Genericity And Nonbounding

Mariya I. Soskova

University of Leeds
Department of Pure Mathematics

03.07.2006

Background

- Cooper, Lee, Sorbi and Yang
  1. Every Δ^0_2 e-degree bounds a minimal pair.
  2. There exists a Σ^0_2 e-degree that does not bound a minimal pair.
- Copestake
  1. Every 2-generic e-degree bounds a minimal pair

Main Result

Theorem

There exists a 1-generic enumeration degree $\mathbf{a}$*, that does not bound a minimal pair in the semi-lattice of the enumeration degrees.*

Definitions

Definition

A set A is 1-generic if for every c.e. set S the following condition holds:

$$\exists \tau \subseteq \chi_A(\tau \in S \vee \forall \mu \supseteq \tau(\mu \notin S)).$$

Definition

Let a and b be two enumeration degrees. We say that a and b form a minimal pair in the semi-lattice of the enumeration degrees if:

1. $a > 0$ and $b > 0$.
2. For every enumeration degree c $(c \leq a \wedge c \leq b \to c = 0)$.

The Requirements

- $G^W : \exists\tau \subseteq \chi_A(\tau \in W \vee \forall\mu \supseteq \tau(\mu \notin W))$
- $R^{\Theta_0\Theta_1} : \Theta_0(A) = X - c.e. \vee \Theta_1(A) = Y - c.e. \vee$
  $\vee \exists \Phi_0, \Phi_1((\Phi_0(X) = \Phi_1(Y) = D) \wedge \forall W - c.e. (W \neq D))$

  $$\Downarrow$$

  $S^W : (X - c.e \vee Y - c.e. \vee (\Phi_0(X) = \Phi_1(Y) = D \wedge$
  $\wedge \exists z(W(z) \neq D(z))))$

The Requirements

- $G^W : \exists \tau \subseteq \chi_A(\tau \in W \vee \forall \mu \supseteq \tau(\mu \notin W))$
- $R^{\Theta_0 \Theta_1} : \Theta_0(A) = X - c.e. \vee \Theta_1(A) = Y - c.e. \vee$
  $\vee \exists \Phi_0, \Phi_1((\Phi_0(X) = \Phi_1(Y) = D) \wedge \forall W - c.e.(W \neq D))$

  $$\Downarrow$$

  $S^W : (X - c.e \vee Y - c.e. \vee (\Phi_0(X) = \Phi_1(Y) = D \wedge$
  $\wedge \exists z(W(z) \neq D(z))))$

The Requirements

- $G^W : \exists \tau \subseteq \chi_A(\tau \in W \vee \forall \mu \supseteq \tau(\mu \notin W))$
- $R^{\Theta_0\Theta_1} : \Theta_0(A) = X - c.e. \vee \Theta_1(A) = Y - c.e. \vee$
  $\vee \exists \Phi_0, \Phi_1((\Phi_0(X) = \Phi_1(Y) = D) \wedge \forall W - c.e.(W \neq D))$

$\Downarrow$

$S^W : (X - c.e \vee Y - c.e. \vee (\Phi_0(X) = \Phi_1(Y) = D \wedge$
$\wedge \exists z(W(z) \neq D(z))))$

The Requirements

- $G^W: \exists \tau \subseteq \chi_A(\tau \in W \vee \forall \mu \supseteq \tau(\mu \notin W))$
- $R^{\Theta_0 \Theta_1}: \Theta_0(A) = X - c.e. \vee \Theta_1(A) = Y - c.e. \vee$
  $\vee \exists \Phi_0, \Phi_1((\Phi_0(X) = \Phi_1(Y) = D) \wedge \forall W - c.e.(W \neq D))$

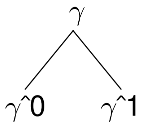
  $$\Downarrow$$

  $S^W: (X - c.e \vee Y - c.e. \vee (\Phi_0(X) = \Phi_1(Y) = D \wedge$
  $\wedge \exists z(W(z) \neq D(z))))$

The Construction

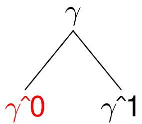
- Priority tree T with nodes labelled by the strategies
- At stage s : δ_s, A_s
- $A_s^0 = \mathbb{N}$
- Approximations to c.e. sets.

The G^W - Strategy



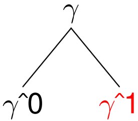
- Choose a finite string λ_γ
- Ask $\exists \mu \supset \lambda_\gamma(\mu \in W)$?
- Yes : outcome 0
- No : outcome 1

The G^W - Strategy



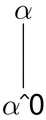
- Choose a finite string λ_γ
- Ask $\exists \mu \supset \lambda_\gamma (\mu \in W)$?
- Yes : outcome 0
- No : outcome 1

The G^W - Strategy



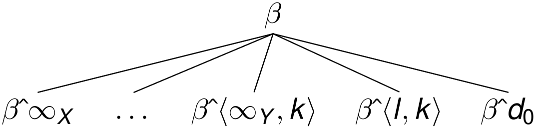
- Choose a finite string λ_γ
- Ask $\exists\mu \supset \lambda_\gamma(\mu \in W)$?
- Yes : outcome 0
- No : outcome 1

The *R* Strategy



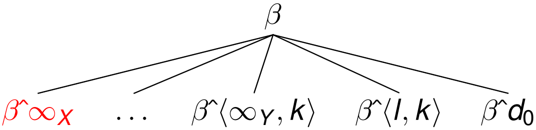
- Check all S^W substrategies.
- At this level operators Φ_0 and Φ_1 are built
- Outcome 0

The S^W Strategy – $o- = \infty_X$



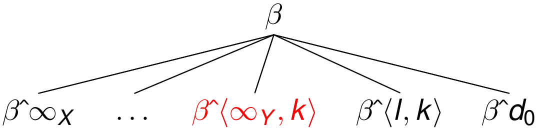
- Build a set U, aiming to prove that X is c.e.
- Scan U for errors.
  - No errors found - outcome ∞_X
  - An error found at point k - build agitator set E_k for k, such that $k \in X \Leftrightarrow E_k \subset A$, Outcome $\langle\infty_Y, k\rangle$

The S^W Strategy – $o- = \infty_X$



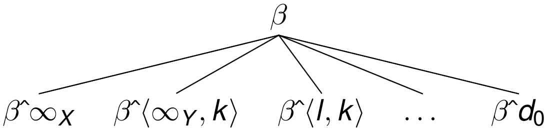
- Build a set U, aiming to prove that X is c.e.
- Scan U for errors.
  - No errors found - outcome ∞_X
  - An error found at point k - build agitator set E_k for k, such that $k \in X \Leftrightarrow E_k \subset A$,
    Outcome $\langle\infty_Y, k\rangle$

The S^W Strategy – $o- = \infty_X$



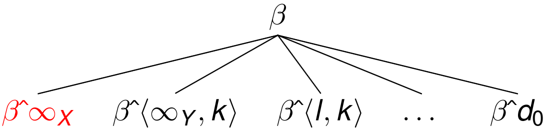
- Build a set U, aiming to prove that X is c.e.
- Scan U for errors.
  - No errors found - outcome ∞_X
  - An error found at point k - build agitator set E_k for k, such that $k \in X \Leftrightarrow E_k \subset A$,
    Outcome $\langle\infty_Y, k\rangle$

The S^W Strategy – $o- = \langle \infty_Y, k \rangle$



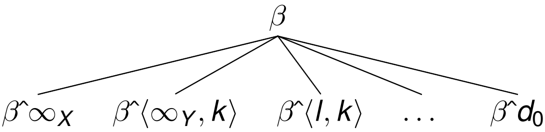
- Check if the agitator for k is still valid. If not the outcome is ∞_X
- Build a set V_k, aiming to prove that Y is c.e.
- Scan V_k for errors.
  - No errors - outcome $\langle \infty_Y, k \rangle$
  - An error found at element l, then build an agitator for l, enumerate a witness z in D: $\langle z, \{k\} \rangle \searrow \Phi_0$ and $\langle z, \{l\} \rangle \searrow \Phi_1$, outcome d_0

The S^W Strategy – $o- = \langle \infty_Y, k \rangle$



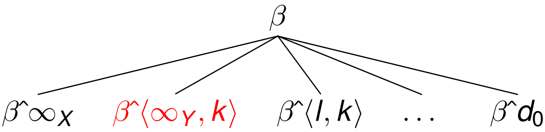
- Check if the agitator for k is still valid. If not the outcome is ∞_X
- Build a set V_k, aiming to prove that Y is c.e.
- Scan V_k for errors.
  - No errors - outcome $\langle \infty_Y, k \rangle$
  - An error found at element l, then build an agitator for l, enumerate a witness z in D: $\langle z, \{k\} \rangle \searrow \Phi_0$ and $\langle z, \{l\} \rangle \searrow \Phi_1$, outcome d_0

The S^W Strategy – $o- = \langle \infty_Y, k \rangle$



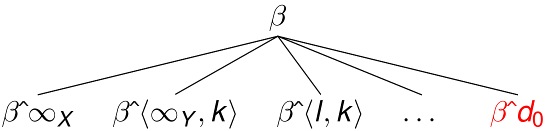
- Check if the agitator for k is still valid. If not the outcome is ∞_X
- Build a set V_k, aiming to prove that Y is c.e.
- Scan V_k for errors.
  - No errors - outcome $\langle \infty_Y, k \rangle$
  - An error found at element l, then build an agitator for l, enumerate a witness z in D: $\langle z, \{k\} \rangle \searrow \Phi_0$ and $\langle z, \{l\} \rangle \searrow \Phi_1$, outcome d_0

The S^W Strategy – $o- = \langle \infty_Y, k \rangle$



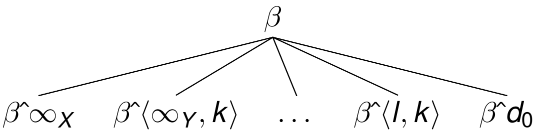
- Check if the agitator for k is still valid. If not the outcome is ∞_X
- Build a set V_k, aiming to prove that Y is c.e.
- Scan V_k for errors.
  - No errors - outcome $\langle \infty_Y, k \rangle$
  - An error found at element l, then build an agitator for l, enumerate a witness z in D: $\langle z, \{k\} \rangle \searrow \Phi_0$ and $\langle z, \{l\} \rangle \searrow \Phi_1$, outcome d_0

The S^W Strategy – $o- = \langle \infty_Y, k \rangle$



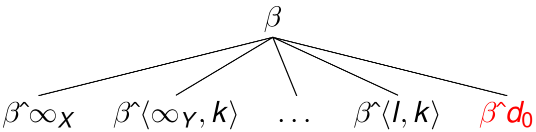
- Check if the agitator for k is still valid. If not the outcome is ∞_X
- Build a set V_k, aiming to prove that Y is c.e.
- Scan V_k for errors.
  - No errors - outcome $\langle \infty_Y, k \rangle$
  - An error found at element l, then build an agitator for l, enumerate a witness z in D: $\langle z, \{k\} \rangle \searrow \Phi_0$ and $\langle z, \{l\} \rangle \searrow \Phi_1$, outcome d_0

The S^W Strategy – $o- = d_0$



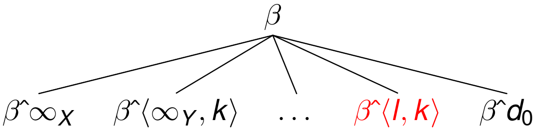
- Check if $z \in W$.
- If not outcome d_0
- If yes, then extract z from D, outcome $\langle l, k\rangle$

The S^W Strategy – $o- = d_0$



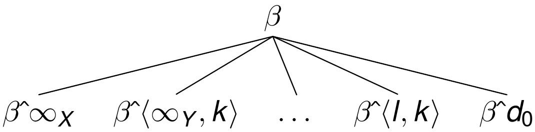
- Check if $z \in W$.
- If not outcome d_0
- If yes, then extract z from D, outcome $\langle l, k\rangle$

The S^W Strategy – $o- = d_0$



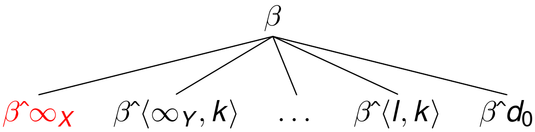
- Check if $z \in W$.
- If not outcome d_0
- If yes, then extract z from D, outcome $\langle l, k\rangle$

The S^W Strategy – $o- = \langle l, k\rangle$



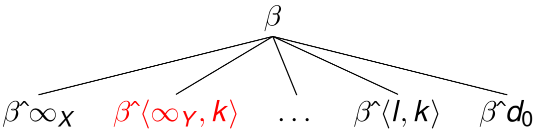
- Check if the agitator for k is valid - if not fix errors in Φ_0 and Φ_1 by enumerating axioms $\langle z, \emptyset\rangle$, outcome ∞_X
- Check if the agitator for l is valid - if not fix errors in Φ_0 and Φ_1 outcome $\langle\infty_Y, k\rangle$
- Otherwise - outcome $\langle l, k\rangle$

The S^W Strategy – $o- = \langle l, k \rangle$



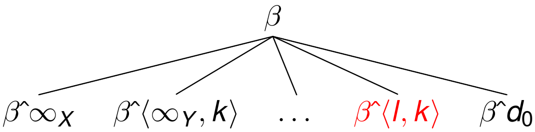
- Check if the agitator for k is valid - if not fix errors in Φ_0 and Φ_1 by enumerating axioms $\langle z, \emptyset \rangle$, outcome ∞_X
- Check if the agitator for l is valid - if not fix errors in Φ_0 and Φ_1 outcome $\langle \infty_Y, k \rangle$
- Otherwise - outcome $\langle l, k \rangle$

The S^W Strategy – $o- = \langle l, k \rangle$



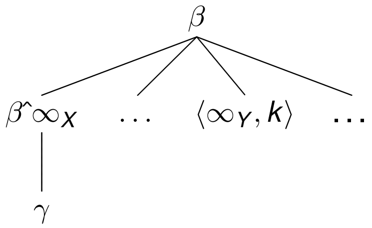
- Check if the agitator for k is valid - if not fix errors in Φ_0 and Φ_1 by enumerating axioms $\langle z, \emptyset \rangle$, outcome ∞_X
- Check if the agitator for l is valid - if not fix errors in Φ_0 and Φ_1 outcome $\langle \infty_Y, k \rangle$
- Otherwise - outcome $\langle l, k \rangle$

The S^W Strategy – $o- = \langle l, k \rangle$



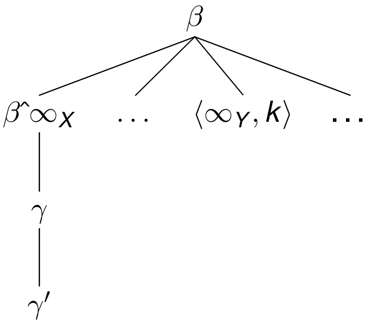
- Check if the agitator for k is valid - if not fix errors in Φ_0 and Φ_1 by enumerating axioms $\langle z, \emptyset \rangle$, outcome ∞_X
- Check if the agitator for l is valid - if not fix errors in Φ_0 and Φ_1 outcome $\langle \infty_Y, k \rangle$
- Otherwise - outcome $\langle l, k \rangle$

Local priority



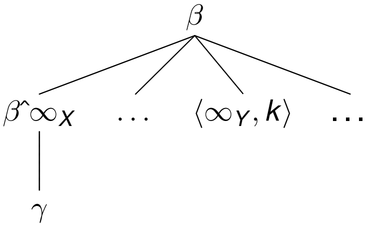
- An element k enters U, relying on axiom $\langle k, E \rangle \in \Theta_0$
- γ extracts k from X by extracting some $e \in E$ from A
- β builds agitator $E_k \supset E$ for k and moves to the right.

Local priority



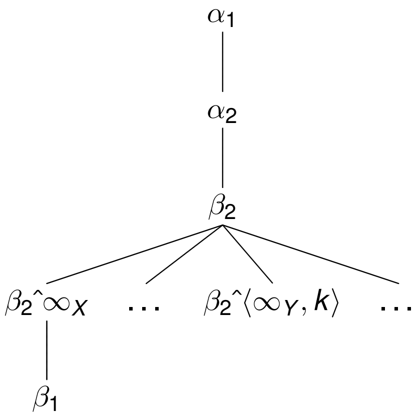
- A new axiom $\langle k, E' \rangle$ makes E_k invalid and fixes the error in U
- A new γ' extracts k from X again by extracting some $e \in E'$ from A

Local priority



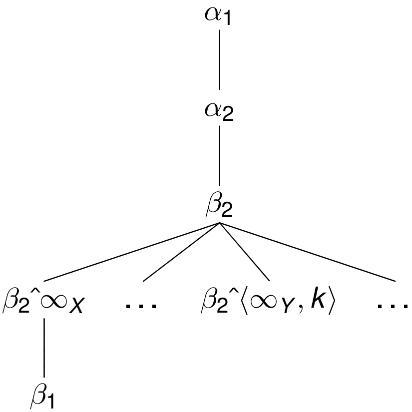
- A computable bijection $\sigma : \Gamma \to \mathbb{N}$ assigns local priority to G^W- strategies below a S^W strategy.
- If an element k enters U and $\sigma(\gamma) > k$, then γ is initialized.

The Structure Watched



- β_1 chooses agitators E_1 and F_1, modifying $(\Phi_0, \Phi_1)_{\alpha_1}$
- β_2 chooses E_2 s.t $E_1 \subset E_2$ and $F_1 \not\subseteq E_2$.
- Agitators are chosen more carefully to include all agitators of nodes with lower priority in the corresponding subtree

The Structure Watched



- Now $(E_1 \cup F_1) \subset E_2$, but E_1 becomes invalid.
- A list $Watched_\alpha$ is kept by all R^W strategies. It fixes mistakes in the operators Φ_0 and Φ_1

Bibliography

- R. I. Soare, *Recursively enumerable sets and degrees*, Springer-Verlag, Heidelberg, 1987.
- S. B. Cooper, *Computability Theory*, Chapman & Hall/CRC Mathematics, Boca Raton, FL, 2004.
- P. G. Odifreddi, *Classical Recursion Theory, Volume II*, North-Holland/Elsevier, Amsterdam, Lausanne, New York, Oxford, Shannon, Singapore, Tokyo 1999.
- K. Copestake, *1-genericity in the enumeration degrees*, J. Symbolic Logic **53** (1988), 878–887.
- S. B. Cooper, Angsheng Li, Andrea Sorbi and Yue Yang, *Bounding and nonbounding minimal pairs in the enumeration degrees*, Journal of Symbolic Logic 70 (2005), 741-766.