

A ω -REA SET FORMING A MINIMAL PAIR WITH $0'$

PETER M. GERDES

Abstract. It is easy to see that no n -REA set can form a (non-trivial) minimal pair with $0'$ and only slightly more difficult to observe that no ω -REA set can form a (non-trivial) minimal pair with $0''$. Shore has asked whether this can be improved to show that no ω -REA set forms a (non-trivial) minimal pair with $0'$. We show that no such improvement is possible by constructing a set C with $\underline{0} <_{\mathbf{T}} C \leq_{\mathbf{T}} 0''$ forming a minimal pair with $0'$. We then show that no α -REA set can form a (non-trivial) minimal pair with $0''$.

§1. Introduction.

1.1. Notation. The notation we use in this proof is largely standard. We use σ, τ, δ to denote partial functions from ω to $\{0, 1\}$ and write $\sigma \prec \tau$ to denote that the function τ extends σ . We identify sets with their characteristic functions so that $\sigma \prec X$ has the expected meaning for $X \in \mathcal{P}(\omega)$. We denote $x \in \text{dom } \sigma$ ($x \notin \text{dom } \sigma$) by $\sigma(x)\downarrow$ ($\sigma(x)\uparrow$) and say σ is incompatible (compatible) with τ , denoted $\sigma \upharpoonright \tau$ ($\sigma \downharpoonright \tau$), if there is some (no) x with $\sigma(x)\downarrow \neq \tau(x)\downarrow$. We let α, β, γ range over elements in $\omega^{<\omega}$, write α^- as shorthand for $\alpha \upharpoonright_{|\alpha|-1}$ and denote the concatenation of α with β by $\alpha\widehat{\beta}$. We denote the length of a α by $|\alpha|$ and the and extend this notation to partial functions by setting $|\delta| = 1 + \max \text{dom } \delta$. Capital roman letters range over subsets of ω which we identify with their characteristic function.

$\Phi_e(Z; x)$ denotes the e -th $\{0, 1\}$ valued partial computable functional applied to oracle Z on the input x . We adopt the convention that if $\Phi_e(Z; x)$ converges in s steps, written $\Phi_e(Z; x)\downarrow_s$, then $\Phi_e(Z \upharpoonright_s; x) = \Phi_e(Z; x)$. W_e^Z is the e -th set c.e. in Z and $W_{e,s}^Z$ is its stage s approximation. We use $\langle x, y \rangle$ to denote the integer code of the pair (x, y) . Capital roman letters range over $\mathcal{P}(\omega)$ and we write \underline{C} for the Turing degree of C , \overline{C} for the compliment of C , C' for the jump of C and use $\leq_{\mathbf{T}}, \equiv, \wedge_{\mathbf{T}}, \vee_{\mathbf{T}}$ to denote Turing reducibility, equivalence, meet and join respectively.

We follow the standard practice of identifying $X^{[n]}$, the n -th column of X , with $\{y \mid \langle n, y \rangle \in X\}$ and $X^{[\leq n]}$ for $\{ \langle m, y \rangle \in X \mid m \leq n \}$. We extend this notation to partial functions by letting $\sigma^{[\leq l]}$ represent the restriction of σ to $\omega^{[\leq l]}$. We also stipulate that $X^{[\leq 0]} = \emptyset$.

Received by the editors in Fall 2010 for consideration for publication in the [Journal for Symbolic Logic](#).

1991 *Mathematics Subject Classification*. Primary 03D25, 03D30; Secondary (03D60).

Key words and phrases. REA, w-REA, Turing reducible, computability.

Partially Supported by NSF EMSW21-RTG-0739007 and EMSW21-RTG-0838506

1.2. Overview. In [1] Jockusch and Shore introduce the α -REA, for $\alpha < \omega_1^{ck}$, sets as the sets produced by effectively iterating the construction of a relatively c.e. set α many times. Since we will restrict our attention here to $\alpha = \omega$ we will use the equivalent (up to Turing degree) definition.

DEFINITION 1.1. $C \subseteq \omega$ is ω -REA iff there is a computable function f such that

$$C^{[n]} = W_{f(n)}^{C^{[<n]}}$$

Professor Shore has observed that if $C \not\leq_T \emptyset$ is ω -REA then $\exists B \leq_T C$ with $\emptyset <_T B <_T 0''$ but asked (private communication) if this would still hold if $0''$ was replaced with $0'$. In this paper we answer this question in the negative by proving the following theorem.

THEOREM 1.2. *There is an ω -REA set $C \not\leq_T \emptyset$ such that $0' \wedge_T C = \emptyset$*

1.3. Failure at $0''$. Before we embark on this construction it is instructive to see why this claim fails for $0''$.

PROPOSITION 1.3. *If C is ω -REA and $C \not\leq_T \emptyset$ then $C \wedge_T 0'' \neq \emptyset$.*

PROOF. Assume C fails the lemma. We first argue that for every n $C^{[\leq n]}$ must be computable. Since $C^{[n+1]}$ is Δ_2^0 in $C^{[\leq n]}$ if n is greatest with $C^{[\leq n]} \leq_T \emptyset$ we must have $\emptyset <_T C^{[n+1]} \leq_T 0' \leq_T 0''$. Hence if $C \wedge_T 0'' = \emptyset$ then $C^{[\leq n]} \leq_T \emptyset$ for all $n \in \omega$.

So suppose $C^{[\leq n]} \leq_T \emptyset$ for all $n \in \omega$. We now argue that $0''$ can compute C . Note that given a c.e. index for a computable set R $0''$ can recover a c.e. index for \bar{R} and from an index for $C^{[\leq n]}$ and $\bar{C}^{[\leq n]}$ one can uniformly recover a c.e. index from $C^{[\leq n+1]}$. Thus if $C^{[\leq n]}$ is computable for all $n \in \omega$ by induction $0''$ can recover i_n, j_n with $C^{[\leq n]} = W_{i_n} = \bar{W}_{j_n}$. Clearly these indexes allow $0''$ to compute C .

□

The lesson to be drawn from this proof is that any C satisfying theorem 1.2 must be the join of a countable collection of computable sets. Thus the non-computability of C must result from the non-uniformity of this join. The difficulty in building C is therefore how to encode enough about $0'$ in $C^{[<n]}$ so $C^{[n]}$ can successfully diagonalize against the Δ_2^0 sets while making sure $C^{[<n]}$ only encodes a finite amount of non-computable information.

§2. Machinery.

2.1. Building ω -REA Sets. Evidently if we are to build $C \notin \Delta_2^0$ we will have to somehow have to uniformly specify an c.e. procedure to build $C^{[n]}$ from $C^{[<n]}$ while dealing with the fact that our approximation to $C^{[<n]}$ will never settle on the correct value. Rather than trying to explicitly give such a procedure upfront we will instead enumerate rules called axioms committing us to enumerate certain elements into $C^{[n]}$ when certain conditions are met by $C^{[<n]}$.

DEFINITION 2.1. A **axiom** is a triple $\langle l : \sigma \rightarrow y \rangle$ where $l \in \omega$, σ is a function from a finite subset of $\omega^{[<l]}$ to $\{0, 1\}$ and $y \in \omega^{[\geq l]}$.

In our construction we will think of the axiom $\langle l : \sigma \rightarrow y \rangle$ as the commitment to place y in C if $\sigma \prec C$. The parameter l serves only to ensure that attempts to enumerate elements in the n -th column of C are only allowed to consult the first $n - 1$ columns of C thus avoiding any circularity. The utility of this definition is made clear by the following lemma.

LEMMA 2.2. *If \mathcal{A} is an c.e. set of axioms then the set C defined by*

$$y \in C \iff (\exists l \in \omega)(\exists \sigma \prec C) \left[\langle l : \sigma \rightarrow y \rangle \in \mathcal{A} \right]$$

is ω -REA

PROOF. Note that

$$\langle n, x \rangle \in C \iff (\exists l \leq n)(\exists \sigma \prec C^{[<l]}) \left[\langle l : \sigma \rightarrow y \rangle \in \mathcal{A} \right]$$

Thus $C^{[n]}$ only depends on $C^{[<n]}$ so C is well defined. Furthermore the above equation explicitly defines $C^{[n]}$ from $C^{[<n]}$ and n via a (uniformly) Σ_1^0 formula. Thus by an application of the s-m-n theorem [2] there is a computable function f satisfying definition 1.1. \dashv

Our construction will proceed by building a c.e. set \mathcal{A} of axioms which will yield an ω -REA set via the preceding lemma. To make proper use of this machinery we introduce two more definitions. We first try and capture the notion that some axiom $\langle l : \sigma' \rightarrow y \rangle$ only has an effect if $\sigma \prec C$.

DEFINITION 2.3. The axiom $\langle l : \sigma' \rightarrow y \rangle$ **depends** on σ if $\sigma \prec \sigma'$. We say the axiom $\langle l : \sigma \rightarrow y \rangle$ is enumerated dependent on δ to mean we enumerate $\langle l : \sigma \cup \delta^{[<l]} \rightarrow y \rangle$ into \mathcal{A} .

We will also speak of an axiom depending on $C(n) = 0$ to mean it depends on the σ defined by $\sigma(n) = 0$. During our construction we will frequently want to satisfy some requirement on the assumption that a guess about how C behaves on some finite number of columns and a finite initial segment is true. We therefore introduce a notion of how the axioms would affect C if such a guess were correct.

DEFINITION 2.4. Given any set $C_\alpha \subseteq \omega^{[<l_\alpha]}$ and a partial function δ_α satisfying $\delta_\alpha^{[<l]} \prec C_\alpha^{[<l_\alpha]}$ (understood as a guess at an initial segment of C) we say that a set of axioms \mathcal{A} yields C over C_α, δ_α if

$$\begin{aligned} \langle n, x \rangle \in C \iff & \left(n < l_\alpha \wedge \langle n, x \rangle \in C_\alpha \right) \vee \\ & \left(\langle n, x \rangle \in \delta_\alpha \right) \vee \\ & \left(\langle n, x \rangle \notin \text{dom } \delta_\alpha \wedge (\exists l \leq n)(\exists \sigma \prec C) \left[\langle l : \sigma \rightarrow \langle n, x \rangle \rangle \in \mathcal{A} \right] \right) \end{aligned}$$

In other words \mathcal{A} yields X over C_α, δ_α if we take C_α, δ_α to be the first l_α columns of C and $\delta_\alpha \prec C$ regardless of what the axioms say and then build the rest of C using the construction from lemma 2.2.

§3. Requirements & Modules. We fix a computable array $V_{e,s}$ of finite sets via the limit lemma [5] such that every Δ_2^0 set is of the form $V_e = \lim_{s \rightarrow \infty} V_{e,s}$ and build C to meet the following requirements.

$$\begin{aligned} \mathcal{R}_{e,i} &:: \Phi_i(C) \neq V_e \text{ or } V_e \leq_{\mathbf{T}} \mathcal{Q} \text{ whenever } V_e \text{ defined.} \\ \mathcal{N}_e &:: W_e \neq \overline{C} \end{aligned}$$

We reserve columns $3\langle e, i \rangle, 3\langle e, i \rangle + 1$ for $\mathcal{R}_{e,i}$ and the column $3e + 2$ for meeting \mathcal{N}_e and grant each requirement the right to modify a finite initial segment of later columns but not earlier columns. Each column of C will be either finite or cofinite thereby making $C^{[<n]}$ computable as our observation required.

As C can't be computable in $0'$ during the construction later requirements won't know, even in the limit, how the earlier requirements are satisfied. To deal with this we perform our construction along a tree assigning to each $\alpha \in \omega^{<\omega}$ in the tree a module P_α tasked with handling a particular requirement on the assumption that α correctly encodes how the higher priority requirements are met. In particular we assign requirements to modules as follows.

$$(3.1) \quad P_\alpha \text{ handles } \begin{cases} \mathcal{R}_{e,i} & \text{if } |\alpha| = 2\langle e, i \rangle \\ \mathcal{N}_e & \text{if } |\alpha| = 2e + 1 \end{cases}$$

Given $\alpha \in \omega^{<\omega}$ we define l_α to be the first column reserved for the requirement handled by P_α . Note that l_α only depends on $|\alpha|$ so all modules tasked with meeting a given requirement share columns. We will associate to each α set $C_\alpha \subset \omega^{[<l_\alpha]}$ intended as a guess at $C^{[<l_\alpha]}$ and a partial function $\delta_\alpha \in 2^{<\omega}$ representing a guess at the finite part of C used by prior requirements. These two guesses will always be compatible, i.e. $\delta_\alpha^{[<l_\alpha]} \prec C^{[<l_\alpha]}$. Implicitly δ_α will function as a restraint as well since P_α won't attempt to change $C(x)$ if $x \in \text{dom } \delta_\alpha$. We regard C_α, δ_α as a description of the ultimate effect of P_α on C .

3.1. Action Along The Tree. As explained above the module P_α will act to meet it's requirement using the information encoded in α about how earlier requirements were met. At each stage we will have some guess at how the various requirements are met and that guess will control which modules are then executed at that stage, i.e., only those modules that appear to have correct guesses execute. Those familiar with Π_2^0 tree constructions may be assured that the tree executes the modules in the standard fashion and skip ahead to the next section while those desiring more details can read on.

More formally we will define a function f , the true path, such that if $\alpha^- \subset f$ then $f(|\alpha|)$ indicates how P_α satisfies it's associated requirement. In an abuse of notation we will write $f(\alpha) = n$ to indicate that if $f \supseteq \alpha$ then $f(|\alpha|) = n$. At any stage s we will have some approximation $f_s \in \omega^{<\omega}$ to the true path with $f = \liminf_s f_s$. We will execute a single module P_α satisfying $f_s \supseteq \alpha$ at every stage s and leave it to P_α to set the value of $f_s(\alpha)$ at such stages. We ensure that if $f_s \supseteq \alpha$ occurs infinitely often then P_α is executed infinitely often as well by starting out at the root node and executing in increasing order the modules

at each node $\alpha \subseteq f_s$ with $|\alpha| \leq l$ before starting over at the root and working out to nodes of length $l + 1$.

§4. The Construction. A full description of the construction will consist of giving the behavior of each module P_α the approximation to it's outcome $f_s(\alpha)$ and the properties C_β, δ_β for each $\beta = \alpha \hat{\setminus} f_s(\alpha)$. We will always define C_β, δ_β at the first stage $f_s \supseteq \beta$ guaranteeing they are always defined when needed. Note that when describing the various modules we will say the P_α stage s to refer to the s -th time the module P_α is executed. We will also adopt the shorthand α^+ for $\alpha \hat{\setminus} f(\alpha)$ whenever $\alpha \subset f$.

4.1. Basic Approach. Before describing the full construction it's useful to informally sketch how each requirement is to be met. The action of the module P_α implementing the strategy \mathcal{N}_e can be thought of as implementing a straightforward finite injury argument as follows. P_α will wait for a chance to enumerate some element from W_e into $C^{[l_\alpha]}$ doing nothing until such an element is found. If no such element is found then both W_e and C fail to cover some element in the column l_α . On the other hand if such an element is found \mathcal{N}_e will enumerate that element into $C^{[l_\alpha]}$ and reset all weaker priority requirements. This reset is accomplished simply by permanently changing $f_s(\alpha)$ from the 0 it had been up till now to 1 thereby abandoning all previously visited modules $P_\beta, \beta \supseteq \alpha$.

The interesting case occurs when P_α implements $\mathcal{R}_{e,i}$. Here our strategy will be to lay dormant (unactivated) as long as the action of weaker requirements never leads us to change our mind about (our approximation to) $\Phi_i(C)$, i.e., yields only compatible computations. If we remain in this situation we will argue that $\Phi_i(C)$ is computable. If we do see a change in $\Phi_i(C; x)$ for some x we will activate $\mathcal{R}_{e,i}$ and work to alternate between the two computations to diagonalize against $V_e(x)$. Later we will show that if we ever change our mind about $\Phi_i(C; x)$ then P_α has the means to roll back the intervening axioms and recover the previous value of $\Phi_i(C; x)$ by enumerating some controlling element into $C^{[l_\alpha+1]}$. P_α can now act to ensure that $\Phi_i(C; x)$ always disagrees with $V_{e,s}(x)$ by taking said element in and out of $C^{[l_\alpha+1]}$. To ensure that P_α can later change take it back out each time P_α enumerates the controlling element into $C^{[l_\alpha+1]}$ it does so dependent on some large number being absent from $C^{[l_\alpha]}$. By latter adding this number to $C^{[l_\alpha]}$, P_α can effectively cancel it's previous commitment and keep $\Phi_i(C; x) \neq V_{e,s}(x)$.

Provided $V_{e,s}(x)$ eventually settles down this provides no problem. Each time $V_{e,s}(x)$ flip-flops we simply set $f_s(\alpha)$ to the next unused value which has the effect of resetting all the subsequence requirements. However, we must accommodate the possibility this limit fails to exist and somehow prevent those $P_\beta, \beta \supset \alpha$ that assume the limit exists from interfering with those that assume it doesn't. The key point here is to ensure that a particular flag element will be in $C^{[l_\alpha+1]}$ iff $\lim_{s \rightarrow \infty} V_{e,s}(x)$ exists. This allows the modules $P_\beta, \beta \supset \alpha$ guessing the limit doesn't exist to predicate all their actions on the absence of this element and vice versa ensuring noninterference. The effect of this is to ensure that if $\alpha \subset f$ once α appears on f_s then no other requirements modify the region of C used by P_α .

4.2. Global Constraints. To ensure the P_α modules interact appropriately we need to impose two minor additional constraints on the construction.

- (I) If P_α enumerates axiom π then π is enumerated dependent on δ_α .
- (II) If P_α wants to enumerate axiom π and $P_\beta, \beta \subsetneq \alpha$ is an unactivated $\mathcal{R}_{e,i}$ module then π is enumerated dependent on the partial function sending $\langle l_\alpha + 1, m \rangle$ to 0 with m larger than anything mentioned so far in the construction .

This first constraint will ensure that if the guess δ_α at an initial segment of C is wrong then the axioms enumerated by P_α have no effect on the construction. In particular it will guarantee that if P_β implements $\mathcal{R}_{e,i}$ the modules $\alpha \supset \beta$ assuming that $\mathcal{R}_{e,i}$ has only finitely many eventful stages and those assuming it has infinitely many such stages don't interfere.

The second constraint will allow P_β implementing $\mathcal{R}_{e,i}$ to 'roll back' axioms enumerated by $\alpha \supset \beta$ while P_β to recover an earlier computation of $\Phi_i(C)$. This will ensure that even if we first see one value for $\Phi_i(C; x)$ and then $\Phi_i(C; x)$ appears to diverge for many stages before converging to an alternate value we will still be able to return to the first value and use it to diagonalize against $\Phi_i(C) = V_e$.

We now give the detailed actions of the various modules with the understanding that they be modified in the obvious way to comply with these two constraints.

4.3. The Basic $\mathcal{R}_{e,i}$ Strategy. Suppose P_α is assigned to handle $\mathcal{R}_{e,i}$. We wait until we observe a P_α stage s (i.e. s -th time P_α is executed), integers x, y , strings $Y_0, Y_1 \in 2^{<\omega}$ such that C would extend Y_1 if nothing is done but Y_0 if $y \notin Y_0$ is added to $C^{[l_\alpha+1]}$ and Y_0, Y_1 yield incompatible computations. More formally

$$\begin{aligned}
 (4.1) \quad & Y_0^{[\leq l_\alpha]} = Y_1^{[\leq l_\alpha]} \wedge \delta_\alpha \prec Y_0, Y_1 \\
 & \Phi_i(Y_0; x) \downarrow_y \neq \Phi_i(Y_1; x) \downarrow \\
 & |Y_0| < y \\
 & y \in \omega^{[l_\alpha+1]} \\
 & A_s \text{ yields an extension of } Y_1 \text{ over } C_\alpha, \delta_\alpha \\
 & A_s \cup \{\langle l_\alpha + 1 : \emptyset \rightarrow y \rangle\} \text{ yields an extension of } Y_0 \text{ over } C_\alpha, \delta_\alpha
 \end{aligned}$$

When this occurs we say that P_α is **activated** at stage s . We will later show that these conditions are equivalent to the informal notion of changing our mind about the value of $\Phi_i(C; x)$.

If P_α is activated at P_α stage s select $q \in \omega^{[l_\alpha+1]}$ larger than any number mentioned so far to serve as our 'flag' by satisfying $q \in C$ iff P_α only acts finitely many times. Also pick k_s larger than any number mentioned so far with the intent of enumerating $\langle l_\alpha, k_t \rangle$ into C to cancel any decision to put y into C at P_α stage s . Now for any $t \geq s + 1$ let $j_t \in \{0, 1\}$ such that $\Phi_i(Y_{j_t}; x) \neq V_{e,s}(x)$. Say that a P_α stage $t > s$ is **eventful** for P_α if $j_t \neq j_{t-1}$. At P_α stage $t \geq s$ P_α acts as follows.

If t is not an eventful P_α stage let σ_t be the partial function defined by $\sigma_t(\langle l_\alpha, k_t \rangle) = 0$, that is σ_t asserts that $\langle l_\alpha, k_t \rangle \notin C$. Enumerate into \mathcal{A} (if not already present) $\langle l_\alpha + 1 : \sigma_t \rightarrow q \rangle$, that is place q into C if $\langle l_\alpha, k_t \rangle \notin C$ thereby indicating that P_α completes after finite action. If $V_{e,t}(x) \neq \Phi_i(Y_1; x)$ do nothing so that without further action we would have $C \succ Y_1$. If $V_{e,t}(x) = \Phi_i(Y_1; x)$ then enumerate $\langle l_\alpha + 1 : \sigma_t \rightarrow y \rangle$ into \mathcal{A} as well. This has the effect of setting $C \supseteq Y_0$ if $\langle l_\alpha, k_t \rangle$ remains out of C . Finally set $k_{t+1} = k_t$.

If t is an eventful P_α stage then enumerate the axiom $\langle l_\alpha : \emptyset \rightarrow \langle l_\alpha, k_t \rangle \rangle$ into \mathcal{A} and set $k_{t+1} = k_t + 1$. This has the effect of canceling the effect of any axiom enumerated by P_α at any earlier stage by placing $\langle l_\alpha, k_t \rangle$ into C . Note that if every element of the form $\langle l_\alpha, k_t \rangle$ is eventually placed in C then no axiom will place q into C .

Now assume that at (global) stage t P_α executes it's s -th stage then define

$$f_t(|\alpha|) = \begin{cases} 0 & \text{if } P_\alpha \text{ hasn't yet been activated} \\ 1 & \text{if } s \text{ is an eventful stage for } P_\alpha \\ n + 2 & \text{if } s \text{ is uneventful and there have been } n \text{ prior eventful stages.} \end{cases}$$

If t is the first stage for which $f_t \geq \alpha \hat{\ } \langle w \rangle$ and t corresponds to P_α stage s then define.

$$\delta_{\alpha \hat{\ } \langle w \rangle} = \begin{cases} \delta_\alpha & \text{if } w = 0 \\ X \upharpoonright_r^{[\geq l_\alpha]} \cup \delta_\alpha \text{ where } r = \langle l_\alpha, k_t \rangle & \text{otherwise} \end{cases}$$

$$C_{\alpha \hat{\ } \langle w \rangle} = \begin{cases} C_\alpha \cup \delta_{\alpha \hat{\ } \langle w \rangle} & \text{if } w \neq 1 \\ C_\alpha \cup \delta_{\alpha \hat{\ } \langle w \rangle} \{z \mid z \geq k_0\} \cup X^{[l_\alpha + 1]} & \text{if } w = 1 \end{cases}$$

Where

$$X = \text{the result of } \mathcal{A}_s \text{ over } C_\alpha, \delta_\alpha$$

Note that $\delta_{\alpha \hat{\ } \langle w \rangle}$ is longer enough to restrain later modules from interfering with Y_0 and Y_1 . Also observe that for $w > 1$ $\delta_{\alpha \hat{\ } \langle w \rangle}$ is defined at an uneventful P_α stage s so $\delta_{\alpha \hat{\ } \langle w \rangle}$ reflects the assumption that whatever axioms are enumerated dependent on $\langle l_\alpha, k_t \rangle$ remain uncancelled. On the other hand for $w = 1$ $\delta_{\alpha \hat{\ } \langle w \rangle}$ is defined at an eventful P_α stage t so behaves as if none of the axioms dependent on $\langle l_\alpha, k_t \rangle$ apply. The definition of $C_{\alpha \hat{\ } \langle w \rangle}$ guesses any unreferenced elements in column $l_\alpha + 1$ are absent and that those in column l_α are present if $w = 1$ and absent otherwise.

4.4. The Basic N_e Strategy. We ensure that C is not computable by ensuring that $\overline{C^{[l_\alpha]}} \neq W_e^{[l_\alpha]}$. We only place finitely many elements into $C^{[l_\alpha]}$ so if $W_e^{[l_\alpha]}$ is also finite the requirement is trivially satisfied. We ensure that if $W_e^{[l_\alpha]}$ is infinite then $W_e^{[l_\alpha]} \cap C^{[l_\alpha]} \neq \emptyset$ by enumerating $\langle l_\alpha : \emptyset \rightarrow y \rangle$ into \mathcal{A} at the first P_α stage s and least $y \geq s$ with $y \in \omega^{[l_\alpha]}$ and $y \notin \delta_\alpha$ for which we observe $y \in W_{e,s}$. We say P_α acts at such a stage and once P_α has acted we never let it do so again.

$$\begin{aligned}
f_t(|\alpha|) &= \begin{cases} 0 & \text{if } N_\alpha \text{ hasn't acted} \\ 1 & \text{if } N_\alpha \text{ has acted} \end{cases} \\
C_{\alpha \hat{\langle} w \rangle} &= C_\alpha \cup X^{[l_\alpha]} \\
\delta_{\alpha \hat{\langle} w \rangle} &= X \upharpoonright_m^{[\geq l_\alpha]} \cup \delta_\alpha
\end{aligned}$$

Where:

X is the result of \mathcal{A}_s over C_α, δ_α
 s is the first stage with $f_s \supseteq \alpha \hat{\langle} w \rangle$
 m is larger than any number mentioned so far.

§5. Verification. We now verify that the construction above produces the desired set C . By lemma 2.2 we have evidently built an ω -REA set so all that remains is to show that $C \not\leq_{\mathbf{T}} \emptyset$ and $C \geq_{\mathbf{T}} X \in \Delta_2^0 \implies X \leq_{\mathbf{T}} \emptyset$.

LEMMA 5.1. $f = \liminf_{s \rightarrow \infty} f_s$ is well defined. Furthermore if $\alpha \subseteq f$ then P_α is executed infinitely often.

PROOF. Suppose n is the least such that the lemma fails for $f \upharpoonright_n = \alpha^+$. Evidently P_α can't implement N_e as $f_s(\alpha)$ would either remain 0 or switch permanently to 1. So assume that P_α implements $\mathcal{R}_{e,i}$. But in this case $f(n)$ could only be undefined if for any m there was some stage t such that if $s > t$ $f_s(n) > m$. However, this would entail there were infinitely many eventful stages. Hence $\liminf_{s \rightarrow \infty} f_s(n) = 1$. The second half of the statement follows directly from the construction. \dashv

LEMMA 5.2. Suppose P_α implements $\mathcal{R}_{e,i}$ then for all n $\delta_{\alpha \hat{\langle} 1 \rangle} \upharpoonright \delta_{\alpha \hat{\langle} n+2 \rangle}$ whenever both are defined.

PROOF. Let $q \in \omega^{[l_\alpha+1]}$ be the flag location selected during the execution of P_α . By the remarks at the end of subsection 4.3 $\delta_{\alpha \hat{\langle} 1 \rangle}(q) = 0$ while $\delta_{\alpha \hat{\langle} n+2 \rangle}(q) = 1$. \dashv

LEMMA 5.3. If $\alpha \subseteq f, f_s$ then for all $\beta \upharpoonright \alpha$ if P_β enumerates the axiom $\langle l : \sigma \rightarrow y \rangle$ after stage s then $\sigma \upharpoonright \delta_\alpha$.

PROOF. Suppose the claim holds for α^- . If P_{α^-} implements N_e then for all $t > s$ $f_t \supseteq \alpha^- \implies f_t \supseteq \alpha$ so the claim holds for α . Similarly if P_{α^-} implements $\mathcal{R}_{e,i}$ and $\alpha = \alpha^- \hat{\langle} w \rangle$ for $w \neq 1$ then the claim holds for α . So suppose $\alpha = \alpha^- \hat{\langle} 1 \rangle$. By construction if $t \geq s$ $f_t \not\supseteq \alpha^- \hat{\langle} 0 \rangle$. Hence only β satisfying $\beta \supseteq \alpha^- \hat{\langle} n+2 \rangle$ for some n are of concern. But by the preceding lemma $\delta_{\alpha^- \hat{\langle} 1 \rangle} \upharpoonright \delta_{\alpha^- \hat{\langle} n+2 \rangle}$. But if $\beta \supseteq \alpha^- \hat{\langle} n+2 \rangle$ then constraint I ensures that if P_β enumerates $\langle l : \sigma \rightarrow y \rangle$ then $\alpha^- \hat{\langle} n+2 \rangle \prec \sigma$ so the lemma also holds for α . \dashv

LEMMA 5.4. Suppose $\alpha \subset f, f_s$ and \mathcal{A}_s yields X_s over C_β, δ_β with $\beta \subset \alpha$. Then $X_s \succ \delta_\alpha$ and $X_s^{[< l_\alpha]} \subset C^{[< l_\alpha]}$. Furthermore for every r there are infinitely many s such that $X_s \succ C \upharpoonright_r$.

PROOF. The first claim follows by straightforward induction on γ with $\beta \subseteq \gamma \subseteq \alpha$. Since $X_s^{[<l_\gamma]} \subset C^{[<l_\gamma]}$ every axiom already enumerated by P_γ applies in a straightforward manner as they only reference elements outside of $\text{dom } \delta_\gamma$ via constraint II, i.e., the axioms have effect if outside of δ_γ we haven't added elements not in C . By lemma 5.3 we don't have to worry about nodes incompatible with α and it is straightforward to check from the construction that when $f_s \supseteq \gamma^+$ the axioms enumerated so far by P_γ cause $X_s \succ \delta_{\gamma^+}$.

To prove the second part of the lemma simply pick $\alpha \subset f$ so large that $\omega \upharpoonright_r \subset \omega^{[<l_\alpha]}$. Now merely choose s such that $f_s \supset \alpha$ large enough that the axioms responsible for placing every element into $C \upharpoonright_r$ have already been enumerated. \dashv

LEMMA 5.5. C is not computable.

PROOF. If C were computable then $\overline{C} = W_e$ for some e . Now pick $\alpha \subseteq f$ such that P_α implements N_e . Now if $W_e^{[l_\alpha]}$ is infinite then there is some stage s such that P_α acts to make $W_e \cap C \neq \emptyset$. On the other hand if $W_e^{[l_\alpha]}$ is finite then as $C^{[l_\alpha]}$ is also finite $\overline{C} \neq W_e$. \dashv

LEMMA 5.6. Suppose that $\alpha \subseteq f$, P_α implements $\mathcal{R}_{e,i}$ and there are stages $s_0 < s_1$ at which P_α not yet activated with $\alpha \subseteq f_{s_0}, f_{s_1}$ such that \mathcal{A}_{s_j} yields C_j over C_α, δ_α for $j = 0, 1$ and $\Phi_i(C_0) \downarrow_{s_0} \mid \Phi_i(C_1) \downarrow_{s_1}$ then there are Y_0, Y_1, x, y such that Y_0, Y_1, x, y satisfy the conditions 4.1 at stage s_1

PROOF. If $Y_j = C_j \upharpoonright_{s_j}$ by our convention on use we may know that $\Phi_i(C_j) \downarrow_{s_j} = \Phi_i(Y_j) \downarrow_{s_j}$. By lemma 5.3 any axiom enumerated by $\beta \mid \alpha$ after s_0 will have no effect on C and by construction the effects of all $\beta \subsetneq \alpha$ are accounted for in C_α, δ_α . As no $\beta \supseteq \alpha$ is allowed to affect column l_α or $l_\alpha + 1$ and P_α has yet to enumerate any axioms we know that $Y_0^{[l_\alpha]} = Y_1^{[l_\alpha]} = \delta_\alpha^{[l_\alpha]}$.

Now select $y = \langle l_\alpha + 1, m \rangle$ where m is the first large number used by constraint II for P_α after stage s_0 . By constraint II every axiom π enumerated by $\beta \supseteq \alpha$ after stage s_0 is enumerated dependent on sending y to 0, i.e., predicated on $y \notin C$. Thus, as $y > s_0$ we have $\mathcal{A}_s \cup \{\langle l_\alpha + 1 : \emptyset \rightarrow y \rangle\}$ yields some $X \succ Y_0$ over C_α, δ_α . The other conditions follow trivially. \dashv

LEMMA 5.7. If $V \in \Delta_2^0$ and $V \leq_{\mathbf{T}} C$ then V is computable.

PROOF. Pick e such that $V = \lim_{s \rightarrow \infty} V_{e,s}$, i such that $\Phi_i(C) = V$ and $\alpha \subset f$ such that P_α implements $\mathcal{R}_{e,i}$. By construction if P_α is ever activated then $\Phi_i(C) \neq V$. So suppose P_α is never activated. We compute $V(x)$ as follows. Wait for a stage s such that $f_s \supseteq \alpha$ such that \mathcal{A}_s yields Y_s over C_α, δ_α and $\Phi_i(Y_s; x) \downarrow_s$. Use this value for $V(x)$.

Such a stage must exist as by lemma 5.4 we can find s such that Y_s is equal to C on the use of $\Phi_i(C; x)$. As the s just mentioned yields the correct value so too must our computation or there are stages s_0, s_1 as in lemma 5.6 so P_α is activated. Contradiction. \dashv

This completes the proof of theorem 1.2.

§6. Generalizations. At this point one might naturally wonder if this result could be improved by moving to ordinals past ω . One might conjecture there

is some $\omega \cdot \omega$ -REA degree C that forms a nontrivial minimal pair with $0''$. Disappointingly this conjecture turns out to be false. We sketch the proof below following the same approach as in lemma 1.3 but now considering limit stages. The notation we use for computable ordinals is from [4] and the definition of α -REA degrees can be found in [1]. Note that for the remainder of the paper we let α, β, λ and γ range over \mathcal{O} , i.e., notations for constructive ordinals

LEMMA 6.1. *Suppose $C_\lambda = \bigoplus_{\beta <_{\mathcal{O}} \lambda} C_\beta$ if λ a limit, $C_\gamma +_{\mathcal{O}} 1 = W_{f(\gamma)}^{C_\gamma} \oplus C_\gamma$ and $C_0 = \emptyset$. If $C_\alpha \leq_{\mathbf{T}} \mathcal{Q}$ and $f(\beta)$ is defined for all $\beta <_{\mathcal{O}} \alpha$ then $0''$ can (uniformly in α) compute an index for C_α as a c.e. set.*

PROOF. We prove this using definition via effective transfinite recursion. We will define a computable function $I(e)$ such that $\Phi_{I(e)}(0''; \gamma) = i_\gamma$ with $W_{i_\gamma} = C_\gamma$ if for all $\beta <_{\mathcal{O}} \gamma$ $\Phi_e(0''; \beta) = i_\beta$ and $W_{i_\beta} = C_\beta$. Then by application of the recursion theorem [3] we recover a fixed point e such that $\Phi_{I(e)}(0'') \simeq \Phi_e(0'')$ is our desired $0''$ computable function.

Before we construct $I(e)$ we observe that there is a total computable function h such that for all β if $C_\beta = W_i$ and $\overline{C_\beta} = W_{\hat{i}}$ then $C_{\beta +_{\mathcal{O}} 1} = W_{h(\beta, i, \hat{i})}$. The existence of h follows immediately from the computability of f and definition of $C_{\beta +_{\mathcal{O}} 1}$. Additionally there is a computable function g such that if $W_i = C_\gamma$ and $\beta <_{\mathcal{O}} \gamma$ then $g(\gamma, \beta, i) = i'$ with $C_\beta = W_{i'}$. As g merely unwraps some number of effective join operations it is straightforward to verify it exists.

If $\gamma = 0$ then $\Phi_{I(e)}(0''; \gamma)$ returns a c.e. index for the empty set. If $\gamma = \beta +_{\mathcal{O}} 1$ then $\Phi_{I(e)}(0''; \gamma)$ first runs $\Phi_e(0''; \beta)$ to extract i_β and then computes an index \hat{i}_β for the compliment of W_{i_β} . The computation then returns $h(\gamma, i_\beta, \hat{i}_\beta)$ as the index for C_γ . Finally if γ is a limit then $\Phi_{I(e)}(0''; \gamma)$ searches through all pairs of indexes i, \hat{i} for complimentary c.e. sets and returns the first i such that:

$$\begin{aligned} (\forall \beta <_{\mathcal{O}} \gamma) [\Phi_e(0''; \beta) = j_0 \wedge g(\gamma, \beta, i) = j_1 \implies W_{j_0} = W_{j_1}] \\ (\forall \beta) [(\exists x) (\langle \beta, x \rangle \in W_i) \implies \beta <_{\mathcal{O}} \gamma] \end{aligned}$$

Now let e be the fixed point of $I(e)$. It is straightforward to trace out the definitions to verify that $\Phi_e(0'')$ behaves correctly at 0 and at every successor and limit stage so by transfinite induction $\Phi_e(0''; \alpha)$ satisfies the lemma. \dashv

PROPOSITION 6.2. *Suppose that C is of non-computable α -REA degree then $C \wedge_{\mathbf{T}} 0'' \neq \mathcal{Q}$.*

PROOF. By the definition of α -REA sets $C = C_\alpha$ where C_α is defined as in lemma 6.1 relative to some computable function f . Thus there is some least $\beta \leq_{\mathcal{O}} \alpha$ such that C_β isn't computable. If β is a successor then just as in proposition 1.3 $C_\beta \leq_{\mathbf{T}} 0'$. So assume β is a limit. By lemma 6.1 we can uniformly find a c.e. code for each C_γ with $\gamma <_{\mathcal{O}} \beta$. To determine if $\langle \gamma, x \rangle \in C_\beta$ we first ask $0''$ if $\gamma <_{\mathcal{O}} \beta$. If not $\langle \gamma, x \rangle \notin C_\beta$. Otherwise ask $0''$ for a c.e. code i for C_γ and report $\langle \gamma, x \rangle \in C_\beta$ iff $0''$ determines $x \in C_\gamma$. Hence in either case $\mathcal{Q} <_{\mathbf{T}} C_\beta \leq_{\mathbf{T}} 0''$ and as $C_\beta \leq_{\mathbf{T}} C$ we have $C \wedge_{\mathbf{T}} 0'' \neq \mathcal{Q}$. \dashv

REFERENCES

- [1] Carl G. Jockusch and Richard A Shore, *Pseudajump operators II: Transfinite iterations, hierarchies and minimal covers*, J. Symbolic Logic **49** (1984), 1205–1236. [MR771789](#)
- [2] S. C. Kleene, *General recursive functions of natural numbers*, Math. Ann. **112** (1936), no. 1, 727–742. [MR1513071](#)
- [3] ———, *On notation for ordinal numbers*, J. Symbolic Logic **3** (1938), no. 4, 150–155.
- [4] Gerald E. Sacks, *Higher recursion theory*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin, 1990. [MR1080970](#)
- [5] J. R. Shoenfield, *On degrees of unsolvability*, Ann. of Math. (2) **69** (1959), 644–653. [MR0105355](#)

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF NOTRE DAME DU LAC
NOTRE DAME, INDIANA 46556

URL: <http://invariant.org/>
E-mail: gerdes@invariant.org