η-REPRESENTATION OF SETS AND DEGREES

KENNETH HARRIS

Abstract. We show that a set has an η-representation in a linear order if and only if it is the range of a $0'$-computable limitwise monotonic function. We also construct a Δ_3 Turing degree for which no set in that degree has a strong η-representation, answering a question posed by Downey.

1. Introduction

Let $A = \{a_0 < a_1 < a_2 < \ldots\}$ which does not include 0 or 1 and η the order type of the rationals. When A can be represented by a computable linear order of the order-type

$$\eta + \mathbf{a_0} + \eta + \mathbf{a_1} + \eta + \mathbf{a_2} + \eta \ldots$$

where each $\mathbf{a_i}$ consists of a_i elements linearly ordered then such a set has a *strong η-representation*. When A can be represented by such a computable linear order but the blocks $\mathbf{a_i}$ may occur in any order and repetitions of blocks of the same size are allowed, then A has an *η-representation*. And when A can be represented by such a computable linear ordering where the blocks $\mathbf{a_i}$ may occur in any order, but only one block of size a_i may occur, then A has a *unique η-representation*. A Turing *degree* has a (strong, unique) η-representation if some set in that degree has one. This paper answers several questions on these representations which have been open since the early days in the investigation of computable linear orders.

Any set with a (unique) η-representation is Σ_3, and any set with a strong η-representation is Δ_3. On the other hand, all Σ_2 and Π_2 sets have even a strong η-representation (see [Dow98].) But Lerman ([Ler81]) produced a Δ_3 set which does not even have an η-representation. So there is no simple characterization of the sets with η-representations in the arithmetic hierarchy.

Question 1 (Downey)**.** *Is there a classification of the sets with η-representations which highlights interesting properties of these sets? What about the sets with unique η-representations? Or the sets with strong η-representations?*

The dynamic nature of constructing η-representations led us to draw their connection with limitwise monotonic functions which have played an important role in several investigations in computable algebra and model theory ([Khi98], [KNS97], [CDK98] [CHKS04]). We show in Theorem 5.4 that the ranges of $0'$-computable limitwise monotonic functions are precisely the sets with η-representations. We show a similar characterization of the sets with unique η representations as the ranges of 1-1 $0'$-computable limitwise monotonic functions (Theorem 6.2.) It might be hoped that there is a characterization of the sets with strong η-representations as the ranges of strictly increasing $0'$-computable limitwise monotonic functions, but we show in Theorem 7.3 that there there is a set with a strong η-representation but which is not the

I would like to thank Rod Downey for the inspiration behind this paper. I have benefitted immeasurably by his enthusiasm for this subject and his wealth of knowledge. I would like to thank Robert Soare for the perspiration required to see this paper through. I am grateful to the referee for his invaluable suggestions.

range of such a function. The problem of classifying strong η-representations is still open.

Representing degrees is considerably easier since this only requires representing some set in that degree. Lerman provided a characterization of the degrees with η-representations: they are the Σ_3-degrees. Lerman additionally asked

Question 2 ([Ler81])**.** *Do the Σ_3 degrees have unique η-representations?*

We prove that every *set* which has an η-representation also has a unique η-representation (Theorem 6.4.) As a consequence, the Σ_3 degrees do have unique η-representations.

Downey asked whether there was a similar characterization of the degrees with a strong η-representation in the arithmetic hierarchy:

Question 3 ([Dow98])**.** *Do the Δ_3 degrees have strong η-representations?*

We show in Theorem 7.1 that there is a Δ_3 degree which has no strong η-represention.

2. Basic Definitions

There are two equivalent formulations of computably presentable linear-ordered sets:

Definition 2.1 (Computable Linear Orderings)**.** *A linear ordering of the natural numbers $\mathbf{L} = (\mathbb{N}, \leq_{\mathbf{L}})$ is **computable** if $\leq_{\mathbf{L}}$ is a computable relation. A linear order $\mathbf{L}$ is **computably presentable** if there is a computable linear ordering of the natural numbers which is order-isomorphic to $\mathbf{L}$. An order-type τ is **computably presentable** if there is a computable linear order whose order-type is τ.*

In this definition of computable (computably presentable) linear order, the domain of the order is taken to be the natural numbers. Let $\mathbb{Q}$ be a computable linear ordering whose order-type is the rational numbers. By effectivising the usual proof that countable linear orders can be isomorphically embedded in the rationals (see [Dow98, Theorem 2.1]) we obtain a second formulation of a computable linear order:

Theorem 2.2. *A linear order is computably presented if and only if it is computably isomorphic to a computable subset of $\mathbb{Q}$ under the usual ordering of the rationals.*

The advantage of either account is that to build a computable linear order we can either construct a computable ordering on the natural numbers, or a computable subset of the rationals $\mathbb{Q}$, whichever is convenient.

Let η be the order-type of the rationals and ζ the order-type of the integers. Note that these order types are computably presentable. For integer n, $\mathbf{n}$ is the order-type of the n-element linear order. Given two order-types α and β, $\alpha + \beta$ is the order type of a linear order $\mathbf{P} = (P, <_{\mathbf{P}})$ with two suborderings $\mathbf{L} = (L, <_{\mathbf{P}} \restriction L \times L)$ and $\mathbf{M} = (M, <_{\mathbf{P}} \restriction M \times M)$ where

(a) The order-type of $\mathbf{L}$ is α, and the order-type of $\mathbf{M}$ is β.
(b) $P = L \cup M$.
(c) For every $a \in L$ and $b \in M$, $a <_{\mathbf{P}} b$.

Our interest in the following definition is when the order type τ is η or ζ:

Definition 2.3. *Let τ be an order-type and $A = \{a_0 < a_1 < \dots\}$ an infinite subset of the natural numbers which does not include 0 or 1. Then*

(i) *A has a* τ***-representation*** *if there is a surjection* $\pi : \mathcal{N} \to \mathcal{N}$ *such that*

$$\sum_i \tau + \mathbf{a}_{\pi(i)}$$

is computably presentable.

(ii) *A has a* ***unique*** τ***-representation*** *if there is a permutation* $\pi : \mathcal{N} \to \mathcal{N}$ *such that*

$$\sum_i \tau + \mathbf{a}_{\pi(i)}$$

is computably presentable.

(iii) *A has a* ***strong*** τ***-representation*** *if*

$$\sum_i \tau + \mathbf{a}_i$$

is computably presentable.

A degree $\mathbf{d}$ *has a (strong, unique)* τ*-representation if there is an* $A \in \mathbf{d}$ *with a (strong, unique)* τ*-representation.*

The sets and degrees with η- and ζ-representations have been the primary focus of investigation. We have stated the definition for *infinite sets*, although it could be extended to finite sets as well; but, any finite set has an η-representation and a ζ-representation, so we will assume in what follows that the sets we consider are infinite. We also assume that $0, 1 \notin A$ since $\eta + \mathbf{0} + \eta = \eta = \eta + \mathbf{1} + \eta$. Let $\mathbf{L} = \left(L, \leq_{\mathbf{L}}\right)$ be a computable linear order. For distinct members x and y of L, say that x and y are *adjacent* if $\neg\exists z(x <_{\mathbf{L}} z <_{\mathbf{L}} y)$, so a $0'$-oracle can decide if x and y are adjacent. A sequence of distinct elements $\langle x_1, \ldots, x_k\rangle$ from L, are a *block* if for each $i < k$, x_i and x_{i+1} are adjacent. For each block σ and number $n \in \omega$, $0'$ can decide if either $\sigma^\frown\langle n\rangle$ is a block or if $\langle n\rangle^\frown\sigma$ is a block in L. A block σ is *maximal* if there is no number $n \in \omega$ such that either $\sigma^\frown\langle n\rangle$ is a block or $\langle n\rangle^\frown\sigma$ is a block. A $0''$-oracle can decide if a block is a maximal block.

Lerman in [Ler81] has given a classification of the sets and degrees with (strong, unique) ζ-representations.

Theorem 2.4. *The sets with (unique, strong)* ζ*-representations are exactly the* Σ_3 *sets. The degrees with (unique, strong)* ζ*-representations are exactly the* Σ_3 *degrees.*

So, in the case of ζ-representations, the three flavors of representations offer no real distinction. This is not the case of η-representations, and the classification problem for the sets and degrees with (strong, unique) η-representations was open at the time we began our investigation.

3. Constructing η-Representations

We now turn to constructing η-representations. The construction presented here is the only method we have seen for constructing η-representations. We formalize the construction, to extract-out the essential features that must be met for the construction to succeed. This will also motivate one of our characterizations of the η-representable sets.

We want to start by building a basic block of the order type $\eta + \mathbf{n}$, where at each stage of the construction we look to a computable approximation for the size of the finite block. Let $f : \omega \to \omega$ be computable, where for $s \in \omega$ we will use the value $f(s)$ as the stage s

approximation of the size of the finite block under construction. Let $\mathbf{L}_s$ be the finite linear order we have built at stage s, and which looks as follows:

$$qqq \ldots qqq\ xxx \ldots xxx$$

where the q represent the part of the order that at stage s is to be the dense η portion, and the x represent $f(s)$ points which are currently expected to be a finite block of adjacencies. At stage $s+1$, if $f(s+1) \leq f(s)$, then we must incorporate some of the xs into the rational part, and *densify* the rational part:

$$pqpqpqp \ldots pqpqpqrprpr\ x \ldots xxx$$

where p represents new points added to ensure we have a copy of η and some of the x which were to be in the finite block, have been corralled into the dense copy (and converted to r) to make the finite block have size $f(s+1)$. If, on the other hand, $f(s+1) > f(s)$, then the corralling must occur in the other direction

$$pqpqpqp \ldots pqpqp\ yyxxx \ldots xxx$$

where y represents points which had been in the dense copy at the previous stage, and now are incorporated into the finite block. We want the limit of this process

$$\eta + \mathbf{n} = \bigcup_s \mathbf{L}_s$$

for some target n. What conditions on f must be met to ensure that this process actually creates the target block? The goal in this section is to show that a necessary and sufficient condition that a block of size $\mathbf{n}$ is constructed is that

$$n = \liminf_s f(s)$$

The following definition makes precise this construction.

Definition 3.1. *Let $f : \omega \to \omega$ and $\mathbf{L}$ a linear order. Then we will say $\mathbf{L}$ is* ***densified according to*** *f, if there is a sequence of finite linear orders $\langle \boldsymbol{L}_s : s \in \omega \rangle$ with $\boldsymbol{L} = \cup_s \boldsymbol{L}_s$ and where $\boldsymbol{L}_s$ is given by:*

- s=0: $\boldsymbol{L}_0 = \{q < x_{f(0)} < \ldots < x_2 < x_1\}$
- s+1: *Let $\boldsymbol{L}_s = \{x_{m_s} < \ldots < x_2 < x_1\}$ where $m_s > f(s)$*
  - (a) *If $f(s+1) \geq |\boldsymbol{L}_s|$ then $\boldsymbol{L}_{s+1} = \mathbf{1} + (\mathbf{f(s+1)} - \mathbf{m_s}) + \boldsymbol{L}_s$.*
  - (b) *Otherwise, let $\boldsymbol{L}_{s+1}$ be given by adding a new element p for each i with $m_s > i \geq f(s+1)$ so that $x_{i+1} < p < x_i$.*

Lemma 3.2. *Let $f : \omega \to \omega$ and $\boldsymbol{L}$ densified according to f. Then for $n \in \omega$, the order-type of $\boldsymbol{L}$ is $\eta + \mathbf{n}$ if and only if $n = \liminf_s f(s)$.*

Proof. Let $\langle \mathbf{L}_s : s \in \omega \rangle$ be a sequence of finite linear orders as in Definition 3.1. Note that $\mathbf{L} = \cup_s \mathbf{L}_s$ is a linear order. Suppose $n = \liminf_s f(s)$. Let s large enough so that for all $t \geq s$, both $n \leq f(t)$ and $n < |\mathbf{L}_t|$. Let $x_n, \ldots, x_1$ be the last n elements in $\mathbf{L}_s$. For each $t > s$ and each $i < n$, there is no p with $x_{i+1} < p < x_i$, since $n \leq f(t)$. Thus, the order-type of $\mathbf{L}$ is $\tau + \mathbf{n}$ for some linear order τ. Let $x, y, z \in \mathbf{L}_t$ where x is in the sub-ordering of type τ and $y < x < z$ in $\mathbf{L}_t$. Let $u > t$ such that $n = f(u)$ and $x = x_i$ in $\mathbf{L}_{u-1}$, so that $m_u > i > f(u) = n$. Then, there are $p, q \in \mathbf{L}_u$ with $y < p < x < q < z$. Thus, $\tau = \eta$, and the order-type of $\mathbf{L}$ is $\eta + \mathbf{n}$.

Conversely, suppose that $\liminf_s f(s)$ does not exist. Then for any n there are only finitely many stages s with $n > f(s)$. Let s be large enough so that $n \leq f(t)$ for all $t \geq s$. From stage s onward, the right-most block $x_n, \ldots, x_1$ of $\mathbf{L}_s$ remains intact. Thus, $\mathbf{L}$ contains a right-most block of adjacencies of size at least n. Since this is true for every n, $\mathbf{L}$ contains a right-most block whose order-type is ω^* (an infinite descending sequence of adjacent elements.) □

Theorem 3.3. *Let $f : \omega \times \omega \to \omega$ be a computable function, $F : \omega \to \omega$ a total function with $F(\cdot) = \liminf_s f(\cdot, s)$. Let $A = ran(F) \setminus \{0, 1\}$. Then*

(a) *A has an η-representation.*
(b) *Suppose that whenever $m \neq n$, $F(m) \neq F(n)$. Then then A has a unique η-representation.*
(c) *Suppose that whenever $m < n$, $F(m) < F(n)$. Then A has a strong η-representation.*

Proof. We will construct a linear order $\mathbf{L} = \cup_n \mathbf{L}_n$ whose domain is a computable subset of $\mathbb{Q}$ and whose order type is $\sum_n \tau_n$ where $\tau_n = \eta + \mathbf{F(n)}$ is the order-type of $\mathbf{L}_n$.

Let $\{q_0, q_1, \ldots, \}$ be a computable enumeration of $\mathbb{Q}$. For each n, the linear-order $\mathbf{L}_n = \cup_s \mathbf{L}_n^s$ where $\langle \mathbf{L}_n^s \rangle_{s \in \omega}$ is a finite sequence of linear orders which are constructed by densifying according to $f(n, \cdot)$, as in Definition 3.1. To ensure $\mathbf{L}$ is computable, any new elements added to $\mathbf{L}_n^{s+1}$ will be enumerated after q_s in the computable enumeration of $\mathbb{Q}$. By Lemma 3.2 the order-type of $\mathbf{L}_n$ will be $\eta + \mathbf{F(n)}$. □

4. What was Known about η-Representable Sets and Degrees

This section summarizes what was known about the sets and degrees with (strong, unique) η-representations. Proofs of all these results are sketched in [Dow98, Section 4].

An upper bound on the complexity of sets with (strong, unique) η-representations is

Theorem 4.1. *All sets with an η-representation or unique η-representation are Σ_3. All sets with a strong η-representation are Δ_3.*

Some lower bounds on the complexity of sets with strong η-representations are

Theorem 4.2. *Every Σ_2 set and every Π_2 set has a strong η-representation.*

Unfortunately, classifying the classes of sets which have (unique, strong) η-representations will not be by means of the arithmetical hierarchy by the following result of [Ler81, Theorem 3.1]

Theorem 4.3. *There is a Δ_3 set which has no η-representation.*

Corollary 5.5 shows that the Δ_3 counterexample set A can be made so that $A \oplus 0'$ is low over $0'$.

Lerman [Ler81] showed though that the η-representable *degrees* could be classified within the arithmetical hierarchy:

Theorem 4.4. *The degrees with η-representations are exactly the Σ_3 degrees.*

Lerman conjectured (see [Ler81, pp. 141-2]) that all Σ_3 degrees actually have a unique η-representation, but the proof of theorem 4.4 is not refined enough for this. We will confirm Lerman's conjecture (Corollary 6.5) by showing that the sets with η-representations are exactly the sets with unique η-representations (Theorem 6.4.)

It might be hoped that the Δ_3 degrees could be characterized by the degrees with strong η-representations, but we will show (Theorem 7.1) that this is not the case.

5. η-Representations [1]

The main result in this section, Theorem 5.4, is the classification of the sets with η-representations as the ranges of $0'$-computable limitwise monotonic functions.

[1]Rod Downey and Denis Hirschfeldt brought to my attention the intriguing possibility of connecting limitwise monotonic functions with η-representations. Asher Kach has independently proved Lemma 5.2 in his investigation of computable shuffle sums.

Definition 5.1. *Let* $\mathbf{d}$ *be a Turing degree.* F *is a* $\mathbf{d}$***-Computable Limitwise Monotonic*** *function (*$\mathbf{d}$*-lmf) if there is a function* $f \leq_T \mathbf{d}$ *satisfying*

(a) $f(n,s) \leq f(n,s+1)$ *for all* $n,s \in \omega$*,*
(b) $\lim_s f(n,s)$ *exists for all* $n \in \omega$*,*
(c) $F(n) = lim_s f(n,s)$*.*

F *is order-preserving if* $F(m) < F(n)$ *whenever* $m < n$*.*

For any $0'$-computable limitwise monotonic function F, there is a ternary computable function g such that

$$F(n) = \lim_s \lim_t g(n,s,t).$$

But, it is possible to do better in this case,

Lemma 5.2. *For any total function* F*, the following are equivalent:*

(i) F *is a* $\mathbf{0}'$*-lmf.*
(ii) *There is a binary computable function* g *such that* $F(n) = \liminf_s g(n,s)$ *for each* $n \in \omega$*.*

Proof. *(ii)* $\Rightarrow$ *(i)*: Given a binary computable function g, the proof will produce a $\mathbf{0}'$-lmf F with $F(n) = \liminf_s g(n,s)$ for each $n \in \omega$. We construct a $0'$-computable function f in stages s, and let $F(\cdot) = \lim_s f(\cdot, s)$.

Constuction.

stage 0: Let $f(n,0) = 0$ for all n.

stage s+1: Given $f(n,s)$ use the $0'$-oracle to decide whether

$$(\exists t > s)\ \big[g(n,t) < g(n,s)\big].$$

If yes, then let $f(n,s) = f(n,s+1)$. If no, then let $f(n,s+1) = g(n,s)$.
This ends the construction.

Verification. f is monotonic: We will show for each n that $f(n,s) \leq g(n,t)$ for every s and every $t \geq s$. Fixing n, the argument is by induction on s. When $s = 0$, $f(n,0) = 0 \leq g(n,t)$ for all $t \geq 0$. Suppose that for a given s, $f(n,s) \leq g(n,t)$ for every $t \geq s$. Either $f(n,s+1) = f(n,s) \leq g(n,t)$ for all $t \geq s+1$, or $f(n,s+1) = g(n,s)$, but in this case $g(n,s) \leq g(n,t)$ for all $t \geq s+1$. Thus, $f(n,s+1) \leq g(n,t)$ for all $t \geq s+1$. Finally, f is monotonic: $f(n,s+1) = f(n,s)$ or $f(n,s+1) = g(n,s)$, but $f(n,s) \leq g(n,s)$.

$\lim_s f(n,s) = \liminf_s g(n,s)$ for each n: Let $G(n) = \liminf_s g(n,s)$. Let s be a stage such that $g(n,s) = G(n)$ and $g(n,t) \geq g(n,s)$ for all $t \geq s$. So, $f(n,s+1) = g(n,s)$. Furthermore, $f(n,t) = f(n,s+1) = G(n)$ for any $t \geq s+1$; and thus, $\lim_s f(n,s) = G(n)$.

(i) $\Rightarrow$ *(ii)*: Given a $\mathbf{0}'$-lmf F the proof will produce a binary computable function g satisfying $F(n) = \liminf_s g(n,s)$ for all $n \in \omega$. Let $F(\cdot) = \lim_s f(\cdot, s)$ where

(a) $f(n,s) \leq f(n,s+1)$ for all $n,s \in \omega$.
(b) $\lim_s f(n,s)$ exists for all $n \in \omega$.
(c) $f \leq_T \mathbf{0}'$.

By the Limit Lemma (see [Soa87], Lemma II.3.3) there is a computable h with

$$F(n) = \lim_t \lim_s h(n,t,s) \qquad \text{for every } n.$$

We will approximate $F(n)$ in stages s using an *s-reasonable* sequence: A sequence σ is s-reasonable if

$$\sigma = \langle h(n,0,s), \ldots, h(n,|\sigma|-1,s)\rangle \qquad \text{where } h(n,i,s) \leq h(n,i+1,s) \text{ for all } i < |\sigma|.$$

The strategy is to let $g(n,s) = \sigma(|\sigma|-1) = h(n,|\sigma|-1,s)$ for some s-reasonable sequence σ. The trick is to decide on how long σ will be. The problem is that if we always choose s-reasonable σ of maximal length we may have $h(n,|\sigma|-1,s) > F(n)$ for all s. So, we need a more conservative guess of how long our approximation sequence to $F(n)$ will be. To formalize the strategy for choosing s-reasonable sequences, prioritize sequences by $\sigma < \tau$ (σ has higher priority than τ) if $|\sigma| < |\tau|$. At each stage $s+1$, the construction chooses the highest priority s-reasonable sequence for each n. Note that there is exactly one s-reasonable sequence of each length for each n.

Construction. The construction produces a binary computable g in stages. We will label each sequence *active* or *retired* during the construction in order to determine which sequences will influence the construction. (This labeling will be carried-out independently for each argument n.)

stage 0: $g(n,0) = 0$ for all n. All sequences are active. Retire ε, the empty sequence.

stage s+1: Suppose $g(n,s)$ has been defined for each n. Let σ be the highest priority active s-reasonable sequence, and let $g(n,s+1) = \sigma(|\sigma|-1)$. Retire σ.
This ends the construction.

Verification. A sequence σ is *correct* (for n) if

$$\sigma = \langle f(n,0), \ldots, f(n,|\sigma|-1)\rangle$$

Sublemma 1. *For every n, each correct sequence for n is eventually retired.*

Proof. Let σ be a correct sequence for n, and suppose that all correct sequences τ with $|\tau| < |\sigma|$ are retired before stage t. Since $f(n,i) = \lim_s h(n,i,s)$ for all $i \in \omega$, there is a stage $s \geq t$ such that for all $i < |\sigma|$, $h(n,i,s) = f(n,i)$. If σ has not yet been retired by stage s, then since σ is the highest priority s-reasonable sequence, σ will determine the construction at stage s and be retired. □

It follows from Sublemma 1, that for each $k \in \omega$ there is a stage s with $g(n,s) = f(n,k)$: Let $\sigma = \langle f(n,0), \ldots, f(n,k)\rangle$. Since σ is correct for n, there is a stage s at which σ is retired. It follows from the construction that $g(n,s) = \sigma(\text{lh}(\sigma)-1) = f(n,k)$.

$\liminf_s g(n,s) = F(n)$: Since for each k there is an s with $g(n,s) = f(n,k)$, it follows that $\liminf_s g(n,s) \leq F(n)$. For the reverse inequality, let $f(n,k) = F(n)$. By the monotonicity of f, for all $j \geq k$ $f(n,j) = F(n)$. Let s be a stage satisfying the following

(i) For all $t \geq s$ and $i \leq k$, $h(n,i,t) = f(n,i)$.
(ii) If $\sigma = \langle f(n,0), \ldots, f(n,i)\rangle$ and $i < k$ then σ has been retired by s.

Then for all $t \geq s$ and all t-reasonable τ,

$$\tau \supseteq \langle f(n,0), \ldots, f(n,k)\rangle,$$

and so, $\tau(|\tau|-1) \geq f(n,k) = F(n)$. Thus, $\liminf_s g(n,s) \geq F(n)$. □

The following lemma was stated (in slightly more generality) and proven in [CDK98]. We learned this after producing our proof.

Lemma 5.3. *If a set has an η-representation then it is the range of a $\mathbf{0}'$-lmf.*

Proof. Let $\mathbf{L}$ be a linear order which η-represents a set A, and whose domain, L, is a computable subset of the rationals $\mathbb{Q}$ with the usual order $<_Q$. Let $\{\langle x_n, y_n\rangle\}_{n\in\omega}$ be a $0'$-enumeration of adjacent elements in L. (The lemma holds for finite sets, so we assume here that A is infinite, and thus that the set of pairs of adjacent elements are also infinite.) The strategy for each $n \in \omega$ is to let $f(n, \cdot)$ start with a pair $\langle x, y\rangle$ which is a block, then try to extend the block by searching for an integer w such that $\langle w, x, y\rangle$ or $\langle x, y, w\rangle$ is a block. If such a w can be found the value of $f(n, s)$ is incremented, and the construction looks for an extension of this new block.

Construction. The construction will build a $0'$-computable function f satisfying for each argument n, $f(n, s) \leq f(n, s+1)$ for every stage s, and will track the following data items:

- $\sigma_{n,s}$: The current block of L being tracked by $f(n, \cdot)$, and satisfying $f(n, s) = |\sigma_{n,s}|$ for each stage s.
- $w_{n,s}$: The current integer value that will be used to try to extend the block $\sigma_{n,s}$.

stage 0: For each n, set

- $f(n, 0) = 0$,
- $\sigma_{n,0} = \varepsilon$ and
- $w_{n,0} = 0$.

stage s+1: The construction has determined for each n, $f(n, s)$, $\sigma_{n,s}$ and $w_{n,s}$. For each $n < s$ do the following: If $\sigma_{n,s}{}^\frown\langle w_{n,s}\rangle$ is a block (or, if $\langle w_{n,s}\rangle{}^\frown\sigma_{n,s}$ is a block) then set

- $f(n, s+1) = 1 + f(n, s)$,
- $\sigma_{n,s+1} = \sigma_{n,s}{}^\frown\langle w_{n,s}\rangle$ (or, $\sigma_{n,s+1} = \langle w_{n,s}\rangle{}^\frown\sigma_{n,s}$) and
- $w_{n,s+1} = 0$.

and otherwise set

- $f(n, s+1) = f(n, s)$,
- $\sigma_{n,s+1} = \sigma_{n,s}$ and
- $w_{n,s+1} = w_{n,s} + 1$.

For $n = s$ set

- $f(s, s+1) = 2$,
- $\sigma_{s,s+1} = \langle x_s, y_s\rangle$ and
- $w_{s,s+1} = 0$.

For $n > s$ set

- $f(s, s+1) = 0$,
- $\sigma_{s,s+1} = \varepsilon$ and
- $w_{s,s+1} = 0$.

This ends the construction.

Verification. For each stage s and $n < s$, $\sigma_{n,s}$ is a block in L and $|\sigma_{n,s}| \leq |\sigma_{n,s+1}|$. Since $f(n, s) = |\sigma_{n,s}|$, $f(n, s) \leq f(n, s+1)$, and as all blocks of $\mathbf{L}$ are finite, $\lim_s f(n, s)$ exists for all n. Let $\sigma_n = \lim_s \sigma_{n,s}$. Then σ_n is a maximal block of L: σ_n is a finite block since each $\sigma_{n,s}$ is a block, and all blocks of L are finite. Let s be the first stage where $\sigma_n = \sigma_{n,s}$, so that $\sigma_{n,s-1}$ is a subblock of $\sigma_{n,s}$. Then $w_{n,s} = 0$, and for $t \geq s$, $w_{n,t} = w_{n,s} + (t - s)$. If σ_n is not maximal then let x be least such that either $\sigma_n{}^\frown\langle x\rangle$ is a block or $\langle x\rangle{}^\frown\sigma_n$ is a block. At stage $t = s + x$, $w_{n,t} = x$ and at stage $t + 1$ the construction will expand $\sigma_{n,t+1}$ from $\sigma_{n,t} = \sigma_{n,s}$ to include x. This contradicts the hypothesis that $\sigma_{n,s} = \sigma_n$. Thus, each σ_n is a maximal block. Let $F(n) = \lim_s f(n, s)$ for all n. Then A is the range of F.

□

We have now established each piece of the classification theorem for η-representable sets:

Theorem 5.4. *Let A be an infinite set which does not include 0 or 1. Then the following are equivalent:*

1. *A has an η-representation.*
2. *A is the range of a $\mathbf{0}'$-lmf.*
3. *There is a total function G and binary computable function g such that $G(\cdot) = \liminf_s g(\cdot, s)$, and A is the range of G.*

Proof. (1→ 2): by Lemma 5.3.
(2→ 3): by Lemma 5.2.
(3→ 1): by Theorem 3.3. □

There is now an alternative means for producing a Δ_3 set which has no η-representation (Theorem 4.3.) [KNS97, Lemma 2.7] produced a Δ_2 set which is not the range of any computable limitwise monotonic function. This construction can be relativized to $0'$. Recently [HMP07, Theorem 3.1] showed that a lowness requirement can be added to the construction of [KNS97] to produce a low set which is not the range of any computable limitwise monotonic function. This construction can also be relativized to $0''$ to yield,

Corollary 5.5. [2] *There is a Δ_3 set A with $(A \oplus 0')' = 0''$ which has no η-representation.*

6. Unique η-Representations

In this section we will establish a corresponding classification of the sets with a unique η-representation and the sets which are the range of a one-to-one $0'$-computable limitwise monotonic function (Theorem 6.2), then use this classification to show that the sets with η-representations have unique η-representations (Corollary 6.4.)

Lemma 6.1. *Every set with a unique η-representation is the range of a 1-1 $\mathbf{0}'$-lmf.*

Proof. Let $\mathbf{L}$ be a linear order which η-represents an infinite set A, and whose domain, L, is a computable subset of the rationals $\mathbb{Q}$ with the usual order $<_{\mathbb{Q}}$. Let $\left\{\langle x_n, y_n\rangle\right\}_{n\in\omega}$ be a $0'$-enumeration of adjacent elements in L. (Since A is infinite, this set is infinite.) The construction of Lemma 5.3 fails to produce a 1-1 function F, so the construction will be modified to ensure uniqueness. Let σ and ρ be two blocks, and say σ and ρ *clash* if they share at least one point in common. The problem in the construction for Lemma 5.3 arises if at some stage s $\sigma_{n,s}$ and $\sigma_{m,s}$ clash for some $n \neq m$. We will resolve clashes assigning a new block of the same size which does not clash with any other blocks being constructed. This is possible since A is infinite, so there are blocks of every length.

Construction. The construction will build a $0'$-computable function f satisfying for each argument n, $f(n, s) \leq f(n, s + 1)$ for every stage s and will track the following data items:

- $\sigma_{n,s}$: The current block of L being tracked by $f(n, \cdot)$, and satisfying $f(n, s) = |\sigma_{n,s}|$ for every s.
- $w_{n,s}$: The current integer value that will be used to try to extend $\sigma_{n,s}$.

stage 0: For each n, set $f(n, 0) = 0$, $\sigma_{n,0} = \varepsilon$ and $w_{n,0} = 0$.

stage s+1: The construction has determined for each n, $f(n, s)$, $\sigma_{n,s}$ and $w_{n,s}$, For each $n < s$ in increasing order perform the following: Provided $\sigma_{n,s}$ does not clash with any $\sigma_{m,s+1}$ for any $m < n$ do the following: if $\sigma_{n,s}{}^\frown\langle w_{n,s}\rangle$ is a block (or, if $\langle w_{n,s}\rangle{}^\frown\sigma_{n,s}$ is a block) set

[2]Thank you to the referee for pointing-out this result of [HMP07, Theorem 3.1].

- $f(n, s+1) = 1 + f(n, s)$,
- $\sigma_{n,s+1} = \sigma_{n,s}{}^\frown\langle w_{n,s}\rangle$ (or, $\sigma_{n,s+1} = \langle w_{n,s}\rangle{}^\frown\sigma_{n,s}$) and
- $w_{n,s+1} = 0$.

If $\sigma_{n,s}$ cannot be extended to a block by adding $w_{n,s}$ then set

- $f(n, s+1) = f(n, s)$,
- $\sigma_{n,s+1} = \sigma_{n,s}$ and
- $w_{n,s+1} = 1 + w_{n,s}$.

If for some $m < n$ there is a clash of $\sigma_{n,s}$ with $\sigma_{m,s+1}$ then do the following:

- Let the size of $\sigma_{n,s}$ be k. Use a $0'$-oracle to find a block, $\langle x_1, \ldots, x_k\rangle$, which does not clash with any other block under construction, and set
$$\sigma_{n,s+1} = \langle x_1, \ldots, x_k\rangle,$$
- $f(n, s+1) = f(n, s)$ and
- $w_{n,s+1} = 0$.

To avoid clashes, we modify the treatment of the $n = s$ case from Lemma 5.3:

- $f(s, s+1) = 2$,
- Let $\langle x, y\rangle$ be least which avoids clashes and set $\sigma_{s,s+1} = \langle x, y\rangle$ and
- $w_{s,s+1} = 0$.

For $n > s$, set $f(s, s+1) = 0$, $\sigma_{s,s+1} = \varepsilon$ and $w_{s,s+1} = 0$.
This ends the construction.

Verification. We'll show that for each n there is a stage s and maximal block σ such that for all $t \geq s$, $\sigma_{n,t} = \sigma$. Suppose this is true for all $m < n$. Let s be a stage such that for all $m < n$ and $t \geq s$, $\sigma_{m,t} = \sigma_m$ (so σ_m is a maximal block.) Then for any $t > s$, $\sigma_{n,t}$ does not clash with any block σ_m where $m < n$, and so $\sigma_{n,s+1}$ is a subblock of $\sigma_{n,t}$ for all $t \geq s+1$. Let σ_n be the maximal block containing $\sigma_{n,s+1}$. Then for each $t \geq s+1$, $\sigma_{n,t} \subseteq \sigma_n$, and so as in Lemma 5.3, $\sigma_n = \lim_s \sigma_{n,s}$. Thus, we also have $\liminf_s f(n,s) = |\sigma_n|$. Since $\mathbf{L}$ is a unique η-representation, $|\sigma_n| \neq |\sigma_m|$ whenever $m \neq n$, and therefore $F(n) = \liminf_s f(n,s)$ is 1-1. □

Theorem 6.2. *Let A be an infinite set which does not include 0 or 1. Then the following are equivalent:*

1. *A has a unique η-representation.*
2. *A is the range of a 1-1 $\mathbf{0}'$-lmf.*
3. *There is a total 1-1 function G and computable function g such that $G(\cdot) = \liminf_s g(\cdot, s)$, and A is the range of G.*

Proof. (1→ 2): Lemma 6.1.
(2→ 3): Lemma 5.2.
(3→ 1): Theorem 3.3. □

The key result of this section is that the ranges of injective $\mathbf{0}'$-lmfs are exactly the ranges of $\mathbf{0}'$-lmfs. This theorem relativizes to other degrees,

Theorem 6.3. *Let A be an infinite set. If A is the range of a $\mathbf{d}$-lmf then A is the range of 1-1 $\mathbf{d}$-lmf.*

Proof. The proof is a movable marker construction, *The Greedy Marker Game*. The game play lasts ω stages. The playing board is ω which is empty at the beginning of the game. A $\mathbf{d}$-computable f is the adversary, which is required to satisfy

(i) $f(n, s) \leq f(n, s+1)$ for all n, s and

(ii) $\lim_s f(n,s)$ exists for all n.

The goal is to construct a **d**-computable function g satisfying the following requirements:

(R.a) $g(n,s) \leq g(n,s+1)$ for all n and s,
(R.b) $\lim_s g(n,s)$ exists for all n,
(R.c) If $m \neq n$ then $\lim_s g(n,s) \neq \lim_s g(m,s)$ and
(R.d) $\{n : (\exists i)[n = \lim_s f(i,s)]\} = \{n : (\exists i)[n = \lim_s g(i,s)]\}$.

The adversary f has an infinite set of markers $\{\Phi_n\}_{n\in\omega}$. The position of the nth marker at stage s is denoted by Φ_n^s, which is determined by the value $f(n,s)$. We are also given an infinite set of markers, $\{\Gamma_n\}_{n\in\omega}$, with the position of the nth marker at stage s denoted by Γ_n^s, and which determines the value of $g(n,s)$. The Greedy Marker Game is given by the rules and conditions for successful play.

Rules of Play.

(1) At the start of the game, stage -1, the board is completely empty, all markers placed on $-\infty$ (off board).
(2) All markers may only move to the right: from smaller board positions to larger board positions (ordered by $<$ on ω).
(3) At each stage $s \geq 0$ of the game, the adversary f places one new marker, Φ_s, on position $f(s,s)$, and moves each marker on the board, Φ_n, to position $f(n,s)$.
(4) We may place an unlimited (finite) number of markers on the board. We may move any marker an unlimited (finite) number of times in any stage of play, but only movement consistent with rule 2.
(5) f may have an unlimited number of markers on a single board position at the end of any stage. We are not allowed to have more than one marker on a board position at the end of any stage.
(6) Each marker of f must eventually stop moving and remain in a fixed position for the duration of play.
(7) At the end of play, f must cover infinitely many board positions with markers.

Conditions for a win. We win if we meet all of the following conditions after ω stages of play:

(W.a) All of our markers are placed on the board.
(W.b) None of our markers are moved infinitely often (so off the board.)
(W.c) Every one of our markers is located on a position occupied by a marker of f.
(W.d) No marker of f lies on a position unoccupied by one of our markers.

Otherwise, our adversary f wins.

Our win at the end of play guarantees that the four requirements for constructing g will be fulfilled. Let $g(n,s) = 0$ for all stages s in which the marker Γ_n is not on the board. Suppose we have fulfilled the four conditions for winning at the end of play.

(R.a) $g(n,s) \leq g(n,s+1)$ for all n and s by adherence to Rule 2.
(R.b) $\lim_s g(n,s)$ exists for all n by (W.b).
(R.c) If $m \neq n$ then $\lim_s g(n,s) \neq \lim_s g(m,s)$ by condition (W.a) and adherence to Rule 5.
(R.d) Conditions (W.c) and (W.d) ensure that

$$\{n : (\exists i)[n = \lim_s f(i,s)]\} = \{n : (\exists i)[n = \lim_s g(i,s)]\}.$$

Play of the Game. We will assign a *priority* to our markers as part of a strategy to guarantee a win. The marker Γ_n has *higher priority* than the marker Γ_m when $n < m$. Let k_s denote the total number of Γ markers on the board at the end of state s. Our strategy will ensure that if Γ_n is on the board, then Γ_m is on the board for all $m < n$ and that at least one new marker is placed on the board in every stage after the first stage.

Stage -1. The board is empty. $k_{-1} = 0$.

Stage s. The markers in play at the beginning of stage s are

- $\{\Phi_i\}_{i=1}^{s-1}$ and
- $\{\Gamma_i\}_{i=1}^{k_{s-1}}$ where $k_{s-1} \geq s$.

f's movement. For each $n < s$, f moves marker Φ_n to position $f(n, s)$, and places marker Φ_s on position $f(s, s)$.

g's movement. The strategy for placing markers will be carried out in several steps.

Greedy movement. Select in order of priority the markers Γ_n in place on the board and move according to the following conditions:

(i) Γ_n lies on a position that is unoccupied by a Φ marker and there is a marker Φ_i at a higher position which is occupied by only lower priority Γ markers or no Γ markers. Move marker Γ_n to the first position Φ_i^s satisfying this condition.

(ii) Γ_n lies on a position occupied by a higher priority marker and there is a marker Φ_i at a higher position which is occupied by only lower priority Γ markers or no Γ markers. Move marker Γ_n to the first position Φ_i^s satisfying this condition.

Conflict resolution. For each marker Γ_n which is on a board position occupied by a higher priority marker, move to the first unoccupied position greater than $\max\{\Phi_m^s : m < n\}$.

Covering. For each marker Φ_n that is on a position unoccupied by a Γ marker, place the highest priority marker not yet on the board in the position Φ_n^s. If no new markers have been put into play, place the highest priority marker not yet on the board in the first open position on the board. Let k_s be the number of Γ markers on the board.

This ends the play for stage s.

Verification. There are several points about the board at the end of each stage s:

(i) $k_s > k_{s-1} \geq s$ by the *Covering* strategy.

(ii) No board position has more than one Γ marker, by the *Conflict resolution* strategy.

(iii) No Γ marker is on a position unoccupied by a Φ marker, if there is a lower priority Γ marker on a higher board position occupied by a Φ marker. This follows from the *Greedy movement* strategy.

(iv) No Φ marker is on a position unoccupied by a Γ marker by the *Covering* strategy.

We verify that our conditions for success, (W.a) to (W.d), are satisfied.

Sublemma 1. *Condition (W.a) is satisfied.*

Proof of Sublemma 1. By the end of stage s, all markers Γ_n for $n < k_s$ are in play. Since $\lim_s k_s = \infty$ by point (i) above, (W.a) is satisfied. □

Sublemma 2. *Conditions (W.b) and (W.c) are satisfied.*

Proof of Sublemma 2. Suppose that for all $m < n$ there is an ℓ such that

$$\lim_s \Gamma_m^s = \lim_s \Phi_\ell^s.$$

Let s be a stage at which for each $m < n$

$$\Gamma_m^s = \Gamma_m^t \qquad \text{for all } t \geq s.$$

By *Conflict resolution*, Γ_n is moved to a higher position if Γ_n lands on a position Γ_m^t for $m < n$, so there is a stage $t \geq s$ where Γ_n will never again move to a position occupied by Γ_m for any $m < n$. Let $u \geq t$ be the first state for which some marker Φ_i lies on a position $\Phi_i^u \geq \Gamma_n^u$, and which is unoccupied by any markers with higher priority than Γ_n. (There must be such a stage u since the range of F is infinite.) By *Conflict resolution*, $\Gamma_n^u = \Gamma_n^t$ at the beginning of stage u, so by *Greedy movement*, $\Gamma_n^u = \Phi_i^u$. Since no higher priority marker will force Γ_n off positions occupied by Φ_i, it follows by *Conflict resolution* that $\Gamma_n^v = \Phi_i^v$ for all stages $v \geq u$. Thus

$$\lim_s \Gamma_n^s = \lim_s \Phi_j^s.$$

□

Sublemma 3. *Condition (W.d) is satisfied.*

Proof of Sublemma 3. Let t be a stage such that $\Phi_i^t = \lim_s \Phi_i^s$. If Φ_i^t is unoccupied by a marker from Γ, then by *Covering* at stage t some marker Γ_n is moved to position Φ_i^t. By point (iv) above, there will be a Γ marker on the position Φ_i^u for all $u \geq t$. It is possible that for some $u \geq t$, Γ_n occupies position Φ_i^u but at stage $u+1$, Γ_m occupies Φ_i^{u+1}. But in this case $m < n$, so that for some n

$$\lim_s \Gamma_n^s = \lim_s \Phi_i^s.$$

□

This completes Theorem 6.3. □

Corollary 6.4. *A set has an η-representation if and only if it has a unique η-representation.*

Corollary 6.5. *The degrees with unique η-representations are exactly the Σ_3 degrees.*

7. Strong η-Representations

This section consists of two negative results: there is a Δ_3 degree with no strong η-representation (Theorem 7.1), and the ranges of order-increasing $\mathbf{0}'$-computable limitwise monotonic functions do not characterize the sets with strong η-representations (Theorem 7.3.)

Although not all Δ_3 *sets* have η-representations (Theorem 4.3), all Σ_3 degrees have η-representations (Theorem 4.4); so, it may be hoped that all Δ_3 degrees have strong η-representations. The next theorem shows this is not the case,

Theorem 7.1. [3] *There is a Δ_3 degree which has no strong η-representation. This degree can be taken to be minimal among the Turing degrees.*

Proof. The proof uses a tree splitting argument, which is a variant of Spector's construction of a Δ_3 minimal degree, as presented in [Soa87, Theorem IV.5.6]. The construction here also produces a minimal degree: a degree $\mathbf{d}$ is *minimal* if whenever $\mathbf{a} < \mathbf{d}$ then $\mathbf{a} = \mathbf{0}$. We allow access to a $\mathbf{0}''$-oracle in the proof. Let $\Phi_0^X, \Phi_1^X, \dots$ be a listing of the Turing functionals and $L_0, L_1, \dots$ be a listing of all computable subsets of $\mathcal{Q}$, where the underlying ordering is taken to be $\leq_{\mathcal{Q}}$. (This enumeration is possible using $\mathbf{0}''$ since the set of indices of computable subsets of $\mathcal{Q}$ is Π_2.) We will construct a set $A \in \Delta_3$ satisfying the requirements:

[3]I have recently learned that Maxim Zubkov reported the existence of a Δ_3 degree with no strong η-representation at the Malcev International Confererence in November, 2005.

$R_{\langle e,i\rangle}$: If L_i is a strong η-representation of the set C, and $\Phi_e^A = C$, then C is computable.

We introduce splitting trees (see [Soa87, Section VI.5]), and then explain how this will be used in the construction to satisfy the requirements. Let $\mathcal{T}$ be a binary branching tree, $\mathcal{T} \subset \{0,1\}^{<\omega}$, and $\sigma \in \mathcal{T}$. We will say that σ *e-splits* in $\mathcal{T}$ if there are $\rho, \tau \in \mathcal{T}$ such that

(i) $\sigma \subset \rho$ and $\sigma \subset \tau$
(ii) Φ_e^ρ and Φ_e^τ are *incompatible*:

$$\exists x \left[\Phi_{e,|\rho|}^{\rho}(x)\downarrow \neq \ \Phi_{e,|\tau|}^{\tau}(x)\downarrow\right]$$

Given a computable tree $\mathcal{T}$ and $\sigma \in \mathcal{T}$, we can use a $\mathbf{0}''$-oracle to determine if σ e-splits in $\mathcal{T}$. If σ does e-split in $\mathcal{T}$ then a computable procedure can produce witnesses $\rho, \tau \in \mathcal{T}$ and argument x such that ρ and τ are incompatible on argument x. A binary branching tree $\mathcal{T}$ is an *e-splitting tree*, if each $\sigma \in \mathcal{T}$ e-splits in $\mathcal{T}$. If $\mathcal{T}$ is a computable tree which contains an e-splitting subtree, then there is a computable subtree of $\mathcal{T}$ which is an e-splitting tree. If $\mathcal{T}$ is a computable tree which does not contain an e-splitting subtree, then there is a string $\rho \in \mathcal{T}$ such that for every path B extending ρ through $\mathcal{T}$, either Φ_e^B is computable or Φ_e^B is not total ([Soa87, Lemma IV.5.2].) We can use $\mathbf{0}''$ to decide whether a computable tree $\mathcal{T}$ has an e-splitting subtree, since this is a Π_2 question.

If we are given a linear ordering L_i which is a strong η-representation of a set C, then C is Δ_3 by Theorem 4.1. To determine if $n \in C$, find a block of n adjacencies, and continue to search for new blocks containing n adjacencies and lying entirely below the given block. Eventually a least block of n adjacencies will be found, and if this is maximal then $n \in C$, and otherwise $n \notin C$. Suppose we have determined in the construction an initial segment σ of A, and want to try to satisfy requirement $R_{\langle e,i\rangle}$. We can use $\mathbf{0}''$ to find extensions ρ and τ of σ, and an argument $n \in \omega$ so that

$$\Phi_e^\rho(n)\downarrow \neq \Phi_e^\tau(n)\downarrow .$$

(If there are no such strings ρ and τ and argument n, then Φ_e^A is either computable or not total–in either case the requirement is satisfied.) If L_i is a strong η-representation of a set C, then we can use $\mathbf{0}''$ to extend σ to ρ or τ so that $\Phi_e^A \neq \chi_C$. The difficulty with meeting requirement $R_{\langle e,i\rangle}$ is that we cannot use our access to $\mathbf{0}''$ to decide if a linear order actually strongly η-represents a set. If a linear order is not a strong η-representation then a search for a maximal block of n adjacencies need not ever terminate. (This could be the case, for example, if there is a ζ block in the linear order and the search finds lower and lower n-sized blocks in this ζ block.) The construction cannot block lower priority requirements from acting. Our solution is to use e-splitting trees to allow lower priority requirements to act without frustrating the search of higher priority requirements.

Suppose we have determined in the construction an initial segment σ of A, and want to try to satisfy requirement R_u, where $u = \langle e,i\rangle$. We can use $\mathbf{0}''$ to determine if there is an e-splitting tree, $\mathcal{T}_u$, whose root is σ. (If there is no e-splitting tree then there is an extension of σ which will ensure that Φ_e^A will be either computable or not total, and so R_u will be satisfied.) The requirement fixes an argument $n \in \omega$ for an e-split in the tree $\mathcal{T}_u$ and uses $\mathbf{0}''$ to search for a maximal block of n adjacencies. Lower priority requirements must restrict their extensions to strings in $\mathcal{T}_u$, so the splitting trees they use must be subtrees of $\mathcal{T}_u$. Suppose a lower priority requirement wants to extend σ to ρ where $\rho \in \mathcal{T}_u$. The subtree of $\mathcal{T}_u$ rooted at ρ is also an e-splitting tree, but it is possible that none of the e-splittings extending ρ split with the current argument n. In this case, there must be an $m > n$ such that m is an argument for which there is an e-splitting in $\mathcal{T}_u$ above ρ. The requirement R_u attempts to extend its current n-sized block of adjacencies to a block of size m. If this is possible, then R_u can allow the lower priority requirement to extend to ρ, and continue its search for m-sized blocks of adjacencies. If the current block cannot be extended to an m-sized block of adjacencies, then we can act to satisfy requirement R_u, since the search for a maximal block of n adjacencies can be restricted (even if L_i is not a strong η-representation.)

Construction.

We will construct A by stages, determining σ_s at stage s whose length is at least s and which extends σ_t for all $t < s$. This will determine the set A by $\chi_A = \lim_s \sigma_s$. For each requirement $R_{\langle e,i\rangle}$ we will associate the following data (which will depend upon the stage s, which is suppressed in the notation)

- $\mathcal{T}_{\langle e,i\rangle}$ is a computable tree (possibly an e-splitting tree),
- $x_{\langle e,i\rangle}$ is a witness to defeat L_i, and
- $X_{\langle e,i\rangle}$ is a block of adjacencies of length $x_{\langle e,i\rangle}$ in the linear order L_i.

A requirement may be satisfied only modulo conditions which may later fail to hold, so we will also maintain a classification of the current *status* of requirements for each stage s in the construction:

- *waiting*: A requirement is waiting for an opportunity to begin actively searching for satisfaction.
- *active*: A requirement is actively searching for satisfaction.
- *retired*: A requirement has been confirmed by $\mathbf{0}''$ to be currently satisfied.

We will ensure the following invariant conditions hold at the end of each stage.

(a) For any requirements R_u and R_v with $u < v$, $\mathcal{T}_u \supseteq \mathcal{T}_v$.
(b) If requirement R_u is waiting, then $\mathcal{T}_u = \emptyset$ and for any $v > u$, R_v is waiting as well.
(c) If a requirement R_u is either active or retired, then $\mathcal{T}_u$ is a nonempty computable tree and $\sigma_s \in \mathcal{T}_u$.
(d) If a requirement R_u is active then
  (i) $\mathcal{T}_u$ is an e-splitting tree,
  (ii) $\mathcal{T}_u$ e-splits for argument x_u, and
  (iii) X_u is a block of adjacencies of size x_u in L_i.

Stage 0: Let $\sigma_0 = \epsilon$ and $\mathcal{T}_{\langle e,i\rangle} = \emptyset$, $x_{\langle e,i\rangle} = 0$, and $X_{\langle e,i\rangle} = \epsilon$ for all e and i. All requirements are waiting.

Stage $s+1$: Let $u_1, \ldots, u_k$ be the active requirements at the end of stage s, and assume all the invariant conditions hold at the end of stage s. We first determine which requirements, if any, are *ready-to-act* for satisfaction. For each $u_j = \langle e_j, i_j\rangle$ use $\mathbf{0}''$ to answer the following question

> Is there a block of adjacencies Y in the linear order L_{i_j} of cardinality x_{u_j} which lies strictly below the block X_{u_j} in the order?

If Yes, update $X_{u_j} = Y$ (the requirement R_{u_j} is not ready-to-act), and otherwise, mark the requirement R_{u_j} as ready-to-act. If no active requirement has been marked ready-to-act, then let $R_{\langle e,i\rangle}$ be the highest priority requirement which is waiting. Let R_u be the lowest priority requirement which is either active or waiting and use $\mathbf{0}''$ to check if there is an e-splitting subtree of $\mathcal{T}_u$. (If there is no such requirement, let $\mathcal{T}_u$ be the full binary branching tree.) If there is no e-splitting subtree of $\mathcal{T}_u$, then mark $R_{\langle e,i\rangle}$ ready-to-act and set $\mathcal{T}_{\langle e,i\rangle} = \mathcal{T}_u$. If there is an e-splitting subtree of $\mathcal{T}_u$ then mark R_u active and set $u_{k+1} = \langle e, i\rangle$, $\mathcal{T}_{u_{k+1}}$ to an e-splitting subtree of $\mathcal{T}_u$ rooted at σ_s, $x_{u_{k+1}}$ to the first witness to a splitting in $\mathcal{T}_u$, and $X_{u_{k+1}}$ to a block of adjacencies of size $x_{u_{k+1}}$ If no such block of adjacencies exists, then mark the requirement $R_{\langle e,i\rangle}$ ready-to-act. (In this case, if L_i does strongly η-represent a set, the set is finite and all its members are smaller than $x_{u_{k+1}}$.) If there is still no requirement ready-to-act then the construction will have to force an extension of σ_s. Note that these assignments preserve the invariant conditions.

A requirement R_u, with $u = \langle e, i\rangle$, which is ready-to-act must choose an extension $\rho \supset \sigma_s$. There are several cases which determine the ρ selected.

(1) If R_u is ready-to-act because no e-splitting tree could be found, then let ρ be the shortest and left-most extension of σ_s in $\mathcal{T}_u$ with no e-splitting above ρ.
(2) If R_u is ready-to-act because either X_u is a maximal block of size x_u, or there is no maximal block of this size to the left of block X_u in the order L_i, then $u = u_i$ for some $i \leq k + 1$. By invariant condition (d.iii) for active requirements, x_u is an argument on the e_u-splitting tree, $\mathcal{T}_u$. If X_u is a maximal block of size x_u, then choose $\rho \in \mathcal{T}_u$ such that

$$\Phi^{\rho}_{e,|\rho|}(x_u) \downarrow= 0$$

and otherwise choose $\rho \in \mathcal{T}_u$ such that

$$\Phi^{\rho}_{e,|\rho|}(x_u) \downarrow= 1$$

(3) If no requirements are currently ready-to-act, then choose the left-most immediate successor node $\rho \supset \sigma_s$ in $\mathcal{T}_{u_{k+1}}$.

Before an extension of σ_s to ρ will be permitted, requirements must seek *permission* among all active higher priority requirements. If no requirement was ready-to-act, then the construction must seek permission from each R_{u_j} for $j \leq k + 1$ before extending σ_s. Suppose the requirements that must grant permission to extend are $R_{u_0},\ldots,R_{u_i}$. For each $j \leq i$ the following check is performed, in order of priority:

(1) Let y be the argument of the first e_{u_j}-splitting in $\mathcal{T}_{u_j}$ above ρ with $y > x_{u_j}$. Such a node exists in the tree since above each node there are infinitely many e_{u_j}-splittings.
(2) Try to extend the block X_{u_j} to a block of adjacencies Y of size y. If this is impossible, then mark requirement R_{u_j} as ready-to-act, and otherwise update $\mathcal{T}_{u_j}$ to the e_{u_j}-splitting subtree rooted at ρ, x_{u_j} to y, and X_{u_j} to Y.
(3) If R_{u_j} has been marked ready-to-act, then it has higher priority, so the procedure of obtaining permission from higher priority requirements continues for R_{u_j}. Otherwise R_{u_j} gives permission to R_u to act.

If all higher priority R_{u_j} give permission then it is permissible to extend σ_s. If a requirement R_u was ready-to-act, then set $\sigma_{s+1} = \rho$ and mark R_u retired. For every requirement R_v of lower priority than R_u which is marked active or retired, re-set $\mathcal{T}_v = \emptyset$, $x_v = 0$, $X_v = \emptyset$ and mark R_v waiting. If no requirement was marked ready-to-act, then set σ_{s+1} to ρ. There are no further changes to the state of the requirements. This ends stage $s + 1$. Note that the construction preserves the invariant conditions for all requirements.

Let $A = \bigcup_s \sigma_s$. For each s, $|\sigma_s| \geq s$ and if $t \geq s$, then $\sigma_t \supset \sigma_s$. Since the construction only uses a $\mathbf{0}''$-oracle, it follows that A is Δ_3.

Verification.

We will say that the status of a requirement R_u is *stable* at stage s if the status of R_u at any stage $t \geq s$ is the same as the status at the beginning of stage s, and R_u is never marked ready-to-act at any stage $t \geq s$.

Sublemma 1. *For each requirement R_u there is a stage s such that R_u is stable at stage s. If s is the least stage such that all higher priority requirements are stable, then R_u is the highest priority requirement waiting at stage s.*

Proof. The proof is by induction on the priority ordering of requirements. Assume for the inductive hypothesis that there is a least stage s such that

For each $v < u$, the requirement R_v is stable at stage s.

So, for each $v < u$ and each stage $t \geq s$, the requirement R_v is either retired at stage t or is active but not marked ready-to-act at stage t. We first show that R_u is the highest priority requirement marked waiting at the beginning of stage s. At stage $s-1$ either some higher priority requirement was moved from waiting to active, so R_u must be waiting at stage s; or some higher priority requirement was ready-to-act, in which case some requirement of higher priority actually extended σ_{s-1} and moved from active to retired, which forced all lower priority requirements (including R_u) to waiting status. Since only one requirement is moved from waiting to active (or retired), R_u is the highest priority requirement waiting at the opening of stage s.

R_u will become active or retired in stage s (since no higher priority requirement will be marked ready-to-act at stage s.) The only way the status of R_u can change from active or retired to waiting is if some higher priority requirement is marked ready-to-act, but this cannot happen after stage $s-1$ by the inductive hypothesis. So, R_u will be active or retired for all stages $t \geq s$. If R_u is never marked ready-to-act at any stage $t \geq s$, then it will be stable at stage $s+1$. If there is a stage $t \geq s$ in which R_u is marked ready-to-act, then all higher priority requirements will give R_u permission to act (the only way a higher priority requirement could fail to give permission is if it becomes marked ready-to-act, which cannot happen after stage $s-1$.) So, R_u will act at stage t and change status to retired, and become stable at stage $t+1$. Therefore, there will be a stage $t \geq s$ such that R_u will be stable. □

Sublemma 2. *Let s be the least stage at which all higher priority requirements than R_u are stable. Then A is a path in $\mathcal{T}_u^s$.*

Proof. Let s be the least stage such that all higher priority requirements than R_u are stable. By Sublemma 1, R_u was waiting at the beginning of stage s, and is the highest priority requirement waiting at stage s. Since all higher priority requirements are stable, a nonempty tree is assigned to R_u at stage s, and for all stages $t \geq s$, $\mathcal{T}_u^s = \mathcal{T}_u^t$ (since all higher priority requirements are stable.) By the invariant condition (c) for nonempty trees, $\sigma_t \in \mathcal{T}_u^t$ for all stages $t \geq s$. Thus, A is a path in $\mathcal{T}_u^s$. □

Sublemma 3. *$R_{\langle e,i \rangle}$ is satisfied for each e and i.*

Proof. Let $u = \langle e, i \rangle$. From Sublemma 1, let s be the least stage such that for all higher priority requirements are stable. At the beginning of stage s, R_u is the highest priority requirement which is marked waiting. R_u will be given a chance to act at stage s, since no higher priority requirement will ever be ready-to-act.

Suppose no e-splitting tree is found for R_u at stage s. Then R_u is marked ready-to-act at stage s. The tree $\mathcal{T}_u$ is a computable tree rooted at σ_s and is a subset of all higher priority trees (by the invariant condition (a).) Since no higher priority requirement will be ready-to-act at stage s (all are stable), R_u will be given permission to act at stage s. The choice of extension $\rho \supset \sigma_s$ by R_u guarantees that for any infinite path $B \supset \rho$ in $\mathcal{T}_u$ either Φ_e^B is computable or Φ_e^B is not total. By Sublemma 2, A will be a path in $\mathcal{T}_u^s$ and extends $\sigma_{s+1} = \rho$. Thus, R_u will be satisfied by the construction.

Suppose R_u is marked active at stage s (so that an e-splitting tree was found.) Then x_u is set to an argument at a splitting of $\mathcal{T}_u$ and X_u is a block of adjacencies of L_i of size x_u (possibly empty if none were found, in which case R_u will be marked ready-to-act.) For all stages $t \geq s$ in which R_u is marked active, the following conditions hold:

1. $\mathcal{T}_u^t = \mathcal{T}_u^s$ and for all R_v of lower priority, $\mathcal{T}_v^t \subseteq \mathcal{T}_u^t$.
2. If R_u does not act at stage t, then one of the following conditions holds at stage $t+1$:
   - (i) $X_u^t \subseteq X_u^{t+1}$ and $\left|X_u^t\right| \leq \left|X_u^{t+1}\right|$, with equality only when R_u is ready-to-act, or
   - (ii) Let $y \in X_u^t$ be smallest in the block X_u^t and $z \in X_u^{t+1}$ be largest in the block X_u^{t+1}, then $z <_{\mathcal{Q}} y$. (The block X_u^{t+1} lies entirely below the block X_u^t.)
3. x_u^t is the cardinality of X_u^t and there is a splitting in $\mathcal{T}_u^t$ with argument x_u^t.

(1) is true by the stability of all higher priority requirements, and the invariant condition (a). (2) is true (that is, one of the conditions (i) or (ii) holds at $t+1$): at the beginning of stage $t+1$, the construction searches for a new block entirely below the block X_u^t. If a block is found then condition (ii) holds at $t+1$; otherwise, R_u is marked ready-to-act and $X_u^t = X_u^{t+1}$ so that condition (i) holds at $t+1$. If R_u must extend permission to another requirement to act at stage $t+1$, then the construction will look to extend the block X_u^{t+1}. (3) is true by invariant condition (d).

Suppose L_i is a strong η-representation. Then for any block of adjacencies B of L_i, there are only finitely many adjacencies in the same block or in blocks located below B. For all $t \geq s$ (2) holds, so by (2i) and (2ii) there will eventually be a stage $t \geq s$ in which X_u^t is either maximal or there is no block of size x_u^t below X_u^t. At stage t, R_u will be marked ready-to-act, and since all higher priority requirements are stable, R_u will extend σ_t to σ_{t+1}. If X_u^t is a maximal block of size x_u^t, then by (3), R_u can choose an extension $\rho \supset \sigma_t$ such that

$$\Phi_{e,|\rho|}^{\rho}(x_u^t) \downarrow = 0.$$

If there is no block of size x_u^t below X_u^t, then there is no maximal block of size x_u^t in L_i and by (3), R_u can choose an extension $\rho \supset \sigma_t$ such that

$$\Phi_{e,|\rho|}^{\rho} x_u^t) \downarrow = 1.$$

The requirement R_u is satisfied since $\rho = \sigma_{t+1} \subset A$. □

Sublemma 4. *A is non-computable.*

Proof. For any computable set C, there is a linear order L_i which strongly η-represents C. Let e be the code of the following program:

$$\Phi_e^{\rho}(n) = \begin{cases} 1 & \text{if } n < |\rho| \text{ and } \rho(n) = 1 \\ 0 & \text{if } n < |\rho| \text{ and } \rho(n) = 0 \\ \uparrow & \text{otherwise} \end{cases}$$

There is an e-splitting subtree of any splitting tree, and so at any stage at which $R_{\langle e,i \rangle}$ is marked ready-to-act, the tree $\mathcal{T}_{\langle e,i \rangle} \neq \emptyset$. By Sublemma 1, $R_{\langle e,i \rangle}$ is eventually satisfied, so that at some stage s, we have strings ρ and τ satisfying

$$\sigma_{s-1} \subset \rho, \tau \in \mathcal{T}_{\langle e,i \rangle},$$

and a number n such that

$$\Phi^{\rho}_{e,|\rho|}(n)\downarrow \neq \Phi^{\tau}_{e,|\tau|}(n)\downarrow .$$

At stage s, σ_s is chosen to be either ρ or τ so that

$$\Phi^{\sigma_s}_e(n) \neq C(n),$$

but $\Phi^{\sigma_s}_e(n) = \sigma_s(n) = A(n)$. □

The theorem follows from Sublemma 3 and Sublemma 4: By Sublemma 3, each requirement

$R_{\langle e,i\rangle}$: If L_i strongly η-represents a set C and $\Phi^A_e = C$, then C is computable.

is satisfied. Suppose L_i is a strong η-representation of C. If for any e, $\Phi^A_e = C$, then C is computable. But by Sublemma 4, A is non-computable, so $A \neq C$. Thus, for $\mathbf{a}$ the degree of A, $\mathbf{a} > 0$ and if $\mathbf{c} \leq \mathbf{a}$ has a strong η-representation, then $\mathbf{c} = \mathbf{0}$.

Sublemma 5. *A is minimal in the Turing degrees.*

Proof. Suppose Φ^A_e is total, but not computable. We will show that $A \leq_{\mathrm{T}} \Phi^A_e$. Let L_i be any computable subset of $\mathcal{Q}$. Let $u = \langle e, i\rangle$. Fix a stage s as in Sublemma 1, so that all requirements with higher priority than R_u are stable. By Sublemma 1, R_u is the highest priority requirement waiting at stage s, and will be given a chance to act at stage s. Let R_v be the lowest priority requirement whose tree is nonempty at the beginning of stage s. Since Φ^A_e is total and not computable, there is an e-splitting subtree of $\mathcal{T}_v$ (otherwise the construction would have chosen σ_{s+1} to ensure Φ^A_e was computable or not total.) By Sublemma 2, A is a path in the e-splitting tree $\mathcal{T}_u$.

We will use Φ^A_e to compute a sequence of strings $(\sigma_s)_{s\in\omega}$ such that $A = \cup_s \sigma_s$. Let $\sigma_0 = \epsilon$. Suppose σ_s has been determined with $\sigma_s \subset A$ (and so, $\sigma_s \in \mathcal{T}_u$.) Let ρ and τ be the shortest pair of strings in $\mathcal{T}_u$ which extend σ_s and such that

$$(\exists x)\ \left[\Phi^{\rho}_{e,|\rho|}(x)\downarrow \neq \Phi^{\tau}_{e,|\tau|}(x)\downarrow\right].$$

(There must be such strings since $\mathcal{T}_u$ is e-splitting.) Finding strings ρ and τ and argument x is computable given σ_s, since $\mathcal{T}_u$ is a computable tree. If

$$\Phi^A_e(x) = \Phi^{\rho}_e(x)$$

then let $\sigma_{s+1} = \rho$ and otherwise let $\sigma_{s+1} = \tau$. Since A must extend ρ or τ, we have $\sigma_{s+1} \subset A$. □

This completes Theorem 7.1. □

By Lemma 5.2 the following equivalence holds

(i) A is the range of an order-preserving $\mathbf{0}'$-lmf.
(ii) There is a total order-preserving G and computable g such that $G(\cdot) = \liminf_s g(\cdot, s)$, and A is the range of G.

By Theorem 3.3 these sets have strong η-representations; but we will show that the reverse inclusion does not hold. Constructing a strong η-representation by densification allows changing from building $\eta + \mathbf{n}$ for some $n > 1$ to building a copy of η for a step. If carried-out infinitely often, a copy of η is produced, which can then be absorbed into an adjacent block of η. This process of *rubbing-out* a block cannot be matched by an order-preserving $\mathbf{0}'$-lmf, and is partly captured in the following lemma:

Lemma 7.2. *Let $g : \omega \times \omega \to \omega$ be a computable function and $G(\cdot) = \liminf_s g(\cdot, s)$ a total function satisfying the following condition*

- *Whenever $m < n$ and $1 < G(m), G(n)$, then $G(m) < G(n)$.*

Then the set ran$(G) \setminus \{0,1\}$ has a strong η representation.

Proof. By Theorem 3.3, there is an η-representation $\mathbf{L}$ of the set $\operatorname{ran}(G) \setminus \{0,1\}$, constructed by densification according to g. The order of the maximal blocks, whose size is greater than one, in the ordering of $\mathbf{L}$ will be strictly increasing in size (by the condition in the Lemma.) But since $\eta + \eta = \eta = \eta + 1 + \eta$, the blocks of size zero or one are absorbed into adjacent copies of η. So, $\mathbf{L}$ is actually a strong η-representation. □

Theorem 7.3. *There is a set which has a strong η-representation but which is not the range of an order-preserving $\mathbf{0}'$-lmf.*

Proof. [4] Let $\psi_0, \psi_1, \ldots$ be an enumeration of partial computable binary functions and let $\Psi_e(\cdot) = \liminf_s \psi_e(\cdot, s)$, where $\Psi_e(n)$ exists if and only if both $\psi_e(n,s)\downarrow$ for every $s \in \omega$ and $\liminf_s \psi(n,s)$ exists. We will produce a set A with a strong η representation but such that for each $e \in \omega$, if Ψ_e is total and order-preserving, then A is not the range of Ψ_e. (This is sufficient by Lemma 5.2.) The construction actually builds a binary computable function $g(\cdot,\cdot)$ satisfying the properties

(a) $g(2e, s) = 0$ or $g(2e, s) = 2e + 2$ for all e and s.
(b) $g(2e + 1, s) = 2e + 3$ for all e and all s.

Let $G(\cdot) = \liminf_s g(\cdot, s)$, so that G is a total function and $\operatorname{ran}(G) \setminus \{0,1\}$ satisfies the condition of Lemma 7.2, so has a strong η-representation.

Let $a_0, a_1, \ldots$ be an enumeration of $\operatorname{ran}(G) \setminus \{0,1\}$ in increasing order. For each $e \in \omega$ there is an index $n(e) \leq 2e + 1$ with $a_{n(e)} = 2e + 3$. The strategy is to place $2e + 3$ in the enumeration to ensure that Ψ_e fails to place $2e + 3$: $\Psi_e(n(e)) \neq 2e + 3 = a_{n(e)}$. We will use arguments $2e$ and $2e + 1$ of G to attack Ψ_e. Consider the strategy against the single function Ψ_0: we will ensure $G(1) = 3$, but $G(0)$ may be either 0 or 2. If $\Psi_0(0) \geq 3$ and Ψ_0 is order-preserving then $\Psi_0(1) > 3$; but, we will construct $G(0) = 2$ and $G(1) = 3$, so that $a_1 = 3$ but $\Psi_0(1) \neq 3 = G(1)$. If $\Psi_0(0) < 3$ then we will construct $G(0) = 0$ and $G(1) = 3$, so that $a_0 = 3$ but $\Psi_0(0) \neq 3 = G(1)$. The way we will construct G, is to set $g(1, s) = 2e + 3$ for all s, but set $g(0, s) = 0$ on stages t where $\psi(0, t) < 3$ and set $g(0, s) = 2$ on stages t where $\psi(0, t) \geq 3$. (The stages t may lag behind the stages s, since the construction must wait for $\psi(0, t)$ to converge.) If $\Psi_0(0) \geq 3$, then then for all sufficiently large stages t, $\psi(0, t) \geq 3$, so that for almost all stages s, $g(0, s) = 2$, and thus $G(0) = 2$. But if $\Psi_0(0) < 3$, then there will be infinitely many stages t in which $\psi(0, t) < 3$, so there will be infinitely many stages s in which $g(0, s) = 0$, and thus $G(0) = 0$.

The construction against two functions throws in an added complication, since we will need to guess which argument we are attacking. Consider the strategy against two functions, Ψ_0 and Ψ_1. In this case $G(0) = 0$ or 2 and $G(1) = 3$ to force Ψ_0 to fail, and $G(2) = 0$ or 4 and $G(3) = 5$ to force Ψ_1 to fail. But now we must also guess whether $G(0) = 0$ or $G(0) = 2$ to determine whether we are constructing against $\Psi_1(1)$ or $\Psi_1(2)$. Suppose that $G(0) = 2$, then it follows that either $a_2 = 5$ or $a_3 = 5$, so we need to construct against $\Psi_1(2)$. But, since $G(0) = 2$, for almost every stage s, $g(0, s) = 2$, so that for almost every stage s we can set $g(2, s)$ against $\psi_1(2, s)$, just as in the case against a single function. But, if $G(0) = 0$ then

[4] Rod Downey suggested exploiting the ability to *rub-out* blocks during densification to diagonalize against increasing $\mathbf{0}'$-lmfs, which is the heart of the construction.

$a_1 = 5$ or $a_2 = 5$, so we need to construct $G(2)$ against $\Psi_1(1)$. The problem is that there may be infinitely stages with $g(0,s) = 2$, so that we incorrectly target $\psi_1(2,s)$, as well as infinitely many stages with $g(0,s) = 0$ so that we correctly target $\psi_1(1,s)$. If Ψ_1 is really order-preserving then $\Psi_1(1) < \Psi_1(2)$, so that if $\Psi_1(1) \geq 5$ then there is no harm done by infinitely often constructing against $\psi_1(2,s)$ since $\Psi_1(2) \geq 5$ as well. If $\Psi_1(1) < 5$ then for infinitely many stages s, we will set $g(2,s) = 0$, so that it will not matter what the construction does on stages against $\psi_1(2,s)$. Thus, the construction will succeed against both Ψ_0 and Ψ_1.

There is one last complication in the construction, which can arise when building against three or more functions. In the previous paragraph, we assumed $\Psi_1(1) \leq \Psi_1(2)$, and there is no harm to the strategy against Ψ_1 since if this is not true, then Ψ_1 fails to be order-preserving and is automatically defeated. But, $\Psi_1(1) > \Psi_1(2)$ could lead to our attacking the wrong argument against Ψ_e where $e \geq 2$. Suppose that $G(0) = 0$, so that $\Psi_1(1)$ is the correct argument to attack, yet it is possible that we infinitely often attack $\Psi_2(2)$ as well. But suppose that $\Psi_1(1) \geq 5$ and $\Psi_1(2) < 5$. Whenever we build against $\Psi_1(1)$ we are guessing $G(0) = 0$, $G(1) = 3$, $G(2) = 4$ and $G(3) = 5$, so that we attack $\Psi_2(3)$; and whenever we build against $\Psi_1(2)$ we are guessing $G(0) = 2$, $G(1) = 3$, $G(2) = 0$ and $G(3) = 5$, so we again attack $\Psi_2(3)$. In fact, $G(0) = 0$ (by hypothesis) and $G(2) = 0$ (provided infinitely often we construct against $\Psi_1(2)$), and so we should be attacking $\Psi_2(2)$. To guard against this possibility, we will build against the minimum of $\Psi_1(1)$ and $\Psi_1(2)$ whenever we build against $\Psi_1(1)$. More generally to defeat Ψ_e we build against the minimum of Ψ_e for all arguments between the current guess (which is at least e, the fewest possible non-zero G values in the first $2e$ arguments of G) and $2e$ (the most possible non-zero G values in the first $2e$ arguments of G.)

Construction. For each $e \in \omega$ and each stage s of the construction we will associate several values

- $n_e^s \in [e, 2e]$, the current argument against Ψ_e.
- $u_{e,m}^s$ where $m \in [e, 2e]$, at stage s we are building against $\psi_e(n_e^s, u_{e,n_e^s}^s)$.
- $g_{2e}(s), g_{2e+1}(s)$ where
  $g(2e,s) = 0$ if $\psi_e(m, u_{e,j}^t) < 2e+3$ for some $m \in [n_e^s, 2e]$ and $t \in [s', s]$ where s' was the last stage (or 0) where $n_e^{s'} = n_e^s$, or
  $g(2e,s) = 2e+2$ if $\psi_e(m, u_{e,j}^t) \geq 2e+3$ for all $m \in [n_e^s, 2e]$ and $t \in [s', s]$ where s' was the last stage (or 0) where $n_e^{s'} = n_e^s$, and
  $g(2e+1, s) = 2e+3$ for all s.

For simplicity in the notation, we will add the following condition for all $e \in \omega$ and $m \in [e, 2e]$

$$\psi_e(j, -1) = 2e + 3.$$

stage 0: Set the following values: $n_e^0 = 2e$ and $u_{e,m} = -1$ for all $m \in [e, 2e]$. Set $g(2e, 0) = 2e+2$ and $g(2e+1, 0) = 2e+3$ for all e.

stage $s+1$: Start with $n = 0$, then for each $e < s+1$ update as follows, starting with $e = 0$:

1. Set $n_e^{s+1} = n$.
2. For each $m \in [e, 2e]$, if $\psi_e(m, u_{e,m}^s + 1)$ converges in $s+1$ steps, then set $u_{e,m}^{s+1} = u_{e,m}^s + 1$; and otherwise, set $u_{e,m}^{s+1} = u_{e,m}^s$.
3. Set $g(2e+1, s+1) = 2e+3$.
4. Let z be the minimum of $\psi_e(m, u_{e,m}^t)$ where $m \in [n, 2e]$ and $t \in [s', s+1]$ where s' was the last stage of the construction when $n_e^{s'} = n$, and $s' = 0$ if $s+1$ is the first stage with $n_e^{s+1} = n$.

If $z \geq 2e+3$, then let $g(2e, s+1) = 2e+2$ and increment n by 2. Otherwise, (where $z < 2e+3$) let $g(2e, s+1) = 0$ and increment n by 1.

(5) Proceed to $e+1$ provided $e+1 < s+1$.

For $e \geq s+1$ set:

- $n_e^{s+1} = n_e^s$, $u_{e,j}^{s+1} = u_{e,j}^s$ for all $j \in [e, 2e]$, and
- $g(2e, s+1) = g(2e, s)$, $g(2e+1, s+1) = 2e+3$.

This ends the construction.

Let $G(\cdot) = \liminf_s g(\cdot, s)$, and let $A = \mathrm{ran}(G) \setminus \{0, 1\}$. Then G is total, and A has a strong η-representation by Lemma 7.2. We will show that for each $e \in \omega$, if Ψ_e is total and order-preserving, then A is not the range of Ψ_e.

Verification.

Sublemma 1. *For each $e \in \omega$, the following conditions hold:*

(a) $n_e^s \in [e, 2e]$ *for all* $s \in \omega$.

(b) *If* $\big|A \cap [0, 2e-1]\big| = n_e$, *then* $\liminf_s n_e^s = n_e$.

Proof. Part (a) is satisfied as is clear from the construction.

The proof of (b) is by induction on e. (b) holds for $e = 0$, since $n_e^s = 0$ for all s. Suppose (b) holds for e, and we will show that if $\big|A \cap [0, 2e+1]\big| = n_{e+1}$ then $\liminf_s n_{e+1}^s = n_{e+1}$. Either $G(2e) = 2e+2$ or $G(2e) = 0$.

Suppose $G(2e) = 2e+2$, so that $n_{e+1} = n_e + 2$. Then $\liminf_t g(2e, t) = 2e+2$ so that for almost every stage t, $g(2e, t) = 2e+2$. Using the induction hypothesis for e, $n_e^s \geq n_e$ for sufficiently large stages s with equality infinitely often. Choose s sufficiently large such that for all $t \geq s$, $g(2e, t) = 2e+2$ and $n_e^t \geq n_e$. Then by construction $n_{e+1}^t = n_e^t + 2 \geq n_e + 2$, with equality holding at the infinitely many stages t where $n_e^t = n_e$. Therefore, $\liminf_s n_{e+1}^s = n_e + 2 = n_{e+1}$.

Suppose $G(2e) = 0$, so that $n_{e+1} = n_e + 1$. Since $G(2e) = 0$, $\liminf_t g(2e, t) = 0$, so that there are infinitely many stages t where $g(2e, t) = 0$. For this to occur there is some $m \in [n_e, 2e]$ with $\liminf_s \psi_e(m, u_{e,m}^s) < 2e+2$, and so for infinitely many states s, $\psi_e(m, u_{e,m}^s) < 2e+2$. By the inductive hypothesis for e, $n_e^s \geq n_e$ for all sufficiently large s with equality at infinitely many stages s. Using step (4) of the construction update procedure, there must be infinitely stages s at which $n_e^s = n_e$ and $\psi_e(m, u_{e,m}^t) < 2e+2$ where $t \in [s', s]$ and $s' < s$ was the last stage for which $n_e^{s'} = n_e$. Thus, at infinitely many stages s, $n_{e+1}^s = n_e^s + 1 = n_e + 1$, and so $\liminf_s n_{e+1}^s = n_{e+1}$. □

Write $A = \{a_0 < a_1 < a_2 < \ldots\}$.

Sublemma 2. *For each $e \in \omega$, if Ψ_e is total and order-preserving, then $\Psi_e(n_e) \neq a_{n_e}$.*

Proof. By Sublemma 1, there is a stage s such that $n_e \leq n_e^t \leq 2e$ for all $t \geq s$. If Ψ_e is total and order-preserving s can be chosen sufficiently large so that for all $t \geq s$,

$$\Psi_e(n_e) \leq \psi_e(n_e^t, u_{e,n_e^t}^t).$$

We may additionally assume that for any stage $t \geq s$, if $n_e^t = m$ then there is a stage s' with $t > s' \geq s$ and $n_e^{s'} = m$. There are two cases to consider:

(a) If $\Psi_e(n_e) \geq 2e+3$, then $\psi_e(m, u_{e,m}^t) \geq 2e+3$ for all stages $t \geq s$ used to compute $g(2e, t)$ on step (4) of the construction. Thus, $g(2e, t) = 2e+2$ for all $t \geq s$, and $G(2e) = 2e+2 = a_{n_e}$, so that $\Psi_e(n_e) \neq a_{n_e}$.

(b) If $\Psi_e(n_e) < 2e+3$, then $g(2e,t) = 0$ for infinitely many t (namely at those stages t with $n_e^t = n_e$ and $\psi_e(n_e, u_{e,n_e}^{t'}) < 2e+3$ for some $t' \in [t'', t]$ where $t'' < t$ was the last stage with with $n_e^{t''} = n_e$.) Thus, $G(2e) = 0$ and $a_{n_e} = 2e+3$, so that $\Psi_e(n_e) \neq a_{n_e}$.

□

It follows from Sublemma 2, that for any $e \in \omega$, if Ψ_e is total and monotonic, then A is not the range of Ψ_e. □

References

[CDK98] R.J. Coles, R. Downey, and B.M. Khoussainov. On initial segments of computable linear orders. *Order*, 14:107–24, 1998.

[CHKS04] B. Csima, D. Hirschfeldt, J.F. Knight, and R.I. Soare. Bounding prime models. *J. Symbolic Logic*, 69(4):1117–42, 2004.

[Dow98] Rod Downey. Computability theory and linear orders. In Y.L. Ershov, S.S. Goncharov, A. Nerode, and J.B. Remmel, editors, *Handbook of Recursive Mathematics*, volume 138 of *Studies in Logic and the Foundations of Mathematics*, chapter 14. Elsevier, 1998.

[HMP07] D. Hirschfeldt, R. Miller, and S. Podzorov. Order-computable sets. *The Notre Dame Journal of Formal Logic*, 48(3):317–347, 2007.

[Khi98] N. G. Khisamiev. Constructive abelian groups. In Yu. L. Ershov, S.S. Goncharov, A. Nerode, and J.B. Remmel, editors, *Handbook of Recursive Mathematics*, volume 138-9 of *Stud. Logic Found. Math.*, pages 1177–1231. Elsevier Science, 1998.

[KNS97] B.M. Khoussainov, A. Nies, and R.A. Shore. Computable models of theories with few models. *Notre Dame J. Formal Logic*, 38:165–78, 1997.

[Ler81] Manuel Lerman. On recursive linear orders. In M. Lerman, J.H. Schmerl, and R.I. Soare, editors, *Logic Year 1979-80*, volume 859 of *Lecture Notes in Mathematics*, pages 132–42, 1981.

[Soa87] R.I. Soare. *Recursively Enumerable Sets and Degrees*. Perspectives in Mathematics. Springer-Verlag, 1987.