Sets With a Self-Modulus Bounding No Non-Recursive Δ^0_α Set

Peter Gerdes

University of California at Berkeley

Graduate Student Conference in Logic, 2007

Outline

Definitions and Notation

- Say $f \succeq g$ if $f(n) \geq g(n)$ everywhere they are both defined.
- So if $f, g \in \omega^\omega$ then $f \succeq g \Leftrightarrow (\forall n)[f(n) \geq g(n)]$

Definitions

Let $f \in \omega^\omega$ and $\mathbf{X}$ a Turing degree.

- f is a *modulus (of computation)* for $\mathbf{X}$ if for all $g \in \omega^\omega$ if $g \succeq f$ then $g \geq_T \mathbf{X}$.
- f is a *self-modulus* if f is a modulus for the turing degree of f.

Definitions and Notation

- Say $f \succeq g$ if $f(n) \geq g(n)$ everywhere they are both defined.
- So if $f, g \in \omega^\omega$ then $f \succeq g \Leftrightarrow (\forall n)[f(n) \geq g(n)]$

Definitions

Let $f \in \omega^\omega$ and $\mathbf{X}$ a Turing degree.

- f is a *modulus (of computation)* for $\mathbf{X}$ if for all $g \in \omega^\omega$ if $g \succeq f$ then $g \geq_T \mathbf{X}$.
- f is a *self-modulus* if f is a modulus for the turing degree of f.

Definitions and Notation

- Say $f \succeq g$ if $f(n) \geq g(n)$ everywhere they are both defined.
- So if $f, g \in \omega^\omega$ then $f \succeq g \Leftrightarrow (\forall n)[f(n) \geq g(n)]$

Definitions

Let $f \in \omega^\omega$ and $\mathbf{X}$ a Turing degree.

- f is a *modulus (of computation)* for $\mathbf{X}$ if for all $g \in \omega^\omega$ if $g \succeq f$ then $g \geq_T \mathbf{X}$.
- f is a *self-modulus* if f is a modulus for the turing degree of f.

Definitions and Notation

- Say $f \succeq g$ if $f(n) \geq g(n)$ everywhere they are both defined.
- So if $f, g \in \omega^\omega$ then $f \succeq g \Leftrightarrow (\forall n)[f(n) \geq g(n)]$

Definitions

Let $f \in \omega^\omega$ and $\mathbf{X}$ a Turing degree.

- f is a *modulus (of computation)* for $\mathbf{X}$ if for all $g \in \omega^\omega$ if $g \succeq f$ then $g \geq_T \mathbf{X}$.
- f is a *self-modulus* if f is a modulus for the turing degree of f.

Basic Facts

- Every Δ^0_2-set has a modulus.
  - Every Δ^0_2-set is a recursive limit ($A(x) = \lim_{s\to\infty} R_e(x, s)$).
  - Every function that looks high enough can compute the limit.
- By induction every Δ^1_1 -set has a modulus.
- Using clever trick from proof that $\Delta^0_2 \to$ h.i.-free we can show every Δ^0_2-degree computes such a function itself.
- By Induction every $0^{(\alpha)}$, $\alpha < \omega_1^{ck}$ has a self-modulus as does any n-REA set.
- So every Δ^1_1 -set has a modulus. Does anything else?

Basic Facts

- Every Δ^0_2-set has a modulus.
  - Every Δ^0_2-set is a recursive limit ($A(x) = \lim_{s\to\infty} R_e(x, s)$).
  - Every function that looks high enough can compute the limit.
- By induction every Δ^1_1 -set has a modulus.
- Using clever trick from proof that $\Delta^0_2 \rightarrow$ h.i.-free we can show every Δ^0_2-degree computes such a function itself.
- By Induction every $0^{(\alpha)}$, $\alpha < \omega_1^{ck}$ has a self-modulus as does any n-REA set.
- So every Δ^1_1 -set has a modulus. Does anything else?

Basic Facts

- Every Δ^0_2-set has a modulus.
  - Every Δ^0_2-set is a recursive limit ($A(x) = \lim_{s\to\infty} R_e(x, s)$).
  - Every function that looks high enough can compute the limit.
- By induction every Δ^1_1 -set has a modulus.
- Using clever trick from proof that $\Delta^0_2 \to$ h.i.-free we can show every Δ^0_2-degree computes such a function itself.
- By Induction every $0^{(\alpha)}$, $\alpha < \omega_1^{ck}$ has a self-modulus as does any n-REA set.
- So every Δ^1_1 -set has a modulus. Does anything else?

Basic Facts

- Every Δ^0_2-set has a modulus.
  - Every Δ^0_2-set is a recursive limit ($A(x) = \lim_{s\to\infty} R_e(x,s)$).
  - Every function that looks high enough can compute the limit.
- By induction every Δ^1_1 -set has a modulus.
- Using clever trick from proof that $\Delta^0_2 \to$ h.i.-free we can show every Δ^0_2-degree computes such a function itself.
- By Induction every $0^{(\alpha)}$, $\alpha < \omega_1^{ck}$ has a self-modulus as does any n-REA set.
- So every Δ^1_1 -set has a modulus. Does anything else?

Basic Facts

- Every Δ^0_2-set has a modulus.
  - Every Δ^0_2-set is a recursive limit ($A(x) = \lim_{s\to\infty} R_e(x,s)$).
  - Every function that looks high enough can compute the limit.
- By induction every Δ^1_1 -set has a modulus.
- Using clever trick from proof that $\Delta^0_2 \rightarrow$ h.i.-free we can show every Δ^0_2-degree computes such a function itself.
- By Induction every $0^{(\alpha)}$, $\alpha < \omega_1^{ck}$ has a self-modulus as does any n-REA set.
- So every Δ^1_1 -set has a modulus. Does anything else?

Basic Facts

- Every Δ^0_2-set has a modulus.
  - Every Δ^0_2-set is a recursive limit ($A(x) = \lim_{s\to\infty} R_e(x,s)$).
  - Every function that looks high enough can compute the limit.
- By induction every Δ^1_1 -set has a modulus.
- Using clever trick from proof that $\Delta^0_2 \to$ h.i.-free we can show every Δ^0_2-degree computes such a function itself.
- By Induction every $0^{(\alpha)}$, $\alpha < \omega_1^{ck}$ has a self-modulus as does any n-REA set.
- So every Δ^1_1 -set has a modulus. Does anything else?

Basic Facts

- Every Δ^0_2-set has a modulus.
    - Every Δ^0_2-set is a recursive limit ($A(x) = \lim_{s\to\infty} R_e(x,s)$).
    - Every function that looks high enough can compute the limit.
- By induction every Δ^1_1 -set has a modulus.
- Using clever trick from proof that $\Delta^0_2 \rightarrow$ h.i.-free we can show every Δ^0_2-degree computes such a function itself.
- By Induction every $0^{(\alpha)}$, $\alpha < \omega^{ck}_1$ has a self-modulus as does any n-REA set.
- So every Δ^1_1 -set has a modulus. Does anything else?

Characterizing Degrees With A Modulus

Theorem (Solovay)

X has a modulus if and only if X is Δ^1_1.

- Completely characterizes sets with a Modulus.
- What about sets with a self-modulus?
- Not every Δ^1_1-set has a self-modulus, e.g., hyperimmune free degrees.

Question

Could the n-REA degrees be the only degrees with a self-modulus?

Characterizing Degrees With A Modulus

Theorem (Solovay)

X has a modulus if and only if X is Δ^1_1.

- Completely characterizes sets with a Modulus.
- What about sets with a self-modulus?
- Not every Δ^1_1-set has a self-modulus, e.g., hyperimmune free degrees.

Question

Could the n-REA degrees be the only degrees with a self-modulus?

Characterizing Degrees With A Modulus

Theorem (Solovay)

X has a modulus if and only if X is Δ^1_1.

- Completely characterizes sets with a Modulus.
- What about sets with a self-modulus?
- Not every Δ^1_1-set has a self-modulus, e.g., hyperimmune free degrees.

Question

Could the n-REA degrees be the only degrees with a self-modulus?

Characterizing Degrees With A Modulus

Theorem (Solovay)

X has a modulus if and only if X is Δ^1_1.

- Completely characterizes sets with a Modulus.
- What about sets with a self-modulus?
- Not every Δ^1_1-set has a self-modulus, e.g., hyperimmune free degrees.

Question

Could the n-REA degrees be the only degrees with a self-modulus?

Characterizing Degrees With A Modulus

Theorem (Solovay)

X has a modulus if and only if X is Δ^1_1.

- Completely characterizes sets with a Modulus.
- What about sets with a self-modulus?
- Not every Δ^1_1-set has a self-modulus, e.g., hyperimmune free degrees.

Question

Could the n-REA degrees be the only degrees with a self-modulus?

Slaman and Groszek's Result

Theorem (Slaman and Groszek)

There is a self-modulus f that computes no non-recursive Δ^0_2-set.

- Need a recursive functional $\Phi : \omega^\omega \mapsto \omega^\omega$ and a function f so that Φ witnesses that f is a self-modulus.
- Build Φ in stages. Each stage specifying the behavior of Φ on bigger ($\succ$) inputs.
- At each stage s we will have a guess, f_s, at f
- f_s is always the value of Φ_s (and Φ) at biggest input considered so far.

Slaman and Groszek's Result

Theorem (Slaman and Groszek)

There is a self-modulus f that computes no non-recursive Δ^0_2-set.

- Need a recursive functional $\Phi : \omega^\omega \mapsto \omega^\omega$ and a function f so that Φ witnesses that f is a self-modulus.
- Build Φ in stages. Each stage specifying the behavior of Φ on bigger ($\succ$) inputs.
- At each stage s we will have a guess, f_s, at f
- f_s is always the value of Φ_s (and Φ) at biggest input considered so far.

Slaman and Groszek's Result

Theorem (Slaman and Groszek)

There is a self-modulus f that computes no non-recursive Δ^0_2-set.

- Need a recursive functional $\Phi : \omega^\omega \mapsto \omega^\omega$ and a function f so that Φ witnesses that f is a self-modulus.
- Build Φ in stages. Each stage specifying the behavior of Φ on bigger ($\succ$) inputs.
- At each stage s we will have a guess, f_s, at f
- f_s is always the value of Φ_s (and Φ) at biggest input considered so far.

Slaman and Groszek's Result

Theorem (Slaman and Groszek)

There is a self-modulus f that computes no non-recursive Δ^0_2-set.

- Need a recursive functional $\Phi : \omega^\omega \mapsto \omega^\omega$ and a function f so that Φ witnesses that f is a self-modulus.
- Build Φ in stages. Each stage specifying the behavior of Φ on bigger ($\succ$) inputs.
- At each stage s we will have a guess, f_s, at f
- f_s is always the value of Φ_s (and Φ) at biggest input considered so far.

Slaman and Groszek's Result

Theorem (Slaman and Groszek)

There is a self-modulus f that computes no non-recursive Δ^0_2-set.

- Need a recursive functional $\Phi : \omega^\omega \mapsto \omega^\omega$ and a function f so that Φ witnesses that f is a self-modulus.
- Build Φ in stages. Each stage specifying the behavior of Φ on bigger ($\succ$) inputs.
- At each stage s we will have a guess, f_s, at f
- f_s is always the value of Φ_s (and Φ) at biggest input considered so far.

Basic Machinery

Definition

An *axiom* is a pair $\langle \sigma \rightarrow \tau \rangle$ where $\sigma, \tau \in \omega^{<\omega}$.

- $\langle \sigma \rightarrow \tau \rangle$ 'means' map any $g \preceq \sigma$ to τ.
- We define Φ by effectively enumerating axioms into it.
- Formally $\Phi(g)(n) = \tau(n)$ where $\langle \sigma \rightarrow \tau \rangle$ first axiom enumerated into Φ with $g \preceq \sigma$ and $n \in \operatorname{dom} \tau$
- We never add an axiom we already know can't work.
- Look for places where $\varphi_e^{f_t}(m) \neq \varphi_e^{f_s}(m)$ to avoid $\varphi_e^f = A_i(x)$.

Lemma

If there is some initial segment, $f \restriction_n$, of f so that we never enumerate two axioms extending $f \restriction_n$ giving different values for $\varphi_e^{\tau}(m)$ then φ_e^f is recursive.

Basic Machinery

Definition

An *axiom* is a pair $\langle \sigma \to \tau \rangle$ where $\sigma, \tau \in \omega^{<\omega}$.

- $\langle \sigma \to \tau \rangle$ 'means' map any $g \preceq \sigma$ to τ.
- We define Φ by effectively enumerating axioms into it.
- Formally $\Phi(g)(n) = \tau(n)$ where $\langle \sigma \to \tau \rangle$ first axiom enumerated into Φ with $g \preceq \sigma$ and $n \in \operatorname{dom} \tau$
- We never add an axiom we already know can't work.
- Look for places where $\varphi_e^{f_t}(m) \neq \varphi_e^{f_s}(m)$ to avoid $\varphi_e^f = A_i(x)$.

Lemma

If there is some initial segment, $f \upharpoonright_n$, of f so that we never enumerate two axioms extending $f \upharpoonright_n$ giving different values for $\varphi_e^\tau(m)$ then φ_e^f is recursive.

Basic Machinery

Definition

An *axiom* is a pair $\langle \sigma \rightarrow \tau \rangle$ where $\sigma, \tau \in \omega^{<\omega}$.

- $\langle \sigma \rightarrow \tau \rangle$ 'means' map any $g \preceq \sigma$ to τ.
- We define Φ by effectively enumerating axioms into it.
- Formally $\Phi(g)(n) = \tau(n)$ where $\langle \sigma \rightarrow \tau \rangle$ first axiom enumerated into Φ with $g \preceq \sigma$ and $n \in \operatorname{dom} \tau$
- We never add an axiom we already know can't work.
- Look for places where $\varphi_e^{f_t}(m) \neq \varphi_e^{f_s}(m)$ to avoid $\varphi_e^f = A_i(x)$.

Lemma

If there is some initial segment, $f \upharpoonright_n$, of f so that we never enumerate two axioms extending $f \upharpoonright_n$ giving different values for $\varphi_e^\tau(m)$ then φ_e^f is recursive.

Basic Machinery

Definition

An *axiom* is a pair $\langle \sigma \to \tau \rangle$ where $\sigma, \tau \in \omega^{<\omega}$.

- $\langle \sigma \to \tau \rangle$ 'means' map any $g \preceq \sigma$ to τ.
- We define Φ by effectively enumerating axioms into it.
- Formally $\Phi(g)(n) = \tau(n)$ where $\langle \sigma \to \tau \rangle$ first axiom enumerated into Φ with $g \preceq \sigma$ and $n \in \operatorname{dom} \tau$
- We never add an axiom we already know can't work.
- Look for places where $\varphi_e^{f_t}(m) \neq \varphi_e^{f_s}(m)$ to avoid $\varphi_e^f = A_i(x)$.

Lemma

If there is some initial segment, $f \upharpoonright_n$, of f so that we never enumerate two axioms extending $f \upharpoonright_n$ giving different values for $\varphi_e^{\tau}(m)$ then φ_e^f is recursive.

Basic Machinery

Definition

An *axiom* is a pair $\langle \sigma \to \tau \rangle$ where $\sigma, \tau \in \omega^{<\omega}$.

- $\langle \sigma \to \tau \rangle$ 'means' map any $g \preceq \sigma$ to τ.
- We define Φ by effectively enumerating axioms into it.
- Formally $\Phi(g)(n) = \tau(n)$ where $\langle \sigma \to \tau \rangle$ first axiom enumerated into Φ with $g \preceq \sigma$ and $n \in \operatorname{dom} \tau$
- We never add an axiom we already know can't work.
- Look for places where $\varphi_e^{f_t}(m) \neq \varphi_e^{f_s}(m)$ to avoid $\varphi_e^f = A_i(x)$.

Lemma

If there is some initial segment, $f \upharpoonright_n$, of f so that we never enumerate two axioms extending $f \upharpoonright_n$ giving different values for $\varphi_e^{\tau}(m)$ then φ_e^f is recursive.

Basic Machinery

Definition

An *axiom* is a pair $\langle \sigma \to \tau \rangle$ where $\sigma, \tau \in \omega^{<\omega}$.

- $\langle \sigma \to \tau \rangle$ 'means' map any $g \preceq \sigma$ to τ.
- We define Φ by effectively enumerating axioms into it.
- Formally $\Phi(g)(n) = \tau(n)$ where $\langle \sigma \to \tau \rangle$ first axiom enumerated into Φ with $g \preceq \sigma$ and $n \in \operatorname{dom} \tau$
- We never add an axiom we already know can't work.
- Look for places where $\varphi_e^{f_t}(m) \neq \varphi_e^{f_s}(m)$ to avoid $\varphi_e^f = A_i(x)$.

Lemma

If there is some initial segment, $f \restriction_n$, of f so that we never enumerate two axioms extending $f \restriction_n$ giving different values for $\varphi_e^\tau(m)$ then φ_e^f is recursive.

Basic Machinery

Definition

An *axiom* is a pair $\langle \sigma \rightarrow \tau \rangle$ where $\sigma, \tau \in \omega^{<\omega}$.

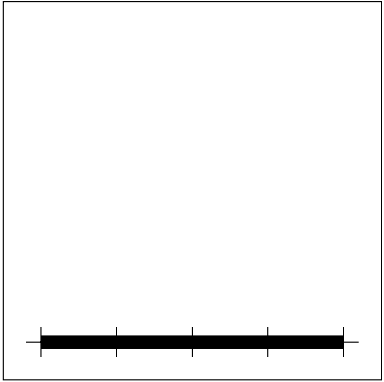
- $\langle \sigma \rightarrow \tau \rangle$ 'means' map any $g \preceq \sigma$ to τ.
- We define Φ by effectively enumerating axioms into it.
- Formally $\Phi(g)(n) = \tau(n)$ where $\langle \sigma \rightarrow \tau \rangle$ first axiom enumerated into Φ with $g \preceq \sigma$ and $n \in \operatorname{dom} \tau$
- We never add an axiom we already know can't work.
- Look for places where $\varphi_e^{f_t}(m) \neq \varphi_e^{f_s}(m)$ to avoid $\varphi_e^f = A_i(x)$.

Lemma

If there is some initial segment, $f\restriction_n$, of f so that we never enumerate two axioms extending $f\restriction_n$ giving different values for $\varphi_e^\tau(m)$ then φ_e^f is recursive.

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



Start with $f_0 = \langle 0, 0, 0, 0 \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

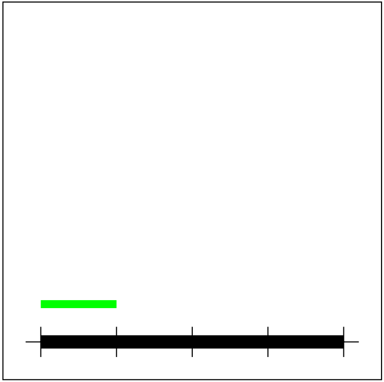
- $\langle (1) \to (0) \rangle$
- $\langle (2,1) \to (0,0) \rangle$
- $\langle (5,5) \to (0,4) \rangle$
- $\langle (8,7,6) \to (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \to (3,2) \rangle$
- $\langle (7,6,4) \to (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



$R_i(x,1) = 0$. Guess $f(0) = 0$.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

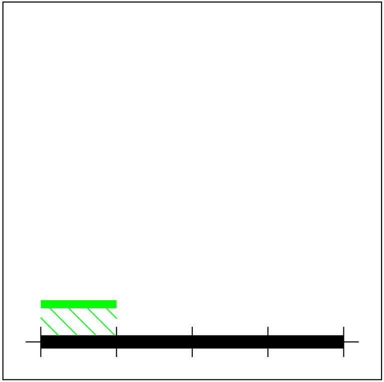
- $\langle (1) \to (0) \rangle$
- $\langle (2,1) \to (0,0) \rangle$
- $\langle (5,5) \to (0,4) \rangle$
- $\langle (8,7,6) \to (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \to (3,2) \rangle$
- $\langle (7,6,4) \to (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



If $g(0) \leq 1$ then $f(0) = 0$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

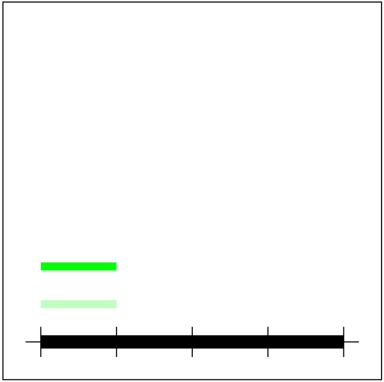
- $\langle(1) \to (0)\rangle$
- $\langle(2,1) \to (0,0)\rangle$
- $\langle(5,5) \to (0,4)\rangle$
- $\langle(8,7,6) \to (0,4,5)\rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle(4,3) \to (3,2)\rangle$
- $\langle(7,6,4) \to (6,4,3)\rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



$R_i(x,2) = 0$ Still Guess $f(0) = 0$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

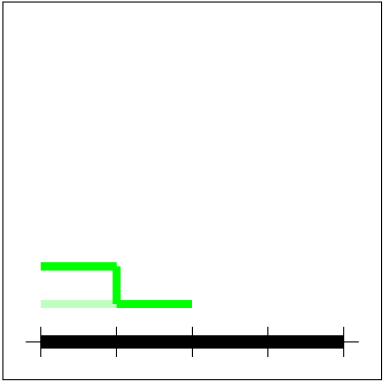
- $\langle(1) \to (0)\rangle$
- $\langle(2,1) \to (0,0)\rangle$
- $\langle(5,5) \to (0,4)\rangle$
- $\langle(8,7,6) \to (0,4,5)\rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle(4,3) \to (3,2)\rangle$
- $\langle(7,6,4) \to (6,4,3)\rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



Don't stop moving right.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

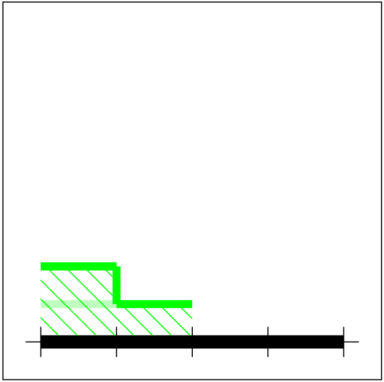
- $\langle (1) \rightarrow (0) \rangle$
- $\langle (2,1) \rightarrow (0,0) \rangle$
- $\langle (5,5) \rightarrow (0,4) \rangle$
- $\langle (8,7,6) \rightarrow (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \rightarrow (3,2) \rangle$
- $\langle (7,6,4) \rightarrow (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s \to \infty} R_i(x, s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s \to \infty} R_i(x, s) = 1 \end{cases}$



If $g(0) \leq 2$ then $f(0) = 0$.

Axioms Guessing: $\lim_{s \to \infty} R_i(x, s) \neq 1$

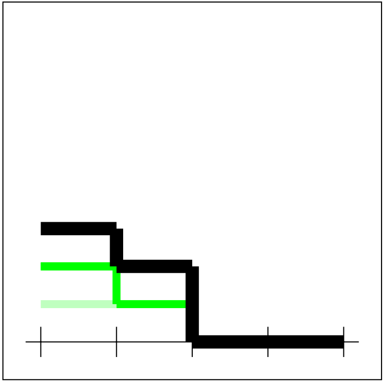
- $\langle (1) \to (0) \rangle$
- $\langle (2, 1) \to (0, 0) \rangle$
- $\langle (5, 5) \to (0, 4) \rangle$
- $\langle (8, 7, 6) \to (0, 4, 5) \rangle$

Axioms Guessing: $\lim_{s \to \infty} R_i(x, s) \neq 1$

- $\langle (4, 3) \to (3, 2) \rangle$
- $\langle (7, 6, 4) \to (6, 4, 3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



$R_i(x,3) = 1$. Obsolete green axioms.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

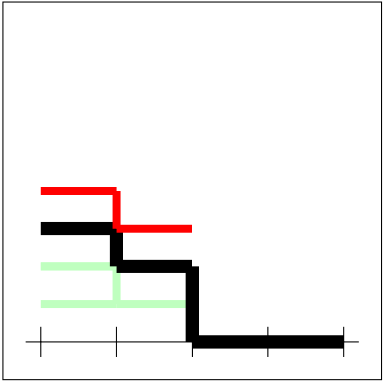
- $\langle (1) \to (0) \rangle$
- $\langle (2,1) \to (0,0) \rangle$
- $\langle (5,5) \to (0,4) \rangle$
- $\langle (8,7,6) \to (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \to (3,2) \rangle$
- $\langle (7,6,4) \to (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



Add axiom guessing $f(0) = 3$.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

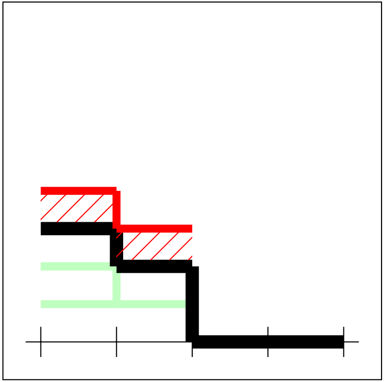
- $\langle (1) \to (0) \rangle$
- $\langle (2,1) \to (0,0) \rangle$
- $\langle (5,5) \to (0,4) \rangle$
- $\langle (8,7,6) \to (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \to (3,2) \rangle$
- $\langle (7,6,4) \to (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



But now need $g(1) \leq 3$ to trigger.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

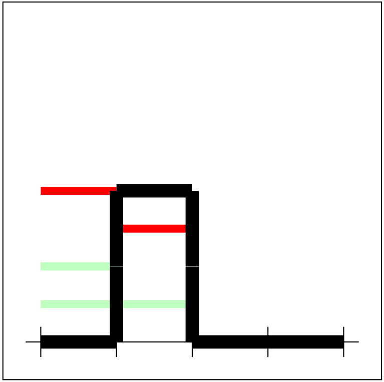
- $\langle (1) \to (0) \rangle$
- $\langle (2,1) \to (0,0) \rangle$
- $\langle (5,5) \to (0,4) \rangle$
- $\langle (8,7,6) \to (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \to (3,2) \rangle$
- $\langle (7,6,4) \to (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



$R_i(x,4) = 0$ Obsolete red axioms.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

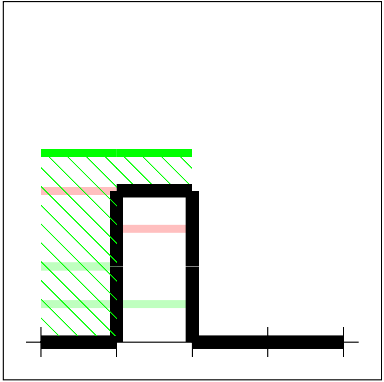
- $\langle (1) \to (0) \rangle$
- $\langle (2,1) \to (0,0) \rangle$
- $\langle (5,5) \to (0,4) \rangle$
- $\langle (8,7,6) \to (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \to (3,2) \rangle$
- $\langle (7,6,4) \to (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



Add green axiom.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

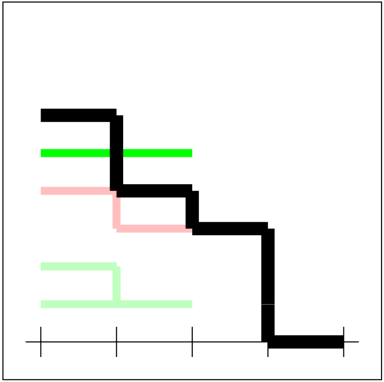
- $\langle (1) \to (0) \rangle$
- $\langle (2,1) \to (0,0) \rangle$
- $\langle (5,5) \to (0,4) \rangle$
- $\langle (8,7,6) \to (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \to (3,2) \rangle$
- $\langle (7,6,4) \to (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



$R_i(x, 5) = 1$. Obsolete green axioms.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

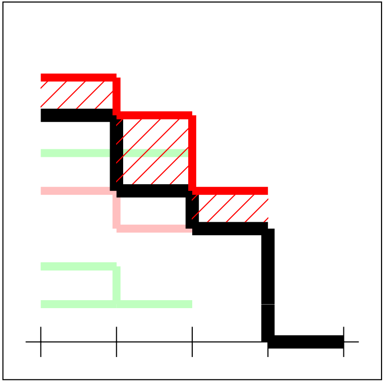
- $\langle (1) \to (0) \rangle$
- $\langle (2,1) \to (0,0) \rangle$
- $\langle (5,5) \to (0,4) \rangle$
- $\langle (8,7,6) \to (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \to (3,2) \rangle$
- $\langle (7,6,4) \to (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



Add red axiom.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

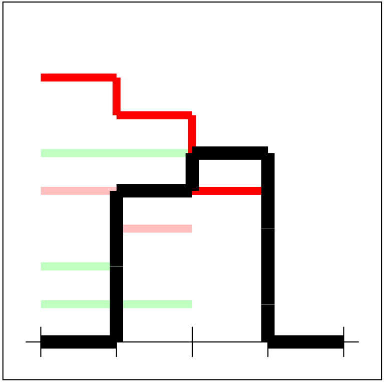
- $\langle(1) \to (0)\rangle$
- $\langle(2,1) \to (0,0)\rangle$
- $\langle(5,5) \to (0,4)\rangle$
- $\langle(8,7,6) \to (0,4,5)\rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle(4,3) \to (3,2)\rangle$
- $\langle(7,6,4) \to (6,4,3)\rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



$R_i(x, 6) = 0$ Obsolete red axioms.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

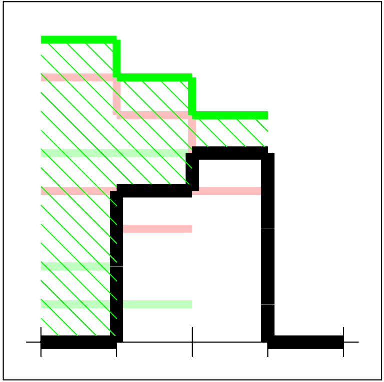
- $\langle (1) \to (0) \rangle$
- $\langle (2,1) \to (0,0) \rangle$
- $\langle (5,5) \to (0,4) \rangle$
- $\langle (8,7,6) \to (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \to (3,2) \rangle$
- $\langle (7,6,4) \to (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



Add green axiom.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

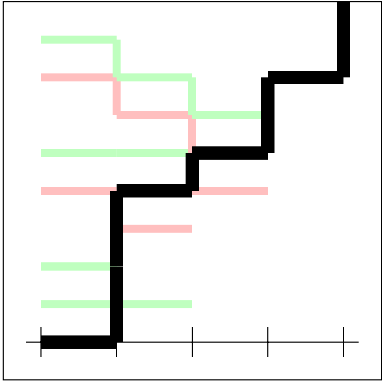
- $\langle (1) \to (0) \rangle$
- $\langle (2,1) \to (0,0) \rangle$
- $\langle (5,5) \to (0,4) \rangle$
- $\langle (8,7,6) \to (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \to (3,2) \rangle$
- $\langle (7,6,4) \to (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



If we keep switching get staircase.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

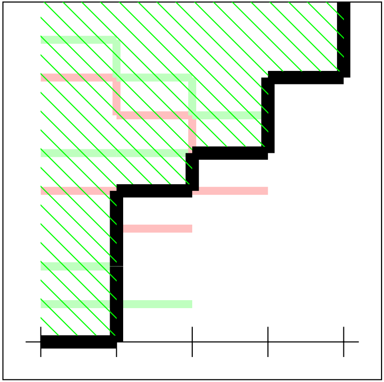
- $\langle (1) \to (0) \rangle$
- $\langle (2,1) \to (0,0) \rangle$
- $\langle (5,5) \to (0,4) \rangle$
- $\langle (8,7,6) \to (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \to (3,2) \rangle$
- $\langle (7,6,4) \to (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



Only green axioms above stairs.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

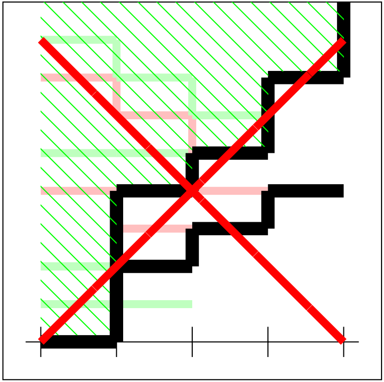
- $\langle (1) \to (0) \rangle$
- $\langle (2,1) \to (0,0) \rangle$
- $\langle (5,5) \to (0,4) \rangle$
- $\langle (8,7,6) \to (0,4,5) \rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle (4,3) \to (3,2) \rangle$
- $\langle (7,6,4) \to (6,4,3) \rangle$

Avoiding One Set

Enumerate axioms so that $\begin{cases} f(0) = 0 & \text{if } \lim_{s\to\infty} R_i(x,s) \neq 1 \\ f(0) \neq 1 & \text{if } \lim_{s\to\infty} R_i(x,s) = 1 \end{cases}$



Later f_s can't go under.

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle(1) \to (0)\rangle$
- $\langle(2,1) \to (0,0)\rangle$
- $\langle(5,5) \to (0,4)\rangle$
- $\langle(8,7,6) \to (0,4,5)\rangle$

Axioms Guessing: $\lim_{s\to\infty} R_i(x,s) \neq 1$

- $\langle(4,3) \to (3,2)\rangle$
- $\langle(7,6,4) \to (6,4,3)\rangle$

Δ^0_2 Injuries

- Avoid $\varphi^f_e = A_i$ with priority $\langle e, i \rangle$.
- Each requirement reserves initial segment plus extra space needed to cancel it.
- While guessing $R_e \neq 0$ only reserve first column.
- Along true path machinery for R_e only takes finite space.
- Lower priorities can't mess with higher ones.
- Lower priorities play a strategy for every initial segment not yet handled.
- Every requirement is injured only finitely.

Δ^0_2 Injuries

- Avoid $\varphi^f_e = A_i$ with priority $\langle e, i \rangle$.
- Each requirement reserves initial segment plus extra space needed to cancel it.
- While guessing $R_e \neq 0$ only reserve first column.
- Along true path machinery for R_e only takes finite space.
- Lower priorities can't mess with higher ones.
- Lower priorities play a strategy for every initial segment not yet handled.
- Every requirement is injured only finitely.

Δ^0_2 Injuries

- Avoid $\varphi^f_e = A_i$ with priority $\langle e, i \rangle$.
- Each requirement reserves initial segment plus extra space needed to cancel it.
- While guessing $R_e \neq 0$ only reserve first column.
- Along true path machinery for R_e only takes finite space.
- Lower priorities can't mess with higher ones.
- Lower priorities play a strategy for every initial segment not yet handled.
- Every requirement is injured only finitely.

Δ^0_2 Injuries

- Avoid $\varphi_e^f = A_i$ with priority $\langle e, i \rangle$.
- Each requirement reserves initial segment plus extra space needed to cancel it.
- While guessing $R_e \neq 0$ only reserve first column.
- Along true path machinery for R_e only takes finite space.
- Lower priorities can't mess with higher ones.
- Lower priorities play a strategy for every initial segment not yet handled.
- Every requirement is injured only finitely.

Δ^0_2 Injuries

- Avoid $\varphi^f_e = A_i$ with priority $\langle e, i \rangle$.
- Each requirement reserves initial segment plus extra space needed to cancel it.
- While guessing $R_e \neq 0$ only reserve first column.
- Along true path machinery for R_e only takes finite space.
- Lower priorities can't mess with higher ones.
- Lower priorities play a strategy for every initial segment not yet handled.
- Every requirement is injured only finitely.

Δ^0_2 Injuries

- Avoid $\varphi_e^f = A_i$ with priority $\langle e, i \rangle$.
- Each requirement reserves initial segment plus extra space needed to cancel it.
- While guessing $R_e \neq 0$ only reserve first column.
- Along true path machinery for R_e only takes finite space.
- Lower priorities can't mess with higher ones.
- Lower priorities play a strategy for every initial segment not yet handled.
- Every requirement is injured only finitely.

Δ^0_2 Injuries

- Avoid $\varphi^f_e = A_i$ with priority $\langle e, i \rangle$.
- Each requirement reserves initial segment plus extra space needed to cancel it.
- While guessing $R_e \neq 0$ only reserve first column.
- Along true path machinery for R_e only takes finite space.
- Lower priorities can't mess with higher ones.
- Lower priorities play a strategy for every initial segment not yet handled.
- Every requirement is injured only finitely.

New Results

Theorem

For every $\alpha < \omega_1^{ck}$ there is a self-modulus above no non-recursive Δ^0_α set.

- Inductively layer machinery from Δ^0_2 construction on top of itself.
- Now every requirement counts on an infinite amount of machinery.
- Must allow supporting machinery to be injured without reseting attempt to meet that requirement.

New Results

Theorem

For every $\alpha < \omega_1^{ck}$ there is a self-modulus above no non-recursive Δ^0_α set.

- Inductively layer machinery from Δ^0_2 construction on top of itself.
- Now every requirement counts on an infinite amount of machinery.
- Must allow supporting machinery to be injured without reseting attempt to meet that requirement.

New Results

Theorem

For every $\alpha < \omega_1^{ck}$ there is a self-modulus above no non-recursive Δ^0_α set.

- Inductively layer machinery from Δ^0_2 construction on top of itself.
- Now every requirement counts on an infinite amount of machinery.
- Must allow supporting machinery to be injured without reseting attempt to meet that requirement.

New Results

Theorem

For every $\alpha < \omega_1^{ck}$ there is a self-modulus above no non-recursive Δ^0_α set.

- Inductively layer machinery from Δ^0_2 construction on top of itself.
- Now every requirement counts on an infinite amount of machinery.
- Must allow supporting machinery to be injured without reseting attempt to meet that requirement.

Infinite Machinery

- Distribute priorities to machinery working for a requirement.
- When supporting machinery is injured just copy it to after the injury.
- The moved machinery still controls the earlier values and thus whether or not the injury 'happens' along the truth path.
- Δ^0_2-proof stuck all the machinery for higher priority before lower priority machinery.
- **PROBLEM!** Now one requirement might injure another infinitely often.

Infinite Machinery

- Distribute priorities to machinery working for a requirement.
- When supporting machinery is injured just copy it to after the injury.
- The moved machinery still controls the earlier values and thus whether or not the injury 'happens' along the truth path.
- Δ^0_2-proof stuck all the machinery for higher priority before lower priority machinery.
- **PROBLEM!** Now one requirement might injure another infinitely often.

Infinite Machinery

- Distribute priorities to machinery working for a requirement.
- When supporting machinery is injured just copy it to after the injury.
- The moved machinery still controls the earlier values and thus whether or not the injury 'happens' along the truth path.
- Δ^0_2-proof stuck all the machinery for higher priority before lower priority machinery.
- **PROBLEM!**Now one requirement might injure another infinitely often.

Infinite Machinery

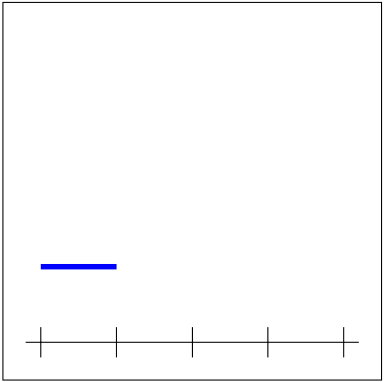
- Distribute priorities to machinery working for a requirement.
- When supporting machinery is injured just copy it to after the injury.
- The moved machinery still controls the earlier values and thus whether or not the injury 'happens' along the truth path.
- Δ^0_2-proof stuck all the machinery for higher priority before lower priority machinery.
- **PROBLEM!** Now one requirement might injure another infinitely often.

Infinite Machinery

- Distribute priorities to machinery working for a requirement.
- When supporting machinery is injured just copy it to after the injury.
- The moved machinery still controls the earlier values and thus whether or not the injury 'happens' along the truth path.
- Δ^0_2-proof stuck all the machinery for higher priority before lower priority machinery.
- **PROBLEM!** Now one requirement might injure another infinitely often.

Infinite Interference

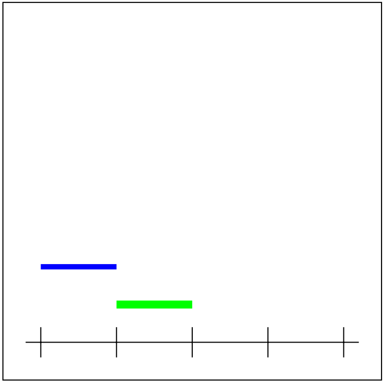
This time all the lines represent guesses at f. Suppose green bars belong to a higher priority requirement but this blue machinery has higher priority than this green machinery.



- Green uses column after blue.
- Intermediate priority requirement acts.
- Green machinery moves.
- Blue column changes guess.
- Same requirement acts again.
- Green moves again.
- Green injured again.

Infinite Interference

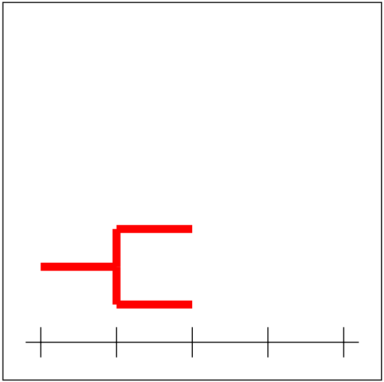
This time all the lines represent guesses at f. Suppose green bars belong to a higher priority requirement but this blue machinery has higher priority than this green machinery.



- Green uses column after blue.
- Intermediate priority requirement acts.
- Green machinery moves.
- Blue column changes guess.
- Same requirement acts again.
- Green moves again.
- Green injured again.

Infinite Interference

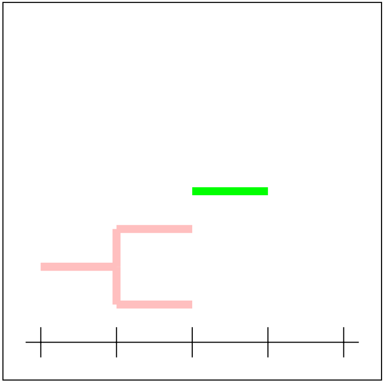
This time all the lines represent guesses at f. Suppose green bars belong to a higher priority requirement but this blue machinery has higher priority than this green machinery.



- Green uses column after blue.
- Intermediate priority requirement acts.
- Green machinery moves.
- Blue column changes guess.
- Same requirement acts again.
- Green moves again.
- Green injured again.

Infinite Interference

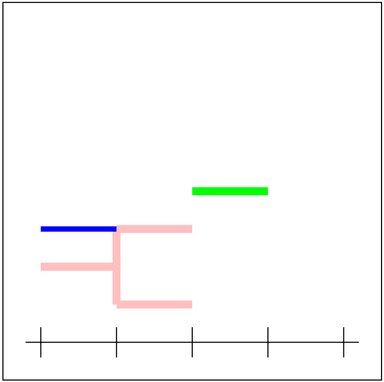
This time all the lines represent guesses at f. Suppose green bars belong to a higher priority requirement but this blue machinery has higher priority than this green machinery.



- Green uses column after blue.
- Intermediate priority requirement acts.
- Green machinery moves.
- Blue column changes guess.
- Same requirement acts again.
- Green moves again.
- Green injured again.

Infinite Interference

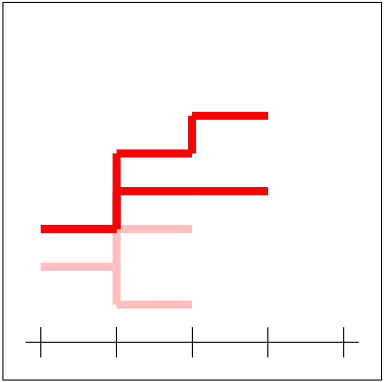
This time all the lines represent guesses at f. Suppose green bars belong to a higher priority requirement but this blue machinery has higher priority than this green machinery.



- Green uses column after blue.
- Intermediate priority requirement acts.
- Green machinery moves.
- Blue column changes guess.
- Same requirement acts again.
- Green moves again.
- Green injured again.

Infinite Interference

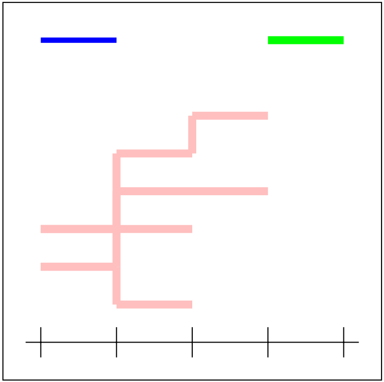
This time all the lines represent guesses at f. Suppose green bars belong to a higher priority requirement but this blue machinery has higher priority than this green machinery.



- Green uses column after blue.
- Intermediate priority requirement acts.
- Green machinery moves.
- Blue column changes guess.
- Same requirement acts again.
- Green moves again.
- Green injured again.

Infinite Interference

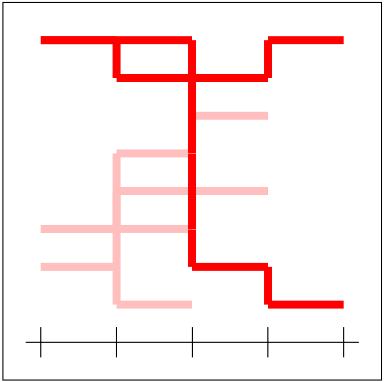
This time all the lines represent guesses at f. Suppose green bars belong to a higher priority requirement but this blue machinery has higher priority than this green machinery.



- Green uses column after blue.
- Intermediate priority requirement acts.
- Green machinery moves.
- Blue column changes guess.
- Same requirement acts again.
- Green moves again.
- Green injured again.

Infinite Interference

This time all the lines represent guesses at f. Suppose green bars belong to a higher priority requirement but this blue machinery has higher priority than this green machinery.



- Green uses column after blue.
- Intermediate priority requirement acts.
- Green machinery moves.
- Blue column changes guess.
- Same requirement acts again.
- Green moves again.
- Green injured again.

Protecting Machinery

- Let higher priority column protect later lower priority columns.
- The bigger the guess at f is in a column the more lower priority columns it can protect.
- Don't want protection to extend indefinitly!
- Only allow protection of columns belonging to higher priority requirements or lower level in same requirement.
- Along true path protection only extends finitely far.
- Everything else can play a different strategy for each value take.

Protecting Machinery

- Let higher priority column protect later lower priority columns.
- The bigger the guess at f is in a column the more lower priority columns it can protect.
- Don't want protection to extend indefinitly!
- Only allow protection of columns belonging to higher priority requirements or lower level in same requirement.
- Along true path protection only extends finitely far.
- Everything else can play a different strategy for each value take.

Protecting Machinery

- Let higher priority column protect later lower priority columns.
- The bigger the guess at f is in a column the more lower priority columns it can protect.
- Don't want protection to extend indefinitly!
- Only allow protection of columns belonging to higher priority requirements or lower level in same requirement.
- Along true path protection only extends finitely far.
- Everything else can play a different strategy for each value take.

Protecting Machinery

- Let higher priority column protect later lower priority columns.
- The bigger the guess at f is in a column the more lower priority columns it can protect.
- Don't want protection to extend indefinitly!
- Only allow protection of columns belonging to higher priority requirements or lower level in same requirement.
- Along true path protection only extends finitely far.
- Everything else can play a different strategy for each value take.

Protecting Machinery

- Let higher priority column protect later lower priority columns.
- The bigger the guess at f is in a column the more lower priority columns it can protect.
- Don't want protection to extend indefinitly!
- Only allow protection of columns belonging to higher priority requirements or lower level in same requirement.
- Along true path protection only extends finitely far.
- Everything else can play a different strategy for each value take.

Protecting Machinery

- Let higher priority column protect later lower priority columns.
- The bigger the guess at f is in a column the more lower priority columns it can protect.
- Don't want protection to extend indefinitly!
- Only allow protection of columns belonging to higher priority requirements or lower level in same requirement.
- Along true path protection only extends finitely far.
- Everything else can play a different strategy for each value take.

Thanks

In no particular order:

- My advisor Leo Harrington for frequently talking with me about the problem.
- Theodore Slaman for suggesting the problem and showing me the Δ^0_2-case.
- The conference organizers for setting this all up.