How to Think About Computable Structures

Wesley Calvert

Banff International Research Station
March 20, 2023

Procedural Notes: Good exposition is sometimes bad history

Procedural Notes: Good exposition is sometimes bad history

- Russian work often paralleled or preceded the western results I give here, especially in the early days.

Procedural Notes: Good exposition is sometimes bad history

- Russian work often paralleled or preceded the western results I give here, especially in the early days.
- It will sound like there are successive periods that opened and closed. This is false.

Question (Van der Waerden, 1930)

How much field theory can be carried out explicitly?

Question (Van der Waerden, 1930)

How much field theory can be carried out explicitly?

Answer (Van der Waerden, 1930)

Well, certainly some.

Definition (Fröhlich-Shepherdson, 1956)

Let R be a ring whose elements are equivalence classes of natural numbers under a computable equivalence relation, and whose operations are computable functions. Then we say that R is explicit.

Definition (Fröhlich-Shepherdson, 1956)

Let R be a ring whose elements are equivalence classes of natural numbers under a computable equivalence relation, and whose operations are computable functions. Then we say that R is explicit.

Theorem (Van der Waerden, 1930)

If K is an explicit field and $F \supseteq K$ is a finite extension, then F is explicit.

Definition (Fröhlich-Shepherdson, 1956)

Let R be a ring whose elements are equivalence classes of natural numbers under a computable equivalence relation, and whose operations are computable functions. Then we say that R is explicit.

Theorem (Van der Waerden, 1930)

If K is an explicit field and $F \supseteq K$ is a finite extension, then F is explicit.

Theorem (Van der Waerden, 1930)

If K is an explict field with a splitting algorithm (that is, an algorithm to decide, for any polynomial, whether it factors nontrivially), and $F \supseteq K$ is a finite extension, then there is a splitting algorithm for F.

Theorem (Van der Waerden, 1930)

If there exists a general splitting algorithm for all explicitly given fields, then every set $S \subseteq \mathbb{N}$ is computable.

Theorem (Fröhlich-Shepherdson, 1956)

If K is an explicit field and $F \supseteq K$ is a finite extension, then F is explicit, and unique up to computable isomorphism.

Theorem (Fröhlich-Shepherdson, 1956)

If K is an explicit field and $F \supseteq K$ is a finite extension, then F is explicit, and unique up to computable isomorphism.

Theorem (Fröhlich-Shepherdson, 1956)

Every explicit field has an explicit extension with no computable transcendence base.

Theorem (Fröhlich-Shepherdson, 1956)

If K is an explicit field and $F \supseteq K$ is a finite extension, then F is explicit, and unique up to computable isomorphism.

Theorem (Fröhlich-Shepherdson, 1956)

Every explicit field has an explicit extension with no computable transcendence base.

Proof.

Extend by indeterminates $t, x_1, x_2, \ldots$. Let λ be a computable one-to-one function with noncomputable range. Then let I be the ideal generated by polynomials of the form $x_{\lambda(n)}^{p_n} - t$, and take the quotient. □

Theorem (Fröhlich-Shepherdson, 1956)

Every explicit field has an explicit extension with no splitting algorithm.

Definition (Rabin 1960)

We say that a group is computable if and only if its elements are equivalence classes of natural numbers under a computable equivalence relation, and its group operation is computable. A ring is computable under similar circumstances.

Definition (Rabin 1960)

We say that a group is computable if and only if its elements are equivalence classes of natural numbers under a computable equivalence relation, and its group operation is computable. A ring is computable under similar circumstances.

Theorem (Rabin 1960)

The quotient of a computable group by a computable normal subgroup is computable, and the natural projection is computable.

Theorem (Rabin 1960)

If F is a computable field, then there exists a computable algebraic closure of F, and a computable embedding of F into its closure.

This line continues:

This line continues:

- Friedman-Simpson-Smith, 1983, and others on Reverse Mathematics

This line continues:

- Friedman-Simpson-Smith, 1983, and others on Reverse Mathematics
- Mileti-Dzhafarov 2018, and others on degree of the set of primes in a UFD.

This line continues:

- Friedman-Simpson-Smith, 1983, and others on Reverse Mathematics
- Mileti-Dzhafarov 2018, and others on degree of the set of primes in a UFD.
- Brown-McNicholl, 2020, and others, on L^p-spaces

Let L be a computable language. We identify a formula with its Gödel number.

Let L be a computable language. We identify a formula with its Gödel number.

Definition

Let L be a computable signature. We say that an L-structure is computable if and only if its atomic diagram is computable.

Let L be a computable language. We identify a formula with its Gödel number.

Definition

Let L be a computable signature. We say that an L-structure is computable if and only if its atomic diagram is computable.

Definition

Let L be a computable signature. We say that an L-structure is decidable if and only if its full elementary diagram is computable.

Theorem (Harrington 1974)

Let T be a complete decidable theory. The following are equivalent:

Theorem (Harrington 1974)

Let T be a complete decidable theory. The following are equivalent:

- *T has a computable prime model.*

Theorem (Harrington 1974)

Let T be a complete decidable theory. The following are equivalent:

- *T has a computable prime model.*
- *There is a computable function f such that given $\varphi(x)$ consistent with T, applying f to the code for φ gives an index for a computable principal type containing φ.*

Theorem (Harrington 1974)

Let T be a complete decidable theory. The following are equivalent:

- *T has a computable prime model.*
- *There is a computable function f such that given $\varphi(x)$ consistent with T, applying f to the code for φ gives an index for a computable principal type containing φ.*

Proof.

Use the Henkin construction with a little care. □

Proposition (T. Millar 1974)

Every type realized in a decidable model is computable.

Proposition (T. Millar 1974)

Every type realized in a decidable model is computable.

Proposition (T. Millar 1974)

Every computable type of a complete decidable theory is realized in some decidable model.

Proposition (T. Millar 1974)

Every type realized in a decidable model is computable.

Proposition (T. Millar 1974)

Every computable type of a complete decidable theory is realized in some decidable model.

Proposition (T. Millar 1974)

Every computable non-principal type of a complete decidable theory is omitted from some decidable model of that theory.

Theorem (T. Millar 1974)

There is a complete decidable theory with all types recursive such that the countable saturated model of the theory is not decidable.

Theorem (Tennenbaum 1974)

If $\mathcal{A}$ is a computable model of Peano Arithmetic, then $\mathcal{A}$ is isomorphic to the standard model.

Theorem (Tennenbaum 1974)

If $\mathcal{A}$ is a computable model of Peano Arithmetic, then $\mathcal{A}$ is isomorphic to the standard model.

Theorem (Goncharov 1976)

There is a decidable ω-stable theory with no computable prime model.

Theorem (Tennenbaum 1974)

If $\mathcal{A}$ is a computable model of Peano Arithmetic, then $\mathcal{A}$ is isomorphic to the standard model.

Theorem (Goncharov 1976)

There is a decidable ω-stable theory with no computable prime model.

Question (Goncharov 1980)

Let T be a decidable strongly minimal theory. Which models of T have computable copies?

Millar said the subject had no future, because "there are too many counterexamples."

This thread also continues:

- Andrews, Lempp, Knight, Medvedev, and others on models of strongly minimal theories
- Csima 2004, on degrees of prime models
- Lange 2008, on degrees of homogeneous models

Theorem (Ash-Nerode 1981)

Let $\mathcal{A}$ be a computable structure, and R a relation on $\mathcal{A}$. Suppose further that for a tuple $\overline{c} \subseteq \mathcal{A}$ and an existential formula φ, we can decide whether tere exists $\bar{a} \notin R$ such that $\mathcal{A} \models (\bar{c}, \bar{a})$. Then the following are equivalent:

Theorem (Ash-Nerode 1981)

Let $\mathcal{A}$ be a computable structure, and R a relation on $\mathcal{A}$. Suppose further that for a tuple $\overline{c} \subseteq \mathcal{A}$ and an existential formula φ, we can decide whether tere exists $\bar{a} \notin R$ such that $\mathcal{A} \models (\bar{c}, \bar{a})$. Then the following are equivalent:

- *R is computably enumerable in every computable isomorphic copy of $\mathcal{A}$.*

Theorem (Ash-Nerode 1981)

Let $\mathcal{A}$ be a computable structure, and R a relation on $\mathcal{A}$. Suppose further that for a tuple $\overline{c} \subseteq \mathcal{A}$ and an existential formula φ, we can decide whether tere exists $\bar{a} \notin R$ such that $\mathcal{A} \models (\bar{c}, \bar{a})$. Then the following are equivalent:

- *R is computably enumerable in every computable isomorphic copy of $\mathcal{A}$.*
- *There is a computably enumerable sequence of existential formulas whose disjunction is equivalent to R.*

Definition (Goncharov 1977)

Let $\mathcal{A}$ be a computable structure. The <u>computable dimension of $\mathcal{A}$</u> is the number of distinct isomorphic computable copies of $\mathcal{A}$ up to computable isomorphism.

Definition (Goncharov 1977)

Let $\mathcal{A}$ be a computable structure. The computable dimension of $\mathcal{A}$ is the number of distinct isomorphic computable copies of $\mathcal{A}$ up to computable isomorphism.

Example

Let $\mathcal{V}$ be a finite-dimensional vector space over a computable field. Then $\mathcal{V}$ has computable dimension 1.

Definition (Goncharov 1977)

Let $\mathcal{A}$ be a computable structure. The computable dimension of $\mathcal{A}$ is the number of distinct isomorphic computable copies of $\mathcal{A}$ up to computable isomorphism.

Example

Let $\mathcal{V}$ be a finite-dimensional vector space over a computable field. Then $\mathcal{V}$ has computable dimension 1.

Example

Let $\mathcal{V}$ be an infinite-dimensional vector space over a computable infinite field. Then $\mathcal{V}$ has computable dimension ∞.

Theorem (Goncharov 1977)

There exists a computable structure $\mathcal{A}$ of computable dimension 2.

Theorem (Goncharov 1977)

There exists a computable structure $\mathcal{A}$ of computable dimension 2.

Theorem (Goncharov 1974)

A Boolean algebra has computable dimension 1 if there are finitely many atoms, and ∞ otherwise.

Theorem (Goncharov 1977)

There exists a computable structure $\mathcal{A}$ of computable dimension 2.

Theorem (Goncharov 1974)

A Boolean algebra has computable dimension 1 if there are finitely many atoms, and ∞ otherwise.

Theorem (Dzgoev-Goncharov 1980)

If $\mathcal{A}$ is a linear ordering, then $\mathcal{A}$ has computable dimension 1 if the successor relation is finite, and ∞ otherwise.

Definition

We say that $\mathcal{A}$ is <u>computably categorical</u> if and only if it has computable dimension 1.

Definition

Let L be a computable signature.

- A computable Σ_0 or Π_0 formula is a first-order quantifier-free formula.

Definition

Let L be a computable signature.

- A computable Σ_0 or Π_0 formula is a first-order quantifier-free formula.
- A computable Σ_α formula is a computable disjunction of formulas of the form $\exists \overline{x} R$, where R is a computable Π_α formula.

Definition

Let L be a computable signature.

- A computable Σ_0 or Π_0 formula is a first-order quantifier-free formula.
- A computable Σ_α formula is a computable disjunction of formulas of the form $\exists\overline{x}R$, where R is a computable Π_α formula.
- A computable Π_α formula is a computable disjunction of formulas of the form $\forall\overline{x}R$, where R is a computable Σ_α formula.

Definition

Let L be a computable signature.

- A computable Σ_0 or Π_0 formula is a first-order quantifier-free formula.
- A computable Σ_α formula is a computable disjunction of formulas of the form $\exists\overline{x}R$, where R is a computable Π_α formula.
- A computable Π_α formula is a computable disjunction of formulas of the form $\forall\overline{x}R$, where R is a computable Σ_α formula.

If we drop the requirements of "computable" everywhere (to countable), we get $L_{\omega_1\omega}$.

Theorem (Ash 1986)

Satisfaction of computable Σ_α (respectively, Π_α formulas in a computable structure is Σ^0_α (respectively, Π^0_α), with all concievable uniformity.

Theorem (Scott 1965)

Every countable structure $\mathcal{A}$ has an $L_{\omega_1\omega}$ sentence whose models are exactly the isomorphic copies of $\mathcal{A}$.

Theorem (Scott 1965)

Every countable structure $\mathcal{A}$ has an $L_{\omega_1\omega}$ sentence whose models are exactly the isomorphic copies of $\mathcal{A}$.

Definition

Let $\bar{a}, \bar{b} \subseteq \mathcal{A}$.

Theorem (Scott 1965)

Every countable structure $\mathcal{A}$ has an $L_{\omega_1\omega}$ sentence whose models are exactly the isomorphic copies of $\mathcal{A}$.

Definition

Let $\bar{a}, \bar{b} \subseteq \mathcal{A}$.

- We say that $\bar{a} \equiv_0 \bar{b}$ if and only if they satisfy the same quantifier-free formulas.

Theorem (Scott 1965)

Every countable structure $\mathcal{A}$ has an $L_{\omega_1\omega}$ sentence whose models are exactly the isomorphic copies of $\mathcal{A}$.

Definition

Let $\bar{a}, \bar{b} \subseteq \mathcal{A}$.

- We say that $\bar{a} \equiv_0 \bar{b}$ if and only if they satisfy the same quantifier-free formulas.
- We say that $\bar{a} \equiv_\alpha \bar{b}$ if and only if for all $\beta < \alpha$, and for each $\bar{c}$, there exists $\bar{d}$ such that $\bar{a}\bar{c} \equiv_\beta \bar{b}\bar{d}$, and symmetrically.

Definition

The Scott rank of a tuple $\bar{a}$ is the least β such that for all $\bar{b}$, the relation $\bar{a} \equiv_\beta \bar{b}$ implies $(\mathcal{A}, \bar{a}) \cong (\mathcal{A}, \bar{b})$.

Definition

The Scott rank of a tuple $\bar{a}$ is the least β such that for all $\bar{b}$, the relation $\bar{a} \equiv_\beta \bar{b}$ implies $(\mathcal{A}, \bar{a}) \cong (\mathcal{A}, \bar{b})$.

Definition

The Scott rank of the structure $\mathcal{A}$ is the least ordinal α greater than the ranks of all tuples in $\mathcal{A}$.

Theorem (Nadel 1980)

Let $\mathcal{A}$ be a computable structure. Then the Scott rank of $\mathcal{A}$ is at most $\omega_1^{CK} + 1$.

Proposition (C.–Harizanov–Knight–Quinn, 2006)

The set of indices for computable $\mathbb{Q}$-vector spaces of infinite dimension is m-complete Π^0_3.

Proposition (C.–Harizanov–Knight–Quinn, 2006)

The set of indices for computable $\mathbb{Q}$-vector spaces of infinite dimension is m-complete Π^0_3.

Proof.

- A Scott sentence is given by the axioms of a vector space, plus

$$\bigwedge_{n\in\mathbb{N}} \exists x_1, \ldots, x_n \bigwedge_{\lambda} \lambda(\bar{x}) \neq 0$$

Proposition (C.–Harizanov–Knight–Quinn, 2006)

The set of indices for computable $\mathbb{Q}$-vector spaces of infinite dimension is m-complete Π^0_3.

Proof.

- A Scott sentence is given by the axioms of a vector space, plus

$$\bigwedge_{n\in\mathbb{N}} \exists x_1, \ldots, x_n \bigwedge_{\lambda} \lambda(\bar{x}) \neq 0$$

- Given a c.e. set S, build a vector space:

Proposition (C.–Harizanov–Knight–Quinn, 2006)

The set of indices for computable $\mathbb{Q}$-vector spaces of infinite dimension is m-complete Π^0_3.

Proof.

- A Scott sentence is given by the axioms of a vector space, plus

$$\bigwedge_{n\in\mathbb{N}} \exists x_1, \ldots, x_n \bigwedge_{\lambda} \lambda(\bar{x}) \neq 0$$

- Given a c.e. set S, build a vector space:
  - Start with a lot of elements that might be a basis.

Proposition (C.–Harizanov–Knight–Quinn, 2006)

The set of indices for computable $\mathbb{Q}$-vector spaces of infinite dimension is m-complete Π^0_3.

Proof.

- A Scott sentence is given by the axioms of a vector space, plus

$$\bigwedge_{n\in\mathbb{N}} \exists x_1, \ldots, x_n \bigwedge_{\lambda} \lambda(\bar{x}) \neq 0$$

- Given a c.e. set S, build a vector space:
  - Start with a lot of elements that might be a basis.
  - Every time a new element enters S, find a fresh linear combination and make it zero.

□

Theorem

The set of indices for computable well-orderings is m-complete Π^1_1.

Theorem

The set of indices for computable well-orderings is m-complete Π^1_1.

Proof.

- "Every subset. . ."

Theorem

The set of indices for computable well-orderings is m-complete Π^1_1.

Proof.

- "Every subset..."
- Given a Π^1_1 set S, we can make a sequence of linear orderings $\mathcal{L}_n$ such that

$$\mathcal{L}_n = \begin{cases} \text{a computable ordinal} & \text{if } n \in S \\ \omega_1^{CK}(1+\mathbb{Q}) & \text{otherwise} \end{cases}$$

□

Question

What if there isn't a computable thing?

Question

What if there isn't a computable thing?

Definition (Richter 1977)

Let $\mathcal{A}$ be a structure. The degree of the isomorphism type of $\mathcal{A}$ is the least degree (if one exists) in which $\mathcal{A}$ has a computable copy.

Question

What if there isn't a computable thing?

Definition (Richter 1977)

Let $\mathcal{A}$ be a structure. The degree of the isomorphism type of $\mathcal{A}$ is the least degree (if one exists) in which $\mathcal{A}$ has a computable copy.

Theorem (Richter 1977)

Let $\mathbf{d}$ be a Turing degree. Then there is a structure whose isomorphism type has degree $\mathbf{d}$.

Question

What if there isn't a computable thing?

Definition (Richter 1977)

Let $\mathcal{A}$ be a structure. The degree of the isomorphism type of $\mathcal{A}$ is the least degree (if one exists) in which $\mathcal{A}$ has a computable copy.

Theorem (Richter 1977)

Let $\mathbf{d}$ be a Turing degree. Then there is a structure whose isomorphism type has degree $\mathbf{d}$.

Theorem (Richter 1977)

There is a structure whose isomorphism type has no Turing degree.

Theorem (Knight 1986)

Let $\mathcal{A}$ be a structure such that no finite tuple $\bar{a}$ such that any permutation of $\mathcal{A}$ fixing $\bar{a}$ pointwise is an automorphism. Then the degrees of copies of $\mathcal{A}$ are closed upwards.

Quite a lot of activity here these days:

- Categoricity with an oracle.
- In what degrees does the structure have a copy?
- What degrees compute the isomorphism?
- Which structures have a computable infinitary Scott sentence?
- Sets of indices for structures with given property

How to Think About Computable Structures

Wesley Calvert

Banff International Research Station

March 20, 2023