η-Representable Sets and Degrees

Kenneth Harris

Department of Computer Science
University of Chicago

http://people.cs.uchicago.edu/~kaharris
kaharris@uchicago.edu

BEGINNINGS

THE BASICS

WHAT IS KNOWN

Computable Linear Orders

- A *linear order* is a structure $\mathbf{L} = (L, \leqslant_L)$ with domain L and binary relation $\leqslant_L$ satisfying the following properties:
  - $x \leqslant_L x$, for all $x \in L$
  - $x \leqslant_L y$ and $y \leqslant_L x \implies x = y$, for all $x, y \in L$.
  - $x \leqslant_L y$ and $y \leqslant_L z \implies x \leqslant_L z$, for all $x, y, z \in L$.
  - $x \leqslant_L y$ or $y \leqslant_L x$, for all $x, y \in L$.
- A linear order $\mathbf{L} = (\mathbb{N}, \leqslant_L)$ is *computably presented* if $\leqslant_L$ is a computable relation on the natural numbers $\mathbb{N}$.
- There is a Δ_3 listing of all computably presented linear orders

$$\mathbf{L}_1, \mathbf{L}_2, \ldots$$

Basic Relations

Let $\mathbf{L} = (\mathbb{N}, \leqslant_L)$ be a computably presented linear order.

Predicates will be color coded:
Π_1 predicates.
Π_2 predicates.

- $\mathrm{adj}(x, y)$: There is no z such that $x <_L z <_L y$.
- $\mathrm{Block}(x_1, \ldots, x_n)$: For each $i < n$, $\mathrm{adj}(x_i, x_{i+1})$.
- $\text{Right-LimitPoint}(x)$: For every $z >_L x$ there is y such that $x <_L y <_L z$.
- $\text{Left-LimitPoint}(x)$: For every $z <_L x$ there is y such that $z <_L y <_L x$.
- $\mathrm{MaximalBlock}(x_1, \ldots, x_n)$: $\mathrm{Block}(x_1, \ldots, x_n)$ and $\text{Left-LimitPoint}(x_1)$ and $\text{Right-LimitPoint}(x_n)$

Linear Order Types

- A *linear order type* is an equivalence class of linear orders which are order-isomorphic. An order type may have have computable presentations.
- η is the order type of a countable dense linear ordering without endpoints (so, any such ordering is order-isomorphic to the rationals.) η has computable presentation.
- ζ is the order type of a countable discrete ordering without endpoints or limit points (so, any such ordering is order-isomorphic to the integers.) ζ has computable presentations.

η-Representable Sets

- Let $A = \{a_0 < a_1 < a_2 < \dots\}$ (0 and 1 are assumed not to be a member of A.)
- A is *strongly η-representable* if there is a computably presented linear order whose order type is given by

$$\eta + a_0 + \eta + a_1 + \eta + a_2 + \eta + \dots$$

  where a_i is a MaximalBlock of size a_i.
- A is *η-representable* if there is a computable linear order where the maximal blocks need not be in order of size, and may have repetitions.
- A is *uniquely η-representable* there is exactly one maximal block for each element of A, but no restriction to the order.

ζ-Representable Sets

Definition: A (strong, unique) ζ-representation is defined just as in (strong, unique) η-representation, except for replacing copies of η with copies of ζ.

We have the best possible result for ζ-representations

Theorem: (Lerman) A set is (strongly, uniquely) ζ-representable iff it is Σ_3.

Moral: We can effectively hide information from a $0''$-oracle using copies of ζ.

Upper Bounds

- (Feiner) If A has an (uniquely) η-representation, then A is Σ_3.

  $n \in A$ *iff* there are $x_1, \ldots, x_n$ such that $\text{MaximalBlock}(x_1, \ldots, x_n)$.

- (Rosenstein) If A has a strong η-representation, then A is Δ_3.

  $n \notin A$ *iff* there are $x_1, \ldots, x_{n+1}$ such that $\text{Block}(x_1, \ldots, x_{n+1})$ and there are no $y_1, \ldots, y_n$ such that $\text{Block}(y_1, \ldots, y_n)$ and $y_n <_L x_1$.

Lower Bounds

- (Rosenstein) Every Σ_2 set has a strong η-representation.
- (Fellner) Every Π_2 set as a strong η-representation.
- (Lerman) There is a Δ_3 set which is *not* η-representable. (Moral: η-representations cannot be so effectively hidden from a $0''$ oracle.)

Representing Degrees

Definition: A Turing degree can be (strongly, uniquely) η-represented if some set in that degree can be so represented.

- (Lerman) Every Σ_3 degree is η-representable.

Question: (Downey) Is every Δ_3 degree strongly η-representable?

Our Contribution

- There is a Δ_3 degree which is *not* strongly η-representable.
- Every η-representable set is uniquely η-representable.
- The η-representable sets are exactly the ranges of $0'$-computable limitwise monotonic functions.

Interlude

A Δ_3 set which is not η-representable

Interlude: Theorem

Theorem (Lerman): There is a Δ_3 set which has no η-representation.

The Players

- Blue: We build a set A using $0''$ oracle satisfying the following requirements:

$$R_n : \qquad \mathbf{L}_n\ \eta\text{-represents } C_n \implies A \neq C_n$$

- Red: Plays all computably presented linear orders. We can list these orders using $0''$

$$\mathbf{L}_1, \mathbf{L}_2, \ldots$$

Blue wins if each requirement R_n is eventually satisfied. Red wins otherwise.

The Game

Stage s: Assume the following in play:

- Red: Computably presented linear orders

$$\mathbf{L}_{s_1}, \ldots, \mathbf{L}_{s_k}$$

  where $s_1, \ldots, s_k < s$.

- Blue: A_s, a finite approximation to A which will agree with A on $x < s$, and finite sets of Blocks

$$B_{s_1}, \ldots, B_{s_k}$$

  where

  - $|B_{s_i}| = s$ or $|B_{s_i}| = 0$ when no block of size s exists in $\mathbf{L}_{s_i}$
  - B_{s_i} is a block in $\mathbf{L}_{s_i}$

The Play

Stage s:

- Red: Puts L_s into play.
- Blue:
  1. Let B_s be any Block of size $s+1$ in $\mathbf{L}_{s_i}$, or set $B_s = \emptyset$ if no block exists.
  2. Expand each block $B_{s_i} \neq \emptyset$ in L_{s_i} to size $s+1$. If this is not possible mark B_{s_i} a MaximalBlock.
  3. Choose t least such that B_t is empty or marked MaximalBlock.
  4. If $B_t = \emptyset$ then let $A_{s+1} = A_s \cup \{s+1\}$; and otherwise (either there is no t or B_t is marked MaximalBlock) let $A_{s+1} = A_s$.
  5. Remove $\mathbf{L}_t$ from play.
     Blue has ensured R_t satisfied.
  6. For any B_{s_i} still marked MaximalBlock, let B_{s_i} be any Block of size $s+1$ in $\mathbf{L}_{s_i}$, or set $B_s = \emptyset$

Verification

- If Blue removes $\mathbf{L}_n$ from play at any stage s, then Blue satisfies requirement R_n.
- If $\mathbf{L}_n$ does η-represent a set C_n, then at stage $n+1$ $\mathbf{L}_n$ is put into play, and at some stage $s \geqslant n+1$ Blue removes $\mathbf{L}_n$ from play.
- Thus, Blue wins!!
- It is possible that $\mathbf{L}_n$ is never removed from play. This will happen if at some stage s, B_n is part of an infinite block inside $\mathbf{L}_n$. In this case $\mathbf{L}_n$ does not η-represent a set at all, so R_n is trivially satisfied, although $0''$ may not be able to decide this at any stage s of the game.

Densification

How to build η-Representations

Building $\eta + \mathbf{n}$

Goal: We want to build part of a computably presented linear order to be

$$\eta + \mathbf{n}$$

Given: A computable guide f for the desired size of the maximal block. At stage s of our construction, f directs us to build a maximal block of size $f(s)$.

Question: What conditions on f will guarantee that we build a computable copy of $\eta + \mathbf{n}$?

Densification

Construction by Densification: Build linear relation $<$ on $\mathbb{N}$

Stage 0:

$$q < b_{f(0)} < \ldots < b_1$$

where

- q is intended to be η
- b_i is intended to be part of a maximal block, currently of size $f(0)$.

Densification

Construction by Densification: Build linear relation $<$ on $\mathbb{N}$

End of Stage s: Assume we have already constructed

$$q < \ldots < q < b_{f(s)} < \ldots < b_1$$

where

- q are members of the dense block, intended to become part of η.
- b_i are intended to be part of a maximal block, currently of size $f(s)$.

Begin Stage s+1: Build according to $f(s+1)$

Densification

Construction by Densification: Build linear relation $<$ on $\mathbb{N}$

Stage s+1: $f(s) = f(s+1)$

$$p < q < p < \ldots < p < q < p < b_{f(s+1)} < \ldots < b_1$$

where

- q were elements in dense block in stage s.
- p are *new* elements to the dense block added to force density.
- b_i are intended to be part of a maximal block, currently of size $f(s+1)$.

Densification

Construction by Densification: Build linear relation $<$ on $\mathbb{N}$

Stage s+1: $f(s) < f(s+1)$

$$p < q < p < \ldots\ldots < p < q < p <$$
$$c_{f(s+1)} < \ldots < c_{f(s)+1} < b_{f(s)} < \ldots < b_1$$

where

- q were elements in dense block in stage s.
- p are *new* elements to dense block added to force density.
- b_i are intended to be part of a maximal block, currently of size $f(s+1)$.
- c_j were members of the dense block in stage s and crossed-over to fill-out the maximal block in stage $s+1$.

Densification

Construction by Densification: Build linear relation $<$ on $\mathbb{N}$

Stage s+1: $f(s) > f(s+1)$

$$p < q < p < \ldots < q < p < c < p < \ldots < c < p < b_{f(s+1)} < \ldots < b_1$$

where

- q were elements in dense block in stage s.
- p are *new* elements added to the dense block in stage $s+1$ to force density.
- c were members of the maximal block in stage s and crossed-over to the dense block in stage $s+1$.
- b_i are intended to be part of a maximal block, currently of size $f(s+1)$.

Densification Lemma

Definition: Let τ be a linear order type and $f : \omega \to \omega$. Then τ is *densified according to f* if $\tau = \bigcup_s \tau_s$ where τ_s is given at stage s of the densification construction.

Densification Lemma: Let $f : \omega \to \omega$ and τ densified according to f. Then

$$\tau = \eta + \mathbf{n} \text{ iff } \liminf_s f(s) = n.$$

Densification Theorem

Theorem: Let $f : \omega \times \omega \to \omega$ be computable and $F : \omega \to \omega$ total with

$$F(n) = \liminf_s f(n, s) \qquad \text{for all } n$$

Then the range of F is η-representable.

Furthermore,

- If F is injective, then the range of F is uniquely η-representable.
- If F is order-preserving, then the range of F is strongly η-representable.

Interlude

All Σ_3 Degrees are

η-Representable

Interlude: Theorem

Theorem (Lerman): Every Σ_3 degree is η-representable.

That is, for every Σ_3 set A, there is a set B Turing equivalent to A such that B is η-representable.

The Goal

Given: Σ_3 set A and computable $g : \omega \times \omega \to \omega$ satisfying

$$n \in A \implies (\exists! y) W_{g(n,y)} = \omega$$
$$n \notin A \implies (\forall y) W_{g(n,y)} \text{ finite}$$

Goal: Show the following set is η-representable

$$B = A \oplus \omega = \{2x \ : \ x \in A\} \cup \{2x + 1 \ : \ x \in \omega\}$$

Notice, B is Turing equivalent to A.

Strategy: Produce a computable $f : \omega \times \omega \to \omega$ such that $\liminf_s f(m, s)$ exists for every m, and B is the range of $\liminf_s f(\cdot, s)$.

Requirements to Satisfy: $R_{\langle n,y \rangle}$

$$W_{g(n,y)} = \omega \implies \liminf_s f(\langle n, y \rangle, s) = 2n$$
$$W_{g(n,y)} \text{ finite} \implies \liminf_s f(\langle n, y \rangle, s) = 2n + 1$$

Construction

$$f(\langle n, y\rangle, s) = \begin{cases} 2n & \text{if } W^{s+1}_{g(n,y)} \neq W^{s}_{g(n,y)} \\ 2n+1 & \text{otherwise} \end{cases}$$

Verification

- For every s, $f(\langle n, y\rangle, s)$ is either $2n$ or $2n+1$.
- If $W_{g(n,y)} = \omega$, then for infinitely many s, $f(\langle n, y\rangle, s) = 2n$. In this case

$$\liminf_s f(\langle n, y\rangle, s) = 2n$$

- If $W_{g(n,y)}$ is finite, then for almost every s, $f(\langle n, y\rangle, s) = 2n+1$. In this case

$$\liminf_s f(\langle n, y\rangle, s) = 2n+1$$

- For at least one y, $W_{g(n,y)}$ is finite.

Strengthening of Theorem

The Lerman construction produces lots of duplicate copies of maximal blocks of size $2n + 1$. But using a more sensitive construction, we eliminate the redundancy and improve the result:

Theorem (Harris): Every Σ_3 degree is uniquely η-representable.

Characterizing η-Representations

η-Representations are ...

the ranges of $0'$-computable limitwise monotonic functions

Limitwise Monotonic Functions

A function $G : \omega \to \omega$ is *limitwise monotonic* if there exists a computable function $g : \omega \times \omega \to \omega$ such that

(a) $g(n, s) \leqslant g(n, s + 1)$ for all $n, s \in \omega$

(b) $\lim_s g(n, s)$ exists for all $n \in \omega$

(c) $G(n) = \lim_s g(n, s)$

More generally, G is **d**-*computable limitwise monotonic* if g is **d**-computable.

Characterization of η-Representable Sets

Theorem: (Harris) For any set A, TFAE:

1. A is η-representable
2. A is the range of a $0'$-computable limitwise monotonic function.
3. There is a computable f such that $\liminf_s f(n, s)$ exists for all n and A is the range of $\liminf_s f(n, s)$.

(3) $\Longrightarrow$ (1)

For any set A, $(3) \Longrightarrow (1)$:

(3) There is a computable f such that $\liminf_s f(n, s)$ exists for all n and A is the range of $\liminf_s f(n, s)$.

(1) A is η-representable

This is just the Densification Theorem.

(3) $\Longrightarrow$ (2)

For any set A, (3) $\Longrightarrow$ (2):

(3) There is a computable f such that $\liminf_s f(n,s)$ exists for all n and A is the range of $\liminf_s f(n,s)$.

(2) A is the range of a $0'$-computable limitwise monotonic function.

Given computable $f(n,s)$ define $0'$-computable $g(n,s)$ by

$$\begin{aligned} g(n,0) &= 0 \\ g(n,s+1) &= \begin{cases} f(n,s) & \text{if } (\forall t \geqslant s)\ f(n,t) \geqslant f(n,s) \\ g(n,s) & \text{otherwise} \end{cases} \end{aligned}$$

(1) $\Longrightarrow$ (2)

For any set A, $(1) \Longrightarrow (2)$:

(1) A is η-representable

(2) A is the range of a $0'$-computable limitwise monotonic function.

Proof by *Growing Blocks*.

The Game

Given: A computable linear order L which η-represents a set A.

Goal: $0'$-computable $g : \omega \times \omega \to \omega$ satisfying

- For each n, s, $g(n, s) \leqslant g(n, s + 1)$
- For each n, there is an N such that $g(n, s) = N$ for almost all s, and there is a MaximalBlock in L of size N.

Strategy: Start with adj(x_i, x_{i+1}) and $g(n, 0) = 2$, and grow in stages s to a MaximalBlock$(x_1, \ldots, x_N)$, so that eventually $g(n, s) = N$ for all sufficiently large s.

The Play

Stage 0:

- Let adj(x, y), and set $B_0 = \langle x, y \rangle$, so Block$(B_0)$.
- $g(n, 0) = 2$
- $w_0 = 0$

Stage $s + 1$: Given B_s a Block and integer w_s.

- If $B_s \hat{}\langle w_s \rangle$ is a Block then
  - $B_{s+1} = B_s \hat{}\langle w_s \rangle$
  - $g(n, s+1) = g(n, s) + 1$
  - $w_{s+1} = 0$

  Similarly, if $\langle w_s \rangle \hat{} B_s$ is a Block.
- Otherwise,
  - $B_{s+1} = B_s$
  - $g(n, s+1) = g(n, s)$
  - $w_{s+1} = 1 + w_s$

Verification

Using $0'$ oracle there is a listing of all pairs of adjacencies in $\mathbf{L}$:

$$\langle x_0, y_0 \rangle, \langle x_1, y_1 \rangle, \ldots$$

where adj(x_n, y_n) for every n.

- For each n, there is a MaximalBlock M of size N with $\langle x_n, y_n \rangle$ a subblock.
- For each s, $g(n, s) = |B_s|$ and B_s is a subblock of B_{s+1}. Thus, $g(n, s+1) \leqslant g(n, s)$.
- For each s, B_s is a subblock of M, so $g(n, s) \leqslant N$.
- There is a stage s, such that $B_s = M$, and this remains true for all $t \geqslant s$. Thus, for all $t \geqslant s$, $g(n, t) = N$.

(2) $\Longrightarrow$ (3)

For any set A, $(2) \Longrightarrow (3)$:

(2) A is the range of a $0'$-computable limitwise monotonic function.

(3) There is a computable f such that $\liminf_s f(n, s)$ exists for all n and A is the range of $\liminf_s f(n, s)$.

Proof by *Good Approximations*.

The Game

Given: $0'$-computable limitwise monotonic g and by the Limit Lemma a computable h which approximates g:

$$g(n,s) = \lim_t h(n,s,t) \qquad \text{for all } n, s$$

Goal: Construct a computable f such that

$$\lim_s g(n,s) = \liminf_s f(n,s)$$

Strategy: Use a *good approximation* at each stage s: For some k with

$$h(n,0,s) \leqslant h(n,1,s) \leqslant \ldots \leqslant h(n,k,s)$$

Let

$$f(n,s) = h(n,k,s)$$

How to choose k?

Choosing k

Let $\omega^{<\omega}$ be tree of finite sequences prioritized by the tree order:

- $\sigma < \tau$ if either $\sigma \subset \tau$ or for i least with $\sigma(i) \neq \tau(i)$, $\sigma(i) < \tau(i)$.

Choosing k at stage s: Any sequence σ which satisfies the following condition is a *good approximation*:

- $\sigma(i) = h(n, i, s)$ for each $i < |\sigma|$
- $\sigma(i-1) \leqslant \sigma(i)$, for each $0 < i < |\sigma|$

Choose the highest priority good approximation σ not yet chosen at a previous stage.

Define

$$f(n, s) = \sigma(|\sigma| - 1)$$

(if no such σ, then let $f(n, s) = f(n, s-1)$.)

Verification

- For each n, s there is a t such that

$$g(n, s') = h(n, s', t') \qquad (\forall s' \leqslant s) \text{ and } (\forall t' \geqslant t)$$

- For each s, there is a stage t such that the sequence

$$\langle g(n,0), \ldots, g(n,s) \rangle$$

  is a good approximation, and will remain a good approximation at all stages $t \geqslant s$.

- For any s there is a stage t such that any good approximation σ in any stage $u \geqslant t$ must properly extend

$$\langle g(n,0), \ldots, g(n,s) \rangle$$

- For any s, there is a stage t such that for all $u \geqslant t$, $f(n,u) \geqslant g(n,s)$.

- There is an s, $g(n,s) = g(n,t)$ for all $t \geqslant s$. Thus, given $t \geqslant s$ there is a sufficiently large $u > t$

$$g(n,t) = f(n,u)$$

- $\lim_g(n,s) = \liminf_s f(n,s)$ for all n.

Uniquely η-Representable Sets

Theorem: (Harris) For any set A, TFAE:

1. A is uniquely η-representable
2. A is the range of an injective $0'$-computable limitwise monotonic function.
3. There is a computable f such that $\liminf_s f(n,s)$ exists for all n, $\liminf_s f(n,s)$ is injective and A is the range of $\liminf_s f(n,s)$.

The only hang-up in the proof is using the *Growing Blocks* strategy for $(2) \implies (1)$. It is possible that two arguments start with different adjacencies in the same maximal block. This difficulty can be fixed though.

Equivalence of Representations

Theorem (Harris) Any η-representable set has a unique η-representation.

Strategy of Proof

Goal: Given a $0'$-computable limitwise monotonic function G whose *range is infinite*, and such that $G(n) = \lim_s g(n, s)$ for $0'$-computable g, construct a function h using $0'$ so that $H(n) = \lim_s h(n, s)$ is an injective $0'$-computable limitwise monotonic.

Strategy: For each n, choose n' and let for each s

$$h(n, s) = g(n', s)$$

Clashes: If for some s and $n < m$,

$$g(n', s) = h(n, s) = h(m, s) = g(m', s)$$

choose a new m'' with $g(m'', s + 1) > h(m, s)$ and set $h(m, s + 1) = g(m'', s + 1)$.

Strongly η-Representable Sets

It might be hoped that the strongly η-representable sets can be characterized by order-preserving $0'$-computable limitwise monotonic functions. But

Theorem (Harris) There is a strongly η-representable set which *is not* the range of an order-preserving $0'$-computable limitwise monotonic function.

Eventually Monotonic Functions

Downey suggested the following modification: A function G is **d**-computable *limitwise eventually monotonic* if there exists a **d**-computable function $g(n, s)$ such that

(a) There is a t such that for all $s > t$,
$g(n, s) \leqslant g(n, s+1)$ for all $n \in \omega$.

(b) $\lim_s g(n, s)$ exists for all $n \in \omega$

(c) $G(n) = \lim_s g(n, s)$

Question: Do the strongly η-representable sets coincide with the ranges of $0'$-computable limitwise eventually monotonic functions, which enumerate in order?

Finale

A Δ_3 degree which is not η-representable

Finale: Theorem

Theorem (Harris): There is a Δ_3 degree with no strong η-representation.

The Players

- Blue: We build a set A using $0''$ oracle satisfying the following requirements $R_{e,n}$ for each e, n

  $\mathbf{L}_n$ strongly η-represents $C_{e,n}$ and $\Phi_e^A = C_{e,n}$ $\implies C_{e,n}$ computable

- Red: Plays all computably presented linear orders. Blue can list these orders using $0''$

$$\mathbf{L}_1, \mathbf{L}_2, \ldots$$

Blue wins if each requirement $R_{e,n}$ is eventually satisfied, *and* makes A *noncomputable*. Red wins otherwise.

e-Splitting Trees

Definition: Let $\mathcal{T}$ be a computable binary branching tree. We say 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^σ and Φ_e^τ are incompatible:

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

For computable $\mathcal{T}$, σ e-splits in $\mathcal{T}$ is Σ_1.

Definition: A binary branching tree $\mathcal{T}$ is an *e-splitting tree* if each $\sigma \in \mathcal{T}$ e-splits in $\mathcal{T}$.

For computable $\mathcal{T}$, $\mathcal{T}$ has an e-splitting subtree is Π_2. In this case, we can *computably produce* the e-splitting subtree.

Golden Rule

Golden Rule:

- **Higher Priority Requirements**: We promise to restrict any extension to the construction of A to be in the splitting trees of higher priority requirements *and* only extend the construction when given permission by higher priority arguments.
- **Lower Priority Requirements**: We shall enforce all extensions to the construction of A to be within our splitting tree *and* only allow extensions to the construction when we and all higher priority requirements give permission.

The Play for $R_{e,n}$

Stage $t = 0$: Assume the following in play:

- Red: Computable linear order $\mathbf{L}_n$.
- Blue: $\sigma_{t=0} \subset 2^{<\omega}$ and computable tree $\mathcal{T}$ which is a subtree of all higher priority splitting trees.

Does $\mathcal{T}$ have an e-splitting subtree?

- **No**. Let $\rho \subset \mathcal{T}$ satisfy the following: For all paths B through $\mathcal{T}$

$$\Phi_e^B \text{ is computable or non-total}$$

  and ask permission to extend to $\sigma_{t=0}$ to ρ.

- **Yes**.
  - Let $\mathcal{T}_{e,n}$ be a computable e-splitting subtree of $\mathcal{T}$.
  - Let x be least argument of first e-splitting on $\mathcal{T}_{e,n}$ and let $B_{e,n}$ be a Block in $\mathcal{L}_{e,n}$ of size x, or empty if no block exists.

The Play for $R_{e,n}$

Stage $t + 1$: Assume the following in play:

- Red: Computable linear order $\mathbf{L}_n$.
- Blue:
  - $\sigma_t \subset 2^{<\omega}$
  - Computable $\mathcal{T}_{e,n}$ which is an e-splitting subtree of all higher priority trees.
  - $B_{e,n}$, a Block in $\mathbf{L}_n$. If non-empty, then there is a splitting in $\mathcal{T}_{e,n}$ on argument x and $|B_{e,n}| \leqslant x$.

Is there a Block of size x below $B_{e,n}$ in $\mathbf{L}_n$?

- **No**. Let $\rho, \tau \subset \mathcal{T}_{e,n}$ satisfy the following:

$$\Phi^{\rho}_{e,|\rho|}(x) \downarrow = 0 \;\neq\; \Phi^{\tau}_{e,|\tau|}(x) \downarrow$$

  and ask permission to extend to ρ if $B_{e,n}$ is a MaximalBlock of size x and ask permission to extend to τ otherwise.

- **Yes**. Let D be such a block and update $B_{e,n} = D$.

Permission Requests

Stage t: Assume the following in play:

- Red: Computable linear order $\mathbf{L}_n$.
- Blue:
  - $\sigma_t \subset 2^{<\omega}$
  - Computable $\mathcal{T}_{e,n}$ which is an e-splitting subtree of all higher priority trees.
  - $B_{e,n}$, a Block in $\mathbf{L}_n$. If non-empty, then there is a splitting in $\mathcal{T}_{e,n}$ on argument x and $|B_{e,n}| \leqslant x$.

Request to move to $\rho \in \mathcal{T}_{e,n}$:

- Let y be first argument greater than x in $\mathcal{T}_{e,n}$ and above ρ where there is an e-splitting. Can we grow $B_{e,n}$ in $\mathbf{L}_n$?
  - **Yes**. Extend $B_{e,n}$ to Block of size y and pass permission on to higher priority requirements.
  - **No**. Use $0''$ to decide if there exists a MaximalBlock of size x in $\mathbf{L}_n$ below $B_{e,n}$ or $\mathbf{L}_n$ is not a strong not η-representation. (This is now a restricted search.) Use e-splitting on x to ask permission to extend σ_t on behalf of $R_{e,n}$.

Permission Granted

Higher priority requirements have granted permission for $R_{e,n}$ to extend σ_t to ρ:

- Let $\sigma_{t+1} = \rho$
- Retire $R_{e,n}$ (Apparent Blue success.)
- Re-set **all** lower priority requirements to begin again.

Re-set **all** lower priority requirements? Even those retired?

- A requirement S may have retired because every extension to a path B on $\mathcal{T}_{i,m}$ for a requirement of lower priority than $R_{e,n}$ satisfies

$$\Phi_i^B \text{ is computable or partial}$$

But the move to σ_t may no longer be on $\mathcal{T}_{i,m}$!!

Verification of $R_{e,n}$

Assume: At $t = 0$ no higher priority requirement will request an extension of $\sigma_{t=0}$. Let $\mathcal{T}$ be a computable splitting subtree of all higher priority requirements. By **assumption** and the Golden Rule, A will be a path in $\mathcal{T}$.

- If there are no e-splitting subtrees of $\mathcal{T}$, then $R_{e,n}$ can extend to ρ to ensure that all paths $\rho \subset B$ in $\mathcal{T}$ satisfy

  $$\Phi_e^B \text{ is computable or partial}$$

  Blue now really satisfies $R_{e,n}$.
- If $\mathbf{L}_n$ strongly η-represents a set, then there are only finitely many adjacencies in MaximalBlocks containing $B_{e,n}$ or below $B_{e,n}$.
- $R_{e,n}$ only grants permission to extend σ_t when there is an e-splitting in $\mathcal{T}_{e,n}$ on argument x with $x \leqslant |B_{e,n}|$.
- If $\mathbf{L}_n$ strongly η-represents a set, then at some stage t, $0''$ will discover a MaximalBlock and will then decide whether there is a MaximalBlock of size x. By using e-splitting on $\mathcal{T}_{e,n}$ for x, Blue guarantees satisfaction of $R_{e,n}$.

NonComputability

- Let e be the program

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

  So, that given any splitting tree $\mathcal{T}$, there is an e-splitting subtree.

- Suppose C is computable, then there is a linear order $\mathbf{L}_n$ which strongly η-represents C.

- Requirement $R_{e,n}$ will be satisfied by ensuring

$$\Phi_e^A \neq C$$

  since $R_{e,n}$ can always find an e-splitting subtree at any stage of the construction.

Bibliography

General Reference:

Rod Downey, ”Computability Theory and Linear Orderings” in *Handbook of Recursive Mathematics*, Chapter 14, pp. 823-975.

My work:

Kenneth Harris, ”η-Representable Sets and Degrees”, preprint at **people.cs.uchicago.edu/∼kaharris/papers/eta-rep.pdf**