}_s}$ to $f^{\mathcal{A}}$ stays the same on the padding elements; we can do this because every block embeds into a later block. For each $e \in \omega$, there will be some collection of coding elements corresponding to e . We divide these into the *initial coding elements* which are the elements first chosen to code whether $e \in X$. It is these elements on which we will check the values of $f^{\mathcal{A}}$ to determine whether e is in X . When we first start coding whether $e \in X$, we will add initial coding elements to \mathcal{A} corresponding (via the current guess at the isomorphism $\mathcal{A} \rightarrow (\omega, <)$) to an interval $[a_i, b_i]$ with i even if

$e \notin X$ or odd if $e \in X$. If, at some later stage, we see that e has entered or exited X , we insert elements into \mathcal{A} so that these initial coding elements now correspond to elements of some $[a_j, b_j]$, $j > i$, with the parity of j depending on whether $e \in X$ or $e \notin X$; moreover, the elements in this interval that they correspond to should be the images, under the φ 's, of the elements they previously corresponded to. It is also possible that some $e' < e$ will enter or exit X , causing us to add elements to \mathcal{A} for the sake of e' , and thus requiring us to take action for the coding elements for e as well. Because the embeddings φ might not be surjective, as the construction progresses, the initial coding elements will “gather” other *subsequent coding elements*. The coding elements for e will be consecutive and we call them the *coding segment* for e .

The goal of the construction is to move the coding elements so that the values of $f^{\mathcal{A}}$ on them mirror whether the corresponding e is in X or not, while also ensuring that once padding elements have been added to the structure, the value of $f^{\mathcal{A}}$ on those elements does not change. Given $e' < e$, the coding segment for e will be greater than the coding segment for e' , so that we can move the segment for e without moving the segment for e' .

Construction: For each e , starting from stage e where they are defined, we will have the initial coding elements $\bar{u}_e \in \mathcal{A}$, the full coding segment $\bar{v}_e \in \mathcal{A}$ containing \bar{u}_e , and a restraint $r_e = \max_{\mathcal{A}} \bar{v}_e \in \mathcal{A}$ which is the \mathcal{A} -largest element of \bar{v} . We write each tuple in \mathcal{A} -increasing order. The initial coding elements \bar{u}_e will never change once defined, while the coding segment \bar{v}_e may have more elements added, and the restraint r_e may increase in the \mathcal{A} order. We will use $\bar{v}_e[s]$ and $r_e[s]$ to refer to the values at stage s . We will always have that if $e' \leq e$ then $\bar{v}_e > r_{e'}$ so that the coding segment for e is free of the restraint of e' .

Recall that at each stage s , given a tuple $\bar{a} \in \mathcal{A}_s$, $\pi_s(\bar{a})$ is the corresponding tuple in $(\omega, <)$. We also have $f^{\mathcal{A}_s}$ which is the function f on \mathcal{A}_s .

From each stage to the next, we will maintain the following properties:

- (1) At each stage $s \geq e$, the coding elements \bar{v}_e of \mathcal{A} correspond to $\pi_s(\bar{v}_e)$ in $(\omega, <)$ which is an interval $[a_i, b_i]$, with i odd if $e \in X_s$ or i even if $e \notin X_s$.
- (2) If at stage $s \geq e$ the coding elements $\bar{v}_e[s]$ corresponded (via $\pi_s: \mathcal{A}_s \rightarrow (\omega, <)$) to an interval $[a_i, b_i]$, and at stage $s + 1$ the coding elements $\bar{v}[s + 1]$ correspond (via $\pi_{s+1}: \mathcal{A}_s \rightarrow (\omega, <)$) to an interval $[a_j, b_j]$, then the coding elements $\bar{v}_e[s]$ at stage s correspond at stage $s + 1$ to the images, under the φ , of the elements of $(\omega, <)$ that they corresponded to at stage s . That is,

$$\pi_{s+1}(\bar{v}_e[s]) = \varphi_{i \rightarrow j}(\pi_s(\bar{v}_e[s])).$$

- (3) If, at stage s an element $a \in \mathcal{A}$ is a padding element, by which we mean that it is not a coding element (i.e., not part of any $\bar{v}_e[s]$), then at stage $s + 1$ it is still not a coding element, and $f^{\mathcal{A}_s}(a) = f^{\mathcal{A}_{s+1}}(a)$.
- (4) From stage $s \geq e$ to stage $s + 1$, elements can only be added to \mathcal{A} below r_e if there is $e' \leq e$ which entered or left X at stage $s + 1$, i.e., with $e' \in X_s \Delta X_{s+1}$. Thus, if $X_{s+1} \upharpoonright_e = X_s \upharpoonright_e$, then the partial isomorphism $\pi_s: \mathcal{A}_s \rightarrow (\omega, <)$ does not change at stage $s + 1$ on elements below r_e .

Stage 0: We begin with \mathcal{A}_0 being empty.

Stage s: At this stage in the construction our partial linear order \mathcal{A}_s can be partitioned into a finite number of coding segments and padding blocks, each of which form an interval:

$$\bar{p}_0 <_{\mathcal{A}} \bar{v}_0[s] <_{\mathcal{A}} \bar{p}_1 <_{\mathcal{A}} \bar{v}_1[s] <_{\mathcal{A}} \cdots .$$

We add new elements to \mathcal{A}_s to code X_{s+1} .

We do not do anything to \bar{p}_0 . We begin by handling $\bar{v}_0[s]$. If $X_{s+1}(0) = X_s(0)$, then we do nothing. But if $X_{s+1}(0) \neq X_s(0)$, then we must act. Suppose that $0 \in X_s$ but $0 \notin X_{s+1}$. Then $\bar{v}_0[s]$ corresponds to the elements of an interval $[a_i, b_i]$ for i even. We insert $a_{i+1} - a_i$ new padding elements above the elements of \bar{p}_0 and below the elements of $\bar{v}_0[s]$. Also add new coding elements ($b_{i+1} + a_i - a_{i+1} - b_i$ of them) to ensure that $\bar{v}_0[s]$ corresponds, in \mathcal{A}_{s+1} , to $\varphi_i(\pi_s(\bar{v}_0[s])) \subseteq [a_{i+1}, b_{i+1}]$. These new elements will be subsequent coding elements; let $\bar{v}_0[s+1]$ consist of $\bar{v}_0[s]$ together with these new elements of \mathcal{A} . If $0 \notin X_s$ but $0 \in X_{s+1}$, then we act similarly but move \bar{v}_0 from an interval $[a_i, b_i]$ for i odd to $[a_{i+1}, b_{i+1}]$ with $i+1$ even.

Now we have added several elements below \bar{p}_1 . We must ensure that the values of f on \bar{p}_1 at stage $s+1$ are the same as they were at stage s . For each block making up \bar{p}_1 , insert new padding elements below the least element of this block and possibly between the elements of this block to move them up to the image of the original f -block in some other f -block into which it embeds. (We use here the fact that each f -block embeds into infinitely many other blocks.) Further, we add enough padding elements on the end to complete these blocks. This ensures that for all padding elements present at stage s , the value of $f^{\mathcal{A}_{s+1}}$ on these elements at stage $s+1$ is the same as it was at stage s .

We continue this process, in turn, with the subsequence coding segments and padding blocks. In increasing order, we ensure that each of these segments from \mathcal{A}_s still satisfy our requirements in \mathcal{A}_{s+1} . However we must now take into account the fact that we may have already inserted elements.

- Given a padding segment \bar{p}_i , consider each set of elements which corresponded at stage s to a block. We must make sure that those same elements correspond in \mathcal{A}_{s+1} to a block of the same type. For each block making up \bar{p}_i , insert new padding elements below the least element of this block and possibly between the elements of this block to move them up to the image of the original f -block in some other f -block into which it embeds. Further, ensure we add enough padding elements on the end to complete this block. This ensures that for all padding elements present at stage s , the value of $f^{\mathcal{A}_s}$ on these elements is the same as at stage $s+1$.
- Given $\bar{v}_e[s]$ the coding segment corresponding to e first check the value of $X_{s+1}(e)$. Even if $X_{s+1}(e) = X_s(e)$ has not changed, we may have added new elements below $\bar{v}_e[s]$ so that it no longer corresponds to the same interval $[a_k, b_k]$ that it did at stage s . Let $\varphi_{k \rightarrow \ell}$ be such that (i) ℓ is even if $e \in X_{s+1}$ and ℓ is odd if $e \notin X_{s+1}$, and (ii) ℓ is sufficiently large that by adding elements below and between the elements of $\bar{v}_e[s]$ that at stage $s+1$ the segment $\bar{v}_e[s]$ corresponds to the image, under $\varphi_{k \rightarrow \ell}$, of $[a_k, b_k] = \pi_s(\bar{v}_e[s])$. (Note that (5) in the definition of a coding sequence—the difference between a weak coding sequence and a coding sequence—is key to obtain such an $[a_\ell, b_\ell]$ with a_ℓ large enough to accommodate the new elements that have been added to the linear order below $\bar{v}_e[s]$.) Any new elements that were inserted and end up in the new interval are added to the collection of coding elements $\bar{v}_e[s+1]$ corresponding to e , and the other newly added elements are padding elements. Further, we ensure that enough new coding elements are after the final

element to ensure we complete the entire interval, putting these elements in $\bar{v}_e[s+1]$ as well. Thus $\bar{v}_e[s+1]$ will now consist of $\pi_{s+1}^{-1}([a_\ell, b_\ell])$.

- Finally, we introduce the coding elements corresponding to s . After we have ensured all previously added elements still satisfy the requirements check the value of $X_{s+1}(s)$ (which we may assume is 0, i.e., $s \notin S_{s+1}$) and identify the next interval of the form $[a_i, b_i]$ such that i is even if $s \in X_{s+1}$ and i is odd if $s \notin X_{s+1}$, and with a_i greater than the length of the linear order at this stage. Insert enough new padding elements to the end of the linear order to extend it to have length a_i , then add $b_i - a_i + 1$ new coding elements corresponding to the interval $[a_i, b_i]$. These are the initial coding elements $\bar{u}_s[s+1]$ for s , and make up the coding segment $\bar{v}_s[s+1]$ at this stage. This ensures we have satisfied requirement R_s at stage s .

Verification: First, to see that \mathcal{A} is really a computable copy of $(\omega, <)$, observe that for any fixed $a \in \mathcal{A}$, only finitely many elements are inserted below a . This follows from (4). Because a was added at stage s in the construction, then there can be at most $s - 1$ coding blocks below it. An element is only inserted below a if some e corresponding to one of these coding blocks enters or leaves X , with a below the restraint. After these $s - 1$ elements of the Δ_2^0 set X stop changing values, elements will stop being inserted below a . Hence, since these are finitely many elements of a Δ_2^0 set, this will occur in a finite number of stages.

To show that $f^{\mathcal{A}} \equiv_T X$, we use the following lemma which follows from (2).

Lemma 3.7. *Suppose that $s < t$. If, at stage s , $\bar{v}_e[s]$ corresponds to $[a_m, b_m]$ and, at stage t , $\bar{v}_e[t]$ corresponds to $[a_n, b_n]$, then the elements of $\bar{v}_e[s]$ corresponds at stage t to the images under $\varphi_{m \rightarrow n}$ of the elements they corresponded to at stage s :*

$$[a_n, b_n] \supseteq \pi_t(\bar{v}_e[s]) = \varphi_{m \rightarrow n}(\pi_s(\bar{v}_e[s])) = \varphi_{m \rightarrow n}[a_m, b_m].$$

In particular, if m and n have the same parity, then $f^{\mathcal{A}_t}(a) = f^{\mathcal{A}_s}(a)$. If m and n have opposite parities, then there is $a \in \bar{u}_e$ such that $f^{\mathcal{A}_t}(a) \neq f^{\mathcal{A}_s}(a)$.

Proof. The first part follows from (2) using induction. The second part follows from the definition of a coding sequence, particularly which $\varphi_{m \rightarrow n}$ are f -preserving. \square

We first use the lemma and (1) to show that $f^{\mathcal{A}} \geq_T X$. Given some element e run the above construction, which is computable, until stage $e + 1$ when the initial coding elements \bar{u}_e corresponding to e are introduced. Now, compute the true value of $f^{\mathcal{A}}$ on these elements. If $f^{\mathcal{A}}$ is the same on \bar{u}_e as it was at stage $e + 1$, then $X(e) = X_e(e + 1)$. Otherwise, if $f^{\mathcal{A}}$ is different on \bar{u}_e than it was at stage $e + 1$, then $X(e) = 1 - X_e(e + 1)$.

Finally, we show that $X \geq_T f^{\mathcal{A}}$. This uses the lemma as well as properties (1) and (3). Given some element $a \in \mathcal{A}$ run the above construction until a is added to the linear order. Using (3) if a is added as a padding element then the construction ensures that $f^{\mathcal{A}}(a)$ does not change and so we take the value at this stage. If a is a coding element corresponding to e then we use (1) and the lemma. We will see that $f^{\mathcal{A}}(a)$ takes on one of two values depending on $X(e)$. Say that a first appears in \mathcal{A} at a stage s in a coding block $\bar{v}_e[s]$, corresponding at that stage to $[a_i, b_i]$ and with $X_s(e) = k$. If $X(e) = X_s(e) = k$, then $f^{\mathcal{A}}(a)$ is the same as it was at stage s . Otherwise, suppose that $X(e) \neq X_s(e) = k$. Then, at some stage $t > s$, we find that $X(e) = X_t(e) = 1 - k$. At this stage, a is an element of $\bar{v}_e[t]$ which corresponds to $[a_j, b_j]$ with j of a different parity from i . Then $f^{\mathcal{A}}(a)$ is the same as it was at stage t .

For the converse direction we must show that if the degree spectrum of f , on a cone, is all Δ_2^0 degrees then there is an infinite f -coding sequence. We work on the cone above α_f (which we suppress) and show that if no infinite f -coding sequences exist then we can produce Δ_2^0 set C such that there is no computable copy \mathcal{L} of $(\omega, <)$ with $f^{\mathcal{L}} \equiv_T C$. To do this, we meet the following requirements

$$R_{e,i,j} : \text{If } \mathcal{L}_e \cong (\omega, <), \text{ then either } \Phi_i^{f^{\mathcal{L}_e}} \neq C \text{ or } \Psi_j^C \neq f^{\mathcal{L}_e}$$

where \mathcal{L}_e is a computable listing of the (possibly partial) linear orders. The construction is a finite injury priority construction. Given a set X , we make use of the notation $X[0, \dots, k]$ for the sequence $\langle X(0), \dots, X(k) \rangle$.

The strategy for $R_{e,i,j}$ is as follows. First, if e th program fails to code a linear order the requirement is automatically satisfied and so we will assume that at all stages s , $\mathcal{L}_{e,s}$ is a linear order.

- (1) To initialize this requirement choose some x that has not yet been restrained, restrain it, and assign it to this requirement. We begin with $C(x) = 0$. We say the strategy is in Phase 0.
- (2) At stage s , if the requirement is in Phase 0, say this requirement requires attention if there are computations

$$\Phi_{i,s}^{f^{\mathcal{L}_{e,s}}}(x) = 0 = C_s(x) \quad \text{with use } u_0$$

and

$$\Psi_{j,s}^{C_s}[0, \dots, m_0] = f^{\mathcal{L}_{e,s}}[0, \dots, m_0] \quad \text{with use } v_0$$

where $m_0 \geq u_0$ is such that all f -blocks that intersect $[0, \dots, u_0]$ in $\mathcal{A}_{e,s}$, and all elements less than them in $\mathcal{L}_{e,s}$, are completely contained in $[0, \dots, m_0]$.

When this requirement acts, restrain $[0, \dots, v_0]$ in C and define $C_{s+1}(x) = 1$. Finally, move the requirement to Phase 1.

- (3) At stage s , if the requirement is in Phase $n + 1$, say this requirement requires attention if there are computations

$$\Phi_{i,s}^{f^{\mathcal{L}_{e,s}}}(x) = C_s(x) \quad \text{with use } u_{n+1}$$

and

$$\Psi_{j,s}^{C_s}[0, \dots, m_{n+1}] = f^{\mathcal{L}_{e,s}}[0, \dots, m_{n+1}] \quad \text{with use } v_{n+1}$$

where $m_{n+1} \geq \max\{m_n, u_{n+1}\}$ such that all f -blocks that intersect the elements $[0, \dots, \max\{m_n, u_{n+1}\}]$, and all elements less than them in $\mathcal{L}_{e,s}$, are completely contained in $[0, \dots, m_{n+1}]$.

When this requirement acts, restrain $[0, \dots, v_{n+1}]$ in C and define $C_{s+1}(x) = 1 - C_s(x)$. Finally, move the requirement to Phase $n + 2$.

Construction of C : At stage s of the construction, consider the first s requirements, in order of decreasing priority. If any requirement requires attention then the one with the highest priority acts according to its strategy, injuring and resetting all lower priority strategies. If no requirement acts, then initialize the lowest priority requirement that has not yet been initialized.

Verification: It is not obvious that the construction is a finite injury construction, as even if a requirement $R_{e,i,j}$ is not injured, it appears on the surface that it might go through infinitely many phases. In this case, our approximation for C would also not come to a limit. We will argue by induction on the requirements that each requirement acts only finitely many times and is eventually satisfied. This argument will use the fact that there is no infinite coding sequence.

Consider a requirement $R_{e,i,j}$ and assume that after some stage it is no longer injured. If \mathcal{L}_e is partial (or not a linear order), then $R_{e,i,j}$ is automatically satisfied and will no longer act after some stage. So we may assume that \mathcal{L}_e is a linear order. If there is some n such that the strategy enters Phase n but never enters Phase $n+1$, then we are also done since $R_{e,i,j}$ will have acted only finitely many times and will also be satisfied. (Otherwise, if $\Phi_i^{f^{\mathcal{L}_e}} = C$ and $\Phi_j^C = f^{\mathcal{L}_e}$, then we would eventually enter Phase $n+1$.) So it is enough to show that the strategy enters finitely many phases as this will also show that it requires attention finitely many times. To do this we show that if, after a strategy is no longer injured, it enters Phase n for arbitrarily large n then we can produce an infinite f -coding sequence; in fact we produce a weak f -coding sequence and then use Lemma 3.5 to obtain an f -coding sequence.

Let s_n be the stage at which the strategy enters Phase n , if it exists, and recall that v_n is the restraint placed at Phase n . We begin by arguing that the restraints are maintained, and that C (up to the restraint) cycles back and forth between two possible configurations depending on whether the phase is odd or even.

Lemma 3.8. *Let $n' > n$ be of the same parity. Then:*

- (1) $C_{s_{n'}}[0, \dots, v_n] = C_{s_n}[0, \dots, v_n]$.
- (2) $f^{\mathcal{L}_{e,s_{n'}}}[0, \dots, m_n] = f^{\mathcal{L}_{e,s_n}}[0, \dots, m_n]$.

Further, for all n ,

- (3) $f^{\mathcal{L}_{e,s_n}}[0, \dots, u_n] \neq f^{\mathcal{L}_{e,s_{n+1}}}[0, \dots, u_n]$.

Proof. We check each of (1), (2), and (3).

- (1) After Phase n , we have restrained the elements $[0, \dots, v_n]$ and so, since the requirement is no longer injured by higher priority arguments, the only elements in $[0, \dots, v_n]$ that can change value is the original x that was restrained for $R_{e,i,j}$. However, since n and n' have the same parity, the construction ensures that $C(x)$ (whether x is in C) is the same as well. Hence, $C_{s_{n'}}[0, \dots, v_n] = C_{s_n}[0, \dots, v_n]$, as desired.
- (2) By construction we have $\Phi_j^{C_{s_n}}[0, \dots, m_n] = f^{\mathcal{L}_{e,s_n}}[0, \dots, m_n]$ and $\Phi_j^{C_{s_{n'}}}[0, \dots, m_n] = f^{\mathcal{L}_{e,s_{n'}}}[0, \dots, m_n]$. Further, the first computation listed has use v_n . Since by (1), this use v_n is the same at Phases n and n' , the computations $\Phi_j^{C_{s_n}}[0, \dots, m_n]$ and $\Phi_j^{C_{s_{n'}}}[0, \dots, m_n]$ must be the same as well and so $f^{\mathcal{L}_{e,s_{n'}}}[0, \dots, m_n] = f^{\mathcal{L}_{e,s_n}}[0, \dots, m_n]$, as desired.
- (3) By construction $\Phi_i^{f^{\mathcal{L}_{e,s_n}}}(x) = C_{s_n}(x)$ and $\Phi_i^{f^{\mathcal{L}_{e,s_{n+1}}}}(x) = C_{s_{n+1}}(x)$ but $C_{s_n}(x) \neq C_{s_{n+1}}(x)$ and so the use of the computation $\Phi_i^{f^{\mathcal{L}_{e,s_n}}}(x)$ must have changed by stage s_{n+1} , otherwise we would recover the same computation. Since the use is u_n , we must have $f^{\mathcal{L}_{e,s_n}}[0, \dots, u_n] \neq f^{\mathcal{L}_{e,s_{n+1}}}[0, \dots, u_n]$, as desired. \square

It will be helpful to consider, at each stage s , the guess $\pi_s: \mathcal{L}_{e,s} \rightarrow (\omega, <)$ at the isomorphism $\mathcal{L}_e \rightarrow (\omega, <)$. Note that (by definition), $f^{\mathcal{L}_{e,s}}$ is the image of f under π_s .

Finally, to produce the weak f -coding sequence we proceed as follows:

- Let $[a_0, b_0]$ be the smallest initial segment of $(\omega, <)$ which contains all of the π_{s_0} -images of the elements $0, \dots, u_0$ in \mathcal{L}_{e,s_0} and all of the blocks which they intersect. Note that all of $[a_0, b_0]$ is contained within the π_{s_0} -images of $[0, \dots, m_0]$.
- Given $[a_i, b_i]$, let $[a_{i+1}, b_{i+1}]$ be the minimal interval in $(\omega, <)$ that contains all of the $\pi_{s_{i+1}}$ -images of the elements $0, \dots, \max\{m_i, u_{i+1}\}$ in $\mathcal{L}_{e,s_{i+1}}$ and all of the blocks which they intersect. Note that all of $[a_i, b_i]$ is contained within the $\pi_{s_{i+1}}$ -images of $[0, \dots, m_{i+1}]$.
- Define the maps $\varphi_i: [a_i, b_i] \rightarrow [a_{i+1}, b_{i+1}]$ by $\varphi_i = \pi_{s_{i+1}} \circ \pi_i^{-1}$. (The way to think about these maps is that, when $[a_i, b_i]$ was defined at stage s_i , these elements of $(\omega, <)$ corresponded to certain elements of \mathcal{L}_e . However by stage s_{i+1} , we have seen more elements enter \mathcal{L}_e , and now those elements of \mathcal{L}_e are instead in correspondence with other elements of $(\omega, <)$, namely some elements of $[a_{i+1}, b_{i+1}]$. The map f_i keeps track of how this correspondence has changed. The element of \mathcal{L}_e that corresponded to $n \in [a_i, b_i]$ now corresponds to $f_i(n) \in [a_{i+1}, b_{i+1}]$.)

This sequence satisfies most of the requirements by construction. We just need to ensure that the φ_i are not f -preserving but the $\varphi_{i+1} \circ \varphi_i$ are. But this is exactly the content of Lemma 3.8. Hence, assuming we enter Phase n for arbitrarily large n we have produced an infinite weak coding sequence. From an infinite weak coding sequence, using Lemma 3.5 we can produce an infinite coding sequence. Thus, by our assumption about the absence of such a sequence it follows that after a strategy is no longer injured it only enters finitely many phases. \square

From this theorem we can easily recover several previous results, such as Theorem 3.10 of [BKW22].

3.3 An example

Now that we have produced necessary and sufficient conditions for the degree spectrum of a unary function being either all c.e. degrees or all Δ_2^0 degrees on a cone, we know exactly which conditions a unary function must satisfy to have an intermediate degree spectrum and can produce an example.

Theorem 3.9. *There is a computable block function f whose degree spectrum on a cone strictly contains the c.e. degrees and is strictly contained within the Δ_2^0 degrees. Moreover, the base of the cone is the computable degree, and so the degree spectrum of this function strictly contains the c.e. degrees and is strictly contained within the Δ_2^0 degrees.*

Proof. Let L_k be the block corresponding to the loop of length k , i.e., the block isomorphic to $[1, \dots, k] \rightarrow [1, \dots, k]$ via $x \mapsto x + 1$ for $x < k$ and $k \mapsto 1$. Consider the block function f where the odd blocks are given by the sequence $L_1, L_2, L_3, L_4, L_5, \dots$ and the even blocks are given by the sequence $L_1, L_1, L_2, L_1, L_2, L_3, \dots$. An initial segment looks like:

$$L_1 + L_1 + L_2 + L_1 + L_3 + L_2 + L_4 + L_1 + L_5 + L_2 + L_6 + L_3 + L_7 + \dots$$

This function is constructed so that it satisfies the following:

- all blocks that occur in the function occur infinitely often,
- no two block types embed in another different block type,
- no two different block types that occur in the function have the same size, and
- no two blocks types are adjacent (in the same order) more than once.

Since each block occurs infinitely often the degree spectrum of f on a cone must contain a non-c.e. degree. To show that the degree spectrum of f on a cone does not contain all Δ_2^0 degrees we show that there is no infinite f -coding sequence. Indeed we will show that any f -coding sequence has length at most five. By Theorems 3.3 and 3.6 this proves the theorem. (For the moreover statement, one must observe that for this particular f no cone is required in the proofs of these theorems.)

Consider any (possibly finite) f -coding sequence $[a_1, b_1], [a_2, b_2], \dots$ and corresponding maps φ_i . We make the following definitions.

- (1) Say that $l_1 < \dots < l_p$ form a *link* in an interval $[a_i, b_i]$ if they form a block of length p in $[a_i, b_i]$. Say that this interval *witnesses* this link.
- (2) Say that a link $l_1 < \dots < l_p$ in $[a_i, b_i]$ is *vulnerable* in $[a_j, b_j]$, $j > i$, if the images $\varphi_{i \rightarrow j}(l_1) < \dots < \varphi_{i \rightarrow j}(l_p)$ are no longer contained in a single block and now intersect two or more different blocks.
- (3) Say that a link $l_1 < \dots < l_p$ in $[a_i, b_i]$ is *broken* in $[a_j, b_j]$, $j > i$, if some element is inserted between the images $\varphi_{i \rightarrow j}(l_1)$ and $\varphi_{i \rightarrow j}(l_p)$, i.e., the l_r are no longer adjacent after applying $\varphi_{i \rightarrow j}$.

We first claim that either there is a link witnessed in $[a_1, b_1]$ that becomes vulnerable in $[a_2, b_2]$, or there is a link witnessed in $[a_2, b_2]$ that becomes vulnerable in $[a_3, b_3]$. The map φ_1 from $[a_1, b_1]$ to $[a_2, b_2]$ is not f -preserving, and so there is a link $l_1 < \dots < l_p$ witnessed in $[a_1, b_1]$ such that φ_1 is not f -preserving on these elements. This link forms a block of size p in $[a_1, b_1]$, and f has only one block type of each size, so either the images of $l_1 < \dots < l_p$ in $[a_2, b_2]$ lie in two different blocks or are contained in some larger block. In the former case, the link $l_1 < \dots < l_p$ has become vulnerable in $[a_2, b_2]$. In the second case, let $k_1 < \dots < k_q$ be this larger block in $[a_2, b_2]$ containing the images of l_1, \dots, l_p . Then $k_1 < \dots < k_q$ is a link witnessed in $[a_2, b_2]$ which becomes vulnerable in $[a_3, b_3]$. This is because $\varphi_{1 \rightarrow 3}$ from $[a_1, b_1]$ to $[a_3, b_3]$ is f -preserving and so the images of $l_1 < \dots < l_p$ form a block in $[a_3, b_3]$; this block is contained in, but not all of, the image of $k_1 < \dots < k_q$ in $[a_3, b_3]$.

Now suppose that we have a link l_1, \dots, l_p which is witnessed in $[a_i, b_i]$ and which becomes vulnerable in $[a_{i+1}, b_{i+1}]$. We claim that it must break in either $[a_{i+2}, b_{i+2}]$ or $[a_{i+3}, b_{i+3}]$. Indeed, in $[a_{i+1}, b_{i+1}]$ the images of the link l_1, \dots, l_p lie in at least two different blocks, and so the same is true in $[a_{i+3}, b_{i+3}]$. Moreover, since the map $\varphi_{i+1 \rightarrow i+3}: [a_{i+1}, b_{i+1}] \rightarrow [a_{i+3}, b_{i+3}]$ is f -preserving, in $[a_{i+3}, b_{i+3}]$ the images of l_1, \dots, l_p lie in blocks of the same block types. However, since $\varphi_{i \rightarrow i+2}: [a_i, b_i] \rightarrow [a_{i+2}, b_{i+2}]$ is f -preserving, the images of l_1, \dots, l_p form a block in $[a_{i+2}, b_{i+2}]$. Thus—as each φ_j is order-preserving— $\varphi_{i+1 \rightarrow i+3}: [a_{i+1}, b_{i+1}] \rightarrow [a_{i+3}, b_{i+3}]$ cannot be the identity. Since each pair of f -blocks only appears adjacent to each other once in f , some element must be inserted between the images in $[a_{i+3}, b_{i+3}]$ of l_1 and l_p ; thus the link must be broken in $[a_{i+3}, b_{i+3}]$ if not earlier.

Now we argue that after a link is broken, the coding sequence must terminate. Suppose that a link l_1, \dots, l_p is witnessed in $[a_i, b_i]$ but broken in $[a_j, b_j]$. Then i and j must be of opposite parities, as if they were of the same parity then the map $\varphi_{i \rightarrow j}: [a_i, b_i] \rightarrow [a_j, b_j]$ would be f -preserving, and the images of l_1, \dots, l_p would form a block in $[a_j, b_j]$, and hence would be a sequence of successive elements. Thus, as i and j have opposite parities, the map $\varphi_{i \rightarrow j+1}$ is f -preserving, so the images of l_1, \dots, l_p in $[a_{j+1}, b_{j+1}]$ must form a block of size p as l_1, \dots, l_p did in $[a_i, b_i]$. However, in $[a_j, b_j]$ and hence in $[a_{j+1}, b_{j+1}]$ the images of l_1, \dots, l_p are no longer successive elements, preventing them from forming such a block. Thus the coding sequence must terminate with $[a_j, b_j]$.

Thus we have a link in $[a_1, b_1]$ or $[a_2, b_2]$ which becomes vulnerable in $[a_2, b_2]$ or $[a_3, b_3]$, and hence broken by $[a_5, b_5]$ or before. There can be no more elements of the coding sequence. \square

4 Many different intermediate spectra

Now that we know that intermediate degree spectrum are possible, we may ask for a more specific description of these intermediate degree spectra. How many possible intermediate degree spectra are there? Are particular interesting classes of degrees possible degree spectra? In this section, we continue to limit the relations we consider to block functions since this is the simplest case. Even so, we find significant complexity. We will particularly focus on the α -c.e. degrees, whose definition we now recall.

Definition 4.1. Let $(\alpha + 1, \prec)$ be (a presentation of) a computable ordinal. A set A is α -c.e. if there are is a computable approximation function $g: \omega^2 \rightarrow \{0, 1\}$ and a computable counting function $r: \omega^2 \rightarrow \alpha + 1$ such that

- (1) for all x , $g(x, 0) = 0$,
- (2) for all x , $r(x, 0) = \alpha$,
- (3) for all x and s , $r(x, s + 1) \preceq r(x, s)$,
- (4) if $g(x, s + 1) \neq g(x, s)$ then $r(x, s + 1) \prec r(x, s)$, and
- (5) $A(x) = \lim_{s \rightarrow \infty} g(x, s)$.

We say that a set is of α -c.e. degree if it is Turing equivalent to an α -c.e. set.

The definition of the α -c.e. sets depends on the chosen the presentation of $\alpha + 1$; indeed, for any Δ_2^0 set, there is some presentation of $\omega^2 + 1$ which makes that set ω^2 -c.e. But for any fixed presentation of $\omega^2 + 1$, there are Δ_2^0 sets which are not ω^2 -c.e.

Any ordinal has a computable presentation on a cone, so when working on a cone we do not need to assume that ordinals are computable. Moreover, by the following remark we also do not need to worry about different presentations for the same ordinal.

Remark 4.2. Let α be an ordinal, and let (P, \prec_P) and (Q, \prec_Q) be two presentations of $\alpha + 1$. We write α_P -c.e. for the α -c.e. sets and degrees defined with respect to the presentation P , and similarly for α_Q -c.e. There is a cone on which (P, \prec_P) and (Q, \prec_Q) are computable, and the α_P -c.e. sets are the same as the α_Q -c.e. sets. This cone is the join of the two presentations and an isomorphism between them. Thus, working on a cone, all presentations for $\alpha + 1$ collapse and so it makes sense to talk about the α -c.e.

sets without reference to presentation. (Note that it is not true that there is a cone on which all presentations are equivalent, but rather that there is in some sense a single presentation up to equivalence in the limit.)

Recall that, while there is no listing of all Δ_2^0 sets, for each (presentation of a) computable ordinal α there is a listing of all of the α -c.e. sets. The following theorem rules out, for block functions, any degree spectrum that can be listed in such a way other than the c.e. sets.

Theorem 4.3. *Let f be a block function which is not intrinsically computable and such that, after some initial segment, all blocks embed into some later block. Let $(X_e^A)_{e \in \omega}$ be, uniformly in A , an A -computable list of (computable approximations to) $\Delta_2^0(A)$ sets each of which computes A . Then, for all A on a cone, the degree spectrum of f relative to A contains some degree which is not Turing equivalent to any set in the listing $(X_e^A)_{e \in \omega}$.*

Corollary 4.4. *Let f be a block function whose degree spectrum on a cone strictly contains the c.e. degrees. Then, for every α , the degree spectrum of f on a cone contains a non- α -c.e. degree.*

Proof of Theorem 4.3. As usual we work on a cone on which we can compute α_f , and assume without loss of generality that all blocks embed into some later block. Suppose $(X_e)_{e \in \omega}$ is a listing as above with $X_{e,s}$ being the limit approximation to X_e . We construct a computable copy \mathcal{A} of $(\omega, <)$ via finite stages, satisfying the requirements

$$R_{e,i,j} : \text{either } \Phi_i^{f^{\mathcal{A}}} \neq X_e \text{ or } \Phi_j^{X_e} \neq f^{\mathcal{A}}$$

The construction is a finite injury priority construction.

The strategy for $R_{e,i,j}$ at stage s is as follows

- (1) To initialize this requirement choose some interval $[a, b]$ which forms an f -block of size greater than one and such that a is greater than the length of the finite linear order \mathcal{A}_{s-1} . Insert new elements at the end of the linear order so that it has length b and restrain the elements in the interval $[a, b]$ for this requirement. Call these elements l_0, \dots, l_{b-a} . We say the requirement is in Phase 0.
- (2) At stage s , if the requirement is in Phase 0, say this requirement requires attention if there are computations

$$\Phi_{i,s}^{X_{e,s}}[l_0, \dots, l_{b-a}] = f^{\mathcal{A}_s}[l_0, \dots, l_{b-a}] \quad \text{with use } u_0$$

and

$$\Psi_{j,s}^{\mathcal{A}_s}[0, \dots, u_0] = X_{e,s}[0, \dots, u_0] \quad \text{with use } v_0.$$

When this requirement acts, restrain all blocks in \mathcal{A}_s which contain or are below some element of the use in the ordering $<_{\mathcal{A}_s}$. Next, insert one element below the block consisting of the elements l_0, \dots, l_{b-a} . For each block that was restrained (except for l_0, \dots, l_{b-a}) check to ensure the value of f on these elements is the same as when it was restrained. If not, then insert new elements below the least element of this block and possibly between the elements of this block to move them up to the image of the original f -block in some other f -block into which it embeds. Further, ensure we add enough new elements on the end to complete this block. This, ensures that for all restrained elements, the value of $f^{\mathcal{A}_{s+1}}$ on these elements is the same in stage s . We say this requirement is in Phase 1.

- (3) At stage s , if the requirement is in Phase $2n$ for $n \geq 1$, say this requirement requires attention if there are computations

$$\Phi_{i,s}^{X_{e,s}}[l_0, \dots, l_{b-a}] = f^{A_s}[l_0, \dots, l_{b-a}] \quad \text{with use } u_{2n}$$

and

$$\Psi_{j,s}^{A_s}[0, \dots, u_{2n}] = X_{e,s}[0, \dots, u_{2n}] \quad \text{with use } v_{2n}$$

When this requirement acts, insert one new element below the block consisting of the elements l_0, \dots, l_{b-a} . For each block that was restrained in Phase 0 (except for l_0, \dots, l_{b-a}) check to ensure the value of f on these elements is the same as when it was restrained. If not, then insert new elements below the least element of this block and possibly between the elements of this block to move them up to the image of the original f -block in some other f -block into which it embeds. Further, ensure we add enough new elements on the end to complete this block. This, ensures that for all restrained elements, the value of $f^{A_{s+1}}$ on these elements is the same in stage s . We say this requirement is in Phase $2n + 1$.

- (4) At stage s , if the requirement is in Phase $2n + 1$ for $n \geq 1$, say this requirement requires attention if there are computations

$$\Phi_{i,s}^{X_{e,s}}[l_0, \dots, l_{b-a}] = f^{A_s}[l_0, \dots, l_{b-a}] \quad \text{with use } u_{2n+1}$$

and

$$\Psi_{j,s}^{A_s}[0, \dots, u_{2n+1}] = X_{e,s}[0, \dots, u_{2n+1}] \quad \text{with use } v_{2n+1}$$

When this requirement acts, insert enough new elements element below the element l_0 and possible between the l_0, \dots, l_{b-a} so that they are moved up to the image of the block containing l_0, \dots, l_{b-a} during Phase $2n$ in some block into which it embeds. This ensures that the value of f on l_0, \dots, l_{b-a} is the same as it was in Phase $2n$. For each block that was restrained in Phase 0 (except for l_0, \dots, l_{b-a}) check to ensure the value of f on these elements is the same as when it was restrained. If not, then insert new elements below the least element of this block and possibly between the elements of this block to move them up to the image of the original f -block in some other f -block into which it embeds. Further, ensure we add enough new elements on the end to complete this block. This ensures that for all restrained elements, the value of f^{A_s} on these elements is the same in stage $s - 1$. We say this requirement is in Phase $2n + 2$.

Note that whenever the requirement is acted on we change the value of f on the l_i , breaking the computation that was found at that stage and ensuring that the requirement is again satisfied when we move to the next stage.

Verification: Since elements are inserted below the l_i corresponding to some requirement only when that requirement or some higher priority requirement requires attention it is enough to check that after a requirement is no longer injured its strategy enters only finitely many phases. This will ensure that this construction produces a copy of $(\omega, <)$ and that all requirements are eventually satisfied, i.e., \mathcal{A} is not equivalent to any of the degrees in the listing.

First, we make the following observations about the construction. Let s_n be the stage where the strategy enters Phase n

- The value of $f^{A_{s_{2n}}}$ is the same on the l_i assigned to this strategy for all n .
- The value of $f^{A_{s_n}}$ on the blocks in the use v_0 which were restrained in Phase 0 is the same for all n .

We now show that X_e changes each time we move to another phase but maintains the same value on some interval for all even phases. In the proof of the previous theorem where we showed the value of the set we were trying to beat oscillated between two possible values but in this case the set must only have a certain value for even phases but is allowed to vary on odd ones.

Lemma 4.5.

- (1) For all even n , $X_{e,s_n}[0, \dots, u_0] = X_{e,s_0}[0, \dots, u_0]$
- (2) For all n , $X_{e,s_n}[0, \dots, u_0] \neq X_{e,s_{n+1}}[0, \dots, u_0]$.

Proof.

- (1) Observe that for the computation $\Psi_{j,s_0}^{A_{s_0}}[0, \dots, u_0] = X_{e,s_0}[0, \dots, u_0]$ its use, except for possibly the l_i , is preserved at all stages and so since the value of $f^{A_{s_n}}$ is the same for all even n , the use of this computation must be preserved at stage s_n for all even n so the original computation must hold, i.e., $\Psi_{j,s_0}^{A_{s_0}}[0, \dots, u_0] = \Psi_{j,s_0}^{A_{s_n}}[0, \dots, u_n]$ and so it follows that $X_{e,s_n}[0, \dots, u_0] = X_{e,s_0}[0, \dots, u_0]$, as desired.
- (2) The computation $\Phi_{i,s_0}^{X_{e,s}}[l_0, \dots, l_{b-a}] = f^{A_{s_0}}[l_0, \dots, l_{b-a}]$ found in Phase 0 has use u_0 and so by part (1) holds at stage s_n for all even n . However, since $f^{A_{s_n}}[l_0, \dots, l_{b-a}] \neq f^{A_{s_{n+1}}}[l_0, \dots, l_{b-a}]$ and $\Phi_{i,s_n}^{X_{e,s}}[l_0, \dots, l_{b-a}] = f^{A_{s_n}}[l_0, \dots, l_{b-a}]$ for all n we must have $\Phi_{i,s_n}^{X_{e,s_n}}[l_0, \dots, l_{b-a}] \neq \Phi_{i,s_{n+1}}^{X_{e,s_{n+1}}}[l_0, \dots, l_{b-a}]$. Hence, the use of the computation must change between stage s_n and s_{n+1} . Further, since either n or $n+1$ is even, the use of one of these computations is u_0 and so we must have $X_{e,s_n}[0, \dots, u_0] \neq X_{e,s_{n+1}}[0, \dots, u_0]$, as desired. \square

This lemma tells us that some element in $[0, \dots, u_0]$ must change value each time the strategy enters a new phase. Now, since X is a Δ_2^0 set, it follows that each finite collection of elements can change only finitely often so this strategy can enter only finitely many stages. \square

Despite the apparent difficulty in describing these intermediate degree spectra one way we can attempt to characterize them is by the least computable ordinal α such that their degree spectra contains all α -c.e. degrees. To do this we introduce the notion of coding trees to better understand which types of finite coding sequences exist for a given function f . To a given f there will be associated two trees, a maximal coding tree and a minimal coding tree. The rank of the maximal coding tree will help to determine whether there is an α -c.e. degree not contained in the degree spectrum, and the rank of the minimal coding tree will help to determine whether every α -c.e. degree is contained in the degree spectrum.

The reason that we will need two different trees is that there are actually two ways in which we might want to be able to move coding elements in a linear order. The first, which was previously mentioned, is the ability to move the coding elements back and forth between two possible values of f to account for changes in a Δ_2^0 set. The second,

which was not as apparent in the case where we had an infinite coding sequence, is the ability to preserve the value of f on the coding sequence while other elements are inserted below them, i.e., the value of f on other coding sequences is changing. So, the rank of the maximal coding tree captures which types of moves are possible when coding only a single element, which is enough for diagonalization, while the minimal coding tree captures which types of moves are possible while ensuring that coding sequences can move independently of each other, which is required for encoding a particular set. However, we will see below that the ranks of the minimal and maximal coding trees only provide an upper and lower bound on the α such that all α -c.e. sets can be coded but it can be much more complicated to determine exactly which sets are in the degree spectrum.

We recall the rank of a tree.

Definition 4.6. Let T be a well-founded tree. Given $\sigma \in T$, the rank of σ is defined by:

- (1) $\text{rank}(\sigma) = 0$ if σ is a leaf.
- (2) $\text{rank}(\sigma) = \sup \{\text{rank}(\tau) + 1 \mid \tau \text{ child of } \sigma\}$.

We define $\text{rank}(T)$ to be the rank of the root node of T , or ∞ if T is non-well-founded.

Definition 4.7. Let f be a block function on $(\omega, <)$. We define its *maximal coding tree* as follows. The elements of the tree will be all finite weak coding sequences, with the empty sequence as the root. Extension on the tree is just extension of coding sequences (including using the same functions φ .) We write, for this tree, $T_{\max}(f)$, and let the *maximal coding rank* of f be $\text{maxrank}(f) = \text{rank}(T_{\max}(f))$.

Theorem 4.8. *Let f be a block function such that every block, except for finitely many, embeds into some later block. If α is the rank of the maximal coding tree for f , then there is some α -c.e. degree which is not in the degree spectrum of f .*

Proof. We work on a cone on which we can compute the usual facts about f as well as a given presentation for α and the maximal coding tree for f together the ranking function on that tree using that presentation of α for the ranks. The construction here is similar to the construction in the second part of the proof of Theorem 3.6 (which shows that if there is no infinite f -coding sequence, then the degree spectrum of f is not all Δ_2^0 degrees). There, we constructed a Δ_2^0 set C while diagonalizing against being Turing equivalent to $f^{\mathcal{A}_e}$ for copies \mathcal{A}_e of $(\omega, <)$. We had requirements

$$R_{e,i,j} : \text{If } \mathcal{L}_e \cong (\omega, <), \text{ then either } \Phi_i^{f^{\mathcal{L}_e}} \neq C \text{ or } \Psi_j^C \neq f^{\mathcal{L}_e}.$$

For each requirement, we chose an element x for which we would change $C(x)$. The fact that there was no infinite f -coding sequence meant that eventually each $R_{e,i,j}$ would be satisfied and $C(x)$ would no longer change; more precisely, we would only change $C(x)$ when we were able to extend the (weak) f -coding sequence that we were building.

The construction now will be exactly the same, except that we will add a rank function $r(x, s) : \omega^2 \rightarrow \alpha + 1$. Given x , we begin with $r(x, 0) = \alpha$. We also maintain at each stage a weak coding sequence $\text{CS}_{x,s}$ with rank in $T_{\max}(f)$ of $\text{rank}(\text{CS}_{x,s}) = r(x, s)$. We begin with $\text{CS}_{x,0}$ as the empty coding sequence, with rank $\alpha = r(x, s)$ in $T_{\max}(f)$. If x is never assigned to a requirement, then $C(x)$ never changes, and we never decrease $r(x, s)$. Otherwise, we do not change $r(x, s)$ and $\text{CS}_{x,s}$ until after x has been assigned to some requirement $R_{e,i,j}$, and then we will only have $r(x, s) < r(x, s - 1)$ at stages s when $R_{e,i,j}$

changes from one phase to the next; these are also the only stages when $C_s(x) \neq C_{s-1}(x)$. The weak coding sequence $\mathbf{CS}_{x,s}$ will be the sequence built by $R_{e,i,j}$ in Theorem 3.6. Suppose that we have defined $r(x, s-1)$ and $\mathbf{CS}_{x,s-1}$ and that at stage s the requirement $R_{e,i,j}$ changes from Phase n to Phase $n+1$. The weak coding sequence $\mathbf{CS}_{x,s-1}$ will have been the sequence of length n constructed at Phase n . In moving to Phase $n+1$, we extend this weak coding sequence to a sequence $\mathbf{CS}_{x,s}$ of length $n+1$ as described in the proof of Theorem 3.6. Define $r(x, s)$ to be the rank of $\mathbf{CS}_{x,s}$ in the maximal coding tree $T_{\max}(f)$. If, at any point, the requirement $R_{e,i,j}$ is injured, then we never again change $C(x)$ and so never again have to decrease $r(x, s)$. Thus we have shown that the set C built is α -c.e., proving the theorem. \square

Note that the condition given in this theorem is sufficient but not necessary. Indeed, suppose that f_k is the function obtained from f by “removing” the first k blocks of f . Then f and f_k have the same degree spectra, but it is possible that $\max\text{rank}(f_k) < \max\text{rank}(f)$.

Example 4.9. Consider the example f from Theorem 3.9. There, we showed that any f -coding sequence has length at most 5. Thus the rank of the maximal coding tree for f is at most 6 (since in the rank of the coding tree we include the empty sequence). Thus by Theorem 4.8 on a cone there is a 6-c.e. degree which is not in the degree spectrum of that function f . Since in that example we can carry out the argument of Theorem 4.8 computably, not working on a cone, there is a 6-c.e. degree which is not in the degree spectrum of f .

We now turn to a condition for the reverse, that is, a coding tree that will allow us to code all α -c.e. sets. This is somewhat more complicated.

Definition 4.10. Let f be a block function on $(\omega, <)$. Suppose that (a) every block embeds into infinitely many later blocks, or equivalently, the degree spectrum of f strictly contains the c.e. degrees on a cone, and (b) there is no infinite f -coding sequence, or equivalently the degree spectrum of f is strictly contained within the Δ_2^0 degrees on a cone.

First, we need an additional definition. Given two coding sequences σ and τ which agree on all but the last interval, which are, say, $[a_n, b_n]$ for σ and $[a'_n, b'_n]$ for τ , we say that σ permits τ if $a_n < a'_n$ and there is a nondecreasing embedding ψ from $[a_n, b_n] \rightarrow [a'_n, b'_n]$ which preserves f and commutes with the maps $\varphi_{n-1}: [a_{n-1}, b_{n-1}] \rightarrow [a_n, b_n]$ and $\varphi'_{n-1}: [a_{n-1}, b_{n-1}] \rightarrow [a'_n, b'_n]$.

$$\begin{array}{ccccccc}
 \cdots & \longrightarrow & [a_{n-2}, b_{n-2}] & \xrightarrow{\varphi_{n-2}} & [a_{n-1}, b_{n-1}] & \xrightarrow{\varphi_{n-1}} & [a_n, b_n] \\
 & & & & & \searrow \varphi'_{n-1} & \downarrow \psi \\
 & & & & & & [a'_n, b'_n]
 \end{array}$$

The fact that each f -block embeds into infinitely many later blocks implies that for any coding sequence σ_0 there is an infinite sequence $\sigma = \sigma_0, \sigma_1, \sigma_2, \dots$ such that each σ_i permits σ_{i+1} .

We define the *minimal coding tree* $T_{\min}(f)$ of f as follows. The elements are all finite coding sequences, with the empty sequence as the root. For the maximal coding tree, we used weak coding sequences, but here we use strong coding sequences. Extension on the

tree is just extension of coding sequences (using the same functions φ). Since there is no infinite f -coding sequence this tree is well-founded.

Define, inductively, a ranking minrank on $T_{\min}(f)$. For a leaf σ , we put $\text{minrank}(\sigma) = 0$. Given a non-leaf $\sigma \neq \emptyset$, we set $\text{minrank}(\sigma) \geq \alpha$ if there is an infinite sequence $\sigma = \sigma_0, \sigma_1, \sigma_2, \dots$ such that each σ_i permits σ_{i+1} and for each $\beta < \alpha$ and each $i = 0, 1, 2, \dots$ there is τ_i a child of σ_i such that $\text{minrank}(\tau_i) \geq \beta$. Then as usual we set $\text{minrank}(\sigma)$ to be the greatest α such that $\text{minrank}(\sigma) \geq \alpha$. We put $\text{minrank}(\emptyset) = \sup_{\sigma \neq \emptyset} \text{minrank}(\sigma) + 1$.

For each non-root-node $\sigma \neq \emptyset$, there is an infinite sequence $\sigma = \sigma_0, \sigma_1, \sigma_2, \dots$ such that for each i , $\text{minrank}(\sigma_i) \geq \text{minrank}(\sigma)$. (For the case when $\text{minrank}(\sigma) = 0$, this is due to the fact that as noted above that because f -block embeds into infinitely many later blocks implies that for any coding sequence σ_0 there is an infinite sequence $\sigma = \sigma_0, \sigma_1, \sigma_2, \dots$ such that each σ_i permits σ_{i+1} .)

We define $\text{minrank}(f)$ to be $\text{minrank}(\emptyset)$ the empty coding sequence which is the root node of $T_{\min}(f)$.

Theorem 4.11. *Let f be a block function such that every block, except for finitely many, embeds into some later block. If $\alpha = \text{minrank}(f)$ is the rank of the minimal coding tree corresponding to f , then, on a cone, the degree spectrum of f contains all β -c.e. degrees for any $\beta < \alpha$.*

Proof of Theorem 4.11. We work on the cone above α_f and the minimal coding tree and its ranking function into a presentation of α . Suppose X is an β -c.e. set, with $\beta < \alpha$, given by functions g, r where g is the computable approximation of X and r is ranking function counting the number of mind changes. We construct a computable copy \mathcal{A} of $(\omega, <)$ such that $f^{\mathcal{A}}$ is Turing equivalent to X . The construction will be almost exactly the same as the first part of Theorem 3.6 but we will be much more careful in how we use our coding sequences. In that construction, we had at each stage s certain elements $\bar{v}_e[s]$ of \mathcal{A}_s which we used to code the value of $X(e)$. In particular, $\bar{v}_e[s]$ corresponded, via the partial isomorphism $\pi_s: \mathcal{A}_s \rightarrow (\omega, <)$, to an interval $[a_i, b_i]$ in a coding sequence. In Theorem 3.6 these intervals $[a_i, b_i]$ were in a fixed infinite f -coding sequence. Now we work within the minimal coding tree. At each stage s , we will have a coding sequence $\text{CS}_{e,s} \in T_{\min}(f)$ with $\text{minrank}(\text{CS}_{e,s}) = r(e, s)$. The length of $\text{CS}_{e,s}$ will be equal to 1 more than the number of times $X(e)$ has changed value before stage s ,

$$\text{length}(\text{CS}_{e,s}) = 1 + \#\{t < s \mid X_t(e) \neq X_{t+1}(e)\}.$$

Suppose that $\text{CS}_{e,s}$ consists of intervals $[a_1, b_1], \dots, [a_n, b_n]$ with $n = 1 + \#\{t < s \mid X_t(e) \neq X_{t+1}(e)\}$. Then we will have $\pi_s(\bar{v}_e[s]) = [a_n, b_n]$. If at stage $s + 1$ we see that $X_{s+1}(e) \neq X_s(e)$ then we know that $r(e, s + 1) < r(e, s)$ and so we can choose an extension $\text{CS}_{e,s+1}$ of $\text{CS}_{e,s}$ with $\text{minrank}(\text{CS}_{e,s+1}) = r(e, s + 1)$ which adds one more interval $[a_{n+1}, b_{n+1}]$. We will add elements to \mathcal{A} so that at stage $s + 1$, $\bar{v}_e[s]$ corresponds to the images in $[a_{n+1}, b_{n+1}]$, under φ_n , of $[a_n, b_n]$:

$$\pi_{s+1}(\bar{v}_e[s]) = \varphi_n(\pi_s(\bar{v}_e[s])).$$

The other possibility is that at stage $s + 1$ we have $X_{s+1}(e) = X_s(e)$ but some elements have already been added to \mathcal{A} below $\bar{v}_e[s]$ for the sake of some $e' < e$. In Theorem 3.6 we had an infinite f -coding sequence and so we had enough room to just move along the sequence to another position $[a_j, b_j]$ with j of the same parity as i . Now, we have limited room; instead, we use the permitting aspect of the tree. There is some other coding

sequence $\text{CS}_{e,s+1}$ which is of the same rank and length as $\text{CS}_{e,s}$, which agrees with $\text{CS}_{e,s}$ except for the last interval $[a'_n, b'_n]$, and which permits $\text{CS}_{e,s}$. There is a non-decreasing and f -preserving map $\psi: [a_n, b_n] \rightarrow [a'_n, b'_n]$. Moreover, we can choose such a coding sequence with a'_n large enough that we can insert elements into \mathcal{A} so that

$$\pi_{s+1}(\bar{v}_e[s]) = \psi(\pi_s(\bar{v}_e[s])).$$

This allows us to adjust \mathcal{A} for e without decreasing the rank of the coding sequence. Thus it is not the case that the coding sequences $\text{CS}_{e,s}$ for e are just growing by extensions; rather, there are also lateral moves when the last interval in $\text{CS}_{e,s}$ changes. Having given the informal idea of the construction, we will now give the formal construction.

Construction: As in Theorem 3.6: For each e , starting from stage e where they are defined, we will have the initial coding elements $\bar{u}_e \in \mathcal{A}$, the full coding segment $\bar{v}_e \in \mathcal{A}$ containing \bar{u}_e , and a restraint $r_e = \max_{\mathcal{A}} \bar{v}_e \in \mathcal{A}$ which is the \mathcal{A} -largest element of \bar{v} . We write each tuple in \mathcal{A} -increasing order. The initial coding elements \bar{u}_e will never change once defined, while the coding segment \bar{v}_e may have more elements added, and the restraint r_e may increase in the \mathcal{A} order. We will use $\bar{v}_e[s]$ and $r_e[s]$ to refer to the values at stage s . We will always have that if $e' \leq e$ then $\bar{v}_e > r_{e'}$ so that the coding segment for e is free of the restraint of e' .

At each stage s for each e we will also have $n_e[s]$ which will be one more than the number of changes⁷ in $X(e)$,

$$n_e[s] = 1 + \#\{t < s \mid X_t(e) \neq X_{t+1}(e)\}$$

and a coding sequence $\text{CS}_{e,s}$ of length $n_e[s]$ and $\text{minrank}(\text{CS}_{e,s}) \geq r(e, s)$. We will generally write $[a_1, b_1], \dots, [a_n, b_n]$ and $\varphi_1, \dots, \varphi_{n-1}$ for the corresponding intervals and maps, taking $n = n_e[s]$ and suppressing e and s for simplicity of notation. Note that $n_e[s]$ will be odd if and only if $X_s(e) = 0$ and even if and only if $X_s(e) = 1$.

Recall that at each stage s , given a tuple $\bar{a} \in \mathcal{A}_s$, $\pi_s(\bar{a})$ is the corresponding tuple in $(\omega, <)$. We also have $f^{\mathcal{A}_s}$ which is the function f on \mathcal{A}_s .

From each stage to the next, we will maintain the following properties:

- (1) At each stage $s \geq e$, the coding elements \bar{v}_e of \mathcal{A} correspond to $\pi_s(\bar{v}_e)$ in $(\omega, <)$ which is the last interval $[a_n, b_n]$ in $\text{CS}_{e,s}$.
- (2) Suppose that at stage $s \geq e$ the coding elements $\bar{v}_e[s]$ corresponded (via $\pi_s: \mathcal{A}_s \rightarrow (\omega, <)$) to the interval $[a_n, b_n]$ in the coding sequence $\text{CS}_{e,s}$. At stage $s + 1$:
 - (a) if $X_{s+1}(e) \neq X_s(e)$, then at stage $s + 1$ the coding elements $\bar{v}[s + 1]$ correspond (via $\pi_{s+1}: \mathcal{A}_{s+1} \rightarrow (\omega, <)$) to the interval $[a_{n+1}, b_{n+1}]$ from the coding sequence $\text{CS}_{e,s+1}$ which extends $\text{CS}_{e,s}$ by this interval. The coding elements $\bar{v}_e[s]$ at stage s correspond at stage $s + 1$ to the images, in the commutative diagram, of the elements of $(\omega, <)$ that they corresponded to at stage s . That is,

$$\pi_{s+1}(\bar{v}_e[s]) = \varphi_{i \rightarrow j}(\pi_s(\bar{v}_e[s])).$$

- (b) if $X_{s+1}(e) = X_s(e)$ then at stage $s + 1$ the coding elements $\bar{v}[s + 1]$ correspond (via $\pi_{s+1}: \mathcal{A}_s \rightarrow (\omega, <)$) to an interval $[a'_n, b'_n]$ which is the last

⁷It will be helpful to assume that $X_s(e) = 0$ for all $s \leq e + 1$.

interval of the coding sequence $\text{CS}_{e,s+1}$ of the same length as $\text{CS}_{e,s}$, with $\text{minrank}(\text{CS}_{e,s+1}) \geq \text{minrank}(\text{CS}_{e,s})$, and such that $\text{CS}_{e,s}$ permits $\text{CS}_{e,s+1}$. Let ψ be the map witnessing this. The coding elements $\bar{v}_e[s]$ at stage s correspond at stage $s+1$ to the images of the elements of $(\omega, <)$ that they corresponded to at stage s . That is,

$$\pi_{s+1}(\bar{v}_e[s]) = \psi(\pi_s(\bar{v}_e[s])).$$

- (3) If, at stage s an element $a \in \mathcal{A}$ is not a coding element (i.e., not part of any $\bar{v}_e[s]$), then at stage $s+1$ it is still not a coding element, and $f^{\mathcal{A}_s}(a) = f^{\mathcal{A}_{s+1}}(a)$.
- (4) From stage $s \geq e$ to stage $s+1$, elements can only be added to \mathcal{A} below r_e if there is $e' \leq e$ which entered or left X at stage $s+1$, i.e., with $e' \in X_s \triangle X_{s+1}$. Thus, if $X_{s+1} \upharpoonright_{e'} = X_s \upharpoonright_{e'}$, then the partial isomorphism $\pi_s: \mathcal{A}_s \rightarrow (\omega, <)$ does not change at stage $s+1$ on elements below r_e .

Stage s : At this stage in the construction our partial linear order \mathcal{A}_s can be partitioned into a finite number of coding segments and padding blocks, each of which form an interval:

$$\bar{p}_0 <_{\mathcal{A}} \bar{v}_0[s] <_{\mathcal{A}} \bar{p}_1 <_{\mathcal{A}} \bar{v}_1[s] <_{\mathcal{A}} \cdots .$$

We add new elements to \mathcal{A}_s to code X_{s+1} .

Starting with $\bar{v}_0[s]$, and continuing in order with $\bar{p}_1, \bar{v}_1[s]$, and so on, we ensure that each of these segments still satisfy our requirements. However we must now take into account the fact that we may have already inserted elements.

- Given a padding segment \bar{p}_i , if no elements have been inserted below this segment, then we do not have to do anything. If there have been other elements, then we must act as follows. Consider each set of elements which corresponded at stage s to a block. We must make sure that those same elements correspond in \mathcal{A}_{s+1} to a block of the same type. For each block making up \bar{p}_i , insert new padding elements below the least element of this block and possibly between the elements of this block to move them up to the image of the original f -block in some other f -block into which it embeds. Further, ensure we add enough padding elements on the end to complete this block. This ensures that for all padding elements present at stage s , the value of $f^{\mathcal{A}_s}$ on these elements is the same as at stage $s+1$.
- Given $\bar{v}_e[s]$ the coding segment corresponding to e first check the value of $X_{s+1}(e)$. We have two cases.
 - If $X_{s+1}(e) \neq X_s(e)$, then $r(e, s+1) < r(e, s)$. Choose a coding sequence $\text{CS}_{e,s+1}$ extending $\text{CS}_{e,s}$ with $\text{minrank}(\text{CS}_{e,s+1}) \geq r(e, s+1)$, which we can do as $r(e, s+1) < r(e, s) \leq \text{minrank}(\text{CS}_{e,s})$. Let $[a_{n+1}, b_{n+1}]$ be the last interval in $\text{minrank}(\text{CS}_{e,s+1})$. We can choose $\text{CS}_{e,s+1}$ such that a_{n+1} is sufficiently large that, by adding elements between and below the elements of $\bar{v}_e[s]$, we can have the segment $\bar{v}_e[s]$ corresponds to the image, under $\varphi_{k \rightarrow \ell}$, of $[a_k, b_k] = \pi_s(\bar{v}_e[s])$. Any new elements that were inserted and end up in the new interval are added to the collection of coding elements $\bar{v}_e[s+1]$ corresponding to e , otherwise the newly added elements are padding elements. Further, we ensure that enough

new coding elements are after the final element to ensure we complete the entire interval, putting these elements in $\bar{v}_e[s+1]$ as well. Thus we have

$$\pi_{s+1}(\bar{v}_e[s]) = \varphi_{i \rightarrow j}(\pi_s(\bar{v}_e[s]))$$

and

$$\pi_{s+1}(\bar{v}_e[s+1]) = [a_{n+1}, b_{n+1}].$$

- If $X_{s+1}(e) = X_s(e)$, then we first ask whether some element has already been inserted below $\bar{v}_e[s]$. If not, then we do not insert any further elements. If so, say k elements have been inserted below $\bar{v}_e[s]$, then we act as follows. We have at this stage a coding sequence $\mathbf{CS}_{e,s}$ with last interval $[a_n, b_n]$. Look for a coding sequence $\mathbf{CS}_{e,s+1}$ of the same length and rank which is permitted by $\mathbf{CS}_{e,s}$, and which has last interval $[a'_n, b'_n]$ with $a'_n \geq a_n + k$. Let ψ be the corresponding non-decreasing f -preserving function $[a_n, b_n] \rightarrow [a'_n, b'_n]$. By inserting elements below and between the elements of $\bar{v}_e[s]$, we can have that the segment $\bar{v}_e[s]$ corresponds, at stage $s+1$, to the image in $[a'_n, b'_n]$, under ψ , of $[a_n, b_n] = \pi_s(\bar{v}_e[s])$. Any new elements that were inserted and end up in the new interval are added to the collection of coding elements $\bar{v}_e[s+1]$ corresponding to e , otherwise the newly added elements are padding elements. Further, we ensure that enough new coding elements are after the final element to ensure we complete the entire interval, putting these elements in $\bar{v}_e[s+1]$ as well. Thus we have

$$\pi_{s+1}(\bar{v}_e[s]) = \psi(\pi_s(\bar{v}_e[s]))$$

and

$$\pi_{s+1}(\bar{v}_e[s+1]) = [a'_n, b'_n].$$

- Finally, we introduce the coding elements corresponding to s . After we have ensured all previously added elements still satisfy the requirements check the value of $X_{s+1}(s)$ (which we may assume is 0, $s \notin X_{s+1}$). Choose a coding sequence \mathbf{CS} of length 1 such that $\text{minrank}(\mathbf{CS}) \geq \beta \geq r(s, s+1)$. Since there is an infinite sequence of such coding sequences each permitting the next, we can choose such a coding sequence \mathbf{CS} such that the last interval $[a, b]$ of that coding sequence has a greater than the length of the linear order \mathcal{A} at this stage. Insert enough new padding elements to the end of \mathcal{A} to extend it to have length a , then add $b - a + 1$ new coding elements corresponding to the interval $[a, b]$. These are the initial coding elements $\bar{u}_s[s+1]$ for s , and make up the coding segment $\bar{v}_s[s+1]$ at this stage. Set $\mathbf{CS}_{s,s+1}$ to be this coding sequence $\mathbf{CS}_{s,s+1} = \mathbf{CS}$.

This ends the construction.

Verification: Much of the verification is the same as in Theorem 3.6. For example, to see that \mathcal{A} is really a computable copy of $(\omega, <)$ we observe as before that for any fixed a there is some e such that elements are only inserted below a when for some stage s and some $e' \leq e$, $X_{s+1}(e) \neq X_s(e)$. Since X is Δ_2^0 , these eventually come to a limit, and so only finitely many elements are inserted below a .

It remains to show that $f^{\mathcal{A}} \equiv_T X$. This is a little more complicated than in Theorem 3.6 due to the lateral movement from one coding sequence to a coding sequence it permits. The analogue of Lemma 3.7 is as follows, and is proved in the same way.

Lemma 4.12. *Suppose that $s < t$.*

- (1) *If, for all $s' with $s \leq s' \leq t$ we have $X_{s'}(e) = X_s(e) = X_t(e)$, then $\text{CS}_{e,s}$ is the same length as and permits $\text{CS}_{e,t}$, say as witnessed by the f -preserving function $\psi: [a_n, b_n] \rightarrow [a'_n, b'_n]$ on the last intervals of these coding sequences. Moreover,$*

$$[a'_n, b'_n] \supseteq \pi_t(\bar{v}_e[s]) = \psi(\pi_s(\bar{v}_e[s])) = \psi[a_n, b_n].$$

- (2) *If $s' \geq s$ is the least stage with $X_{s'+1}(e) \neq X_{s'}(e)$ then $\text{CS}_{e,s}$ is the same length (say m) as and is either equal to or permits $\text{CS}_{e,s'}$, as witnessed by the f -preserving function $\psi: [a_m, b_m] \rightarrow [a'_m, b'_m]$ on the last intervals of these coding sequences, and $\text{CS}_{e,t}$ (say of length $n > m$) strictly extends $\text{CS}_{e,s'}$. (If $\text{CS}_{e,s}$ and $\text{CS}_{e,s'}$ are the same then ψ is the identity.) Moreover,*

$$[a_n, b_n] \supseteq \pi_t(\bar{v}_e[s]) = \varphi_{m \rightarrow n} \circ \psi(\pi_s(\bar{v}_e[s])) = \varphi_{m \rightarrow n} \circ \psi[a_m, b_m].$$

In particular, if m and n are the same parity then $\varphi_{m \rightarrow n} \circ \psi$ is f -preserving.

First we show that $f^A \geq_T X$. Given some element $e \in X$ run the above construction, which is computable, until stage $e + 1$ when the initial coding elements \bar{u}_e corresponding to e are added. Now, compute the value of f^A on these elements. If f^A is the same on \bar{u}_e as it was at stage $e + 1$, then $X(e) = X_e(e + 1)$. Otherwise, if f^A is different on \bar{u}_e than it was at stage $e + 1$, then $X(e) = 1 - X_e(e + 1)$.

Finally, we show that $X \geq_T f^A$. Given some element $a \in \mathcal{A}$ run the above construction until a is added to the linear order. If a is added as a padding element then the construction ensures that $f^A(a)$ does not change and so we take the value at this stage. If a is a coding element corresponding to e then $f^A(a)$ takes on one of two values depending on $X(e)$. Say that a first appears in \mathcal{A} at a stage s in a coding block $\bar{v}_e[s]$, corresponding at that stage to $[a_i, b_i]$ and with $X_s(e) = k$. If $X(e) = X_s(e) = k$, then $f^A(a)$ is the same as it was at stage s . Otherwise, suppose that $X(e) \neq X_s(e) = k$. Then, at some stage $t > s$, we find that $X(e) = X_t(e) = 1 - k$. At this stage, a is an element of $\bar{v}_e[t]$ which corresponds to $[a_j, b_j]$ with j of a different parity from i . Then $f^A(a)$ is the same as it was at stage t . \square

Remark 4.13. Indeed, the proof can be used to show that all α -computable degrees are contained in the degree spectrum of f , where a set A is α -computable if there are is a computable approximation function $g: \omega^2 \rightarrow \{0, 1\}$ and a computable counting function $r: \omega^2 \rightarrow \alpha$ such that

- (1) for all x , $r(x, 0) < \alpha$,
- (2) for all x and s , $r(x, s + 1) \preceq r(x, s)$,
- (3) if $g(x, s + 1) \neq g(x, s)$ then $r(x, s + 1) \prec r(x, s)$, and
- (4) $A(x) = \lim_{s \rightarrow \infty} g(x, s)$.

The difference is that we do not require that $g(x, 0) = 0$, so that we may begin with $x \notin A$ or $x \in A$. An α -computable set is α -c.e. but not necessarily vice versa. For degrees, the difference is mostly important when α is a limit ordinal.

Example 4.14. Consider the function f from Theorem 3.9. It was shown that, given any f -coding sequence $[a_1, b_1], [a_2, b_2], \dots$, there must a link which is vulnerable in either $[a_2, b_2]$ or $[a_3, b_3]$. We show that the minimal coding tree of f has $\text{minrank} \leq 3$ by showing any coding sequence with a vulnerable link has rank 0. Consider the case of a coding sequence $[a_1, b_1], [a_2, b_2], [a_3, b_3]$ with a link ℓ_1, \dots, ℓ_k witnessed in $[a_2, b_2]$ which is vulnerable in $[a_3, b_3]$. (The case of a link witnessed in $[a_1, b_1]$ which is vulnerable in $[a_2, b_2]$ is similar.) Consider a coding sequence $[a_1, b_1], [a_2, b_2], [a'_3, b'_3]$ which is permitted by $[a_1, b_1], [a_2, b_2], [a_3, b_3]$. The image in $[a_3, b_3]$ under φ_2 of the link ℓ_1, \dots, ℓ_k in $[a_2, b_2]$ intersects, in $[a_3, b_3]$, at least two blocks. Since any two blocks appear adjacent to each other in f at most once, the image of these blocks in $[a'_3, b'_3]$ under ψ are not adjacent and so have some element between them. But $\psi \circ \varphi_2$ is the map $\varphi'_2: [a_2, b_2] \rightarrow [a'_3, b'_3]$ from the coding sequence $[a_1, b_1], [a_2, b_2], [a'_3, b'_3]$, and so in $[a'_3, b'_3]$ the image of ℓ_1, \dots, ℓ_k under φ'_2 intersects two blocks which have had some element, not in the image of the link, between them. That is, ℓ_1, \dots, ℓ_k is actually broken in $[a'_3, b'_3]$. Thus the coding sequence $[a_1, b_1], [a_2, b_2], [a'_3, b'_3]$ has no extensions and is of minrank zero. Since $[a_1, b_1], [a_2, b_2], [a_3, b_3]$ only permits coding sequences of minrank zero, it also has minrank zero. Thus $\text{minrank}(f) \leq 3$ as any coding sequence with three elements has minrank zero.

It is not hard to show that $\text{minrank}(f) \geq 3$ so that $\text{minrank}(f) = 3$. Let $\ell < m < n$. Consider the coding sequences $[a_1, b_1]$ where $[a_1, b_1]$ has block type L_ℓ , an ℓ -loop. For any two such coding sequences, the lesser one permits the greater one. These coding sequences can be extended to coding sequences $[a_1, b_1], [a_2, b_2]$ where $[a_2, b_2]$ has block type L_m , and m -loop, say under the map $\varphi_1: [a_1, b_1] \mapsto [a_2, a_2 + b_1 - a_1] \subseteq [a_2, b_2]$. For any two such coding sequences with the same first interval $[a_1, b_1]$, the lesser one permits the greater one. Finally, any such sequence $[a_1, b_1], [a_2, b_2]$ extends to a sequence $[a_1, b_1], [a_2, b_2], [a_3, b_3]$ where the image of $[a_2, b_2]$ in $[a_3, b_3]$ is made up of two blocks, one of block type L_ℓ , and the other of block type L_n . $[a_3, b_3]$ may have further blocks in between these. This shows that any $[a_1, b_1], [a_2, b_2]$ as above has minrank one, and any $[a_1, b_1]$ as above has minrank two. Thus $\text{minrank}(f) = 3$.

Theorems 4.8 and 4.11 only tell us that on a cone the degree spectrum of this function f contains all 2-c.e. degrees but not all 6-c.e. degrees, but does not give any information about the degrees between these. A more ad-hoc argument can be given to show that, in fact, the degree spectrum of f does not contain all 3-c.e. degrees. This argument relies on the fact that any sequence permitting a vulnerable sequence has rank 0 and so this can be accounted for ahead of time in the priority argument by reserving an element which will force the coding sequence produced by the function to move to one which is permitted by it. However, one could imagine this will not happen more generally for any function of minrank three as this argument depends on the number of times one sequence permits a vulnerable sequence before decreasing its maximal rank. In fact, we believe that one could produce an example of a function with minimal rank 3 and maximal rank 6 but whose degree spectrum contains all 4-c.e. degrees. So, the difference between the minimal and maximal ranks arises not from a difference in the lengths of the coding sequences, but from the more complicated behavior underlying which sequences permit each other. We believe that by coming up examples with different permitting behaviours, one might be able to find, e.g., incomparable degree spectra on a cone.

Conjecture 4.15. *There are block functions f and g on $(\omega, <)$ whose degree spectra on a cone are incomparable.*

We will give some examples in this paper but we will focus on cases in which the

minimal and maximal coding trees match up at various ordinals α . We say that an ordinal α is even if it is of the form $\lambda + n$ where λ is a limit ordinal and $n \in \omega$ is even; otherwise, we say that α is odd, in which case it is of the form $\lambda + n$ with n odd.

Theorem 4.16. *Fix $\alpha \geq 6$ even. There is a block function f on $(\omega, <)$ whose degree spectrum on a cone contains all of the β -c.e. degrees for $\beta < \alpha$ and does not contain all of the α -c.e. degrees.*

Proof. Let $T \subseteq \omega^{<\omega}$ be a tree of rank α , with rank function $r: T \rightarrow \alpha \cup \{\alpha\}$. Thus $r(\emptyset) = \alpha$ is even. We may choose T such that for each $\sigma \in T$, $r(\sigma)$ has the same parity as the length of σ , i.e., if σ is of even length then $r(\sigma)$ is an even ordinal, and if σ is of odd length then $r(\sigma)$ is an odd ordinal.

Fix some Gödel number ℓ of the elements of $\omega^{<\omega}$ such that if $|\sigma| < |\tau|$, $\ell(\sigma) < \ell(\tau)$. For each $\sigma = \langle a_0, \dots, a_n \rangle \in \omega^{<\omega}$, with $\sigma \neq \emptyset$, associate with σ the following block type B_σ :

$$x_0 + L_{2^{\ell(\sigma|_0)+2}} + L_{2^{\ell(\sigma|_1)+2}} + \dots + L_{2^{\ell(\sigma|_n)+2}} + x_1 + x_2 + x_3$$

where if $|\sigma|$ is even we let

$$x_0 \rightarrow x_3 \rightarrow x_2 \rightarrow x_1 \rightarrow x_1$$

and if $|\sigma|$ is odd we let:

$$x_0 \rightarrow x_3 \rightarrow x_2 \rightarrow x_1 \rightarrow x_2$$

We call x_0, x_1, x_2, x_3 the sandwich elements of the block.

Notice that each of these blocks have different lengths (in base 2 their lengths have 1's in positions uniquely determined by the initial segments of σ) and that each of their lengths are even. Now, we define f as follows: The even blocks are the sequence L_1, L_3, L_5, \dots of loops of odd lengths. The odd blocks are given by $B_{\sigma_1}, L_{j(1)}, B_{\sigma_2}, L_{j(1)}, \dots$ where $\sigma_1, \sigma_2, \dots$ is a recursive enumeration of the elements of T where each element appears infinitely often. Similarly, j is a recursive enumeration of the odd integers such that each element appears infinitely often but with the additional condition that $j(i) \leq 2i + 1$ for all i . As in Theorem 3.9, f satisfies the following properties:

- all blocks that occur in the function occur infinitely often
- no two different block types that occur in the function have the same size
- no two blocks types are adjacent (in the same order) more than once

There was one property in Theorem 3.9 that now fails: previously, no block type embedded into any other block type, but now there are many block types which embed into each other. The reader should note that the first bullet point is the reason that we include the blocks $L_{j(i)}$, and is what allows the degree spectrum of f on a cone to contain non-c.e. degrees. They play no other role in the proof.

Now, notice that T embeds into the minimal coding tree of f . Given $\sigma \in T$ we can associate to it the coding sequence $[a_1, b_1], \dots, [a_{|\sigma|}, b_{|\sigma|}]$ where $[a_i, b_i]$ is an interval with $a_i > b_{i-1}$ and which is a block of the form $B_{\sigma|_i}$. (Note that there is no block for the empty sequence, but rather the root of T maps to the empty coding sequence.) The maps $\varphi_i : B_{\sigma|_i} \rightarrow B_{\sigma|_{(i+1)}}$ are the natural maps sending each x_j in one block to the corresponding x_j in the other block, and sending each loop to the loop of the same length. For each σ this will be an f -coding sequence and it will be in the minimal coding tree since each interval in the sequence is a single block and so, since each block appears

infinitely often, is always permitted by some later copy of the block. By Theorem 4.11, the degree spectra of f contains all β -c.e. degrees for $\beta < \alpha$.

We consider two possible types of coding sequences. In $[a_1, b_1]$ there must be some element x such that $\varphi_1(f(x)) \neq f(\varphi_1(x))$. In $[a_1, b_1]$ this element is part of a block, say of type I . Consider the image of this block in $[a_2, b_2]$ under φ_1 . If the image is split between two different blocks, then we have a link in $[a_1, b_1]$ that becomes vulnerable in $[a_2, b_2]$. Otherwise, the image is contained entirely in one block, say of type J . Now this block embeds, by φ_2 , in $[a_3, b_3]$. If the image is split between two different blocks, then we have a link in $[a_2, b_2]$ that becomes vulnerable in $[a_3, b_3]$. Otherwise, the entire image is contained in a single block, say of type K . We will concentrate on this case, returning later to the two cases of a link that becomes vulnerable.

We may assume, without loss of generality by shrinking intervals, that $[a_0, b_0]$ is a block of type I , and that each $[a_{n+1}, b_{n+1}]$ is the closure, under blocks, of the image of $[a_n, b_n]$ under φ_n . In particular, $[a_1, b_1]$ is a block of type J , and $[a_2, b_2]$ is a block of type K . Since no two distinct block types have the same size, J must be larger than I , and so K must be larger than I . In particular, I must be a block of type B_{σ_1} and K of type B_{σ_3} for some $\sigma_1 \prec \sigma_3$ whose lengths have the same parity. Let x_0, x_1, x_2, x_3 be the four sandwich elements from I with either

$$x_0 \rightarrow x_3 \rightarrow x_2 \rightarrow x_1 \rightarrow x_1$$

or

$$x_0 \rightarrow x_3 \rightarrow x_2 \rightarrow x_1 \rightarrow x_2.$$

The only elements of this kind are the four sandwich elements from a block B_τ . Thus the f -preserving map $\varphi_{1 \rightarrow 3}: [a_1, b_1] \rightarrow [a_3, b_3]$ must map the sandwich elements of I to those of K , i.e.,

$$a_1 \mapsto a_3, \quad b_1 - 2 \mapsto b_3 - 2, \quad b_1 - 1 \mapsto b_3 - 1, \quad b_1 \mapsto b_3.$$

Then φ_1 must map the sandwich elements of I to those of J ,

$$a_1 \mapsto a_2, \quad b_1 - 2 \mapsto b_2 - 2, \quad b_1 - 1 \mapsto b_2 - 1, \quad b_1 \mapsto b_2,$$

and φ_2 must map the sandwich elements of J to those of K ,

$$a_2 \mapsto a_3, \quad b_2 - 2 \mapsto b_3 - 2, \quad b_2 - 1 \mapsto b_3 - 1, \quad b_2 \mapsto b_3.$$

(For example, if a_1 did not map to a_2 , then we would have $\varphi_1(a_1) > a_2$ and so $a_3 = \varphi_2(\varphi_1(a_1)) > \varphi_2(a_2) \geq a_3$.) That is, in the maps $I \rightarrow J \rightarrow K$ induced by φ_1 and φ_2 , the sandwich elements are mapped to each other. The map $\varphi_{2 \rightarrow 4}: [a_2, b_2] \rightarrow [a_4, b_4]$, since it preserves f , must map the sandwich elements of J to the sandwich elements of another block, say of type M . Moreover, arguing similarly, φ_3 must map the elements of K inside of M while mapping sandwich elements to sandwich elements. Continuing in this way, the sequence $[a_1, b_1], [a_2, b_2], \dots$ must consist of blocks of the form $B_{\sigma_1}, B_{\tau_1}, B_{\sigma_2}, B_{\tau_2}, \dots$ where $\sigma_1 \prec \sigma_2 \prec \dots$ and $\tau_1 \prec \tau_2 \prec \dots$. Moreover, the lengths of the σ_i are all of the same parity as each other, and the lengths of the τ_i are also of the same parity as each other. None of the σ_i and τ_i are \emptyset , the root of T . Thus in T , $r(\sigma_i), r(\tau_i) < \alpha$.

Now we return to the cases we skipped where there is a link which becomes vulnerable, and argue that any such f -coding sequence has length at most 5: For any f -coding sequence $[a_1, b_1], [a_2, b_2], \dots$ and maps φ_i , as shown in Theorem 3.9, if there is some link

in $[a_i, b_i]$ which is vulnerable in $[a_j, b_j]$ then the link will be broken by $[a_{j+2}, b_{j+2}]$ and so the coding sequence must terminate.

Thus, in the maximal coding tree, we have certain (finite) maximal paths of length ≤ 6 , e.g., those consisting of an f -coding sequence where a link becomes vulnerable,⁸ and all other paths are induced by a pair of paths through T . This will imply, since $\alpha \geq 5$, that the rank of the maximal coding tree is at most $\alpha + 1$. Indeed, consider the tree T^* of pairs of sequences $\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \sigma_n, \tau_n \rangle$ of non-root elements of T such that (a) $\sigma_1 \prec \sigma_2 \prec \dots$ and $\tau_1 \prec \tau_2 \prec \dots$, (b) the lengths of the σ_i are all of the same parity as each other, and (c) the lengths of the τ_i are also of the same parity as each other. By (b) and (c), the ranks of the σ_i are either all even or all odd, and similarly for the τ_i . Since each σ_i and τ_i is not the root \emptyset of T , their ranks are all $< \alpha$. Let rank^* be the rank function on this tree T^* , which is clearly well-founded. Then we claim that

(1) Recalling that α is even,

$$\text{rank}^*(\emptyset) \leq \alpha.$$

(2) If σ_1 has even length then

$$\text{rank}^*(\langle \sigma_1 \rangle) \leq r(\sigma_1) + 1,$$

and if σ_1 has odd length then

$$\text{rank}^*(\langle \sigma_1 \rangle) \leq r(\sigma_1)$$

(3) If $r(\sigma_n)$ has even length then

$$\text{rank}^*(\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \tau_{n-1}, \sigma_n \rangle) \leq r(\sigma_n) + 1,$$

and if $r(\sigma_n)$ has odd length then

$$\text{rank}^*(\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \tau_{n-1}, \sigma_n \rangle) \leq r(\sigma_n).$$

If $r(\tau_{n-1})$ is even then

$$\text{rank}^*(\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \tau_{n-1}, \sigma_n \rangle) \leq r(\tau_{n-1}),$$

and if $r(\tau_{n-1})$ is odd then

$$\text{rank}^*(\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \tau_{n-1}, \sigma_n \rangle) \leq r(\tau_{n-1}) - 1.$$

(4) If $r(\tau_n)$ has even length then

$$\text{rank}^*(\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \sigma_n, \tau_n \rangle) \leq r(\tau_n) + 1,$$

and if $r(\tau_n)$ has odd length then

$$\text{rank}^*(\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \sigma_n, \tau_n \rangle) \leq r(\tau_n).$$

If $r(\sigma_n)$ is even then

$$\text{rank}^*(\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \sigma_n, \tau_n \rangle) \leq r(\sigma_n),$$

and if $r(\sigma_n)$ is odd then

$$\text{rank}^*(\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \sigma_n, \tau_n \rangle) \leq r(\sigma_n) - 1.$$

⁸All f -coding sequences have length 5, but in the tree this path starts with the empty f -coding sequence and so this corresponds to a path of length 6.

We argue inductively for (3). Suppose that we have $\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \tau_{n-1}, \sigma_n, \tau_n \rangle$ an extension of rank β extending $\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \tau_{n-1}, \sigma_n \rangle$. We have several cases to check.

- Suppose that τ_{n-1} and τ_n both have even length (and hence $r(\tau_{n-1})$ and $r(\tau_n)$ are even). Then, inductively, $\beta \leq r(\tau_n) + 1$. Since τ_n is an extension of τ_{n-1} and the length of τ_n is at least two greater than the length of τ_{n-1} , $\beta + 1 \leq r(\tau_n) + 2 \leq r(\tau_{n-1})$ and so

$$\text{rank}^*(\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \tau_{n-1}, \sigma_n \rangle) \leq r(\tau_{n-1}).$$

- Suppose that τ_{n-1} and τ_n both have odd length (and hence $r(\tau_{n-1})$ and $r(\tau_n)$ are odd). Then, inductively, $\beta \leq r(\tau_n)$. Since τ_n is an extension of τ_{n-1} and the length of τ_n is at least two greater than the length of τ_{n-1} , $\beta + 1 \leq r(\tau_n) + 1 \leq r(\tau_{n-1}) - 1$ and so

$$\text{rank}^*(\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \tau_{n-1}, \sigma_n \rangle) \leq r(\tau_{n-1}) - 1.$$

- Suppose that σ_n has even length so that $r(\sigma_n)$ is even. Then, inductively, $\beta \leq r(\sigma_n)$ and so $\beta + 1 \leq r(\sigma_n) + 1$. Thus

$$\text{rank}^*(\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \tau_{n-1}, \sigma_n \rangle) \leq r(\sigma_n) + 1.$$

- Suppose that σ_n has odd length so that $r(\sigma_n)$ is odd. Then, inductively, $\beta \leq r(\sigma_n) - 1$ and so $\beta + 1 \leq r(\sigma_n)$. Thus

$$\text{rank}^*(\langle \sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \tau_{n-1}, \sigma_n \rangle) \leq r(\sigma_n).$$

(4) can be argued similarly. To argue (2), we have two cases. Suppose $\langle \sigma_1, \tau_1 \rangle$ is an extension of $\langle \sigma_1 \rangle$ of rank β .

- If σ_1 has even length, then $\beta \leq r(\sigma_1)$ and so $\beta + 1 < r(\sigma_1) + 1$. Thus

$$\text{rank}^*(\langle \sigma_1 \rangle) \leq r(\sigma_1) + 1.$$

- If σ_1 has odd length, then $\beta \leq r(\sigma_1) - 1$ and so $\beta + 1 \leq r(\sigma_1)$. Thus

$$\text{rank}^*(\langle \sigma_1 \rangle) \leq r(\sigma_1).$$

Finally, we argue (1). Suppose that $\langle \sigma_1 \rangle$ is an extension of \emptyset of rank β . Then we have two cases.

- If σ_1 has odd length, then $\beta \leq r(\sigma_1) < \alpha$. Thus $\beta + 1 \leq \alpha$ and so

$$\text{rank}^*(\emptyset) \leq \alpha.$$

- If σ_1 has even length, then $\beta \leq r(\sigma_1) + 1$. Since σ_1 has even length, $r(\sigma_1)$ is an even ordinal, and $r(\sigma_1) < \alpha$ where α is also an even ordinal. Thus $r(\sigma_1) + 1 < \alpha$. Hence $\beta + 1 \leq \alpha$ and so

$$\text{rank}^*(\emptyset) \leq \alpha.$$

This finishes the argument. In particular, we have shown that $\text{rank}^*(T^*) = \text{rank}^*(\emptyset) \leq \alpha$.

Thus, since $\alpha \geq 5$, we have shown that the rank of the maximal coding tree $T_{\max}(f)$ is at most $\alpha + 1$. By Theorem 4.8, on a cone the degree spectrum of f does not contain all $(\alpha + 1)$ -c.e. degrees. \square

We do not know if this can be proved for α an odd ordinal, and there is some reason to think that it might not be true for odd ordinals.

Question 4.17. Is the statement of Theorem 4.16 provable for α an odd ordinal?

We end this paper with several other questions and conjectures. Though we state these for block functions, we are also interested more generally in any relation R on $(\omega, <)$.

Question 4.18. If the degree spectrum on a cone of a block function f contains the non-c.e. degrees, must it contain all 2-c.e. degrees?

Question 4.19. Is there an infinite descending sequence of degree spectra on a cone of block functions?

Question 4.20. Is there an exact classification of when the degree spectrum of a block function f on a cone contains all of the α -c.e. degrees?

References

- [AHTS21] Uri Andrews, Matthew Harrison-Trainor, and Noah Schweber. The property “arithmetic-is-recursive” on a cone. *J. Math. Log.*, 21(3):Paper No. 2150021, 18, 2021.
- [BK23] Nikolay Bazhenov and Dariusz Kalociński. Degree Spectra, and Relative Acceptability of Notations. In Bartek Klin and Elaine Pimentel, editors, *31st EACSL Annual Conference on Computer Science Logic (CSL 2023)*, volume 252 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [BKW22] Nikolay Bazhenov, Dariusz Kalociński, and Michał Wrocławski. Intrinsic complexity of recursive functions on natural numbers with standard order. In *39th International Symposium on Theoretical Aspects of Computer Science*, volume 219 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 8, 20. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2022.
- [BKW24] Nikolay Bazhenov, Dariusz Kalociński, and Michał Wrocławski. Degrees of relations on canonically ordered natural numbers and integers. *Archive for Mathematical Logic*, 2024.
- [CHT17] Barbara F. Csima and Matthew Harrison-Trainor. Degrees of categoricity on a cone via η -systems. *J. Symb. Log.*, 82(1):325–346, 2017.
- [DKMY09] R. Downey, B. Khoussianov, J. Miller, and L. Yu. Degree spectra of unary relations on $(\omega, <)$. *proceedings of the International Congress in Logic, Methodology and Philosophy of Science, Beijing, 2007*, pages 36–55, 2009.
- [HT18] Matthew Harrison-Trainor. Degree spectra of relations on a cone. *Mem. Amer. Math. Soc.*, 253(1208):v+107, 2018.

- [Kno09] C. Knoll. Degree spectra of unary relations on ω and ζ . Master's thesis, University of Waterloo, 2009.
- [Mar68] Donald A. Martin. The axiom of determinateness and reduction principles in the analytical hierarchy. *Bull. Amer. Math. Soc.*, 74:687–689, 1968.
- [Mar75] Donald A. Martin. Borel determinacy. *Ann. of Math. (2)*, 102(2):363–371, 1975.
- [Mon13] Antonio Montalbán. A computability theoretic equivalent to Vaught's conjecture. *Adv. Math.*, 235:56–73, 2013.
- [Mon19] Antonio Montalbán. Martin's conjecture: a classification of the naturally occurring Turing degrees. *Notices Amer. Math. Soc.*, 66(8):1209–1215, 2019.
- [Sco65] Dana Scott. Logic with denumerably long formulas and finite strings of quantifiers. In *Theory of Models (Proc. 1963 Internat. Sympos. Berkeley)*, pages 329–341. North-Holland, Amsterdam, 1965.
- [Wri18] Matthew Wright. Degrees of relations on ordinals. *Computability*, 7(4):349–365, 2018.