Comparing Notions of Robust Computability On ω^ω and 2^ω

Peter M. Gerdes

ASL North American Meeting, July 2026

Outline

Notions of Robust Computation

- Recently [1, 2], there has been interest in studying notions of robust computation — reductions that in some sense measure how much information is encoded in an object in a way that is resilient to small errors.

Robust Reduction

An object X robustly computes Y if every $\widehat{X} \approx X$ computes some $\hat{Y} \approx Y$.

- Robust reductions understand $\approx$ to mean nearly equal.
- We will focus on two notions of robust computation — coarse and effective dense reducibility.

Outline

Small Sets

Definition (Density)

$$\rho(X) \stackrel{\text{def}}{=} \lim_{l\to\infty} \frac{|X\restriction_l|}{l} \qquad \bar{\rho}(X) \stackrel{\text{def}}{=} \limsup_{l\to\infty} \frac{|X\restriction_l|}{l} \qquad \bar{\rho}_a(X) \stackrel{\text{def}}{=} \sup_{l\geq a} \frac{|X\restriction_l|}{l}$$

- $\rho(X)$ and $\bar{\rho}(X)$ are the density and upper density of X respectively.
- $\bar{\rho}_a(X)$ is the upper density of X above a.
- ▯ is the ideal of sets with density 0

- So $S \subset \omega$ is small (in ▯) if the fraction of numbers it contains goes to 0.
- We will work mostly in terms of upper density.
  - $\bar{\rho}(X) = 0 \iff \rho(X) = 0 \iff \rho\left(\overline{X}\right) = 1.$
  - So ▯ is also the ideal of sets with upper density 0.

Coarse Reductions

Definition

$h \in \omega^\omega$ is a coarse description of $f \in \omega^\omega$, if

$$f \,\Delta\, h \in \text{▯} \text{ where } f \,\Delta\, h \overset{\text{def}}{=} \{x \mid f(x) \neq h(x)\}$$

- Actually an equivalence relation — we will sometimes say coarse equivalence.

Understanding ‘computes’ to mean the existence of a Turing reduction yields

Definition

$g \in \omega^\omega$ is non-uniformly coarsely reducible to $f \in \omega^\omega$, written $f \geq_{\mathbf{nc}} g$, if every coarse description of f Turing computes a coarse description of g.

- If this is witnessed by a single computable functional then g is uniformly coarsely reducible to f — denoted $f \geq_{\mathbf{uc}} g$.

Coarse Reductions

Definition

$h \in \omega^\omega$ is a coarse description of $f \in \omega^\omega$, if

$$f \mathbin{\Delta} h \in \mathbb{I} \text{ where } f \mathbin{\Delta} h \overset{\text{def}}{=} \{x \mid f(x) \neq h(x)\}$$

- Actually an equivalence relation — we will sometimes say coarse equivalence.

Understanding ‘computes’ to mean the existence of a Turing reduction yields

Definition

$g \in \omega^\omega$ is non-uniformly coarsely reducible to $f \in \omega^\omega$, written $f \geq_{\mathbf{nc}} g$, if every coarse description of f Turing computes a coarse description of g.

- If this is witnessed by a single computable functional then g is uniformly coarsely reducible to f — denoted $f \geq_{\mathbf{uc}} g$.

Coarse Reductions

Definition

$h \in \omega^\omega$ is a coarse description of $f \in \omega^\omega$, if

$$f \mathbin{\Delta} h \in \square \text{ where } f \mathbin{\Delta} h \overset{\text{def}}{=} \{x \mid f(x) \neq h(x)\}$$

- Actually an equivalence relation — we will sometimes say coarse equivalence.

Understanding ‘computes’ to mean the existence of a Turing reduction yields

Definition

$g \in \omega^\omega$ is non-uniformly coarsely reducible to $f \in \omega^\omega$, written $f \geq_{\mathbf{nc}} g$, if every coarse description of f Turing computes a coarse description of g.

- If this is witnessed by a single computable functional then g is uniformly coarsely reducible to f — denoted $f \geq_{\mathbf{uc}} g$.

Effective Dense Descriptions

Effective dense descriptions (EDD) are like coarse descriptions but must declare the locations at which we don't know the value — we use $\square$ to denote such a location.

Definition (Effective Dense Description)

For $f, h \in (\omega \cup \{\square\})^\omega$ we write $f \cong^0_\square h$ just when

- $(\forall x)(f(x) = h(x) \lor f(x) = \square \lor h(x) = \square)$
- $\{x \mid f(x) = \square \lor h(x) = \square\} \in \mathbb{I}$

When $f \in \omega^\omega$ we say h is an EDD of f.

Lemma

h is an EDD of $f \in \omega^\omega$ iff $h = f^{\square S}$ for some $S \in \mathbb{I}$ where

$$f^{\square S} \overset{\text{def}}{=} \begin{cases} f(x) & \text{if } x \notin S \\ \square & \text{otherwise} \end{cases}$$

Effective Dense Descriptions

Effective dense descriptions (EDD) are like coarse descriptions but must declare the locations at which we don't know the value — we use $\square$ to denote such a location.

Definition (Effective Dense Description)

For $f, h \in (\omega \cup \{\square\})^\omega$ we write $f \cong^0_\square h$ just when

- $(\forall x)(f(x) = h(x) \lor f(x) = \square \lor h(x) = \square)$
- $\{x \mid f(x) = \square \lor h(x) = \square\} \in \mathbb{I}$

When $f \in \omega^\omega$ we say h is an EDD of f.

Lemma

h is an EDD of $f \in \omega^\omega$ iff $h = f^{\square S}$ for some $S \in \mathbb{I}$ where

$$f^{\square S} \stackrel{\text{def}}{=} \begin{cases} f(x) & \text{if } x \notin S \\ \square & \text{otherwise} \end{cases}$$

Effective Dense Reducibility

Definition

$g \in \omega^\omega$ is non-uniformly effectively densely reducible to $f \in \omega^\omega$ — written $f \geq_{\mathbf{ned}} g$ — iff every effective dense description of f computes an effective dense description of g.

- If the reduction is witnessed by a single computable functional then g is uniformly effectively densely reducible to f — written $f \geq_{\mathbf{ued}} g$.

- Uniformity is about the reduction while effective is about the description.
  - An EDD is effective in comparison to a dense description which can simply fail to be defined on a density 0 set.

Effective Dense Reducibility

Definition

$g \in \omega^\omega$ is non-uniformly effectively densely reducible to $f \in \omega^\omega$ — written $f \geq_{\mathbf{ned}} g$ — iff every effective dense description of f computes an effective dense description of g.

- If the reduction is witnessed by a single computable functional then g is uniformly effectively densely reducible to f — written $f \geq_{\mathbf{ued}} g$.

- Uniformity is about the reduction while effective is about the description.
  - An EDD is effective in comparison to a dense description which can simply fail to be defined on a density 0 set.

Summary

- Coarse and effective dense reductions both say that $f \geq g$ when we can take a version of f that is correct except on a small set and get a version of g that is correct except on a small set.
  - Coarse reducibility allows us to just get it wrong on that small set.
  - Effective Dense reducibility allows us to declare we don't know on that small set — but we can never be wrong.
- An important theme of this talk is that small seeming change results in some very different behavior — though not always in the way you might think.

Sets vs Functions

The question that motivated this work (from [1] but also a talk at Midwest Computability) is the following:

Question

Does every degree of the above types (uniform/non-uniform coarse, uniform/non-uniform effective dense) contain a set?

Here we identify sets with their characteristic functions, i.e., sets are elements of $2^\omega \subset \omega^\omega$.

- Most research on these reductions up to this point has focused on sets.
- If every degree contains a set, we lose nothing by restricting out attention to sets.
- Our focus here will more be on using the answer to this question to distinguish coarse and effective dense reductions.

Sets vs Functions

The question that motivated this work (from [1] but also a talk at Midwest Computability) is the following:

Question

Does every degree of the above types (uniform/non-uniform coarse, uniform/non-uniform effective dense) contain a set?

Here we identify sets with their characteristic functions, i.e., sets are elements of $2^\omega \subset \omega^\omega$.

- Most research on these reductions up to this point has focused on sets.
- If every degree contains a set, we lose nothing by restricting out attention to sets.
- Our focus here will more be on using the answer to this question to distinguish coarse and effective dense reductions.

Results: Sets vs. Functions

Theorem (ED degrees without sets)

If $f \in \omega^\omega$ is 3-generic then there is no set X such that $f \equiv_{\mathbf{ned}} X$.

- Obviously, if f is (Cohen) 3-generic then $f \not\equiv_{\mathbf{ued}} X$ for any X

In contrast

Theorem (Coarse degrees contain sets)

Every uniform (and hence non-uniform) coarse degree contains a set.

- We show this holds in the most uniform way possible by producing a pair of computable functionals $\Gamma, \Gamma^\dagger$ that witness the claim (i.e. respect coarse equivalence).

Theorem (ED degrees without sets)

If $f \in \omega^\omega$ is 3-generic then there is no set X such that $f \equiv_{\mathbf{ned}} X$.

- Obviously, if f is (Cohen) 3-generic then $f \not\equiv_{\mathbf{ued}} X$ for any X

In contrast

Theorem (Coarse degrees contain sets)

Every uniform (and hence non-uniform) coarse degree contains a set.

- We show this holds in the most uniform way possible by producing a pair of computable functionals $\Gamma, \Gamma^\dagger$ that witness the claim (i.e. respect coarse equivalence).

Results: Complexity

This research also lead to a more dramatic distinction between effective dense and coarse reductions.

Theorem (Coarse reducibility is arithmetic)

$f \geq_{\mathbf{nc}} g$ (and $f \geq_{\mathbf{uc}} g$) are arithmetic properties of $f \oplus g$.

Theorem (ED Reducibility is Π^1_1 complete)

$f \geq_{\mathbf{ned}} g$ is a Π^1_1 complete property of f, g.

- Might have thought it would be the opposite — coarse reductions should be complex because wrong answers can hide.
- Pretty much everything not immediately implied by this slide is open.

Outline

Initial Difficulty

- First try: assume f forces $f \equiv_{\mathbf{ned}} X$ for some $X \in 2^{\omega}$ and derive a contradiction.
  - Forcing requires some kind of definability. Can we assume such an X is definable from f?

Question

If $f \equiv_{\mathbf{ned}} X \in 2^{\omega}$ must there be $Y \in 2^{\omega}$ with Y hyperarithmetic in f and $f \equiv_{\mathbf{ned}} Y$?

- f can compute some $X^{\square S}$. Can't we just fill in some random junk on S?
  - That junk might cause reductions in other direction to guess wrong (infinitely often).
- Can we search for a way to fill in values on S that preserves reductions?
  - Naive approach starts looking Σ^1_1 complete.

Initial Difficulty

- First try: assume f forces $f \equiv_{\mathbf{ned}} X$ for some $X \in 2^{\omega}$ and derive a contradiction.
  - Forcing requires some kind of definability. Can we assume such an X is definable from f?

Question

If $f \equiv_{\mathbf{ned}} X \in 2^{\omega}$ must there be $Y \in 2^{\omega}$ with Y hyperarithmetic in f and $f \equiv_{\mathbf{ned}} Y$?

- f can compute some $X^{\square S}$. Can't we just fill in some random junk on S?
  - That junk might cause reductions in other direction to guess wrong (infinitely often).
- Can we search for a way to fill in values on S that preserves reductions?
  - Naive approach starts looking Σ^1_1 complete.

Initial Difficulty

- First try: assume f forces $f \equiv_{\mathbf{ned}} X$ for some $X \in 2^\omega$ and derive a contradiction.
    - Forcing requires some kind of definability. Can we assume such an X is definable from f?

Question

If $f \equiv_{\mathbf{ned}} X \in 2^\omega$ must there be $Y \in 2^\omega$ with Y hyperarithmetic in f and $f \equiv_{\mathbf{ned}} Y$?

- f can compute some $X^{\square S}$. Can't we just fill in some random junk on S?
    - That junk might cause reductions in other direction to guess wrong (infinitely often).
- Can we search for a way to fill in values on S that preserves reductions?
    - Naive approach starts looking Σ^1_1 complete.

Initial Difficulty

- First try: assume f forces $f \equiv_{\mathbf{ned}} X$ for some $X \in 2^\omega$ and derive a contradiction.
  - Forcing requires some kind of definability. Can we assume such an X is definable from f?

Question

If $f \equiv_{\mathbf{ned}} X \in 2^\omega$ must there be $Y \in 2^\omega$ with Y hyperarithmetic in f and $f \equiv_{\mathbf{ned}} Y$?

- f can compute some $X^{\square S}$. Can't we just fill in some random junk on S?
  - That junk might cause reductions in other direction to guess wrong (infinitely often).
- Can we search for a way to fill in values on S that preserves reductions?
  - Naive approach starts looking Σ^1_1 complete.

Initial Difficulty

- First try: assume f forces $f \equiv_{\mathbf{ned}} X$ for some $X \in 2^\omega$ and derive a contradiction.
  - Forcing requires some kind of definability. Can we assume such an X is definable from f?

Question

If $f \equiv_{\mathbf{ned}} X \in 2^\omega$ must there be $Y \in 2^\omega$ with Y hyperarithmetic in f and $f \equiv_{\mathbf{ned}} Y$?

- f can compute some $X^{\square S}$. Can't we just fill in some random junk on S?
  - That junk might cause reductions in other direction to guess wrong (infinitely often).
- Can we search for a way to fill in values on S that preserves reductions?
  - Naive approach starts looking Σ^1_1 complete.

Strategy

Generate contradiction from assumption that $X_e \stackrel{\text{def}}{=} \Phi_e(f) \in \{0, 1, \square\}^\omega$ is an EDD of some $X \equiv_{\mathbf{ned}} f$.

Goal

Produce $S, U \in$ ▯ s.t. for all e, i, j the following are incompatible

1. $X_e \in \{0, 1, \square\}^\omega$ and $\bar{\rho}\left(\left\{z \mid X_e(z) = \square\right\}\right) = 0$
2. $\Phi_i\left(f^{\square S}\right) \cong^0_\square X_e$.
3. $\Phi_j\left(X_e^{\square U}\right) \cong^0_\square f$

If we had $f \equiv_{\mathbf{ned}} X$ for some X then:

1. f is an EDD of f so for some e, X_e must be (forced to satisfy) (1).
2. $f^{\square S}$ is also an EDD of f and all EDDs of X must be compatible.
3. ▯ is an ideal so $X_e^{\square U}$ is an EDD of X and must compute an EDD of f.

Strategy

Generate contradiction from assumption that $X_e \stackrel{\text{def}}{=} \Phi_e(f) \in \{0, 1, \square\}^\omega$ is an EDD of some $X \equiv_{\mathbf{ned}} f$.

Goal

Produce $S, U \in$ ▯ s.t. for all e, i, j the following are incompatible

1. $X_e \in \{0, 1, \square\}^\omega$ and $\bar{\rho}\left(\left\{z \mid X_e(z) = \square\right\}\right) = 0$
2. $\Phi_i\left(f^{\square S}\right) \cong^0_\square X_e$.
3. $\Phi_j\left(X_e^{\square U}\right) \cong^0_\square f$

If we had $f \equiv_{\mathbf{ned}} X$ for some X then:

1. f is an EDD of f so for some e, X_e must be (forced to satisfy) (1).
2. $f^{\square S}$ is also an EDD of f and all EDDs of X must be compatible.
3. ▯ is an ideal so $X_e^{\square U}$ is an EDD of X and must compute an EDD of f.

Intuition

Building S, U trying to ensure either X_e or $\Phi_i(f^{\square S})$ couldn't be an EDD of some X or $\Phi_j(X_e^{\square U})$ isn't an EDD of f.

- Threaten to place x into S. Let V_x be the set $\Phi_i(f^{\square S})$ would return $\square$ on in response.
  - Must have $V_x \in \mathbb{I}$ or we add x to S and win.
  - Note: If there is any possible change to $f(x)$ that causes $X_e(z)$ to change then z must be in V_x or we make $\Phi_i(f^{\square S})$ guess wrong (disagree with X_e).
- Find y, z such that $f(x) = y$ or $f(x) = z$ yield same $X_e\restriction_l$ for some really large l.
  - Use pigeonhole as $X_e\restriction_l \in \{0, 1, \square\}^l$ so finitely many choices.
- Add $V_x\restriction_{\geq l}$ to U picking l large enough to keep $U \in \mathbb{I}$
  - If we ever have $\Phi_j(X_e^{\square U}; x) \neq \square$ set $f(x)$ to whichever of y or z disagrees since $V_x\restriction_{\geq l}$ hides any differences.
  - If we always have $\Phi_j(X_e^{\square U}; x) = \square$ repeat this process for a set of x with positive upper density.

Intuition

Building S, U trying to ensure either X_e or $\Phi_i\left(f^{\square S}\right)$ couldn't be an EDD of some X or $\Phi_j\left(X_e^{\square U}\right)$ isn't an EDD of f.

- Threaten to place x into S. Let V_x be the set $\Phi_i\left(f^{\square S}\right)$ would return $\square$ on in response.
  - Must have $V_x \in \mathbb{I}$ or we add x to S and win.
  - Note: If there is any possible change to $f(x)$ that causes $X_e(z)$ to change then z must be in V_x or we make $\Phi_i\left(f^{\square S}\right)$ guess wrong (disagree with X_e).
- Find y, z such that $f(x) = y$ or $f(x) = z$ yield same $X_e\restriction_l$ for some really large l.
  - Use pigeonhole as $X_e\restriction_l \in \{0, 1, \square\}^l$ so finitely many choices.
- Add $V_x\restriction_{\geq l}$ to U picking l large enough to keep $U \in \mathbb{I}$
  - If we ever have $\Phi_j\left(X_e^{\square U}; x\right) \neq \square$ set $f(x)$ to whichever of y or z disagrees since $V_x\restriction_{\geq l}$ hides any differences.
  - If we always have $\Phi_j\left(X_e^{\square U}; x\right) = \square$ repeat this process for a set of x with positive upper density.

Intuition

Building S, U trying to ensure either X_e or $\Phi_i\left(f^{\square S}\right)$ couldn't be an EDD of some X or $\Phi_j\left(X_e^{\square U}\right)$ isn't an EDD of f.

- Threaten to place x into S. Let V_x be the set $\Phi_i\left(f^{\square S}\right)$ would return $\square$ on in response.
  - Must have $V_x \in \mathbb{I}$ or we add x to S and win.
  - Note: If there is any possible change to $f(x)$ that causes $X_e(z)$ to change then z must be in V_x or we make $\Phi_i\left(f^{\square S}\right)$ guess wrong (disagree with X_e).
- Find y, z such that $f(x) = y$ or $f(x) = z$ yield same $X_e \restriction_l$ for some really large l.
  - Use pigeonhole as $X_e \restriction_l \in \{0, 1, \square\}^l$ so finitely many choices.
- Add $V_x \restriction_{\geq l}$ to U picking l large enough to keep $U \in \mathbb{I}$
  - If we ever have $\Phi_j\left(X_e^{\square U}; x\right) \neq \square$ set $f(x)$ to whichever of y or z disagrees since $V_x \restriction_{\geq l}$ hides any differences.
  - If we always have $\Phi_j\left(X_e^{\square U}; x\right) = \square$ repeat this process for a set of x with positive upper density.

Density Lemma

Lemma

If $V_x \in$ ▯ for all x then there is a function $l(x)$ such that

$$\bigcup_{x\in\omega} V_x\restriction_{[l(x),\infty)} \in \text{▯}$$

Shows we can pick l large enough to keep $U \in$ ▯

Proof.

- Pick $l(x)$ large enough that $\bar{\rho}_0\left(V_x\restriction_{[l(x),\infty)}\right) \leq 2^{-x-1}$.
- Contribution from $\bigcup_{x>i} V_x\restriction_{[l(x),\infty)}$ is at most 2^{-i-1}.
- Eventually contribution from V_x with $x \leq i$ is below 2^{-i-1}.
- So we can push upper density as low as we want.

□

Density Lemma

Lemma

If $V_x \in$ ▯ for all x then there is a function $l(x)$ such that

$$\bigcup_{x\in\omega} V_x\restriction_{[l(x),\infty)} \in \text{▯}$$

Shows we can pick l large enough to keep $U \in$ ▯

Proof.

- Pick $l(x)$ large enough that $\bar{\rho}_0\left(V_x\restriction_{[l(x),\infty)}\right) \leq 2^{-x-1}$.
- Contribution from $\bigcup_{x>i} V_x\restriction_{[l(x),\infty)}$ is at most 2^{-i-1}.
- Eventually contribution from V_x with $x \leq i$ is below 2^{-i-1}.
- So we can push upper density as low as we want.

∎

More Genericity

- How do we deal with potentially partial computations $\Phi_i\left(f^{\Box S}\right)$?
  - What if $\Phi_i\left(f^{\Box S}\right)$ is only partial when we threaten to change S?
- Make S a generic element of ▯

Definition

A set is generic for ▯ if it is generic with respect to conditions of the form (σ, k) with $\sigma \in 2^{<\omega}$ and $k \in \omega$ where

$$S \leq_{▯} (\sigma, k) \iff \sigma \prec S \land \bar{\rho}_{|\sigma|}(S) \leq 2^{-k}$$

$$(\sigma', k') \leq_{▯} (\sigma, k) \iff \sigma \prec \sigma' \land k' \geq k \land \bar{\rho}_{|\sigma|}\left(\sigma'\right) \leq 2^{-k}$$

- In other words, Cohen conditions plus a bound on how large subsequent density can get.
- Filters without a maximum value for k — such as all slightly generic filters — correspond to sets in ▯

More Genericity

- How do we deal with potentially partial computations $\Phi_i\left(f^{\square S}\right)$?
  - What if $\Phi_i\left(f^{\square S}\right)$ is only partial when we threaten to change S?
- Make S a generic element of 𝕀

Definition

A set is generic for 𝕀 if it is generic with respect to conditions of the form (σ, k) with $\sigma \in 2^{<\omega}$ and $k \in \omega$ where

$$S \leq_{𝕀} (\sigma, k) \iff \sigma \prec S \wedge \bar{\rho}_{|\sigma|}(S) \leq 2^{-k}$$

$$(\sigma', k') \leq_{𝕀} (\sigma, k) \iff \sigma \prec \sigma' \wedge k' \geq k \wedge \bar{\rho}_{|\sigma|}\left(\sigma'\right) \leq 2^{-k}$$

- In other words, Cohen conditions plus a bound on how large subsequent density can get.
- Filters without a maximum value for k — such as all slightly generic filters — correspond to sets in 𝕀

More Genericity

- How do we deal with potentially partial computations $\Phi_i\left(f^{\square S}\right)$?
  - What if $\Phi_i\left(f^{\square S}\right)$ is only partial when we threaten to change S?
- Make S a generic element of ▯

Definition

A set is generic for ▯ if it is generic with respect to conditions of the form (σ, k) with $\sigma \in 2^{<\omega}$ and $k \in \omega$ where

$$S \leq_{▯} (\sigma, k) \iff \sigma \prec S \wedge \bar{\rho}_{|\sigma|}(S) \leq 2^{-k}$$

$$(\sigma', k') \leq_{▯} (\sigma, k) \iff \sigma \prec \sigma' \wedge k' \geq k \wedge \bar{\rho}_{|\sigma|}\left(\sigma'\right) \leq 2^{-k}$$

- In other words, Cohen conditions plus a bound on how large subsequent density can get.
- Filters without a maximum value for k — such as all slightly generic filters — correspond to sets in ▯

Benefits of Forcing for S

- Work above condition $p = (\sigma, k)$ forcing $\Phi_i\left(f^{\square S}\right)$ to be total, $\bar{\rho}\left(\left\{z \mid \Phi_i\left(f^{\square S}; z\right) = \square\right\}\right) = 0$ and compatible with X_e.
- Above p we can repeatedly threaten to add a clump of locations with upper density 2^{-k-1} to S with $V_x \in \mathbb{I}$ for all x in clump.
  - Conditions above p can't contradict what p forces.
- Clumps don't actually enter S — would violate $S \in \mathbb{I}$ — they become locations with $\Phi_j\left(X_e^{\square U}; x\right) = \square$.
- Recall, this requires placing the $f \oplus S$ computable tails $V_x\restriction_{>l}$ into U while ensuring $U \in \mathbb{I}$.
  - U gets built like S except we can also commit to adding finitely many $f \oplus S$ computable sets in $\mathbb{I}$ into U.
  - Fair bit of technical annoyance, e.g, using product forcing to consider extending conditions for f, S and U simultaneously.

Benefits of Forcing for S

- Work above condition $p = (\sigma, k)$ forcing $\Phi_i\left(f^{\square S}\right)$ to be total, $\bar{\rho}\left(\left\{z \mid \Phi_i\left(f^{\square S}; z\right) = \square\right\}\right) = 0$ and compatible with X_e.
- Above p we can repeatedly threaten to add a clump of locations with upper density 2^{-k-1} to S with $V_x \in$ ▯ for all x in clump.
    - Conditions above p can't contradict what p forces.
- Clumps don't actually enter S — would violate $S \in$ ▯ — they become locations with $\Phi_j\left(X_e^{\square U}; x\right) = \square$.
- Recall, this requires placing the $f \oplus S$ computable tails $V_x\restriction_{>l}$ into U while ensuring $U \in$ ▯.
    - U gets built like S except we can also commit to adding finitely many $f \oplus S$ computable sets in ▯ into U.
    - Fair bit of technical annoyance, e.g, using product forcing to consider extending conditions for f, S and U simultaneously.

Benefits of Forcing for S

- Work above condition $p = (\sigma, k)$ forcing $\Phi_i\left(f^{\Box S}\right)$ to be total, $\bar{\rho}\left(\left\{z \mid \Phi_i\left(f^{\Box S}; z\right) = \Box\right\}\right) = 0$ and compatible with X_e.
- Above p we can repeatedly threaten to add a clump of locations with upper density 2^{-k-1} to S with $V_x \in \mathbb{I}$ for all x in clump.
  - Conditions above p can't contradict what p forces.
- Clumps don't actually enter S — would violate $S \in \mathbb{I}$ — they become locations with $\Phi_j\left(X_e^{\Box U}; x\right) = \Box$.
- Recall, this requires placing the $f \oplus S$ computable tails $V_x\restriction_{>l}$ into U while ensuring $U \in \mathbb{I}$.
  - U gets built like S except we can also commit to adding finitely many $f \oplus S$ computable sets in $\mathbb{I}$ into U.
  - Fair bit of technical annoyance, e.g, using product forcing to consider extending conditions for f, S and U simultaneously.

Benefits of Forcing for S

- Work above condition $p = (\sigma, k)$ forcing $\Phi_i\left(f^{\square S}\right)$ to be total, $\bar{\rho}\left(\left\{z \mid \Phi_i\left(f^{\square S}; z\right) = \square\right\}\right) = 0$ and compatible with X_e.
- Above p we can repeatedly threaten to add a clump of locations with upper density 2^{-k-1} to S with $V_x \in \mathbb{I}$ for all x in clump.
  - Conditions above p can't contradict what p forces.
- Clumps don't actually enter S — would violate $S \in \mathbb{I}$ — they become locations with $\Phi_j\left(X_e^{\square U}; x\right) = \square$.
- Recall, this requires placing the $f \oplus S$ computable tails $V_x\restriction_{>l}$ into U while ensuring $U \in \mathbb{I}$.
  - U gets built like S except we can also commit to adding finitely many $f \oplus S$ computable sets in $\mathbb{I}$ into U.
  - Fair bit of technical annoyance, e.g, using product forcing to consider extending conditions for f, S and U simultaneously.

Benefits of Forcing for S

- Work above condition $p = (\sigma, k)$ forcing $\Phi_i\left(f^{\square S}\right)$ to be total, $\bar{\rho}\left(\left\{z \mid \Phi_i\left(f^{\square S}; z\right) = \square\right\}\right) = 0$ and compatible with X_e.
- Above p we can repeatedly threaten to add a clump of locations with upper density 2^{-k-1} to S with $V_x \in \mathbb{I}$ for all x in clump.
  - Conditions above p can't contradict what p forces.
- Clumps don't actually enter S — would violate $S \in \mathbb{I}$ — they become locations with $\Phi_j\left(X_e^{\square U}; x\right) = \square$.
- Recall, this requires placing the $f \oplus S$ computable tails $V_x\restriction_{>l}$ into U while ensuring $U \in \mathbb{I}$.
  - U gets built like S except we can also commit to adding finitely many $f \oplus S$ computable sets in $\mathbb{I}$ into U.
  - Fair bit of technical annoyance, e.g, using product forcing to consider extending conditions for f, S and U simultaneously.

Outline

Yes, Uniformly Yes

Does every coarse degree contain a set?

Theorem

Every uniform (and therefore non-uniform) coarse degree contains a set.

Forcing considerations suggest that the theorem can only be true uniformly.

Proposition

There are computable functionals $\Gamma : \omega^\omega \mapsto 2^\omega$ and $\Gamma^\dagger$ such that for all $f \in \omega^\omega$

- $\Gamma(f)$ *is total and* $\Gamma^\dagger \circ \Gamma(f) = \mathrm{id}$
- Γ *is a uniform course reduction of* $\Gamma(f)$ *to* f *.*
- $\Gamma^\dagger$ *is a uniform coarse reduction of* f *to* $\Gamma(f)$

Yes, Uniformly Yes

Does every coarse degree contain a set?

Theorem

Every uniform (and therefore non-uniform) coarse degree contains a set.

Forcing considerations suggest that the theorem can only be true uniformly.

Proposition

There are computable functionals $\Gamma : \omega^\omega \mapsto 2^\omega$ and $\Gamma^\dagger$ such that for all $f \in \omega^\omega$

- $\Gamma(f)$ *is total and* $\Gamma^\dagger \circ \Gamma(f) = \mathrm{id}$
- Γ *is a uniform course reduction of* $\Gamma(f)$ *to* f *.*
- $\Gamma^\dagger$ *is a uniform coarse reduction of* f *to* $\Gamma(f)$

Yes, Uniformly Yes

Does every coarse degree contain a set?

Theorem

Every uniform (and therefore non-uniform) coarse degree contains a set.

Forcing considerations suggest that the theorem can only be true uniformly.

Proposition

There are computable functionals $\Gamma : \omega^\omega \mapsto 2^\omega$ *and* $\Gamma^\dagger$ *such that for all* $f \in \omega^\omega$

- $\Gamma(f)$ *is total and* $\Gamma^\dagger \circ \Gamma(f) = \mathrm{id}$
- Γ *is a uniform course reduction of* $\Gamma(f)$ *to* f *.*
- $\Gamma^\dagger$ *is a uniform coarse reduction of* f *to* $\Gamma(f)$

Naive Attempt

Try partitioning ω into countably many 'columns' and encode $f(x)$ into $\omega^{[x]}$ *independently* of other columns.

- Fails for same reason some effective dense degrees don't contain sets.
  - Morally, $\lim_{x \to \infty} \bar{\rho}\left(\omega^{[=x]}\right) = 0$
    - Think of this as the part of column which actually gets used.
  - Same pigeonhole argument as above lets us pick values y, z for $f(x)$ which agree on a long initial segment of $\omega^{[x]}$.
  - By delaying disagreement we can build $f, \hat{f}$ disagree everywhere but $\Gamma(f) \,\Delta\, \Gamma(\hat{f}) \in \mathbb{I}$
- Lesson is that Γ can't encode $f(x)$ separately from $f(x')$.
- Real challenge is to encode f in a way that codes changes in $f(x)$ and $f(x')$ about as close together as x and x'.

Naive Attempt

Try partitioning ω into countably many 'columns' and encode $f(x)$ into $\omega^{[x]}$ *independently* of other columns.

- Fails for same reason some effective dense degrees don't contain sets.
  - Morally, $\lim\limits_{x \to \infty} \bar{\rho}\left(\omega^{[=x]}\right) = 0$
    - Think of this as the part of column which actually gets used.
  - Same pigeonhole argument as above lets us pick values y, z for $f(x)$ which agree on a long initial segment of $\omega^{[x]}$.
  - By delaying disagreement we can build $f, \hat{f}$ disagree everywhere but $\Gamma(f) \Delta \Gamma(\hat{f}) \in$ ▯
- Lesson is that Γ can't encode $f(x)$ separately from $f(x')$.
- Real challenge is to encode f in a way that codes changes in $f(x)$ and $f(x')$ about as close together as x and x'.

Naive Attempt

Try partitioning ω into countably many 'columns' and encode $f(x)$ into $\omega^{[x]}$ *independently* of other columns.

- Fails for same reason some effective dense degrees don't contain sets.
  - Morally, $\lim_{x\to\infty} \bar{\rho}\left(\omega^{[=x]}\right) = 0$
    - Think of this as the part of column which actually gets used.
  - Same pigeonhole argument as above lets us pick values y, z for $f(x)$ which agree on a long initial segment of $\omega^{[x]}$.
  - By delaying disagreement we can build $f, \hat{f}$ disagree everywhere but $\Gamma(f) \,\Delta\, \Gamma(\hat{f}) \in \mathbb{I}$
- Lesson is that Γ can't encode $f(x)$ separately from $f(x')$.
- Real challenge is to encode f in a way that codes changes in $f(x)$ and $f(x')$ about as close together as x and x'.

Naive Attempt

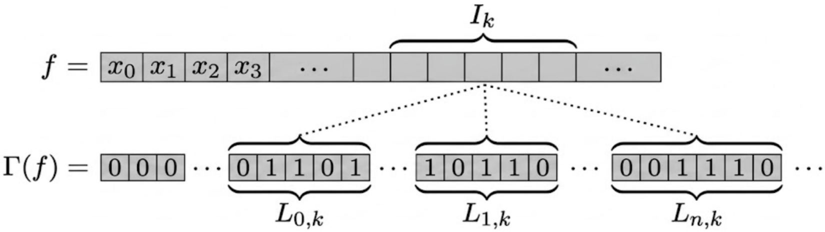
Try partitioning ω into countably many 'columns' and encode $f(x)$ into $\omega^{[x]}$ *independently* of other columns.

- Fails for same reason some effective dense degrees don't contain sets.
  - Morally, $\lim_{x \to \infty} \bar{\rho}\left(\omega^{[=x]}\right) = 0$
    - Think of this as the part of column which actually gets used.
  - Same pigeonhole argument as above lets us pick values y, z for $f(x)$ which agree on a long initial segment of $\omega^{[x]}$.
  - By delaying disagreement we can build $f, \hat{f}$ disagree everywhere but $\Gamma(f) \,\Delta\, \Gamma(\hat{f}) \in \mathbb{I}$
- Lesson is that Γ can't encode $f(x)$ separately from $f(x')$.
- Real challenge is to encode f in a way that codes changes in $f(x)$ and $f(x')$ about as close together as x and x'.

Chunking Up

We can encode sufficiently long chunks of f independently.

- Encode $f\restriction_{I_k}$ where $I_k = [2^k, 2^{k+1})$ into intervals $L_{k,n}$. Separate with long intervals of 0s to avoid interactions.

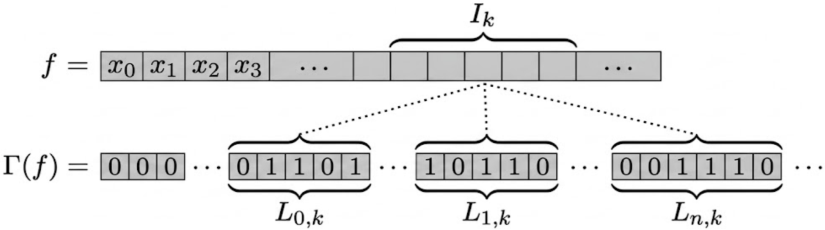


- Define $\Gamma(f)$ on $L_{n,k}$ to be enough copies of some binary string $\mathfrak{e}_n(f\restriction_{I_k})$ to have same effect on upper density independent of n.

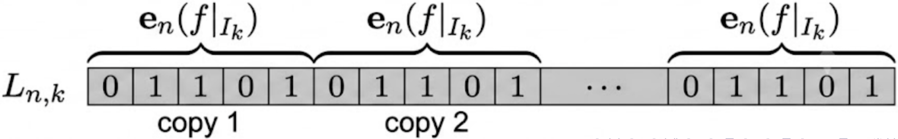
Chunking Up

We can encode sufficiently long chunks of f independently.

- Encode $f\restriction_{I_k}$ where $I_k = [2^k, 2^{k+1})$ into intervals $L_{k,n}$. Separate with long intervals of 0s to avoid interactions.



- Define $\Gamma(f)$ on $L_{n,k}$ to be enough copies of some binary string $\mathfrak{e}_n(f\restriction_{I_k})$ to have same effect on upper density independent of n.



Combinatorial Core

- Define computable function taking $\gamma \in \omega^{(2^k)}$ (representing $f\restriction_{I_k}$) to a binary string $\mathfrak{e}_n(\gamma) \in 2^{(2^k)}$ for each n.
- For each n, $\mathfrak{e}_n(\gamma)$ represents a chance to match $|\gamma \,\Delta\, \hat{\gamma}|$ with $|\mathfrak{e}_n(\gamma) \,\Delta\, \mathfrak{e}_n(\hat{\gamma})|$
  - If we change γ on m locations we want there to be some n for which we change $\mathfrak{e}_n(\gamma)$ on $O(m)$ locations and vice versa.
  - Specifically, we want

$$|\gamma \,\Delta\, \hat{\gamma}| \asymp \sup_{n\in\omega} |\mathfrak{e}_n(\gamma) \,\Delta\, \mathfrak{e}_n(\hat{\gamma})|$$

- Also for all γ we want $(\exists r)(\forall i > r)\big(\mathfrak{e}_i(\gamma) = \varnothing\big)$ (i.e. 0 string)
  - Ensures finite changes to f result in finite changes to $\Gamma(f)$
  - Will allow us to build $\Gamma^\dagger$ by searching for γ that gives closest match once we see $\mathfrak{e}_i(\gamma) = \varnothing$ for long enough.

Combinatorial Core

- Define computable function taking $\gamma \in \omega^{(2^k)}$ (representing $f\restriction_{I_k}$) to a binary string $\mathfrak{e}_n(\gamma) \in 2^{(2^k)}$ for each n.
- For each n, $\mathfrak{e}_n(\gamma)$ represents a chance to match $|\gamma \mathbin{\Delta} \hat{\gamma}|$ with $|\mathfrak{e}_n(\gamma) \mathbin{\Delta} \mathfrak{e}_n(\hat{\gamma})|$
  - If we change γ on m locations we want there to be some n for which we change $\mathfrak{e}_n(\gamma)$ on $O(m)$ locations and vice versa.
  - Specifically, we want

$$|\gamma \mathbin{\Delta} \hat{\gamma}| \asymp \sup_{n \in \omega} |\mathfrak{e}_n(\gamma) \mathbin{\Delta} \mathfrak{e}_n(\hat{\gamma})|$$

- Also for all γ we want $(\exists r)(\forall i > r)\big(\mathfrak{e}_i(\gamma) = \varnothing\big)$ (i.e. 0 string)
  - Ensures finite changes to f result in finite changes to $\Gamma(f)$
  - Will allow us to build $\Gamma^{\dagger}$ by searching for γ that gives closest match once we see $\mathfrak{e}_i(\gamma) = \varnothing$ for long enough.

Desiderata

Goal

1. $\mathfrak{e}_n(\gamma)\downarrow \in 2^{|\gamma|}$ and is a computable function of $n \in \omega, \gamma \in \omega^{(2^k)}$.
2. $(\exists r)(\forall i > r)\big(\mathfrak{e}_i(\gamma) = \varnothing\big)$ (i.e. 0 string)
3. $|\gamma \,\Delta\, \zeta| = d \implies \left\lceil \frac{d}{2} \right\rceil \leq \sup_{n\in\omega} |\mathfrak{e}_n(\gamma) \,\Delta\, \mathfrak{e}_n(\zeta)| \leq d$

- 2 will ensure that finite changes to f result in finite changes to $\Gamma(f)$ — slightly stronger than strictly necessary.
- 3 ensures changes remain similarly close.
  - Full proof requires some more properties, e.g., ability to computably recover a guess at r from k and a coarse descriptions of $\Gamma(f)$, which our solution will satisfy.

Non-Solutions

Problem might seem easy so assume that $\gamma, \zeta \in \omega^{(2^k)}$ and try:

- $\mathfrak{e}_n(\gamma; x) = 1 \iff \gamma(x) > n$
  - Let $\gamma(x) = x$ and $\hat{\gamma}(x) = x + 1$, disagree everywhere but $|\mathfrak{e}_n(\gamma) \,\Delta\, \mathfrak{e}_n(\hat{\gamma})| \leq 1$
- $\mathfrak{e}_{\ulcorner \zeta \urcorner}(\gamma; x) = 1$ iff $\gamma(x) > \zeta(x)$.
  - Gets disagreement size right (take $\zeta(x) = \min \gamma(x), \hat{\gamma}(x)$).
  - Fails finite effect property, since if $\gamma(0) > 0$ there are infinitely many strings ζ with $\gamma(0) > \zeta(0)$.
- $\mathfrak{e}_{\ulcorner \zeta \urcorner}(\gamma; x) = 1$ iff $\gamma(x) > \zeta(x)$ and $\max_x \gamma(x) > \max_x \zeta(x)$
  - Now $\mathfrak{e}_n(\gamma) = \varnothing$ for almost all n.
  - But changing γ at one location can now change $\mathfrak{e}_n(\gamma)$ everywhere.

Non-Solutions

Problem might seem easy so assume that $\gamma, \zeta \in \omega^{(2^k)}$ and try:

- $\mathfrak{e}_n(\gamma; x) = 1 \iff \gamma(x) > n$
  - Let $\gamma(x) = x$ and $\hat{\gamma}(x) = x + 1$, disagree everywhere but $|\mathfrak{e}_n(\gamma) \,\Delta\, \mathfrak{e}_n(\hat{\gamma})| \leq 1$
- $\mathfrak{e}_{\ulcorner \zeta \urcorner}(\gamma; x) = 1$ iff $\gamma(x) > \zeta(x)$.
  - Gets disagreement size right (take $\zeta(x) = \min \gamma(x), \hat{\gamma}(x)$).
  - Fails finite effect property, since if $\gamma(0) > 0$ there are infinitely many strings ζ with $\gamma(0) > \zeta(0)$.
- $\mathfrak{e}_{\ulcorner \zeta \urcorner}(\gamma; x) = 1$ iff $\gamma(x) > \zeta(x)$ and $\max_x \gamma(x) > \max_x \zeta(x)$
  - Now $\mathfrak{e}_n(\gamma) = \varnothing$ for almost all n.
  - But changing γ at one location can now change $\mathfrak{e}_n(\gamma)$ everywhere.

Non-Solutions

Problem might seem easy so assume that $\gamma, \zeta \in \omega^{(2^k)}$ and try:

- $\mathfrak{e}_n(\gamma; x) = 1 \iff \gamma(x) > n$
  - Let $\gamma(x) = x$ and $\hat{\gamma}(x) = x + 1$, disagree everywhere but $|\mathfrak{e}_n(\gamma) \,\Delta\, \mathfrak{e}_n(\hat{\gamma})| \leq 1$
- $\mathfrak{e}_{\ulcorner \zeta \urcorner}(\gamma; x) = 1$ iff $\gamma(x) > \zeta(x)$.
  - Gets disagreement size right (take $\zeta(x) = \min \gamma(x), \hat{\gamma}(x)$).
  - Fails finite effect property, since if $\gamma(0) > 0$ there are infinitely many strings ζ with $\gamma(0) > \zeta(0)$.
- $\mathfrak{e}_{\ulcorner \zeta \urcorner}(\gamma; x) = 1$ iff $\gamma(x) > \zeta(x)$ and $\max_x \gamma(x) > \max_x \zeta(x)$
  - Now $\mathfrak{e}_n(\gamma) = \varnothing$ for almost all n.
  - But changing γ at one location can now change $\mathfrak{e}_n(\gamma)$ everywhere.

Idea

Let's fix our last attempt:

- $\mathfrak{e}_n(\gamma; x)$ still checks if $\gamma(x) > \zeta(x)$ for some ζ coded by n.
- Still want to ensure that if $\max_x \zeta(x) \geq \max_x \gamma(x)$ we return $\emptyset$ to ensure finite effect.
- How can we prevent the value of γ at a single location from affecting $\mathfrak{e}_n(\gamma)$ everywhere by changing $\max_x \gamma(x)$?

Idea

- Each $x < 2^k$ is assigned some number of certificates $y < 2^k$ (can't be reused).
- $\mathfrak{e}_n(\gamma; x) = 1$ just when $\gamma(x) > \zeta(x)$ and x is assigned a certificate y with $\gamma(y) > \max_x \zeta(x)$ (a valid certificate).

- Non-reuse keeps any location in γ from affecting too many locations in $\mathfrak{e}_n(\gamma)$.
- We can afford to use two disagreements in our domain to produce a single disagreement.

Idea

Let's fix our last attempt:

- $\mathfrak{e}_n(\gamma; x)$ still checks if $\gamma(x) > \zeta(x)$ for some ζ coded by n.
- Still want to ensure that if $\max_x \zeta(x) \geq \max_x \gamma(x)$ we return $\emptyset$ to ensure finite effect.
- How can we prevent the value of γ at a single location from affecting $\mathfrak{e}_n(\gamma)$ everywhere by changing $\max_x \gamma(x)$?

Idea

- Each $x < 2^k$ is assigned some number of certificates $y < 2^k$ (can't be reused).
- $\mathfrak{e}_n(\gamma; x) = 1$ just when $\gamma(x) > \zeta(x)$ and x is assigned a certificate y with $\gamma(y) > \max_x \zeta(x)$ (a valid certificate).

- Non-reuse keeps any location in γ from affecting too many locations in $\mathfrak{e}_n(\gamma)$.
- We can afford to use two disagreements in our domain to produce a single disagreement.

Solution

Assume $\ulcorner(\zeta, q)\urcorner$ codes $\omega^{(2^k)} \times \left\{q \mid q : 2^k \mapsto 2^k \land q \text{ is idempotent}\right\}$ bijectively into ω — recall idempotence means $q(q(z)) = q(z)$.

Definition

For $x < 2^k$, $\mathfrak{e}_{\ulcorner(\zeta,q)\urcorner}(\gamma; x) = 1$ iff

$$\gamma(x) > \zeta(x) \land \left(\exists z < 2^k\right)\left(q(z) = x \land \gamma(z) > \max_{i<2^k} \zeta(i)\right)$$

and is otherwise 0.

- $q^{-1}(x)$ is the set of certificates for x.
- Idempotence ensures $\gamma(x)$ affects at most one location in $\mathfrak{e}_n(\gamma)$.
  - x must choose either to act as a certificate for itself or to certify some other location but go without certification itself (so $\mathfrak{e}_n(\gamma; x) = 0$).
    - If $q(x) = y$ then $q(z) = x$ would mean $q(q(z)) = y \neq x = q(y)$

Solution

Assume $\ulcorner(\zeta, q)\urcorner$ codes $\omega^{(2^k)} \times \left\{q \mid q : 2^k \mapsto 2^k \land q \text{ is idempotent}\right\}$ bijectively into ω — recall idempotence means $q(q(z)) = q(z)$.

Definition

For $x < 2^k$, $\mathfrak{e}_{\ulcorner(\zeta,q)\urcorner}(\gamma; x) = 1$ iff

$$\gamma(x) > \zeta(x) \land \left(\exists z < 2^k\right)\left(q(z) = x \land \gamma(z) > \max_{i<2^k} \zeta(i)\right)$$

and is otherwise 0.

- $q^{-1}(x)$ is the set of certificates for x.
- Idempotence ensures $\gamma(x)$ affects at most one location in $\mathfrak{e}_n(\gamma)$.
  - x must choose either to act as a certificate for itself or to certify some other location but go without certification itself (so $\mathfrak{e}_n(\gamma; x) = 0$).
    - If $q(x) = y$ then $q(z) = x$ would mean $q(q(z)) = y \neq x = q(y)$

Verification

Suppose $\ulcorner(\zeta, q)\urcorner = n$ and $\gamma, \hat{\gamma} \in \omega^{(2^k)}$.

- $\mathfrak{e}_n(\gamma) = \varnothing$ for almost all n .
  - Once $\max_x \zeta(x) > \max_x \gamma(x)$ we guaranteed $\mathfrak{e}_n(\gamma) = \varnothing$
  - Turns out that any coarse description of $\Gamma(f)$ will have a point after which most copies of $\mathfrak{e}_n(\gamma) = \varnothing$ and that eventually triggers $\Gamma^\dagger$ to make a good guess at γ.
- $|\mathfrak{e}_n(\gamma) \,\Delta\, \mathfrak{e}_n(\hat{\gamma})| \leq |\gamma \,\Delta\, \hat{\gamma}|$
  - Enough to show that image of $\gamma \,\Delta\, \hat{\gamma}$ under q contains $\mathfrak{e}_n(\gamma) \,\Delta\, \mathfrak{e}_n(\hat{\gamma})$.
  - Suppose $\mathfrak{e}_n(\gamma; x) \neq \mathfrak{e}_n(\hat{\gamma}; x)$ — WLOG $\mathfrak{e}_n(\gamma; x) = 1$ and $\mathfrak{e}_n(\hat{\gamma}; x) = 0$.
    - If both $\gamma, \hat{\gamma}$ have valid certificates for x then $q(x) = x$ and $\gamma(x) > \zeta(x) \geq \hat{\gamma}(x)$.
    - If only γ has a valid certificate z for x then $q(z) = x$ and $\gamma(z) > \max_x \zeta(x) \geq \hat{\gamma}(z)$.
    - Either way $\gamma(y) \neq \hat{\gamma}(y)$ for some y with $q(y) = x$

Verification (cont)

Given $\gamma, \hat{\gamma} \in \omega^{(2^k)}$. We produce $n = \ulcorner(\zeta, q)\urcorner$ with

$$|\mathfrak{e}_n(\gamma) \,\Delta\, \mathfrak{e}_n(\hat{\gamma})| \geq \frac{|\gamma \,\Delta\, \hat{\gamma}|}{2}$$

- Let $S = \gamma \,\Delta\, \hat{\gamma}$, $\min(x) = \min \gamma(x), \hat{\gamma}(x)$ and $\bar{z}$ be the median of $\min[S]$.
- Split S into two halves (morally), $H = \{x \mid x > \bar{z}\}$ and $L = \{x \mid x < \bar{z}\}$
  - For simplicity we assume $\bar{z} \notin \min[S]$, o/w need a third set E.
- We use elements of H as certificates for elements of L by defining q to send H injectively onto L — while preserving idempotence by being the identity on L.
- We ensure all those certificates are valid by setting $\zeta(x) = \min(x)$ on L and $\zeta(x) = 0$ everywhere else — so $\max_x \zeta(x) \leq \bar{z}$.
- If $x \in L$ then x gets a valid certificate and as exactly one of $\gamma(x) > \min(x)$ or $\hat{\gamma}(x) > \min(x)$ we have $\mathfrak{e}_n(\gamma; x) \neq \mathfrak{e}_n(\hat{\gamma}; x)$
- $|L| = \frac{|S|}{2} = \frac{|\gamma \Delta \hat{\gamma}|}{2}$

Outline

Contrasting Complexity

Theorem

The relation $f \geq_{\mathbf{ned}} X$ is Π^1_1 complete

- Kinda ugly explicit construction, see paper draft on arxiv.
- Complexity is unknown if
  - We replace $\geq_{\mathbf{ned}}$ with $\equiv_{\mathbf{ned}}$.
  - We replace $\geq_{\mathbf{ned}}$ with $\geq_{\mathbf{ued}}$.
  - We restrict our attention to sets.

Couldn't be more different with coarse reductions

Theorem

Both $f \geq_{\mathbf{nc}} g$ and $f \geq_{\mathbf{uc}} g$ are $\Sigma^0_4(f, g)$.

- Controlled by behavior on generic coarse descriptions of f (for a suitable notion of generic).

Contrasting Complexity

Theorem

The relation $f \geq_{\mathbf{ned}} X$ is Π^1_1 complete

- Kinda ugly explicit construction, see paper draft on arxiv.
- Complexity is unknown if
  - We replace $\geq_{\mathbf{ned}}$ with $\equiv_{\mathbf{ned}}$.
  - We replace $\geq_{\mathbf{ned}}$ with $\geq_{\mathbf{ued}}$.
  - We restrict our attention to sets.

Couldn't be more different with coarse reductions

Theorem

Both $f \geq_{\mathbf{nc}} g$ and $f \geq_{\mathbf{uc}} g$ are $\Sigma^0_4(f, g)$.

- Controlled by behavior on generic coarse descriptions of f (for a suitable notion of generic).

Contrasting Complexity

Theorem

The relation $f \geq_{\mathbf{ned}} X$ is Π^1_1 complete

- Kinda ugly explicit construction, see paper draft on arxiv.
- Complexity is unknown if
  - We replace $\geq_{\mathbf{ned}}$ with $\equiv_{\mathbf{ned}}$.
  - We replace $\geq_{\mathbf{ned}}$ with $\geq_{\mathbf{ued}}$.
  - We restrict our attention to sets.

Couldn't be more different with coarse reductions

Theorem

Both $f \geq_{\mathbf{nc}} g$ and $f \geq_{\mathbf{uc}} g$ are $\Sigma^0_4(f, g)$.

- Controlled by behavior on generic coarse descriptions of f (for a suitable notion of generic).

Generic Coarse Descriptions

Intuition: A generic coarse description of f is a generic set $S \in \mathbb{I}$ plus a Cohen generic specification of how to change f on S.

Definition

$\breve{f} \in \omega^\omega$ is a generic coarse description of f if there is some generic $S \in \mathbb{I}$ and some Cohen generic $g \in \omega^\omega$ with $\breve{f}\restriction_{\overline{S}} = f\restriction_{\overline{S}}$ and $\breve{f}$ is $g(k)$ on the k-th element of S. Conditions are pairs of S, g specifying the same length result.

- $\Vdash_f$ is forcing relation (note f relative).
- $f[q] \in \omega^{<\omega}$ is the result of applying the condition q to f.

- When f is clear we will talk about $\hat{f}$ being above p to mean that $\hat{f}$ could be built from f by changes that extend p, i.e., $\hat{f} = \lim_{i\to\infty} f[p_i]$ for some p_i.
- Note: If $p_{i+1} \leq p_i$ and density bounds don't stabilize then $f[p_\infty] \stackrel{\text{def}}{=} \lim_{i\to\infty} f[p_i]$ is a coarse description of f whether or not p_i is generic.

Generic Coarse Descriptions

Intuition: A generic coarse description of f is a generic set $S \in \mathbb{I}$ plus a Cohen generic specification of how to change f on S.

Definition

$\check{f} \in \omega^\omega$ is a generic coarse description of f if there is some generic $S \in \mathbb{I}$ and some Cohen generic $g \in \omega^\omega$ with $\check{f}\restriction_{\overline{S}} = f\restriction_{\overline{S}}$ and $\check{f}$ is $g(k)$ on the k-th element of S. Conditions are pairs of S, g specifying the same length result.

- $\Vdash_f$ is forcing relation (note f relative).
- $f[q] \in \omega^{<\omega}$ is the result of applying the condition q to f.

- When f is clear we will talk about $\hat{f}$ being above p to mean that $\hat{f}$ could be built from f by changes that extend p, i.e., $\hat{f} = \lim_{i\to\infty} f[p_i]$ for some p_i.
- Note: If $p_{i+1} \leq p_i$ and density bounds don't stabilize then $f[p_\infty] \stackrel{\text{def}}{=} \lim_{i\to\infty} f[p_i]$ is a coarse description of f whether or not p_i is generic.

Generic Coarse Descriptions

Intuition: A generic coarse description of f is a generic set $S \in \mathbb{I}$ plus a Cohen generic specification of how to change f on S.

Definition

$\check{f} \in \omega^\omega$ is a generic coarse description of f if there is some generic $S \in \mathbb{I}$ and some Cohen generic $g \in \omega^\omega$ with $\check{f} \restriction_{\overline{S}} = f \restriction_{\overline{S}}$ and $\check{f}$ is $g(k)$ on the k-th element of S. Conditions are pairs of S, g specifying the same length result.

- $\Vdash_f$ is forcing relation (note f relative).
- $f[q] \in \omega^{<\omega}$ is the result of applying the condition q to f.

- When f is clear we will talk about $\hat{f}$ being above p to mean that $\hat{f}$ could be built from f by changes that extend p, i.e., $\hat{f} = \lim_{i \to \infty} f[p_i]$ for some p_i.
- Note: If $p_{i+1} \leq p_i$ and density bounds don't stabilize then $f[p_\infty] \stackrel{\text{def}}{=} \lim_{i \to \infty} f[p_i]$ is a coarse description of f whether or not p_i is generic.

Generic Coarse Descriptions

Intuition: A generic coarse description of f is a generic set $S \in \mathbb{I}$ plus a Cohen generic specification of how to change f on S.

Definition

$\check{f} \in \omega^\omega$ is a generic coarse description of f if there is some generic $S \in \mathbb{I}$ and some Cohen generic $g \in \omega^\omega$ with $\check{f}\restriction_{\overline{S}} = f\restriction_{\overline{S}}$ and $\check{f}$ is $g(k)$ on the k-th element of S. Conditions are pairs of S, g specifying the same length result.

- $\Vdash_f$ is forcing relation (note f relative).
- $f[q] \in \omega^{<\omega}$ is the result of applying the condition q to f.

- When f is clear we will talk about $\hat{f}$ being above p to mean that $\hat{f}$ could be built from f by changes that extend p, i.e., $\hat{f} = \lim_{i\to\infty} f[p_i]$ for some p_i.
- Note: If $p_{i+1} \leq p_i$ and density bounds don't stabilize then $f[p_\infty] \stackrel{\text{def}}{=} \lim_{i\to\infty} f[p_i]$ is a coarse description of f whether or not p_i is generic.

Coarse Reductions Want To Be Forced

Proposition

$$f \geq_{\mathbf{nc}} g \iff (\exists e, q_0)\left[q_0 \Vdash_f \left(\Phi_e(\check{f}) \mathbin{\Delta} g\right) \in \Box\right]$$

$$f \geq_{\mathbf{uc}} g \iff q_0 \textit{ is the empty condition.}$$

- Forcing (sentences relative to $f \oplus g$) and counting quantifiers gives us theorem.
- $\implies$ direction is straightforward.
- $\impliedby$: Use e, q_0 and finite information about f to compute a coarse description of g from a coarse description $\hat{f}$ of f.
  - How much information about f is needed can depend on $\hat{f}$ — allowing for non-uniformity.

Coarse Reductions Want To Be Forced

Proposition

$$f \geq_{\mathbf{nc}} g \iff (\exists e, q_0)\left[q_0 \Vdash_f (\Phi_e(\check{f}) \,\Delta\, g) \in \mathbb{I}\right]$$

$$f \geq_{\mathbf{uc}} g \iff q_0 \textit{ is the empty condition.}$$

- Forcing (sentences relative to $f \oplus g$) and counting quantifiers gives us theorem.
- $\implies$ direction is straightforward.
- $\impliedby$: Use e, q_0 and finite information about f to compute a coarse description of g from a coarse description $\hat{f}$ of f.
  - How much information about f is needed can depend on $\hat{f}$ — allowing for non-uniformity.

Using Forcing To Compute

How does knowing $q_0 \Vdash_f \left(\Phi_e\left(\check{f}\right) \Delta\, g\right) \in \mathbb{I}$ help even f compute a coarse description of g?

- f is a coarse description of itself but it's not a generic one so we can't assume $\Phi_e(f)$ is a coarse description of g.

Strategy:

- Define a property — **without using** g — of a chain of conditions $p_i \leq q_0$ which ensures $\lim_{i \to \infty} \Phi_e\left(f\,[p_i]\right)$ must be a coarse description of g.
- Show that we can find $p_i = \lim_{s \to \infty} p_{i,s}$ satisfying property with $p_{i,s}$ f computable.
- Give algorithm for computing a coarse description of g from $p_{i,s}$.

Using Forcing To Compute

How does knowing $q_0 \Vdash_f \left(\Phi_e\left(\breve{f}\right) \,\Delta\, g\right) \in \square$ help even f compute a coarse description of g?

- f is a coarse description of itself but it's not a generic one so we can't assume $\Phi_e(f)$ is a coarse description of g.

Strategy:

- Define a property — **without using** g — of a chain of conditions $p_i \leq q_0$ which ensures $\lim_{i\to\infty} \Phi_e\left(f[p_i]\right)$ must be a coarse description of g.
- Show that we can find $p_i = \lim_{s\to\infty} p_{i,s}$ satisfying property with $p_{i,s}$ f computable.
- Give algorithm for computing a coarse description of g from $p_{i,s}$.

Cauchy In Density

Definition

p_i is an f-Cauchy sequence of conditions if

1. $p_i \leq q_0$.
2. $p_{i+1} \leq p_i$ and p_{i+1} has smaller density bound than p_i.
3. $\Phi_e\big(f[p_{i+1}]\big) \succsim \Phi_e\big(f[p_i]\big)$
4. $\big(\forall q, r \leq p_i\big)\big[\bar{\rho}_l\big(\Phi_e(f[q]) \,\Delta\, \Phi_e(f[r])\big) \leq 2^{-i}\big]$ where $l = |\Phi_e\big(p_i\big)|$

2. As density bounds don't stabilize, $f[p_\infty]$ is a coarse description of f.
3. Ensures $\Phi_e\big(f[p_\infty]\big)$ is total.
4. Ensures $\Phi_e\big(f[p_i]\big)$ converges in density to some function.

Cauchy In Density

Definition

p_i is an f-Cauchy sequence of conditions if

1. $p_i \leq q_0$.
2. $p_{i+1} \leq p_i$ and p_{i+1} has smaller density bound than p_i.
3. $\Phi_e\big(f[p_{i+1}]\big) \gneqq \Phi_e\big(f[p_i]\big)$
4. $\big(\forall q, r \leq p_i\big)\big[\bar{\rho}_l\big(\Phi_e(f[q]) \,\Delta\, \Phi_e(f[r])\big) \leq 2^{-i}\big]$ where $l = |\Phi_e\big(p_i\big)|$

2. As density bounds don't stabilize, $f[p_\infty]$ is a coarse description of f.
3. Ensures $\Phi_e\big(f[p_\infty]\big)$ is total.
4. Ensures $\Phi_e\big(f[p_i]\big)$ converges in density to some function.

Cauchy In Density

Definition

p_i is an f-Cauchy sequence of conditions if

1. $p_i \leq q_0$.
2. $p_{i+1} \leq p_i$ and p_{i+1} has smaller density bound than p_i.
3. $\Phi_e\big(f[p_{i+1}]\big) \succsim\!\!\!\!\!/ \; \Phi_e\big(f[p_i]\big)$
4. $(\forall q, r \leq p_i)\left[\bar{\rho}_l\big(\Phi_e(f[q]) \,\Delta\, \Phi_e(f[r])\big) \leq 2^{-i}\right]$ where $l = |\Phi_e(p_i)|$

2. As density bounds don't stabilize, $f[p_\infty]$ is a coarse description of f.
3. Ensures $\Phi_e\big(f[p_\infty]\big)$ is total.
4. Ensures $\Phi_e\big(f[p_i]\big)$ converges in density to some function.

Cauchy In Density

Definition

p_i is an f-Cauchy sequence of conditions if

1. $p_i \leq q_0$.
2. $p_{i+1} \leq p_i$ and p_{i+1} has smaller density bound than p_i.
3. $\Phi_e\big(f[p_{i+1}]\big) \succnsim \Phi_e\big(f[p_i]\big)$
4. $\big(\forall q, r \leq p_i\big)\big[\bar{\rho}_l\big(\Phi_e(f[q])\,\Delta\,\Phi_e(f[r])\big) \leq 2^{-i}\big]$ where $l = |\Phi_e\big(p_i\big)|$

2. As density bounds don't stabilize, $f[p_\infty]$ is a coarse description of f.
3. Ensures $\Phi_e\big(f[p_\infty]\big)$ is total.
4. Ensures $\Phi_e\big(f[p_i]\big)$ converges in density to some function.

Acting Generic

We note that even if the f-Cauchy sequence p_i isn't generic $\Phi_e\big(f[p_\infty]\big)$ is still a coarse description of g.

- Totality is clear so we just need to prove $\bar{\rho}\big(\Phi_e\big(f[p_\infty]\big) \,\Delta\, g\big) = 0$.
- To see that $\Phi_e\big(f[p_\infty]\big)$ and g don't differ on a set of upper density more than 2^{-w} consider a generic coarse description $\check{f}$ of f extending $p_{w+1} \leq q_0$.
  - $\Phi_e\big(f[p_\infty]\big)$ and $\Phi_e\big(\check{f}\big)$ can only differ on a set of upper density less than 2^{-w-1}.
  - q_0 forces $\Phi_e\big(\check{f}\big) \,\Delta\, g$ to have upper density 0.
  - Desired conclusion follows by noting the triangle inequality applies
- Since w is arbitrary we can conclude $\Phi_e\big(f[p_\infty]\big)$ is a coarse description of g.

Acting Generic

We note that even if the f-Cauchy sequence p_i isn't generic $\Phi_e\big(f[p_\infty]\big)$ is still a coarse description of g.

- Totality is clear so we just need to prove $\bar{\rho}\big(\Phi_e\big(f[p_\infty]\big) \,\Delta\, g\big) = 0$.
- To see that $\Phi_e\big(f[p_\infty]\big)$ and g don't differ on a set of upper density more than 2^{-w} consider a generic coarse description $\check{f}$ of f extending $p_{w+1} \leq q_0$.
  - $\Phi_e\big(f[p_\infty]\big)$ and $\Phi_e\big(\check{f}\big)$ can only differ on a set of upper density less than 2^{-w-1}.
  - q_0 forces $\Phi_e\big(\check{f}\big) \,\Delta\, g$ to have upper density 0.
  - Desired conclusion follows by noting the triangle inequality applies
- Since w is arbitrary we can conclude $\Phi_e\big(f[p_\infty]\big)$ is a coarse description of g.

Acting Generic

We note that even if the f-Cauchy sequence p_i isn't generic $\Phi_e\big(f[p_\infty]\big)$ is still a coarse description of g.

- Totality is clear so we just need to prove $\bar{\rho}\big(\Phi_e\big(f[p_\infty]\big) \Delta\, g\big) = 0$.
- To see that $\Phi_e\big(f[p_\infty]\big)$ and g don't differ on a set of upper density more than 2^{-w} consider a generic coarse description $\check{f}$ of f extending $p_{w+1} \leq q_0$.
  - $\Phi_e\big(f[p_\infty]\big)$ and $\Phi_e\big(\check{f}\big)$ can only differ on a set of upper density less than 2^{-w-1}.
  - q_0 forces $\Phi_e\big(\check{f}\big) \Delta\, g$ to have upper density 0.
  - Desired conclusion follows by noting the triangle inequality applies
- Since w is arbitrary we can conclude $\Phi_e\big(f[p_\infty]\big)$ is a coarse description of g.

Extending f-Cauchy sequences

We now show any partial f-Cauchy sequence can be extended — gives existence and definability.

- As q_0 forces totality we can always find an extension which grows $\Phi_e\big(f[p_i]\big)$. Other requirements trivial except for final density requirement.
- If p_n ends sequence then every $p_{n+1} \leq p_n$ is extended by conditions q, r pushing $\bar{\rho}_l\big(\Phi_e(f[q]) \,\Delta\, \Phi_e(f[r])\big)$ above 2^{-n-1}.
  - $l = |\Phi_e\big(f[p_{n+1}]\big)|$ and thus under our control.
- But then we could build generic coarse descriptions of f, $\check{f}_0, \check{f}_1$ above p_n with $\bar{\rho}\big(\Phi_e\big(\check{f}_0\big) \,\Delta\, \Phi_e\big(\check{f}_1\big)\big) > 0$.
  - Can't keep all extensions of a pair of conditions (u_0, u_1) close to each other if each u_i has extensions far from each other.
- Contradicts fact that $\bar{\rho}\big(\Phi_e\big(\check{f}_i\big) \,\Delta\, g\big) = 0$ for $i = 0, 1$.

Extending f-Cauchy sequences

We now show any partial f-Cauchy sequence can be extended — gives existence and definability.

- As q_0 forces totality we can always find an extension which grows $\Phi_e\big(f[p_i]\big)$. Other requirements trivial except for final density requirement.
- If p_n ends sequence then every $p_{n+1} \leq p_n$ is extended by conditions q, r pushing $\bar{\rho}_l\big(\Phi_e(f[q]) \,\Delta\, \Phi_e(f[r])\big)$ above 2^{-n-1}.
  - $l = |\Phi_e\big(f[p_{n+1}]\big)|$ and thus under our control.
- But then we could build generic coarse descriptions of f, $\breve{f}_0, \breve{f}_1$ above p_n with $\bar{\rho}\big(\Phi_e\big(\breve{f}_0\big) \,\Delta\, \Phi_e\big(\breve{f}_1\big)\big) > 0$.
  - Can't keep all extensions of a pair of conditions (u_0, u_1) close to each other if each u_i has extensions far from each other.
- Contradicts fact that $\bar{\rho}\big(\Phi_e\big(\breve{f}_i\big) \,\Delta\, g\big) = 0$ for $i = 0, 1$.

Extending f-Cauchy sequences

We now show any partial f-Cauchy sequence can be extended — gives existence and definability.

- As q_0 forces totality we can always find an extension which grows $\Phi_e\big(f[p_i]\big)$. Other requirements trivial except for final density requirement.
- If p_n ends sequence then every $p_{n+1} \leq p_n$ is extended by conditions q, r pushing $\bar{\rho}_l\big(\Phi_e(f[q]) \,\Delta\, \Phi_e(f[r])\big)$ above 2^{-n-1}.
  - $l = |\Phi_e\big(f[p_{n+1}]\big)|$ and thus under our control.
- But then we could build generic coarse descriptions of f, $\breve{f}_0, \breve{f}_1$ above p_n with $\bar{\rho}\big(\Phi_e\big(\breve{f}_0\big) \,\Delta\, \Phi_e\big(\breve{f}_1\big)\big) > 0$.
  - Can't keep all extensions of a pair of conditions (u_0, u_1) close to each other if each u_i has extensions far from each other.
- Contradicts fact that $\bar{\rho}\big(\Phi_e\big(\breve{f}_i\big) \,\Delta\, g\big) = 0$ for $i = 0, 1$.

Approximation

Recall/Observe

- The property $p_i, i \leq n$ is f-Cauchy is uniformly $\Pi^0_1(f, e, q_0)$.
- If $p_i, i \leq n$ is f-Cauchy it can be extended to a longer f-Cauchy sequence.
- If $p_i, i \in \omega$ is f-Cauchy then $\Phi_e\big(f[p_\infty]\big)$ is a coarse description of g.

Therefore, by the limit lemma, we have an f-computable sequence $p_{i,s}$ such that

- $\lim_{s\to\infty} p_{i,s} = p_i$ where p_i is f-Cauchy.
- $p_{i+1,s} \leq p_{i,s} \leq q_0$
- $|\Phi_e\big(f[p_{k,k}]\big)| \geq 2^{k+1}$
- $\Phi_e\big(f[p_\infty]\big)$ is a coarse description of g

The Algorithm

Let h be given by $\Phi_e\big(f[p_{k,k}]\big)$ on the interval $I_k = [2^k, 2^{k+1})$.

- $p_{i,s}$ may converge too slowly for h to be $\Phi_e\big(f[p_\infty]\big)$ but this is close enough.
- For almost all s, $p_{s,s} \leq p_i$
- Hence for almost all k, the upper density of $h \,\Delta\, \Phi_e\big(f[p_\infty]\big)$ on I_k must be less than 2^{-k}.
- Since $\Phi_e\big(f[p_\infty]\big)$ is a coarse description of g same triangle inequality argument implies h is a coarse description of g as well.

As h is clearly computable from f, f computes a coarse description of g.

The Algorithm

Let h be given by $\Phi_e\big(f[p_{k,k}]\big)$ on the interval $I_k = [2^k, 2^{k+1})$.

- $p_{i,s}$ may converge too slowly for h to be $\Phi_e\big(f[p_\infty]\big)$ but this is close enough.
- For almost all s, $p_{s,s} \leq p_i$
- Hence for almost all k, the upper density of $h \,\Delta\, \Phi_e\big(f[p_\infty]\big)$ on I_k must be less than 2^{-k}.
- Since $\Phi_e\big(f[p_\infty]\big)$ is a coarse description of g same triangle inequality argument implies h is a coarse description of g as well.

As h is clearly computable from f, f computes a coarse description of g.

Other Coarse Descriptions

How does $\hat{f}$, a coarse description of f compute a coarse description of g?

- Observe that any $\hat{f}$ condition $\hat{p}$ can be translated into an f condition p with $\hat{f}[\hat{p}] = f[p]$.
- Since $\hat{f}$ is a coarse description of f we can find some $\hat{p}_0$ such that all extensions of $\hat{p}_0$ translate to extensions of q_0.
- Straightforward — but annoying — bookkeeping reveals that every $\hat{f}$ Cauchy sequence extending $\hat{p}_0$ translates to an f-Cauchy sequence extending q_0.
- Since $f[p_\infty] = \hat{f}[\hat{p}_\infty]$ same algorithm allows $\hat{f}$ to compute coarse description of g

Other Coarse Descriptions

How does $\hat{f}$, a coarse description of f compute a coarse description of g?

- Observe that any $\hat{f}$ condition $\hat{p}$ can be translated into an f condition p with $\hat{f}[\hat{p}] = f[p]$.
- Since $\hat{f}$ is a coarse description of f we can find some $\hat{p}_0$ such that all extensions of $\hat{p}_0$ translate to extensions of q_0.
- Straightforward — but annoying — bookkeeping reveals that every $\hat{f}$ Cauchy sequence extending $\hat{p}_0$ translates to an f-Cauchy sequence extending q_0.
- Since $f[p_\infty] = \hat{f}[\hat{p}_\infty]$ same algorithm allows $\hat{f}$ to compute coarse description of g

Uniformity and Non-Uniformity

- Given q_0, e and $\hat{p}_0$ we can uniformly compute a coarse description of g from a coarse description $\hat{f}$ of f.
- But $\hat{p}_0$ needs to encode (most of) the difference between $\hat{f}$ and f until they commit to staying very close.
- Very close is (roughly) within the density bound required by q_0.
- If q_0 is the empty condition then $\hat{p}_0$ can be taken to be empty as well and the reduction is uniform.
- Otherwise, the reduction may non-uniformly depend on knowing a sufficiently long initial segment of f.

Outline

- I'd like to emphasize that robust computation is just a small sliver of the larger space of notions of the form: when does every description of X compute a description of Y.

Question

Is the reduction defined by $f \geq_{\Pi^0_1} g$ iff every tree $T \subset \omega^{<\omega}$ with $f \in [T]$ and $[T]$ countable (uniformly?) computes $\hat{T}$ with $[\hat{T}]$ countable and $g \in [\hat{T}]$ interesting?

  - $f \leq_{\Pi^0_1} g$ clearly implies that g is hyperarithmetic in f
  - Genericity and results by Harrington show that this notion differs both from hyperarithmetic reducibility and being a relative Π^0_1 singleton.
- I suspect there are some interesting notions hiding in this area.

Robust Computation Questions

Question (From [1])

Does every degree for (uniform or non-uniform) generic reducibility or dense reducibility contain a set?

- These notions understand computes in terms of enumeration reductions rather than Turing reductions.

Question

Do the effective dense or coarse degrees (uniform or non-uniform) always contain an element of least Turing degree?

- One of many questions you could ask relating these different degree structures.

Robust Computation Questions

Question (From [1])

Does every degree for (uniform or non-uniform) generic reducibility or dense reducibility contain a set?

- These notions understand computes in terms of enumeration reductions rather than Turing reductions.

Question

Do the effective dense or coarse degrees (uniform or non-uniform) always contain an element of least Turing degree?

- One of many questions you could ask relating these different degree structures.

More Questions

Question

What is the complexity of $f \equiv_{\mathbf{ned}} g$, $f \equiv_{\mathbf{ued}} g$ and $f \geq_{\mathbf{ued}} g$? What about on 2^ω? Is $Y \geq_{\mathbf{ned}} X$ still Π^1_1 complete for $X, Y \in 2^\omega$?

- I suspect the approach I used to show $f \geq_{\mathbf{ned}} X$ is Π^1_1 complete could be extended to 2^ω but I have no idea about the others — other than a vague intuition that $\equiv_{\mathbf{ned}}$ is probably simpler than $\geq_{\mathbf{ned}}$.

Question

Can we characterize the class of functions whose (non-uniform) effective dense degree contains a set? Can any function f not dominated by a computable function be equivalent to a set?

More Questions

Question

What is the complexity of $f \equiv_{\mathbf{ned}} g$, $f \equiv_{\mathbf{ued}} g$ and $f \geq_{\mathbf{ued}} g$? What about on 2^{ω}? Is $Y \geq_{\mathbf{ned}} X$ still Π^1_1 complete for $X, Y \in 2^{\omega}$?

- I suspect the approach I used to show $f \geq_{\mathbf{ned}} X$ is Π^1_1 complete could be extended to 2^{ω} but I have no idea about the others — other than a vague intuition that $\equiv_{\mathbf{ned}}$ is probably simpler than $\geq_{\mathbf{ned}}$.

Question

Can we characterize the class of functions whose (non-uniform) effective dense degree contains a set? Can any function f not dominated by a computable function be equivalent to a set?

Even More Questions

Question

If the non-uniform effective dense degree of f contains a set must it contain a set hyperarithmetic in f? What about arithmetic in f or even $\Delta^0_2(f)$?

- One way to attack this question would be to show that if every EDD of f computes something that looks like an EDD of some h we can always fill in the missing pieces.

Question

Suppose that g is a function that returns $\square$ on a set of density 0, every EDD of $f \in \omega^\omega$ computes $\hat{g} \cong^0_\square g$ and if $S \in \mathbb{I}$ then $g^{\square S}$ computes an EDD of f. Must there be some $h \in \omega^\omega$ with $h \cong^0_\square g$ and $f \equiv_{\mathbf{ned}} h$?

Even More Questions

Question

If the non-uniform effective dense degree of f contains a set must it contain a set hyperarithmetic in f? What about arithmetic in f or even $\Delta^0_2(f)$?

- One way to attack this question would be to show that if every EDD of f computes something that looks like an EDD of some h we can always fill in the missing pieces.

Question

Suppose that g is a function that returns $\square$ on a set of density 0, every EDD of $f \in \omega^\omega$ computes $\hat{g} \cong^0_\square g$ and if $S \in \mathbb{I}$ then $g^{\square S}$ computes an EDD of f. Must there be some $h \in \omega^\omega$ with $h \cong^0_\square g$ and $f \equiv_{\mathbf{ned}} h$?

References I

[1] Eric P. Astor, Denis R. Hirschfeldt, and Carl G. Jockusch. "Dense computability, upper cones, and minimal pairs". en. In: *Computability* 8.2 (2019), pp. 155–177. ISSN: 2211-3568, 2211-3576. DOI: 10.3233/COM-180231. URL: https://journals.sagepub.com/doi/10.3233/COM-180231.

[2] Damir D. Dzhafarov and Gregory Igusa. "Notions of robust information coding". EN. In: *Computability* 6.2 (2017), pp. 105–124. ISSN: 2211-3568. DOI: 10.3233/COM-160059. URL: https://doi.org/10.3233/COM-160059.