Effective properties of uncountable structures

Noam Greenberg

Victoria University of Wellington

15$^{\text{th}}$ December 2011

Computable algebra

Computable algebra examines the effective properties of countable structures. Typical questions are:

- How hard to perform are standard algebraic constructions?
- How hard is it to find isomorphisms between two effectively presented copies of a structure?
- What information can be coded into isomorphism classes of structures?

Examples

Theorem (Metakides,Nerode)

There is a computable, infinite-dimensional vector space with no infinite, c.e., linearly independent set.

Theorem (Remmel;Dzgoev)

Let $\mathcal{L}$ be a computable linear ordering. There is an effective isomorphism between $\mathcal{L}$ and any given computable copy of $\mathcal{L}$ if and only if $\mathcal{L}$ contains only finitely many successor pairs.

Theorem (Richter)

For any nonzero Turing degree $\mathbf{a}$ *and every linear ordering $\mathcal{L}$, there is a copy of $\mathcal{L}$ which does not compute* $\mathbf{a}$.

Uncountable structures

The restriction to countable structures comes from computability theory, not from algebra.

Yet there are tools to measure the complexity of uncountable sets as well. We use admissible computability, given by Σ_1 definitions over L_α.

For simplicity, work with structures of size $\aleph_1$. Assume that all reals are constructible.

Finite vs. countable

Theorem (G,Knight)

For every ω_1-Turing degree **a***, there is a linear ordering whose copies lie precisely in the degrees above* **a***.*

The key is the difference between the number of possible cuts in a finite linear ordering, compared to the number of possible cuts in a countable one.

Different methods of coding

Theorem (G,Knight)

There is an ω_1-computable field F such that for every computable $K \cong F$, the collection of irreducible polynomials in $K[x]$ computes the halting problem.

This is analogous to a theorem of Fröhlich and Shepherdson (following van der Waerden). However, their coding method, using prime numbers, cannot generalise to the uncountable case. Instead, we use Borel coding of graphs into fields (Friedman and Stanley). Each bit of information about the graph is coded by a countable piece of the field, and so can be recovered by an ω_1-computation.

Computably categorical linear orderings

Theorem (G,Kach,Lempp,Turetsky)

The following are equivalent for an ω_1-computable linear ordering $\mathcal{L}$:

- *$\mathcal{L}$ is computably categorical: for any ω_1-computable linear ordering $\mathcal{K}$ which is isomorphic to $\mathcal{K}$, there is an ω_1-computable isomorphism between $\mathcal{L}$ and $\mathcal{K}$.*
- *There is a countable set $Q \subset \mathcal{L}$ such that:*
  1. *Every Q-interval of $\mathcal{L}$ is either finite, or is ω_1-saturated; and*
  2. *For each n, the collection of cuts of Q which define $\mathcal{L}$-intervals which are either for size n, or are saturated, is computably enumerable.*

Heavy usage of the dichotomy scattered / nonscattered among the countable linear orderings.

Question

Which ω_2-computable linear orderings are ω_2-computably categorical?

Computable model theory

Computable model theory examines the effective content of theorems of model theory. For instance, it looks at definability:

Theorem (Ash,Knight,Manasse,Slaman;Chisholm)

Let $\mathcal{A}$ be a countable structure, and let R be a relation on $\mathcal{A}$. The following are equivalent:

- *For every isomorphism $f : \mathcal{A} \to \mathcal{B}$, the image $f[R]$ is $\Sigma^0_1(\mathcal{B})$;*
- *R is defined in $\mathcal{A}$ by an effective infinitary existential formula (in the language of $\mathcal{A}$).*

The same theorem holds in the uncountable context (G,Knight). The defining formula is obtained by building a generic copy (forcing with partial isomorphisms) and then examining the forcing relation. In the uncountable case, we use the fact that the forcing relation is countably closed.

Scott families

Theorem (Scott)

The following are equivalent for a countable structure $\mathcal{A}$:

- *For every copy $\mathcal{B}$ of $\mathcal{A}$, there is an isomorphism between $\mathcal{A}$ and $\mathcal{B}$ which is computable in $\mathcal{A} \vee \mathcal{B}$.*
- *After naming finitely many constants, the orbits of $\mathcal{A}$ are definable by an $\mathcal{A}$-c.e. family of existential formulas.*

A similar theorem holds in the uncountable setting (G,Knight), with one added feature: restricted to a closed unbounded subset of $[\mathcal{A}]^{\aleph_0}$, the Scott family is continuous (think of the back-and-forth construction).

So far, examples for the necessity of continuity have relations with countable arities. Is this necessary?

The successor relation

For a linear ordering $\mathcal{L}$, the collection $\mathrm{Succ}(\mathcal{L})$ of all successor pairs in $\mathcal{L}$ is $\Pi^0_1(\mathcal{L})$, and so if $\mathcal{L}$ is computable, $\mathrm{Succ}(\mathcal{L})$ has c.e. degree.

In the countable context, a theorem of Downey, Lempp and Wu (extending work by Frolov) states that for any computable $\mathcal{L}$, for every c.e. degree $\mathbf{a} \geqslant \deg_{\mathrm{T}}(\mathrm{Succ}(\mathcal{L}))$, there is a computable copy $\mathcal{L}'$ of $\mathcal{L}$ such that $\deg_{\mathrm{T}}(\mathrm{Succ}(\mathcal{L}')) = \mathbf{a}$.

This is not so for linear orderings of size $\aleph_1$. Interesting things happen if there is a countable subset Q of $\mathcal{L}$ for which every Q-interval of $\mathcal{L}$ is either finite or dense.

Questions

- Do the effective versions of combinatorial objects like diamond sequences or Souslin lines affect the theory?
- Does the wealth of countably closed forcing notions give rise to interesting "generic structures"?
- What happens if we do not assume that all reals are constructible? Should we then move away from L_{ω_1}?

Thank you