Logic(s) in the computable context

Noah Schweber (Proof School)

October 12, 2023

Motivation

Introducing the logics

Some results

A guilty coda: forcing

Why not just classical logic limited to computable structures

Classical logic is isomorphism invariant, but there are lots of interesting questions which are only *computable*-isomorphism-invariant.

Why not just classical logic limited to computable structures

Classical logic is isomorphism invariant, but there are lots of interesting questions which are only *computable*-isomorphism-invariant.
For instance:

Definition

A computable structure $\mathcal{A}$ has **computable dimension** *k (for $k \leq \omega + 1$) iff $\mathcal{A}$ has exactly k copies up to computable isomorphism.*

In a precise sense (McCoy), the only "natural" computable dimensions are 1 and ω.

Why not just classical logic limited to computable structures

Classical logic is isomorphism invariant, but there are lots of interesting questions which are only *computable*-isomorphism-invariant.
For instance:

Definition

A computable structure $\mathcal{A}$ has **computable dimension** *k (for $k \leq \omega + 1$) iff $\mathcal{A}$ has exactly k copies up to computable isomorphism.*

In a precise sense (McCoy), the only "natural" computable dimensions are 1 and ω. But computable-isomorphism-type counting is not the end of the story!

A hard question

- $(\omega; <)$ is "productive for $\cong_c$:" given any computable list of copies of $(\omega; <)$ we can computably biuild a new copy not $\cong_c$ any on the list.
- There are c.d.∞ structures for which no infinite collection of pairwise-$\not\cong_c$ copies exists. (Turetsky)
- There are also c.d.∞ structures for which a computable list of computable isomorphism types exists (Hirschfeldt/Khoussainov/Shore)

Question

What does the set of "cardinalities" of computable isomorphism types look like, as a poset?

Number realizability in structures, 1/2

Suppose $\mathcal{A}$ is a computable structure with domain ω in a finite (or at worst computable) language.

Number realizability in structures, 1/2

Suppose $\mathcal{A}$ is a computable structure with domain ω in a finite (or at worst computable) language.We define the relation $e : \mathcal{A} \models_0 \varphi$ ("e realizes φ in $\mathcal{A}$") by recursion as follows:

- If φ is a literal, then $e : \mathcal{A} \models_0 \varphi$ iff $\mathcal{A} \models \varphi$ in the usual sense.
- If $\varphi \equiv \psi \wedge \theta$, then $e : \mathcal{A} \models_0 \varphi$ iff $e = \langle a, b \rangle$ with $a : \mathcal{A} \models \psi$ and $b : \mathcal{A} \models \theta$.
- If $\varphi \equiv \psi \vee \theta$, then $e : \mathcal{A} \models_0 \varphi$ iff $e = \langle i, a \rangle \in 2 \times \omega$ with $i = 0 \implies a : \mathcal{A} \models \psi$ and $i = 1 \implies a : \mathcal{A} \models \theta$.
- If $\varphi \equiv \forall x \psi(x)$ then $e : \mathcal{A} \models_0 \varphi$ iff e is an index for a total computable function f with $f(a) : \mathcal{A} \models \psi(a)$ for each $a \in \omega$.
- If $\varphi \equiv \exists x \psi(x)$ then $e : \mathcal{A} \models_0 \varphi$ iff $e = \langle a, b \rangle$ with $a : \mathcal{A} \models \psi(b)$.

Number realizability in structures, 1/2

Suppose $\mathcal{A}$ is a computable structure with domain ω in a finite (or at worst computable) language.We define the relation $e : \mathcal{A} \models_0 \varphi$ ("e realizes φ in $\mathcal{A}$") by recursion as follows:

- If φ is a literal, then $e : \mathcal{A} \models_0 \varphi$ iff $\mathcal{A} \models \varphi$ in the usual sense.
- If $\varphi \equiv \psi \wedge \theta$, then $e : \mathcal{A} \models_0 \varphi$ iff $e = \langle a, b \rangle$ with $a : \mathcal{A} \models \psi$ and $b : \mathcal{A} \models \theta$.
- If $\varphi \equiv \psi \vee \theta$, then $e : \mathcal{A} \models_0 \varphi$ iff $e = \langle i, a \rangle \in 2 \times \omega$ with $i = 0 \implies a : \mathcal{A} \models \psi$ and $i = 1 \implies a : \mathcal{A} \models \theta$.
- If $\varphi \equiv \forall x \psi(x)$ then $e : \mathcal{A} \models_0 \varphi$ iff e is an index for a total computable function f with $f(a) : \mathcal{A} \models \psi(a)$ for each $a \in \omega$.
- If $\varphi \equiv \exists x \psi(x)$ then $e : \mathcal{A} \models_0 \varphi$ iff $e = \langle a, b \rangle$ with $a : \mathcal{A} \models \psi(b)$.

Note: φ can have natural number (= domain-element) parameters but must be finitary first-order *and* with no implications or "non-atomic" negations.

Number realizability in structures, 1/2

Suppose $\mathcal{A}$ is a computable structure with domain ω in a finite (or at worst computable) language.We define the relation $e : \mathcal{A} \models_0 \varphi$ ("e realizes φ in $\mathcal{A}$") by recursion as follows:

- If φ is a literal, then $e : \mathcal{A} \models_0 \varphi$ iff $\mathcal{A} \models \varphi$ in the usual sense.
- If $\varphi \equiv \psi \wedge \theta$, then $e : \mathcal{A} \models_0 \varphi$ iff $e = \langle a, b \rangle$ with $a : \mathcal{A} \models \psi$ and $b : \mathcal{A} \models \theta$.
- If $\varphi \equiv \psi \vee \theta$, then $e : \mathcal{A} \models_0 \varphi$ iff $e = \langle i, a \rangle \in 2 \times \omega$ with $i = 0 \implies a : \mathcal{A} \models \psi$ and $i = 1 \implies a : \mathcal{A} \models \theta$.
- If $\varphi \equiv \forall x \psi(x)$ then $e : \mathcal{A} \models_0 \varphi$ iff e is an index for a total computable function f with $f(a) : \mathcal{A} \models \psi(a)$ for each $a \in \omega$.
- If $\varphi \equiv \exists x \psi(x)$ then $e : \mathcal{A} \models_0 \varphi$ iff $e = \langle a, b \rangle$ with $a : \mathcal{A} \models \psi(b)$.

Note: φ can have natural number (= domain-element) parameters but must be finitary first-order *and* with no implications or "non-atomic" negations. "$\exists e[e : \mathcal{A} \models_0 \varphi]$" is abbreviated "$\mathcal{A} \models_0 \varphi$."

Number realizability in structures, 2/2

We can extend the above in a couple obvious ways to get $\mathcal{R}_c, \mathcal{R}_\to$, and finally $\mathcal{R}$:

- $\models_c$: allow computable infinitary Boolean combinations. (Already standard in computable structure theory, e.g. "rice=Σ^c_1")
- $\models_\to$: allow implications (with non-atomic negations being shorthand for $\varphi \to \bot$).
- $\models_\star$: do both.

Interpreting infinitary Boolean combinations is straightforward ("stay effective"). Implication is more subtle:

- We will set $e : \mathcal{A} \models_\to \varphi \to \psi$ iff e is an index for a partial computable function p such that, whenever $c : \mathcal{A} \models_\to \varphi$, we have

$$p(c)\downarrow : \mathcal{A} \models_\to \psi.$$

Wait, implication seems weird ...

- We will set $e : \mathcal{A} \models_{\to} \varphi \to \psi$ iff e is an index for a partial computable function p such that, whenever $c : \mathcal{A} \models_{\to} \varphi$, we have

$$p(c)\downarrow : \mathcal{A} \models_{\to} \psi.$$

This is a "for-structures" version of **Kleene's number realizability**. Historically only the start, and very unsatisfying for constructive semantics. Unsatisfying because it lets classical negation sneak in through the back door. In particular, by putting double negations everywhere we recover full classical logic! Probably a lot of interesting things to do beyond this (cf. J. Moschovakis or van Oosten). But for now, we'll stick with number realizability.

Complexity hierarchies

There is no single complexity hierarchy for $\mathcal{R}$ (or even $\mathcal{R}_0$) which makes sense. Consider (for $\mathcal{A}$ a computable structure):

- Every $\exists\forall\exists$ sentence which is classically true is computably true.
- Markov's principle ("Every Turing machine either halts or doesn't on input 0") is **not** computably true but is classically Σ_3.

Even finitary disjunctions carry nontrivial computability-theoretic strength once inside a quantifier.
There are **at least three** hierarchies which seem relevant: *quantifier rank*, *tree rank*, and (for $\mathcal{R}_\to$ and $\mathcal{R}$) *moral type*.

The three hierarchies

- **Quantifier rank** is just the classical alternation-of-quantifiers count, adapted to non-prenex formulas by taking max at Booleans. Very "lossy" but still has a role: plays well with EF-games.
- **Tree rank** is the rank of the formula as a well-founded tree, so that quantifiers and Booleans contribute equally. This captures things better but gives anomalous results for very low complexity things (e.g. the disjunction of two literals should have rank zero but doesn't).
- Finally, the **moral type** of a sentence is the intentional behavior of a realizer for that sentence. E.g. even though all realizers are numbers, if $\varphi, \psi \in \mathcal{R}_0$ we intuitively think of a realizer for $\varphi \to \psi$ as a function rather than a number. This suggests

$$\mathsf{MT}(\varphi \to \psi) = \max\{\mathsf{MT}(\varphi) + 1, \mathsf{MT}(\psi)\},$$

with quantifiers having no impact and taking max(/sup) at Booleans.

A bit more on moral type

Moral type plays an important role in index set calculations: if $\varphi \in \mathcal{R}_c$ then the set of (indices of) structures satisfying φ is Σ_3 even if the tree rank of φ is large.

Proposition

If $\mathrm{MT}(\varphi) = n$ *then the set of structures satisfying* φ *is* Σ_{3+n}*, and this is sharp in general.*

(We can generalize this to infinitary sentences, it just takes some notational care.) See also Leivant on implicational complexity.

Computable copies of $(\omega; <)$

Let $\mathfrak{N}$ be the set of computable copies of $(\omega; <)$. Some good test questions:

- In what way is the standard copy optimal?
- Is $\mathfrak{N}$ "elementary," and if not what is the model class of its theory?
- How do things change when we generalize from $\mathcal{R}$ to $\mathcal{R}_c$ or $\mathcal{R}_{\rightarrow}$?

Optimality of "obvious" ω

The first question has an easy answer:

- The standard version of ω is characterized up to computable isomorphism by the realizability of "Every element has a successor," which is an $\mathcal{R}$-sentence.
- Moreover, every classically-true first-order sentence is realizable in the standard version (can even go further, e.g. include $+$), so no other copy (up to $\cong_c$) can be $\mathcal{R}$-characterized.
- Allowing infinitary sentences breaks this horribly, and allowing implication *probably* breaks this horribly.

Not-quite-concrete characterizations of copies of ω, 1/2

Proposition

Let $\mathcal{A}$ be a computable copy of $(\omega; <)$. Then there is a $\varphi \in \mathcal{R}_c$ such that $\mathcal{A} \models_c \varphi$ and any $\mathcal{B} \models_c \varphi$ has $\mathcal{A} \cong_c \mathcal{B}$.

Proof sketch.

We will have

$$\varphi \equiv \forall x \bigvee_{i \in \mathbb{N}} \psi_i(x),$$

where each ψ_i "follows" element i: e.g. if 7 starts out looking like the initial element but then 23 gets enumerated before it, ψ_7 will look like

$$\forall y[x \leq y \vee x \leq y \vee x \leq y \vee \ldots \vee x \leq y \vee \psi_{23}(y)].$$

Note that we avoid "loops" so everything stays well-founded. □

Not-quite-concrete characterizations of copies of ω, 2/2

Proposition

Let $\mathcal{A}$ be a computable copy of $(\omega; <)$. Then there is a $\varphi \in \mathcal{R}_c$ such that $\mathcal{A} \models_c \varphi$ and any $\mathcal{B} \models_c \varphi$ has $\mathcal{A} \cong_c \mathcal{B}$.

I don't know what happens if we allow implication but keep things finitary:

Question

Is there a computable copy of $(\omega; <)$ not computably isomorphic to the usual one, which is $\cong_c$-characterized by an $\mathcal{R}_\to$ sentence?

Some more questions about $\mathfrak{N}$

Rolling back to $\mathcal{R}_0$, there are still interesting open questions. Most immediately:

Definition

Let $\sqsubseteq$ be the preorder on $\mathfrak{N}$ given by $\mathcal{A} \sqsubseteq \mathcal{B}$ iff $Th_{\mathcal{R}_0}(\mathcal{A} \subseteq Th_{\mathcal{R}_0}(\mathcal{B})$.

Canonical ω is maximal in this preorder.

- Are there minimal elements?
- Are there elements with no common lower bound?
- What are the possible sizes of the resulting equivalence classes?

To me the first question is most interesting; I've spent some time trying to get a **negative** answer without success.

The model class of $Th_0(\mathfrak{N})$

Which structures satisfy all the $\mathcal{R}_0$-sentences true in every copy of $(\omega; <)$?

Theorem

Suppose $\mathcal{L}$ is a discrete linear order with a first element and no last element. The for every $n \in \omega$, there is a computable copy $\mathcal{A}$ of $(\omega; <)$ such that Duplicator computably wins the length-n EF-game between $\mathcal{L}$ and $\mathcal{A}$.
Consequently, if $\mathcal{L} \models_0 \varphi$ and φ has tree rank $< n$ then $\mathcal{A} \models_0 \varphi$.

Proof sketch.

Build $\mathcal{A}$ and the winning Duplicator strategy simultaneously. Requirements correspond to intervals "matching up," but this is a **weaker** condition than having the same size: intervals k-match if they have the same size or are each sufficiently large (basically $> 2^k$), with k a parameter keeping track of how many moves are left in the game. □

Since

Re: Scott's isomorphism theorem

Theorem (Scott)

Classically, every countable structure is characterized up to isomorphism by an $\mathcal{L}_{\omega_1,\omega}$ sentence.

This fails in a bad way in our setting!

Re: Scott's isomorphism theorem

Theorem (Scott)

Classically, every countable structure is characterized up to isomorphism by an $\mathcal{L}_{\omega_1,\omega}$ sentence.

This fails in a bad way in our setting! Due to changed semantics, this isn't a trivial corollary of the existence of computable structures of high (classical) Scott rank.

Re: Scott's isomorphism theorem

Theorem (Scott)

Classically, every countable structure is characterized up to isomorphism by an $\mathcal{L}_{\omega_1,\omega}$ sentence.

This fails in a bad way in our setting! Due to changed semantics, this isn't a trivial corollary of the existence of computable structures of high (classical) Scott rank.

Theorem

There are non-computably-isomorphic computable structures which have the same $\mathcal{R}_\star$-theories.

Failure of Scott phenomenon

Theorem

There are non-computably-isomorphic computable structures which have the same $\mathcal{R}_\star$-theories.

Proof sketch.

For each computable ordinal α there are non-computably-isomorphic (indeed, non-isomorphic) computable ordinals such that Duplicator has a winning strategy in the counting-down-along-α EF-game.

Failure of Scott phenomenon

Theorem

There are non-computably-isomorphic computable structures which have the same $\mathcal{R}_\star$-theories.

Proof sketch.

For each computable ordinal α there are non-computably-isomorphic (indeed, non-isomorphic) computable ordinals such that Duplicator has a winning strategy in the counting-down-along-α EF-game.Applying Barwise compactness we get a pair of (classically isomorphic) non-computably-isomorphic Harrison orders such that Duplicator computably wins every computable-length EF-game between them. □

Question

What is the relationship between being half of such a pair and having high Scott rank?

Scrutinizing $\mathcal{R}_0$

$\mathcal{R}_0$ is motivated significantly by preexisting interest in FOL.

Scrutinizing $\mathcal{R}_0$

$\mathcal{R}_0$ is motivated significantly by preexisting interest in FOL. This is grounded — at least partly — in two tameness properties, downward Lowenheim-Skolem and compactness.

Scrutinizing $\mathcal{R}_0$

$\mathcal{R}_0$ is motivated significantly by preexisting interest in FOL. This is grounded — at least partly — in two tameness properties, downward Lowenheim-Skolem and compactness. DLS is meaningless for us, and compactness fails ...

Theorem (Tennenbaum)

There is a computable first-order theory with no computable models such that every finite subtheory has a computable model.

... even within a classical isomorphism type!

Theorem

There is a computable structure $\mathcal{A}$ and a computable $\mathcal{R}_0$-theory T such that for every finite $F \subseteq T$ there is a computable $\mathcal{B} \cong \mathcal{A}$ with $\mathcal{B} \models_0 F$ but no computable structure satisfies$_0$ T.

(Non-)Compactness for computability

Theorem

There is a computable structure $\mathcal{A}$ and a computable $\mathcal{R}_0$-theory T such that for every finite $F \subseteq T$ there is a computable $\mathcal{B} \cong \mathcal{A}$ with $\mathcal{B} \models_0 F$ but no computable structure satisfies$_0$ T.

Basically, "simultaneous computable presentability" is interesting.

Proof sketch.

We can produce a computable sequence of Π_2 indices for graphs which happen to have computable copies admitting no computable sequence of *computable* indices for isomorphic copies. Via Marker extensions, attach a "Π_2 copy" of the ith graph in this sequence to element i in a copy of $(\omega; 0, \text{succ})$. □

Question

Does computable dimension ∞ imply "not compact for computability" (in any sense)?

Is compactness-for-computability interesting?

Question

Does computable dimension ∞ imply "not compact for computability" (in any sense)?

Only relevant result I know:

Theorem (Harrison-Trainor, Proposition 4.14's proof)

For any family of rice predicates which is finitely computably realizable on a copy of $(\omega, <)$, there is a copy of $(\omega, <)$ in which all of the relations are computable.

But this relies crucially on riceness.

Thinking categorically 1/2: strategories

Thinking categorically 1/2: strategories

That's only the *second*-worst term for it: Benda 1979 introduced the (non-effective) one-object version, called "modeloids."

Thinking categorically 1/2: strategories

That's only the *second*-worst term for it: Benda 1979 introduced the (non-effective) one-object version, called "modeloids." Categorical naming convention then suggests ... **modeloidoids**.

Thinking categorically 1/2: strategories

That's only the *second*-worst term for it: Benda 1979 introduced the (non-effective) one-object version, called "modeloids." Categorical naming convention then suggests ... **modeloidoids**.

Over all, the effective EF-games perspective seems more robust and justified (to me anyways!) than any particular "sentential" definition of logic I am aware of.

Thinking categorically 1/2: strategories

That's only the *second*-worst term for it: Benda 1979 introduced the (non-effective) one-object version, called "modeloids." Categorical naming convention then suggests ... **modeloidoids**.

Over all, the effective EF-games perspective seems more robust and justified (to me anyways!) than any particular "sentential" definition of logic I am aware of. And HT/M/M/M suggests a different approach entirely:

Theorem (Harrison-Trainor/Melnikov/Miller/Montalbán)

Functors between categories of isomorphic copies of computable structures correspond to effective infinitary interpretations.

(Uneducated comment: may be fruitful to compare with "conceptual completeness" results in categorical logic, and see Chen 2017.)

Thinking categorically 1/2: strategories

That's only the *second*-worst term for it: Benda 1979 introduced the (non-effective) one-object version, called "modeloids." Categorical naming convention then suggests ... **modeloidoids**.

Over all, the effective EF-games perspective seems more robust and justified (to me anyways!) than any particular "sentential" definition of logic I am aware of. And HT/M/M/M suggests a different approach entirely:

Theorem (Harrison-Trainor/Melnikov/Miller/Montalbán)

Functors between categories of isomorphic copies of computable structures correspond to effective infinitary interpretations.

(Uneducated comment: may be fruitful to compare with "conceptual completeness" results in categorical logic, and see Chen 2017.)

This suggests looking at a logic as an appropriately-invariant assignment of games to pairs of structures so that strategies are composable.

Thinking categorically 1/2: strategories

That's only the *second*-worst term for it: Benda 1979 introduced the (non-effective) one-object version, called "modeloids." Categorical naming convention then suggests ... **modeloidoids**.

Over all, the effective EF-games perspective seems more robust and justified (to me anyways!) than any particular "sentential" definition of logic I am aware of. And HT/M/M/M suggests a different approach entirely:

Theorem (Harrison-Trainor/Melnikov/Miller/Montalbán)

Functors between categories of isomorphic copies of computable structures correspond to effective infinitary interpretations.

(Uneducated comment: may be fruitful to compare with "conceptual completeness" results in categorical logic, and see Chen 2017.)

This suggests looking at a logic as an appropriately-invariant assignment of games to pairs of structures so that strategies are composable. Interpretations are then "nice" functors between such categories.

Thinking categorically 1/2: strategories

That's only the *second*-worst term for it: Benda 1979 introduced the (non-effective) one-object version, called "modeloids." Categorical naming convention then suggests ... **modeloidoids**.

Over all, the effective EF-games perspective seems more robust and justified (to me anyways!) than any particular "sentential" definition of logic I am aware of. And HT/M/M/M suggests a different approach entirely:

Theorem (Harrison-Trainor/Melnikov/Miller/Montalbán)

Functors between categories of isomorphic copies of computable structures correspond to effective infinitary interpretations.

(Uneducated comment: may be fruitful to compare with "conceptual completeness" results in categorical logic, and see Chen 2017.)

This suggests looking at a logic as an appropriately-invariant assignment of games to pairs of structures so that strategies are composable. Interpretations are then "nice" functors between such categories. Sentential structure is (if present at all) an emergent and secondary phenomenon.

Computable forcing

This is only tangentially connected with the story so far, **but**:

Question

What do forcing arguments look like in the computable universe?

Can injury-freely prove Friedberg-Muchnik relative to a Cohen generic, so this is tempting! (Nies-Shore-Slaman barrier notwithstanding)

Computable forcing

This is only tangentially connected with the story so far, **but**:

Question

What do forcing arguments look like in the computable universe?

Can injury-freely prove Friedberg-Muchnik relative to a Cohen generic, so this is tempting! (Nies-Shore-Slaman barrier notwithstanding) There is some material here which seems potentially relevant to building structures in more transparent ways, but is not widely known.

Computable forcing

This is only tangentially connected with the story so far, **but**:

Question

What do forcing arguments look like in the computable universe?

Can injury-freely prove Friedberg-Muchnik relative to a Cohen generic, so this is tempting! (Nies-Shore-Slaman barrier notwithstanding) There is some material here which seems potentially relevant to building structures in more transparent ways, but is not widely known.

The main ideas:

- V is to REC as a c.t.m. M is to a **subrecursive class**
- Not much to say about bare posets (all effectively equivalent or trivial); instead, look at posets equipped with enumeration operators ("virtual c.e. sets")
- Rasiowa-Sikorski is much more nuanced!

Computable forcing

This is only tangentially connected with the story so far, **but**:

Question

What do forcing arguments look like in the computable universe?

Can injury-freely prove Friedberg-Muchnik relative to a Cohen generic, so this is tempting! (Nies-Shore-Slaman barrier notwithstanding) There is some material here which seems potentially relevant to building structures in more transparent ways, but is not widely known.

The main ideas:

- V is to REC as a c.t.m. M is to a **subrecursive class**
- Not much to say about bare posets (all effectively equivalent or trivial); instead, look at posets equipped with enumeration operators ("virtual c.e. sets")
- Rasiowa-Sikorski is much more nuanced!

I want to focus on this last bulletpoint here.

A stumbling block for computable RS

Classically, RS says "Every countable family of dense sets is met by a filter"

Warning

A naive effectivization of RS is false: there is no filter through $2^{<\omega}$ which meets every primitive recursive dense set, since if $f = \Phi_e$ is computable then

$$\{\sigma : \exists n < |\sigma|(\Phi_e(n)[|\sigma|] \downarrow \neq \sigma(n))\}$$

is p.r. and dense.

Note that this parallels the failure of **choiceless** RS, so this isn't too surprising (we'll get into more trouble shortly).
There are **two** effectivizations of RS that seem interesting/useful and are actually true. The weaker one is much more intuitive and has nicer "algebraic" properties.

Two computable RS theorems

There are **two** effectivizations of RS that seem interesting/useful and are actually true. Let $\mathbb{P}$ be a computable poset and let $F : \omega^2 \to \omega$ be total, enumerating $\mathbf{C}(F) := \{\lambda x.F(n, x) : n \in \omega\}$.

Two computable RS theorems

There are **two** effectivizations of RS that seem interesting/useful and are actually true. Let $\mathbb{P}$ be a computable poset and let $F : \omega^2 \to \omega$ be total, enumerating $\mathbf{C}(F) := \{\lambda x.F(n, x) : n \in \omega\}$.

Lemma (Folklore, trivial)

There is a computable filter G such that $G \cap D \neq \emptyset$ whenever (the characteristic function of) D is a dense set in $\mathbf{C}(F)$ and some $f \in \mathbf{C}(F)$ has $f(p) \in D_{\leq p}$ for each p.

Two computable RS theorems

There are **two** effectivizations of RS that seem interesting/useful and are actually true. Let $\mathbb{P}$ be a computable poset and let $F : \omega^2 \to \omega$ be total, enumerating $\mathbf{C}(F) := \{\lambda x.F(n, x) : n \in \omega\}$.

Lemma (Folklore, trivial)

There is a computable filter G such that $G \cap D \neq \emptyset$ whenever (the characteristic function of) D is a dense set in $\mathbf{C}(F)$ and some $f \in \mathbf{C}(F)$ has $f(p) \in D_{\leq p}$ for each p.

Lemma (Maass 1982)

There is a computable filter G such that $G \cap D \neq \emptyset$ whenever (the characteristic function of) D is a dense set in $\mathbf{C}(F)$ and some $f \in \mathbf{C}(F)$ has the more complicated property that

Two computable RS theorems

There are **two** effectivizations of RS that seem interesting/useful and are actually true. Let $\mathbb{P}$ be a computable poset and let $F : \omega^2 \to \omega$ be total, enumerating $\mathbf{C}(F) := \{\lambda x.F(n, x) : n \in \omega\}$.

Lemma (Folklore, trivial)

There is a computable filter G such that $G \cap D \neq \emptyset$ whenever (the characteristic function of) D is a dense set in $\mathbf{C}(F)$ and some $f \in \mathbf{C}(F)$ has $f(p) \in D_{\leq p}$ for each p.

Lemma (Maass 1982)

There is a computable filter G such that $G \cap D \neq \emptyset$ whenever (the characteristic function of) D is a dense set in $\mathbf{C}(F)$ and some $f \in \mathbf{C}(F)$ has the more complicated property that for every computable maximal filter H there are cofinal-in-H-many $p \in H$ with $f(p) \in D_{\leq p}$.

Two computable RS theorems

There are **two** effectivizations of RS that seem interesting/useful and are actually true. Let $\mathbb{P}$ be a computable poset and let $F : \omega^2 \to \omega$ be total, enumerating $\mathbf{C}(F) := \{\lambda x.F(n,x) : n \in \omega\}$.

Lemma (Folklore, trivial)

There is a computable filter G such that $G \cap D \neq \emptyset$ whenever (the characteristic function of) D is a dense set in $\mathbf{C}(F)$ and some $f \in \mathbf{C}(F)$ has $f(p) \in D_{\leq p}$ for each p.

Lemma (Maass 1982)

There is a computable filter G such that $G \cap D \neq \emptyset$ whenever (the characteristic function of) D is a dense set in $\mathbf{C}(F)$ and some $f \in \mathbf{C}(F)$ has the more complicated property that for every computable maximal filter H there are cofinal-in-H-many $p \in H$ with $f(p) \in D_{\leq p}$.

Call such filters "naively F-generic" and "Maassively F-generic" respectively. (Sorry ...) Each notion has a (basically tautological) game characterization.

Comparing genericity notions

First let's see why naive genericity is weak:

Proposition

Let f be a computable function with properly c.e. range. Then no matter what F we pick, there are naively F-generic filters G_0, G_1 through the poset of "permission below f" which enumerate in the obvious way sets of degree $\mathbf{0}$ and $\mathbf{0}'$ respectively.

Compare Maass 1982: Maassive genericity doesn't suffer the same weakness!
So why (besides messiness, e.g. no choiceless parallel for Maassive genericity) is naive genericity still useful?

Thinking categorically 2/2: virtual c.e. sets

The proof of the proposition on the previous side involves "adding constraints" to the forcing in two incompatible ways: one forces the generic to enumerate something in $\mathbf{0}$, and the other something in $\mathbf{0}'$.

Thinking categorically 2/2: virtual c.e. sets

The proof of the proposition on the previous side involves "adding constraints" to the forcing in two incompatible ways: one forces the generic to enumerate something in $\mathbf{0}$, and the other something in $\mathbf{0}'$. The result then follows from monotonicity of forcing with respect to appropriate maps of posets: "enriching the poset" only ever forces more things, **as long as** "forces" means "makes true of all sufficiently-*naively*-generic filters."

Thinking categorically 2/2: virtual c.e. sets

The proof of the proposition on the previous side involves "adding constraints" to the forcing in two incompatible ways: one forces the generic to enumerate something in $\mathbf{0}$, and the other something in $\mathbf{0}'$. The result then follows from monotonicity of forcing with respect to appropriate maps of posets: "enriching the poset" only ever forces more things, **as long as** "forces" means "makes true of all sufficiently-*naively*-generic filters."
This monotonicity **fails** for Maassive genericity, as a consequence precisely of the power of Maass' genericity notion.

Definition

A **virtual c.e. set** $(\mathbb{P}, \nu)$ *is a computable poset equipped with a computable and monotonic enumeration operator.*

Morphisms between virtual c.e. sets

Definition

A **virtual c.e. set** $(\mathbb{P}, \nu)$ *is a computable poset* $\mathbb{P}$ *equipped with a computable and monotonic enumeration operator* ν*. A* **morphism** *of v.c.e. sets*

$$(\mathbb{P}, \nu) \to (\mathbb{Q}, \mu)$$

is a computable function $m : \mathbb{Q} \to \mathbb{P}$ *such that the m-preimage of any* $\mathbb{P}$*-dense set is* $\mathbb{Q}$*-dense and such that* $\mu = \nu \circ m$*.*

Morphisms between virtual c.e. sets

Definition

A **virtual c.e. set** $(\mathbb{P}, \nu)$ *is a computable poset* $\mathbb{P}$ *equipped with a computable and monotonic enumeration operator* ν*. A* **morphism** *of v.c.e. sets*

$$(\mathbb{P}, \nu) \to (\mathbb{Q}, \mu)$$

is a computable function $m : \mathbb{Q} \to \mathbb{P}$ *such that the m-preimage of any* $\mathbb{P}$*-dense set is* $\mathbb{Q}$*-dense and such that* $\mu = \nu \circ m$*.*

(Idea: "adding clauses" to a poset in a non-geneiricity-preventing way gives rise to a morphism)
While Maassive genericity gives us more control over the behavior of individual virtual c.e. sets, the category of virtual c.e. sets is better behaved in terms of naive genericity.

Thanks!

- Ash, Knight. **Computable structures and the hyperarithmetical hierarchy**. 2000
- Barwise. **Admissible sets and structures**. 1975
- Benda. "Modeloids 1." 1979
- Chen. "Borel functors, interpretations, and strong conceptual completeness for $\mathcal{L}_{\omega_1,\omega}$" 2017
- Harrison-Trainor. **Degree spectra of relations on a cone**. 2014
- Harrison-Trainor, Melnikov, Miller, Montalbán. "Computable functors and effective interpretability." 2015
- Hirschfeldt/Khoussainov/Shore. "A computably categorical structure whose expansion by a constant has infinite computable dimension." 2003
- Leivant. "Implicational complexity in intuitionistic arithmetic." 1981
- Maass. "Recursively enumerable generic sets." 1982
- McCoy. "Finite computable dimension does not relativize." 2002
- Moschovakis. "Kleene's Readability and 'Divides' Notions for Formalized Intuitionistic Mathematics." 1980
- Turetsky. "Coding in the automorphism group of a computably categorical structure." 2019
- Van Oosten. **Realizability: An Introduction to its Categorical Side**. 2008