A ω-REA Set Forming A Minimal Pair With $0'$

Peter M. Gerdes

http://invariant.org/math.html

2018, Madison Talk

Outline

Notation

- $A^{[n]} = \{y | \langle n, y \rangle \in A\}, \uparrow$
- $A^{[<n]} = \{\langle m, y \rangle | m < n\}, A^{[=n]} = \{\langle m, y \rangle | m = n\}$, etc..
- $\sigma, \tau, \nu, \delta$ range over $\{0, 1, \uparrow\}^{<\omega}$ (partial binary valued functions with finite domain).
- We write $\sigma \prec \tau$ if τ extends σ and $\sigma \prec X$ if σ is extended by the characteristic function of X.
- Let $V_i(x) \stackrel{\text{def}}{=} \lim_{s \to \infty} \Phi_i(x, s)$ be an enumeration of the Δ^0_2 sets.

C.E. operations are nice!

- We define X' to be c.e. in X.
- Many properties of the jump are really properties of c.e. operations.
- But not all c.e. operations are very jump like.
- Sometimes $W_e^X \not\geq_{\mathbf{T}} X$

1-REA Operations

Definition (Jockusch and Shore)

A 1-REA operator is of the form

$$J_e(A) = A \oplus W_e^A$$

- Jockusch and Shore proved jump inversion holds for all 1-REA operators.
- Naturally we want to iterate to get an analog of the α jump operation.
- To act like an α-jump the sequence of 1-REA operators should be effective.

α-REA

Definition (Jockusch and Shore)

Given a computable function f define:

$$\mathcal{J}_f^0(A) = A$$
$$\mathcal{J}_f^{\beta+1}(A) = J_{f(\beta)}(A) = J_f^{\beta}(A) \oplus W_e^A$$
$$\mathcal{J}_f^{\lambda}(A) = \oplus_{\gamma <_{\mathcal{O}} \lambda} \mathcal{J}_f^{\gamma}(A) \text{ For } \lambda \text{ a limit}$$

The set C is α-REA if $C = \mathcal{J}_f^{\alpha}(\emptyset)$.

Theorem (Jockusch and Shore)

Every Δ_2^0 operator is an α-REA operator for some α. (Sets too!)

Converse

Theorem (Jockusch and Shore)

If $C = \mathcal{J}_f^{\gamma}(\emptyset)$ *is* α*-REA either* $C \leq_{\mathbf{T}} \mathbb{0}$ *or for some* $\kappa <_{\mathcal{O}} \gamma$ *with* $\mathbb{0} <_{\mathbf{T}} \mathcal{J}_f^{\kappa}(\emptyset) \leq_{\mathbf{T}} \mathbb{0}''$ *with* $\mathcal{J}_f^{\kappa}(\emptyset) \leq_{\mathbf{T}} C$.

Proof.

Least κ with $\mathbb{0} <_{\mathbf{T}} \mathcal{J}_f^{\kappa}(\emptyset)$ satisfies.

- If $\kappa = \beta + 1$ then $\mathcal{J}_f^{\beta+1}(\emptyset) = J_e(\mathcal{J}_f^{\alpha}(\emptyset)) \leq_{\mathbf{T}} \mathbb{0}'$.
- Can effectively go from computable description of $\mathcal{J}_f^{\alpha}(\emptyset)$ to computable description of $\mathcal{J}_f^{\beta}(\emptyset)$ with $\beta <_{\mathcal{O}} \alpha$.
- If κ a limit then use $\mathbb{0}''$ to decide if $\langle \beta, x \rangle \in \mathcal{J}_f^{\kappa}(\emptyset)$ by searching for computable index satisfying defining equation of $\mathcal{J}_f^{\alpha}(\emptyset)$ for each $\alpha <_{\mathcal{O}} \beta$.

Limit Levels

Question (Shore)

If $C \not\leq_{\mathbf{T}} \mathbb{0}$ is α-REA must C compute some non-computable Δ^0_2 set?

Theorem

There is an ω-REA set $C >_{\mathbf{T}} \mathbb{0}$ forming a minimal pair with $\mathbb{0}'$.

Corollary

There is an ω-REA operator $\mathcal{J}^\omega$ so that for every A, $\mathcal{J}^\omega(A) >_{\mathbf{T}} A$ but if $X \leq_{\mathbf{T}} A'$ and $X \leq_{\mathbf{T}} \mathcal{J}^\omega(A)$ then $X \leq_{\mathbf{T}} A$.

Representing C

Observation

If $C^{[i]} = W_{f(i)}^{C^{[<i]}}$ for a computable function f then C is of ω-REA degree.

- Clearly $C^{[i]}$ must always be computable for us to succeed.
- We ensure that $C^{[i]} =^* \omega$ or $C^{[i]} =^* \emptyset$.
- Need way to build C in a constructive fashion.
- Produce f by enumerating elements into $C^{[i]}$ contingent on $C^{[<i]}$ having a given finite initial segment.

α-REA operators are c.e. sets

Definition

A (ω-REA) axiom is a pair $\langle \sigma \to z \rangle$ with $z = \langle l, y \rangle$ and $\operatorname{dom} \sigma \subset \omega^{[<l]}$

- The axiom $\langle \sigma \to \langle l, y \rangle \rangle$ represents a commitment to place y into $C^{[l]}$ if $\sigma \prec C^{[<l]}$.

Proposition

If $\mathcal{A}$ is a c.e. set of axioms then there is a unique ω-REA set $\mathcal{C}$ defined by

$$\langle l, y \rangle \in \mathcal{C} \iff [\exists\, \sigma](\langle \sigma \to \langle l, y \rangle \rangle \in \mathcal{A} \wedge \sigma \prec C)$$

- f witnesses $\mathcal{C}(\mathcal{A})^{[l]}$ is uniformly $\Sigma^0_1(C^{[<l]})$.

Enumerating Axioms

- We build C by giving c.e. construction for $\mathcal{A}$.
- We write C_s for the set that $\mathcal{A}_s$ would produce if no further axioms were enumerated.

Notation

- To enumerate $\langle \sigma \rightarrow \langle I, y \rangle \rangle$ into $\mathcal{A}$ dependent on δ we request that $\langle \delta \cup \sigma \rightarrow \langle I, y \rangle \rangle$ be placed into $\mathcal{A}$ (requires $\delta \nmid \sigma$)

Observation

If $\delta \not\prec C$ then nothing enumerated dependent on δ will place elements into C

- This will be key to avoiding conflict between requirements.

Overview

- We work to meet the requirements $\mathcal{N}_e$ and $\mathcal{R}_{i,e}$.
  - $\mathcal{N}_e$ works to ensure $\overline{C} \neq W_e$
  - $\mathcal{R}_{i,e}$ works to ensure that if V_i total either $C \neq V_i$ or $V_i \leq_{\mathbf{T}} \mathbb{0}$
- $\mathcal{P}_j = \begin{cases} \mathcal{N}_e & \text{if } j = 2e \\ \mathcal{R}_{i,e} & \text{if } j = 2\langle i, e\rangle + 1 \end{cases}$
- Each requirement owns 2 columns $\boldsymbol{\ell}_j, \boldsymbol{\ell}_j + 1$ in C.
- We first describe strategy for each requirement as if it was highest priority requirement.
- Later we implement $\mathcal{P}_j$ with nodes α of length j on tree for standard Π^0_2 approximation argument.

Basic $\mathcal{N}_e$ module

Basic Strategy for $\mathcal{P}_j$ implementing $\mathcal{N}_e$

- Try for Σ^0_1 win by looking for $z \in W_e$ in our column $\boldsymbol{\ell}_j$.
  - If found enumerate $\langle \emptyset \to z \rangle$ into $\mathcal{A}$.
  - Has effect of putting $z \in C \cap W_e$.
- Keep column empty while searching to prevent C from eating up $\overline{W_e}$ for Π^0_1 win.

- Note we always leave $\omega^{[\boldsymbol{\ell}_j]} =^* \omega^{[\boldsymbol{\ell}_j+1]} =^* \emptyset$.
- In real construction will have to respect finite restraint.
- At stage s we guess outcome is Π^0_1 unless Σ^0_1 win has been enumerated.

Basic $\mathcal{R}_{i,e}$ module: Overview

Basic Strategy for $\mathcal{P}_j$ implementing $\mathcal{R}_{i,e}$

- Search for stages s, s' with $\Phi_{e,s}(C_s; x)\downarrow \neq \Phi_{e,s'}(C'_s; x)\downarrow$ taking $\Phi_e(C) \leq_{\mathbf{T}} \mathbb{0}$ for Π^0_1 win if not.
- Once found switch between computations maintaining disagreement with $V_{i,s}(x)$ to try for Σ^0_2 win.
- Accept Π^0_2 win causing $V_i(x)\uparrow$ if we keep switching.

- Guess Π^0_1 outcome until disagreement found.
- Later guess at Π^0_2 victory whenever $V_{i,s}(x)$ changes otherwise guess it never changes again for Σ^0_2 win.
- Note that we conceptualize each potential Σ^0_2 win as a different outcome.

Basic $\mathcal{R}_{i,e}$ module: Rollback

Question

How do we ensure we can return to computation $\Phi_{e,s}(C_s; x)\downarrow$ if we later observe $\Phi_{e,s'}(C'_s; x)\downarrow$ to disagree?

- Somehow we must preserve the ability to rollback any changes made to C on the use of $\Phi_{e,s}(C_s; x)$.

Idea

- At stage s pick $q_s \in \omega^{[l_\alpha+1]}$ larger than any use observed so far.
- At stage s' restrain any weaker computations from further affecting use of either computation.
- Thus we choose computation $\Phi_{e,s}(C_s; x)$ by putting q_s in C and $\Phi_{e,s'}(C'_s; x)\downarrow$ by keeping it out.

Rollback Illustration

0	0	0	0	0
⋮	⋮	⋮	⋮	⋮
0	0	0	0	0
⋮	⋮	⋮	⋮	⋮
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

- Preserve first two columns (ours) during search.
- At s we observe $\Phi_{e,s}(C_s; x)\downarrow$ with some use.
- Later change dependent on $C(q_s) = 0$.
- At $s' > s$ we observe $\Phi_{e,s'}(C_{s'}; x')\downarrow$ with some use.
- Later change dependent on both $C(q_s) = 0$ and $C(q_{s'}) = 0$.
- Enumerate $C(q_{s'}) = 1$ recover $\Phi_{e,s'}(C_{s'}; x')\downarrow$.
- Enumerate $C(q_s) = 1$ recover $\Phi_{e,s}(C_s; x)\downarrow$.

Rollback Illustration

0	0	0	0	0
⋮	⋮	⋮	⋮	⋮
0	0	0	0	0
⋮	⋮	⋮	⋮	⋮
0	0	0	0	0
0	0	0	0	0
0	0	0	0	1
0	0	0	0	1

- Preserve first two columns (ours) during search.
- At s we observe $\Phi_{e,s}(C_s; x)\downarrow$ with some use.
- Later change dependent on $C(q_s) = 0$.
- At $s' > s$ we observe $\Phi_{e,s'}(C_{s'}; x')\downarrow$ with some use.
- Later change dependent on both $C(q_s) = 0$ and $C(q_{s'}) = 0$.
- Enumerate $C(q_{s'}) = 1$ recover $\Phi_{e,s'}(C_{s'}; x')\downarrow$.
- Enumerate $C(q_s) = 1$ recover $\Phi_{e,s}(C_s; x)\downarrow$.

Rollback Illustration

0	0	0	0	0
⋮	⋮	⋮	⋮	⋮
0	0	0	0	0
⋮	⋮	⋮	⋮	⋮
0	0	0	0	0
0	0	0	0	0
0	0	0	1	1
0	0	0	0	1

- Preserve first two columns (ours) during search.
- At s we observe $\Phi_{e,s}(C_s; x)\downarrow$ with some use.
- Later change dependent on $C(q_s) = 0$.
- At $s' > s$ we observe $\Phi_{e,s'}(C_{s'}; x')\downarrow$ with some use.
- Later change dependent on both $C(q_s) = 0$ and $C(q_{s'}) = 0$.
- Enumerate $C(q_{s'}) = 1$ recover $\Phi_{e,s'}(C_{s'}; x')\downarrow$.
- Enumerate $C(q_s) = 1$ recover $\Phi_{e,s}(C_s; x)\downarrow$.

Rollback Illustration

0	0	0	0	0
⋮	⋮	⋮	⋮	⋮
0	0	0	1	0
⋮	⋮	⋮	⋮	⋮
0	0	0	0	1
0	0	0	0	0
0	0	0	1	1
0	0	0	0	1

- Preserve first two columns (ours) during search.
- At s we observe $\Phi_{e,s}(C_s; x)\downarrow$ with some use.
- Later change dependent on $C(q_s) = 0$.
- At $s' > s$ we observe $\Phi_{e,s'}(C_{s'}; x')\downarrow$ with some use.
- Later change dependent on both $C(q_s) = 0$ and $C(q_{s'}) = 0$.
- Enumerate $C(q_{s'}) = 1$ recover $\Phi_{e,s'}(C_{s'}; x')\downarrow$.
- Enumerate $C(q_s) = 1$ recover $\Phi_{e,s}(C_s; x)\downarrow$.

Rollback Illustration

0	0	0	0	0
⋮	⋮	⋮	⋮	⋮
0	0	0	1	0
⋮	⋮	⋮	⋮	⋮
0	0	0	0	1
0	0	1	1	0
0	0	1	1	1
0	0	0	0	0

- Preserve first two columns (ours) during search.
- At s we observe $\Phi_{e,s}(C_s; x)\downarrow$ with some use.
- Later change dependent on $C(q_s) = 0$.
- At $s' > s$ we observe $\Phi_{e,s'}(C_{s'}; x')\downarrow$ with some use.
- Later change dependent on both $C(q_s) = 0$ and $C(q_{s'}) = 0$.
- Enumerate $C(q_{s'}) = 1$ recover $\Phi_{e,s'}(C_{s'}; x')\downarrow$.
- Enumerate $C(q_s) = 1$ recover $\Phi_{e,s}(C_s; x)\downarrow$.

Rollback Illustration

0	1	0	0	0
⋮	⋮	⋮	⋮	⋮
0	0	0	1	0
⋮	⋮	⋮	⋮	⋮
0	0	0	0	1
0	0	0	0	0
0	0	0	1	1
0	0	0	0	1

- Preserve first two columns (ours) during search.
- At s we observe $\Phi_{e,s}(C_s; x)\downarrow$ with some use.
- Later change dependent on $C(q_s) = 0$.
- At $s' > s$ we observe $\Phi_{e,s'}(C_{s'}; x')\downarrow$ with some use.
- Later change dependent on both $C(q_s) = 0$ and $C(q_{s'}) = 0$.
- Enumerate $C(q_{s'}) = 1$ recover $\Phi_{e,s'}(C_{s'}; x')\downarrow$.
- Enumerate $C(q_s) = 1$ recover $\Phi_{e,s}(C_s; x)\downarrow$.

Rollback Illustration

$$
\begin{array}{ccccc}
0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 1 & & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1
\end{array}
$$

- Preserve first two columns (ours) during search.
- At s we observe $\Phi_{e,s}(C_s; x)\downarrow$ with some use.
- Later change dependent on $C(q_s) = 0$.
- At $s' > s$ we observe $\Phi_{e,s'}(C_{s'}; x')\downarrow$ with some use.
- Later change dependent on both $C(q_s) = 0$ and $C(q_{s'}) = 0$.
- Enumerate $C(q_{s'}) = 1$ recover $\Phi_{e,s'}(C_{s'}; x')\downarrow$.
- Enumerate $C(q_s) = 1$ recover $\Phi_{e,s}(C_s; x)\downarrow$.

Basic $\mathcal{R}_{i,e}$ module: Changing Our Minds

Question

How do we preserve option to switch back to other computation?

- Must ensure there is always some axiom letting us change $C_t(q_s)$.

Idea

- Can put q_s into C_{t+1} by enumerating $\langle\langle k, 0\rangle \to q_s\rangle$ for least available $k \notin C_t$ in column ℓ_j.
- As long as we don't act k won't enter C causing $q_s \in C$ if k enumerated into C we cancel enumeration of q_s.
- If we eventually stop $C^{[l_i]} =^* \emptyset$ otherwise $C^{[l_i]} =^* \omega$.

Basic $\mathcal{R}_{i,e}$ module: Changing Our Minds

Question

How do we preserve option to switch back to other computation?

- Must ensure there is always some axiom letting us change $C_t(q_s)$.

Idea

- Can put q_s into C_{t+1} by enumerating $\langle\langle k, 0\rangle \to q_s\rangle$ for least available $k \notin C_t$ in column $\boldsymbol{\ell}_j$.
- As long as we don't act k won't enter C causing $q_s \in C$ if k enumerated into C we cancel enumeration of q_s.
- If we eventually stop $C^{[l_i]} =^* \emptyset$ otherwise $C^{[l_i]} =^* \omega$.

Alternating Value Illustration

0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0

AXIOMS

- Let red location be q_s.
- $\langle\langle k_1, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_1\rangle$
- $\langle\langle k_2, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_2\rangle$
- $\langle\langle k_3, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_3\rangle$
- ...

Alternating Value Illustration

0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	1

AXIOMS

- Let red location be q_s.
- $\langle\langle k_1, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_1\rangle$
- $\langle\langle k_2, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_2\rangle$
- $\langle\langle k_3, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_3\rangle$
- ...

Alternating Value Illustration

0	0
0	0
0	0
0	0
0	0
0	0
1	0
0	0

AXIOMS

- Let red location be q_s.
- $\langle\langle k_1, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_1\rangle$
- $\langle\langle k_2, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_2\rangle$
- $\langle\langle k_3, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_3\rangle$
- ...

Alternating Value Illustration

0	0
0	0
0	0
0	0
0	0
0	0
1	0
0	1

AXIOMS

- Let red location be q_s.
- $\langle\langle k_1, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_1\rangle$
- $\langle\langle k_2, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_2\rangle$
- $\langle\langle k_3, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_3\rangle$
- ...

Alternating Value Illustration

0	0
0	0
0	0
0	0
0	0
1	0
1	0
0	0

AXIOMS

- Let red location be q_s.
- $\langle\langle k_1, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_1\rangle$
- $\langle\langle k_2, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_2\rangle$
- $\langle\langle k_3, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_3\rangle$
- ...

Alternating Value Illustration

0	0
0	0
0	0
0	0
0	0
1	0
1	0
0	1

AXIOMS

- Let red location be q_s.
- $\langle\langle k_1, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_1\rangle$
- $\langle\langle k_2, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_2\rangle$
- $\langle\langle k_3, 0\rangle \to q_s\rangle$
- $\langle\emptyset \to k_3\rangle$
- ...

Alternating Value Illustration

0	0
0	0
0	0
0	0
1	0
1	0
1	0
0	0

AXIOMS

- Let red location be q_s.
- $\langle\langle k_1, 0\rangle \rightarrow q_s\rangle$
- $\langle\emptyset \rightarrow k_1\rangle$
- $\langle\langle k_2, 0\rangle \rightarrow q_s\rangle$
- $\langle\emptyset \rightarrow k_2\rangle$
- $\langle\langle k_3, 0\rangle \rightarrow q_s\rangle$
- $\langle\emptyset \rightarrow k_3\rangle$
- ...

Alternating Value Illustration

1	0
1	0
1	0
1	0
1	0
1	0
1	0
0	0

AXIOMS

- Let red location be q_s.
- $\langle\langle k_1, 0\rangle \rightarrow q_s\rangle$
- $\langle\emptyset \rightarrow k_1\rangle$
- $\langle\langle k_2, 0\rangle \rightarrow q_s\rangle$
- $\langle\emptyset \rightarrow k_2\rangle$
- $\langle\langle k_3, 0\rangle \rightarrow q_s\rangle$
- $\langle\emptyset \rightarrow k_3\rangle$
- ...

Construction Overview

- Construction organized as Π^0_2 tree argument.
- Node $\alpha \subseteq f$ knows how all prior columns look when first visited.
- So $\alpha \subseteq f$ so can compute effect of axioms in $\mathcal{A}_s$ enumerated by $\beta \supseteq \alpha$ (replaces C_s).
- Only the Π^0_2 outcome of $\mathcal{R}_{i,e}$ can experience more than finite injury.
- Finite injury easy....work above affected finite initial segment.
- Still need to make sure descendants of Π^0_2 outcome of $\mathcal{R}_{i,e}$ and the Σ^0_2 don't interfere.

Managing Interaction

Idea

Ensure that if $\mathcal{R}_{i,e}$ realizes Π^0_2 outcome no node guessing it has Σ^0_2 outcome affects C.

- This will let us ignore everything but finite injury.
- We do this by mandating that all axioms enumerated by β guessing Σ^0_2 outcome get revoked if we change our mind again.
- Specifically β is enumerated dependent on current values of $C(q_s)$ and $C(k_n)$.
- Thus if β guessed incorrectly all axioms enumerated by β are dependent on $\sigma \mid C$ and ineffective.
- This ensures that if $\alpha \subseteq f, f_s$ then C is just what α has guessed plus effect of $\beta \supseteq \alpha$.

Final Remarks

- Construction is effective so really we've built $\mathcal{J}^\omega$ such that for all X

$$X^{(2)} \geq_{\mathbf{T}} \mathcal{J}^\omega(X) >_{\mathbf{T}} X$$
$$\mathcal{J}^\omega(X) \wedge_{\mathbf{T}} X' \equiv_{\mathbf{T}} X$$

- May have relation to question of whether there is a α-REA set of minimal arithmetic degree.

RecThy Package

- This presentation was prepared using the rec-thy.sty package.
- Download a copy from `https://ctan.org/pkg/rec-thy`
- Help development at `https://github.com/TruePath/Recursion-Theory-Latex-Package/blob/master/rec-thy.sty`