Degree spectra of functions on ω on a cone

Jad Damaj
Mentor: Matthew Harrison-Trainor*

July 29, 2023

Abstract

We examine the degree spectra of unary functions on $(\omega, <)$ viewed as a computable structure. We answer a question posed by Wright about the possible degree spectrum on this structure and in order to do so we look more closely at unary block functions and classify their degree spectra on a cone into one of 3 classes: the c.e. degrees, all Δ_2^0 degrees, or intermediate (strictly containing the c.e. degrees but not containing all Δ_2^0 degrees). Further, we examine these intermediate degree spectra more closely and offer a description as to what the possible degrees they contain are, as well as providing an example of a function with intermediate degree spectra.

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 $(\omega, <)$. 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.

In this paper we follow a series of papers [Wri18, BKWa22, HT18] studying the degree spectra of 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 $S^{\mathcal{L}}$. $S^{\mathcal{L}}$ 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 $f_{\mathcal{L}} \colon \mathcal{L} \to (\omega, <)$, 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 $\mathcal{R}^{\mathcal{L}} \leq_T f_{\mathcal{L}} \leq_T S^{\mathcal{L}} \leq_T \varnothing'$. For many relations R, we also always have $R^{\mathcal{L}} \geq_T S^{\mathcal{L}}$, so that the degree

*This work was supported by the National Science Foundation under Grant DMS-2153823. Jad Damaj was supported by an REU at the University of Michigan funded under this grant.

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 x is the successor of y if and only if there is $z > y$ such that z is the double-successor of x. This 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.)

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^0_2 degrees (such as an infinite and co-infinite unary relation). Because $(\omega, <)$ is $0'$-categorical, every degree spectrum of a computable relation on $(\omega, <)$ is contained within the Δ^0_2 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 Δ^0_2 degrees. Thus no degree spectrum could be intermediate between the computable degree and the c.e. degrees. Wright left open the question of the which degree spectra intermediate between the c.e. degrees and the Δ^0_2 degrees were possible.

In [BKWa22], 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 diagonalization argument. It is ...

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" formalism to capture this notion. Degree spectra on a cone were first studied in the second author's monograph [HT18], where relations on $(\omega, <)$ were specifically considered.[1] In this paper 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? While [BKWa22] resolved Wright's question, it does not resolve the on-a-cone version; the degree spectrum of that relation is the c.e. degrees on a cone.

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

Theorem 1.1. *There are computable relations on $(\omega, <)$ whose degree spectrum is strictly in between the c.e. degrees and the Δ^0_2 degrees on a cone.*

In particular, there is a natural relation of intermediate degree spectrum. To illustrate how natural these examples are, we describe our example. Our example will also be a unary function f. We write I_n for the loop of length n. Then f consists of the following blocks:

$$I_1 I_1 I_2 I_1 I_3 I_2 I_4 I_1 I_5 I_2 I_6 I_3 I_7 I_1 I_8 \ldots .$$

The pattern here is that the blocks in odd positions follow the pattern $I_1 I_2 I_3 I_4 \ldots$ enumerating the natural numbers in increasing order, while the blocks in even positions $I_1 I_1 I_2 I_1 I_2 I_3 \ldots$ 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 only once.

While we can describe our example simply, the example of [BKWa22] 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 how one does Gödel numberings.

[1] Though 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].

2 Preliminaries

Definition 2.1. Given a computable structure $\mathcal{A}$ and a relation R on $\mathcal{A}$, we define the degree spectrum of R, $\mathrm{DgSp}_{\mathcal{A}}(R)$, to be the set of degrees

$$\{\varphi(R)\colon (\varphi,\mathcal{B}) \text{ where } \varphi : \mathcal{A} \cong \mathcal{B} \text{ and } \mathcal{B} \text{ is computable}\}$$

ie. the images of R is all computable copies of $\mathcal{A}$ under all isomorphisms.

If $\mathcal{A}$ is X-computable, we define the degree spectrum of R relative to X, $\mathrm{DgSp}^X_{\mathcal{A}}(R)$, to be the set of degrees

$$\{\varphi(R) \oplus X\colon (\varphi,\mathcal{B}) \text{ where } \varphi : \mathcal{A} \cong \mathcal{B} \text{ and } \mathcal{B} \leq X\}.$$

Definition 2.2. We say that the degree spectra of two relations R and S on $\mathcal{A}$ and $\mathcal{B}$ are equal on a cone if there is some X such that for all $Y \geq_T X$, $\mathrm{DgSp}^Y_{\mathcal{A}}(R) = \mathrm{DgSp}^Y_{\mathcal{A}}(S)$.

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 $Y \geq_T X$, $\mathrm{DgSp}^Y_{\mathcal{A}}(R)$ is the set of Y-c.e. degrees above Y. Similarly, we can define computable or Δ^0_2 on a cone.

Definition 2.3. We say that a function $f : \omega \to \omega$ 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 containing n.

Note that since each f-block is finite we can fix some computable enumeration of the f-blocks. We denote the nth block in this enumeration by I_n and will use this to provide an alternate way to represent an arbitrary block function.

Definition 2.4. Given a block function f, we define sequence, α_f, of natural numbers called the string corresponding to f by $\alpha_f(i) = n$ where I_n is the ith block that occurs in f.

3 Results

In order to determine which block functions have intermediate degree spectra it is useful to first determine the conditions under which the degree spectrum of a function is the c.e. degrees or all Δ^0_2 degrees. We begin by providing a necessary and equivalent condition for when the degree spectrum of a function is exactly the c.e. degrees using a more general criterion for when the degree spectrum of some relation contains a non-c.e degree.

Definition 3.1. We say that $\bar{a}$ is difference-free[2] (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 $\bar{c},\bar{a}'',\bar{b}''$.

Theorem 3.2 (Proposition 3.4 of [HT18])**.** *Theorem that says that if d-free elements exist, then not intrinsically of c.e. degree.*

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

Theorem 3.3. *Suppose f is a block function which is not intrinsically computable. 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. For the forward direction of this proof, we work on the cone above α_f, i.e., relative to any degree computing α_f and the reverse direction we work on the cone above α_f and the indicator function which tells us whether the ith block embeds into finitely many other blocks.

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.

Then, we show that the degree spectrum of f contains some non-c.e. degree. To do this, we show that for any tuple $\overline{c}$, there is some tuple $\overline{a}$ which is d-free over $\overline{c}$ and that we can find $\overline{a}$ effectively. Given $\overline{c}$, let $\overline{a}$ be some f-block of size greater than one such that all its elements are greater than those of $\overline{c}$. Since we assumed that f was not intrinscially computable, it cannot be the identiy almost everywhere and so there must be infinitely many blocks of size greater than one. We claim that $\overline{a}$ is d-free over $\overline{c}$.

Now, with $\overline{c}$, $\overline{a}$ as above, suppose there is some quantifier-free formula $\varphi(\overline{x}, \overline{u}, \overline{v})$ and tuple $\overline{b}$ such that $\varphi(\overline{c}, \overline{a}, \overline{b})$ 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 $\overline{b} = \overline{b}_1\overline{b}_2 \cdots \overline{b}_n$.

Now define $\overline{a}' = \overline{a} + 1$ and $\overline{b}' = \overline{b} + 1$. Then $\overline{c}\,\overline{a}'\overline{b}'$ has the same order type as $\bar{c}, \bar{a}, \bar{b}$ and so still satisfies $\varphi(\overline{x}, \overline{u}, \overline{v})$. However the values of f on $\overline{a}$ are not the same as f on $\overline{a}$ as the image greatest element of $\overline{a}$ is in $\overline{a}$ but this is not true of $\overline{a}'$. Furthermore, suppose $\psi(\overline{x}, \overline{u}, \overline{v})$ is some existential formula satisfied by $\overline{c}$, $\overline{a}'$, $\overline{b}'$. We can again replace this by quantifier free formula $\chi(\overline{x}, \overline{u}, \overline{v}, \overline{w})$ which is satisfied by $\overline{c}$, $\overline{a}'$, $\overline{b}', \overline{e}$ for some tuple $\overline{e}$. 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 $\overline{a}''$ be the image of $\bar{a}$ in some block into which it embeds, and similarly let $\bar{b}''_1, \ldots, \bar{b}''_n$ be the images of $\bar{b}_1, \ldots, \bar{b}_n$ 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.

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

In the other direction, suppose that there are infinitely many blocks which embed into no larger blocks. We show that in all computable copies $\mathcal{A}$, $f^{\mathcal{A}}$ satisfies the conditions of the following lemma and so is always of c.e. degree.

Lemma 3.4. *Suppose that some relation $R^{\mathcal{A}}$ in a computable copy $(\mathcal{A}, <_{\mathcal{A}})$ can compute the nth element in the ordering $<_{\mathcal{A}}$ for arbitrarily large n. Then $R^{\mathcal{A}}$ is of c.e. degree.*

Proof. It is enough to show that $R^{\mathcal{A}}$ computes the successor. Let k be arbitrary. To compute the $<_{\mathcal{A}}$ successor of k, use $R^{\mathcal{A}}$ to find the $<_{\mathcal{A}}$ nth element for $n > k$ then list out elements of ω till $n-1$ elements less than the nth elements are found. These elements will form an initial segment of $<_{\mathcal{A}}$ of length n and so to find the successor of k it is enough to check these finitely many elements. $\square$

Given arbitrary m, there must be a block greater than m which does not embed into any greater blocks and suppose there are k elements $<_{\mathcal{A}}$ less than it. To find the elements of this block in the ordering $<_{\mathcal{A}}$ begin listing elements and their values under $f^{\mathcal{A}}$ until some collection is found that is isomorphic to this block and such that there are k elements $<_{\mathcal{A}}$ less than it. After this search terminates, we have found the position of some element greater than m as desired. $\square$

Similarly, we now provide a sufficient and necessary condition for when the degree spectrum of a function is large as possible, ie. contains all Δ^0_2 degrees. First, however, we introduce the following definition to help formalize the way Δ^0_2 sets can be coded in computable copies of $(\omega, <)$.

Definition 3.5. Given a block function f, we say two sequences of intervals $[a_1, b_1], [a_2, b_2], \ldots$ and $[c_1, d_1], [c_2, d_2], \ldots$ in $(\omega, <)$ and a collection of maps $\varphi_i, \psi_i, f_i, g_i$ form an f-coding sequence $[a_1, b_1], [c_1, d_1], [a_2, b_2], [c_2, d_2], \ldots$ if they satisfy the following conditions

- Each interval completely contains all f-blocks it intersects
- $\varphi_i : [a_i, b_i] \to [a_{i+1}, b_{i+1}]$ and $\psi_i : [c_i, d_i] \to [c_{i+1}, d_{i+1}]$ are non-decreasing embeddings which preserve both f and the ordering. Further, we require the sequence $a_1, c_1, a_2, c_2, \ldots$ to be non-decreasing.
- $f_i : [a_i, b_i] \to [c_i, d_i]$ and $g_i : [c_i, d_i] \to [a_{i+1}, b_{i+1}]$ are non-decreasing embeddings which preserve the order but do not preserve f and such that they commute with the embeddings, ie. $g_i \circ f_i = \varphi_i$ and $f_{i+1} \circ g_i = \psi_i$.

This definition can be motivated as follows. In order to compute some Δ^0_2 from a linear order we need to introduce certain coding elements for each element of ω. Further, we build this linear order via finite stages to that the values of f on these elements should reflects the value of the element they code at that stage. So, they must take on different f values at stages where the value of the element they code changes. We also want to ensure that the Δ^0_2 set computes this linear order so we require that the value of f be the same on the coding elements whenever at a stages in which the element they code has the same value. As the value of an element oscillates between 0 and 1 in the approximation of the Δ^0_2, this produces the back and forth alternation between two values of f on the corresponding elements in the linear order which can be seen in the definition of Δ^0_2.

It will be shown below that only one infinite coding sequence is necessary to code a Δ^0_2 set as all element of ω can be moved along this same sequence. Further, we will show that the absence of such a sequence is enough to miss a Δ^0_2 set. This is because, if all coding sequences are finite, then by producing a Δ^0_2 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 following proof.

Theorem 3.6. *If f is 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^0_2 degrees if and only if there is an infinite f-coding sequence*

Proof. First, suppose there is an infinite f-coding sequence $[a_1, b_1], [c_1, d_1], [a_2, b_2], [c_2, d_2], \ldots$. We show that, on the cone above α_f and the sequence, the degree spectrum of f is all Δ^0_2 degrees. To do this, suppose X is a Δ^0_2 set. We will build a computable copy $\mathcal{A}$ of $(\omega, <)$ which is turing equivalent to X via finite stages. During this construction, the elements we place into the linear order will fall into one of two categories: coding elements and padding elements. For each $n \in \omega$, there will be some collection of coding elements corresponding to n. These coding elements will be an interval, and consist of some collection of blocks; thus we call the collection of these coding elements a coding segment for n. The goal of the construction is to move the coding elements so that they mirror the values they correspond to in X while ensuring that once padding elements have been added in the value f on those elements does not change. Given $n < m$, the coding segment for m will be greater than the coding segment for n, so that we can move the segment for m without moving the segment for n.

Construction: $\mathcal{A}$ will constructed such that at stage s, $\mathcal{A}_s$ satisfies the requirements R_e for $e \leq s$ where R_e states that the coding elements corresponding to e form an interval of the form $[a_i, b_i]$ iff $X_s(e) = 0$ and the coding elements corresponding to e form an interval of the form $[c_i, d_i]$ iff $X_s(e) = 1$.
Stage s: At this stage in the construction our partial linear order can be partitioned into a finite number of coding segements and padding blocks, each of which form an interval. In increasing order, we ensure that each of these intervals in $\mathcal{A}_{s-1}$ still satisfy our requirements

- If some collection of elements forms a padding f-block in $\mathcal{A}_{s-1}$, check to ensure they still do in $\mathcal{A}_s$. If not, then 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 in stage $s-1$, the value of $f^{\mathcal{A}_s}$ on these elements is the same in stage $s-1$.

- If some collection of elements forms the coding block corresponding to e first check the value of $X_s(e)$ then identify the next interval, either of the form $[a_i, b_i]$ or $[c_i, d_i]$, satisfying the requirement R_e. By repeatedly applying the f's and g's to the elements of the coding block we can find the image of the coding elements in this new interval and, as above, by inserting new elements below the least coding element and possibly between them we can move the elements to their images. Any new elements that were inserted and end up in the new interval are added to the collection of coding elements 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. This ensures we have satisfied requirement R_e at stage s for $e < s$.

- 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 $X_s(s)$ and identify the next interval, either of the form $[a_i, b_i]$ or $[c_i, d_i]$, satisfying the requirement R_s and is greater in length the 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 or c_i (depending on which interval was chosen), then add in new coding element corresponding to s in order to extend to the length of the interval. 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 n, only finitely many elements are inserted below n. This is because an element is only inserted below n if the value of some $k \in \omega$ corresponding to a coding segment below k changes. If n was added at stage s in the construction, then there can be at most $s-1$ coding blocks below it and so after these $s-1$ elements of the Δ^0_2 set X stop changing values, elements will stop being inserted below n. Hence, since these are finitely many elements of a Δ^0_2 set, this will occur in a finite number of stages.
Next, we show that $f^{\mathcal{A}} \geq X$. Given some element $n \in X$ run the above construction, which is computable, until stage n when the first coding elements corresponding to n are added. Now, compute the value of f on these elements. If it is the same as the value of f at the stage they were added then $X(n) = X_n(n)$, otherwise $X(n) = 1 - X_n(n)$.

Finally, we show that $X \geq f^{\mathcal{A}}$. Given some element $n \in \omega$ run the above construction till n is added to the linear order. If n is added as a padding element then the construction ensures that $f^{\mathcal{A}}(n)$ does not change so take the value at this stage. If n is a coding element corresponding to k then $f^{\mathcal{A}}(n)$ takes on one of two values depending on $X(k)$ since the conditions on the coding sequence ensures $f^{\mathcal{A}}(n)$ is the same whenever it is in a block of the form $[a_i, b_i]$ and similarly for $[c_i, d_i]$. Hence, after determining the coding block n first appears in and applying f_i or g_i we can determine both possible values for $f^{\mathcal{A}}(n)$ computably. To determine the actual value we compute $X(n)$ then choose the corresponding value of $f^{\mathcal{A}}$.

For the converse direction we work on the cone above α_f and show that if no infinite f-coding sequences exist then we can produce Δ^0_2 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

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

where $\mathcal{L}_e$ is a computable listing of the (possibly partial) linear orders. The construction is a finite injury priority construction.

The strategy for $R_{e,i,j}$ is as follows. First, if φ_e 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 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

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

where $m_0 \geq u_0$ such that all f-blocks that intersect $[0, \ldots, u_0]$ are completely contained in $[0, \ldots, m_0]$.

When this requirement acts, restrain $[0, \ldots, v_0]$ 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

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

where $m_{n+1} \geq \max\{m_n, u_{n+1}\}$ such that all f-blocks that intersect $[0,\ldots,\max\{m_n,u_{n+1}\}]$ are completely contained in $[0,\ldots,m_{n+1}]$.

When this requirement acts, restrain $[0,\ldots,v_{n+1}]$ 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.

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 after a strategy is no longer injured, it enters Phase n for arbitrarily large n, then we can produce an infinite 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 3.7. *Let $n' > n$ be of the same parity. Then:*

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

Further, for all n,

1. $f^{\mathcal{L}_{e,s_n}}[0,\ldots,m_n] \neq f^{\mathcal{L}_{e,s_{n+1}}}[0,\ldots,m_n]$

Proof. 1. After phase n, we have restrained the elements $[0,\ldots,v_n]$ and so, since the requirement is no longer injured by higher priority arguments, the only elements in $[0,\ldots,v_n]$ 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 the value of x is the same as well. Hence, $C_{s_{n'}}[0,\ldots,v_n] = C_{s_n}[0,\ldots,v_n]$, 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 stages n and n', the computations $\Phi_j^{C_{s_n}}[0,\dots,m_n] = f^{\mathcal{L}_{e,s_n}}[0,\dots,m_n]$ and $\Phi_j^{C_{s_{n'}}}[0,\dots,m_n]$ 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. Hence, since the use is contained in $[0,\dots,m_n]$, we must have $f^{\mathcal{L}_{e,s_n}}[0,\dots,m_n] \neq f^{\mathcal{L}_{e,s_{n+1}}}[0,\dots,m_n]$, as desired.

$\square$

In particular, this tells us for every once some elements appear below some m_n the value of f on these elements is the same at all stages with the same parity.

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

- Let $[a_0, b_0] = [0,\dots,m_0]$
- Given $[a_i, b_i]$, let $[c_i, d_i]$ be the minimal interval in $[0,\dots,m_{2i+1}]$ that contains all elements that were in $[a_i, b_i]$ at stage s_{2i} and which completely contains all blocks it intersects.
- Similarly, given $[c_i, d_i]$, let $[a_{i+1}, b_{i+1}]$ be the minimal interval in $[0,\dots,m_{2i+2}]$ that contains all elements that were in $[a_i, b_i]$ at stage s_{2i+1} and which completely contains all blocks it intersects.
- Define the maps $f_i : [a_i, b_i] \to [c_i, d_i]$ by $f_i(n)$ is the position of the n'th element at stage s_{2i} at stage s_{2i+1} and similarly for g_i.
- Finally, define the φ_i, ψ_i to be the compositions $g_i \circ f_i$ and $f_{i+1} \circ g_i$, respectively.

This sequence satisfies most of the requirements by construction we just need to ensure that φ_i, ψ_i are f-preserving and the f_i, g_i are not. However, this is exactly the content of the previous lemma. Hence, assuming we enter Phase n for arbitrarily large n we have produced 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. $\square$

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 condition must satisfy to have an intermediate degree spectrum. We give a more specific example of this in Example 4.2 below.

Next, we may ask for a more specific description of these intermediate degree spectra are but the following theorem shows that these degree spectra are not any easily definable class of degrees.

Theorem 3.8. *If f is a block function which is not intrinsically computable and such that, after some initial segment, all blocks embed into some later block then for any computable listing of indices for Δ^0_2 sets the degree spectrum of f contains some degree which is not in the listing.*

Proof. Suppose φ is a listing as above. Then, we construct a computable copy $\mathcal{A}$ of $(\omega, <)$ via finite stages, satisfying the requirements

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

where X_n is the Δ^0_2 set coded by Φ_n. 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 $l_0, \ldots, l_{b-a}$. 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

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

   and

   $$\Phi_{j,s}^{\mathcal{A}_s}[0, \ldots, u_0] = X_{\varphi(e),s}[0, \ldots, u_0] \quad \text{with use } v_0$$

   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 $l_0, \ldots, l_{b-a}$. For each block that was restrained (except for $l_0, \ldots, l_{b-a}$) 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

   $$\Phi_{i,s}^{X_{\varphi(e),s}}[l_0, \ldots, l_{b-a}] = f^{\mathcal{A}_s}[l_0, \ldots, l_{b-a}] \quad \text{with use } u_{2n}$$

   and

   $$\Phi_{j,s}^{\mathcal{A}_s}[0, \ldots, u_{2n}] = X_{\varphi(e),s}[0, \ldots, u_{2n}] \quad \text{with use } v_{2n}$$

   When this requirement acts, insert one new element below the block consisting of the elements $l_0, \ldots, l_{b-a}$. For each block that was restrained in Phase 0 (except for $l_0, \ldots, l_{b-a}$) 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 $2n+1$.

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

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

and

$$\Phi_{j,s}^{\mathcal{A}_s}[0, \dots, u_{2n+1}] = X_{\varphi(e),s}[0, \dots, u_{2n+1}] \quad \text{with use } v_{2n+1}$$

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 $f^{\mathcal{A}_{s+1}}$ on these elements is the same in stage s. 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, ie. $\mathcal{A}$ 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 $f^{\mathcal{A}_{s_{2n}}}$ is the same on the l_i assigned to this strategy for all n.
- The value of $f^{\mathcal{A}_{s_n}}$ on the blocks in the use v_0 which were restrained in Phase 0 is the same for all n.

We now show that $X_{\varphi(e)}$ 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 3.9.

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

Proof.

1. Observe that for the computation $\Phi_{j,s_0}^{\mathcal{A}_{s_0}}[0,\ldots,u_0] = X_{\varphi(e),s_0}[0,\ldots,u_0]$ its use, except for possibly the l_i, is preserved at all stages and so since the value of $f^{\mathcal{A}_{s_n}}$ 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, ie. $\Phi_{j,s_0}^{\mathcal{A}_{s_0}}[0,\ldots,u_0] = \Phi_{j,s_0}^{\mathcal{A}_{s_n}}[0,\ldots,u_n]$ and so it follows that $X_{\varphi(e),s_n}[0,\ldots,u_0] = X_{\varphi(e),s_0}[0,\ldots,u_0]$, as desired.

2. The computation $\Phi_{i,s_0}^{X_{\varphi(e),s}}[l_0,\ldots,l_{b-a}] = f^{\mathcal{A}_{s_0}}[l_0,\ldots,l_{b-a}]$ found in Phase 0 has use u_0 and so by part (1) holds at stage s_n for all even n. However, since $f^{\mathcal{A}_{s_n}}[l_0,\ldots,l_{b-a}] \neq f^{\mathcal{A}_{s_{n+1}}}[l_0,\ldots,l_{b-a}]$ and $\Phi_{i,s_n}^{X_{\varphi(e),s}}[l_0,\ldots,l_{b-a}] = f^{\mathcal{A}_{s_n}}[l_0,\ldots,l_{b-a}]$ for all n we must have $\Phi_{i,s_n}^{X_{\varphi(e),s_n}}[l_0,\ldots,l_{b-a}] \neq \Phi_{i,s_{n+1}}^{X_{\varphi(e),s_{n+1}}}[l_0,\ldots,l_{b-a}]$. 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 $X_{\varphi(e),s_n}[0,\ldots,u_0] \neq X_{\varphi(e),s_{n+1}}[0,\ldots,u_0]$, as desired. □

This lemma tells us that some element in $[0,\ldots,u_0]$ 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. □

Despite the apparent difficulty in describing these intermediate degree spectra one way we can attempt to characterize them is by the 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.

Definition 3.10. Given a block function f we can define its maximal coding tree as follows. The vertices of the tree will be all finite coding sequences, including the empty sequence. There is an edge between two coding sequences if one coding sequence is one interval than the other, and they are the same sequence on the intervals they have in common. We can also define the minimal coding tree to be the subtree of the maximal coding tree which consist of coding sequences such that for all intervals that occur in the sequence there are infinitely many intervals which are f-isomorphic, except for possibly the last interval in the sequence.

Note that we have defined both a maximal coding tree and a minimal coding tree instead of just one coding tree from which α-c.e sets are possible. This is because there are actually two ways in which we need 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, ie. 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 while the minimal coding tree captures which types of moves are possible while ensuring that coding sequences can move independently of each other. 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 α are possible.

Theorem 3.11. *If α is the rank of the maximal coding tree corresponding to some block function f, then there is some α-c.e. degree which is not in the degree spectrum of f.*

Proof. To show this, we consider the set constructed in the proof of Theorem 3.8 and show that it is $\alpha + 1$-c.e. on the cone above α_f and the maximal tree. Call this set C. Suppose $x \in \omega$ restrained for some strategy $R_{e,i,j}$ then, as shown in the proof, if this strategy reaches Phase k without being injured we can produce a coding sequence of length k. let s_i be the stage the strategy enters Phase i. To show this set is α-c.e we take $g(x, s)$ to be the value of n at stage s in the construction and let $n(x, s)$ be the rank of the vertex corresponding to the sequence produced at Phase k where s_k is the greatest s_i less than s. If s is less than all s_i let $n(x, s) = \alpha$. □

Theorem 3.12. *If α is the rank of the minimal coding tree corresponding to some block function f, then, on a cone, the degree spectrum of f contains all β-c.e. degrees for any $\beta < \alpha$.*

Proof. We work on the cone above α_f and the minimal coding tree. Suppose X is an β-c.e. set, with $\beta < \alpha$, given by functions g, n where g is the value of X and n is the number of changes. We constuct a computable copy, $\mathcal{A}$, of $(\omega, <)$ which is turing equivalent to X. This construction will mirror that of Theorem 3.6 in the sense that we will have two types of elements padding and coding elements and we will move the coding elements to reflect changes in X while maintaining the value of the padding elements. However, in this case we will also be choosing the path the coding elements take through the minimal tree in order to account for the number of changes X can make to the value of some element.

Construction: $\mathcal{A}$ will be constructed such that for $x \in \omega$ there is some collection of coding elements corresponding to x in $\mathcal{A}$. Further, the position of these elements as s increases will correspond to a path through the minimal tree. This construction will be carried out so that at stage s, $\mathcal{A}_s$ satisfies the requirements R_e for $e \leq s$ where R_e states that the path corresponding to the coding elements of e is of even length iff $X_s(e) = 0$ and the path is of odd length iff $X_s(e) = 1$. R_e also requires that, in the minimal coding tree, the sequence corresponding to these coding elements has rank at least $n(e, s)$. To do this, suppose we have the finite ordering $\mathcal{A}_{s-1}$. In increasing order, we ensure that each of the intervals in $\mathcal{A}_{s-1}$ still satisfy our desired conditions

- Suppose that the value e has changed at stages $s_0 < s_1 < s_2 < \ldots < s_k$ then the sequence corresponding to the coding elements of e is the intervals that the coding elements occupy at stages $s_0 - 1, s_1 - 1, \ldots$. The construction will ensure that this really does form a coding sequence.

- If some collection of elements forms a padding f-block in $\mathcal{A}_{s-1}$ check to ensure they still do in $\mathcal{A}_{s-1}$. If not, then insert new padding elements below the least element of this block and possibly between the elements 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 the block. This ensures that for all padding elements present in stage $s-1$, the value of $f^{\mathcal{A}_s}$ on these elements is the same as in stage $s-1$.

- If some collection elements forms the coding interval corresponding to e first check the value of $X_s(e)$. If $X_s(e) = X_{s-1}(e)$ then we want to ensure $f^{\mathcal{A}_{s-1}} = f^{\mathcal{A}_s}$ on this coding interval. If necessary, insert new padding element below the least element of the block to move it up to another interval which is f-isomorphic to the coding interval. If $X_s(e) = X_{s-1}(e)$ then $n(e, s-1) > n(e, s) = \gamma$. At this stage in the construction, the coding block corresponding to e has produced a coding sequence with length equal to the number of changes. Further, its corresponding vertex in the minimal coding tree has rank $n(e, s-1)$ and so there must be some other vertex, corresponding to a coding sequence one interval longer, with rank $\geq \gamma$ which is connected to this vertex. Choose one such sequence and insert elements below and possibly between the elements of the coding interval to move the elements to the image of the current interval under either f or g. Further, ensure that enough new coding element are inserted after the final element so that we complete the entire interval. This ensures we have satisfied requirement R_e at stage s for $e < s$.

- Finally, we introduce the coding elements corresponding to s. After we have ensured all previously added elements satisfy the requirements, choose some coding sequence of length one whose corresponding vertex has rank $\geq \beta$. Further, we choose this sequence so that the least element of its interval is greater than the length $\mathcal{A}_s$ at this point in the construction. Insert enough new padding elements to the end of the linear order to extend it to beginning of this interval, then add new coding elements corresponding to s in order to extend to the length of the full interval. 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 n, only finitely many elements are inserted below n. This is because an element is only inserted below n if the value of some $k \in \omega$ corresponding to a coding segment below k changes. If n was added at stage s in the construction, then there can be at most $s-1$ coding blocks below it and so after these $s-1$ elements of the Δ^0_2 set X stop changing values, elements will stop being inserted below n. Hence, since these are finitely many elements of a Δ^0_2 set, this will occur in a finite number of stages.
Next, we show that $f^{\mathcal{A}} \geq X$. Given some element $n \in X$ run the above construction, which is computable, until stage n when the first coding elements corresponding to n are added. Now, compute the value of f on these elements. If it is the same as the value of f at the stage they were added then $X(n) = X_n(n)$, otherwise $X(n) = 1 - X_n(n)$.

Finally, we show that $X \geq f^{\mathcal{A}}$. Given some element $n \in \omega$ run the above construction till n is added to the linear order. If n is added as a padding element then the construction ensures that $f^{\mathcal{A}}(n)$ does not change so take the value at this stage. If n is a coding element corresponding to k then $f^{\mathcal{A}}(n)$ takes on one of two values depending on $X(k)$ since the conditions on the coding sequence ensures $f^{\mathcal{A}_s}(n)$ is the same whenever $X_s(k)$ is the same. Hence, after determining the coding block n first appears we can determine the value of $f^{\mathcal{A}}(n)$ if $X(k)$ equals the value at this stage. Otherwise, if $X(k)$ is the opposite value, run the construction till $X_s(k)$ changes values. Then, the value of $f^{\mathcal{A}_s}(n)$ at this stage will be the final value of $f^{\mathcal{A}}(n)$. $\square$

4 Examples

We conclude this paper with a few applications of *Theorem* 3.6; the first of which is an instance in which the degree spectrum of a function is all Δ^0_2 degrees. Note that in this example we work with non-pairwise embeddable blocks so all f-embedings in the coding sequence are uniquely determined up to shifting of blocks.

Proposition 4.1. *Given a block function f with non-pairwise embeddable blocks such that after some point all blocks occur occur infinitely often. Then, if there exists distinct strings σ, τ that occur infinitely often in α_f such that σ and τ have the same underlying lengths (ie. the sum of the lengths their blocks is the same) then the degree spectrum is all Δ^0_2 degrees.*

Proof. Suppose σ and τ are as above. Our coding sequence will involve shifting a set amount of elements between occurrences σ and τ as they move up ω. Inductively define the intervals as follows:

- Given $[a_0, b_0], [c_0, d_0], \ldots, [a_{i-1}, b_{i-1}], [c_{i-1}, d_{i-1}]$, let $[a_i, b_i]$ be some interval such that, when viewed as a substring of α_f, the blocks that make up this interval are an instance of σ and ensure that $d_{i-1} < a_i$. Define $g_i : [c_{i-1}, d_{i-1}] \to [a_i, b_i]$ by $x \mapsto x + (a_i - c_i)$ and $\varphi_{i-1} : [a_{i-1}, b_{i-1}]$ by $x \mapsto (a_i - a_{i-1})$.
- Similarly, given $[a_0, b_0], [c_0, d_0], \ldots, [a_i, b_{i-1}]$, let $[c_i, d_i]$ be some interval such that, when viewed as a substring of α_f, the blocks that make up this interval are an instance of τ and ensure that $b_i < c_i$. Define $f_i : [a_i, b_i] \to [c_i, d_i]$ by $x \mapsto x+(a_i-c_i)$ and $\varphi_{i-1} : [a_{i-1}, b_{i-1}]$ by $x \mapsto (a_i - a_{i-1})$.

Since σ and τ have the same underlying length, all intervals have the same length so it follows that the maps satisfy the desired conditions. Hence, we have produced a coding sequence and so the degree spectrum of f must be all Δ^0_2 degrees on a cone. □

The second application of this theorem again works with non-pairwise embeddable blocks but considers a case in which the first example fails. Since there exist no σ and τ as above, we can use this fact to show that the intervals in any coding sequence must increase in size. This combined with restrictions placed on when two blocks may be adjacent in the function will be used to show that all coding sequences for the below function must be finite.

Example 4.2. Let I_k be the block corresponding to the loop of length k, ie. the block isomorphic to $[1, \ldots, k] \to [1, \ldots, k]$ 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 $I_1, I_2, I_3, I_4, I_5, \ldots$ and the even blocks are given by the sequence $I_1, I_1, I_2, I_1, I_2, I_3, \ldots$, eg. an initial segment looks like:

$$I_1 + I_1 + I_2 + I_1 + I_3 + I_2 + I_4 + I_1 + I_5 + I_2 + I_6 + I_3 + I_7 + \cdots$$

This function is constructed so that it satisfies the following:

- all blocks that occur in the function occur infinitely often
- no 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

Since each block occurs infinitely often the degree spectrum of f must contain a non-c.e. degree. To show that the degree spectrum of f does not contain all Δ_2^0 degrees we show that f contains no infinite coding sequence.
Given any finite coding sequence $[a_1, b_1], [c_1, d_1], \ldots$ we make the following definitions. Note that we will identify the elements in an interval with their images under the maps f_i, g_i in all greater intervals.

- Say that $l_1 < \ldots < l_p$ form a *link* if they form a block of length p in the k'th interval. Say the kth interval *witnesses* this link.
- Say that a link $l_1 < \ldots < l_p$ is *vulnerable* if in some interval, after the first interval witnessing this link, the $l_1 < \ldots < l_p$ are contained in two or more blocks.
- Say that a link $l_1 < \ldots < l_p$ is broken in some interval, after the first interval witnessing this link, if some element is inserted between l_1 and l_p, i.e., the l_i are no longer adjacent.

First, observe that since the blocks in f are non-pairwise embedable if some link is witnessed at some stage k then every $m > k$ of the same parity must also witness this link. This also implies that after a link is broken the coding sequence must terminate. To see why this is the case, note that a link cannot be broken at a stage witnessing the link so by the above observation the next interval added to this sequence must witness the link. However, since there have been elements inserted between the l_i and the embeddings f_i, g_i must be order preserving, their images cannot form a block in any intervals and so the coding sequence must terminate. Next, observe that after a link becomes vulnerable it must break in one of the next two intervals in the coding sequence or the coding sequence must terminate. Given any link $l_1 < \ldots < l_p$ in order for this link to become vulnerable there must be some interval in the l_i's lie in two blocks or more blocks. Say this occurs in the kth interval in the sequence. In the $k+2$th interval the value of f on the l_i must be the same and so the each l_i must be in the same block-type as in the kth interval. However, by construction of this function these blocks can no longer be adjacent so some elements must have been inserted between them, ie. the link must have been broken. Finally, it remains to show that in any coding sequence some link must become vulnerable. We claim this must occur in the second or third interval. Since the map from the first to the second interval is not f-preserving there must be some link $l_1 < \ldots < l_p$ witnessed in the first interval which no longer forms a block of size p in the second interval. Either these $l_1 < \ldots < l_p$ lie in two blocks and are already vulnerable or they are all contained in some larger block. Suppose they are contained in some larger block $k_1 < \ldots < k_q$ then this link must become vulnerable in the third interval. This is because the $l_1 < \ldots < l_p$ must form a block of size $p < q$ in this interval and so the $l_1 < \ldots < l_q$ cannot all lie in the same block.

References

[BKWa22] Nikolay Bazhenov, Dariusz Kalociński, and MichałWrocł awski. 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.

[HT18] Matthew Harrison-Trainor. Degree spectra of relations on a cone. *Mem. Amer. Math. Soc.*, 253(1208):v+107, 2018.

[Wri18] Matthew Wright. Degrees of relations on ordinals. *Computability*, 7(4):349–365, 2018.