Computable structures categorical relative to a few jumps

Alexander Melnikov

Victoria University of Wellington

Novosibirsk, 2013.

This talk is based on ideas of Mal'cev

This talk is based on ideas of Mal'cev

$\downarrow$

who was a supervisor of Larisa L'vovna Maksimova

A short introduction to computable algebra

Computable algebra

In algebra, intuitively effective procedures had been used long before mathematicians agreed on a formal definition of an algorithm:

Computable algebra

In algebra, intuitively effective procedures had been used long before mathematicians agreed on a formal definition of an algorithm:

- In 1880, Kroneker showed that the membership to any ideal of $\mathbb{Z}[X_1, \ldots, X_k]$ is algorithmically *decidable*.

Computable algebra

In algebra, intuitively effective procedures had been used long before mathematicians agreed on a formal definition of an algorithm:

- In 1880, Kroneker showed that the membership to any ideal of $\mathbb{Z}[X_1, \ldots, X_k]$ is algorithmically *decidable*.
- In 1912, Dehn showed that the Word Problem is *decidable* for fundamental groups of certain manifolds.

Computable algebra

In algebra, intuitively effective procedures had been used long before mathematicians agreed on a formal definition of an algorithm:

- In 1880, Kroneker showed that the membership to any ideal of $\mathbb{Z}[X_1, \ldots, X_k]$ is algorithmically *decidable*.
- In 1912, Dehn showed that the Word Problem is *decidable* for fundamental groups of certain manifolds.
- Around 1930, van der Waerden conjectured that there is no universal algorithm for factorizing polynomials over algorithmically ("explicitly") presented fields.

Nonethless, showing that a problem is algorithmically undecidable requires a formal definition of an algorithm.

Computable algebra

Only in the late 1930's mathematicians agreed on what is meant by a computable process:

Definition (Turing, Kleene, Markov, Church, an others)

A function : $\mathbb{N} \to \mathbb{N}$ is *computable* if it can be computed by a Turing machine.

Immediately after the definition was accepted, a number of algebraic problems were proved *undecidable*. We mention two:

Computable algebra

Only in the late 1930's mathematicians agreed on what is meant by a computable process:

Definition (Turing, Kleene, Markov, Church, an others)

A function : $\mathbb{N} \to \mathbb{N}$ is *computable* if it can be computed by a Turing machine.

Immediately after the definition was accepted, a number of algebraic problems were proved *undecidable*. We mention two:

- Novikov-Boone ($\sim$1955) constructed a f.p. group having undecidable Word Problem.

Computable algebra

Only in the late 1930's mathematicians agreed on what is meant by a computable process:

Definition (Turing, Kleene, Markov, Church, an others)

A function : $\mathbb{N} \to \mathbb{N}$ is *computable* if it can be computed by a Turing machine.

Immediately after the definition was accepted, a number of algebraic problems were proved *undecidable*. We mention two:

- Novikov-Boone ($\sim$1955) constructed a f.p. group having undecidable Word Problem.
- Frolich and Shtepherdson (1956) formally clarified the above mentioned ideas of van der Waerden.

The formal definition also has *positive* applications in algebra, possibly the best known one is:

Theorem (Higman, 1961)

A finitely generated group G is embeddable into a finitely presented one if, and only if, the defining relations of G can be *computably enumerated (listed)*.

Higman called such groups "recursively presented".

Computable algebra

The following definition captures most of the effective algebraic examples we've discussed so far:

Definition 1 (Mal'cev 1961, Rabin 1960)

Let $\mathcal{A} = (A, g_1, \ldots, g_k, =)$ be a countably infinite algebra.

- A computably enumerable (c.e.) presentation of $\mathcal{A}$ is any surjective map $\nu : \mathbb{N} \to A$ such that g_i become computable functions on the respective numbers (names) of the inputs.
- A a c.e. presentation is of $\mathcal{A}$ is computable if $=$ on $\mathcal{A}$ is decidable on the respective numbers of elements.

Computable algebra

Examples:

Computable algebra

Examples:

- Computable fields are exactly the “explicitly presented” ones (v.d. W. 1930, F.&S.1956).

Examples:

- Computable fields are exactly the “explicitly presented” ones (v.d. W. 1930, F.&S.1956).
- C.e.-presented groups are the *“recursively presented” groups* (Higman 1961).

Computable algebra

Examples:

- Computable fields are exactly the “explicitly presented” ones (v.d. W. 1930, F.&S.1956).
- C.e.-presented groups are the *“recursively presented” groups* (Higman 1961).
- Computably presented groups are exactly those “recursively presented” ones having decidable Word Problem (Dehn 1912).

Isomorphisms between computable presentations

Non-computable isomorphisms arise naturally

in computable algebra.

Isomorphisms

When working with computable presentations, we often use the following *equivalent* definition:

Definition

A countable algebra $\mathcal{C} = (C, f_1, \ldots, f_k)$ is computable if its domain C is $\mathbb{N}$, and its operations and relations are computable. An algebra $\mathcal{A}$ is computably presentable if it has a computable isomorphic copy.

Isomorphisms

When working with computable presentations, we often use the following *equivalent* definition:

Definition

A countable algebra $\mathcal{C} = (C, f_1, \dots, f_k)$ is computable if its domain C is $\mathbb{N}$, and its operations and relations are computable. An algebra $\mathcal{A}$ is computably presentable if it has a computable isomorphic copy.

Mal'cev discovered the following interesting phenomenon:

There are two computable presentations of the group $\bigoplus_{i \in \mathbb{N}} \mathbb{Q}$ that are not *computably* isomorphic.

Only infinitely generated algebras may have this property.

Isomorphisms

Definition (Mal'cev)

A computably presented algebra is autostable or computably categorical if any two computable presentations of the algebra agree up to a *computable* isomorphism.

Isomorphisms

Definition (Mal'cev)

A computably presented algebra is autostable or computably categorical if any two computable presentations of the algebra agree up to a *computable* isomorphism.

There has been a lot of work on computably categorical algebraic structures (LaRoche, Goncharov, Remmel, Nurtazin, Dobrica, Smith, and others).

Isomorphisms

Definition (Mal'cev)

A computably presented algebra is autostable or computably categorical if any two computable presentations of the algebra agree up to a *computable* isomorphism.

There has been a lot of work on computably categorical algebraic structures (LaRoche, Goncharov, Remmel, Nurtazin, Dobrica, Smith, and others).

In most of the "common" classes, computable categoricity is equal to being "computably homogeneous" after fixing finitely many elements. (It can be clarified using relative computable categoricity.)

Isomorphisms

Definition (Mal'cev)

A computably presented algebra is autostable or computably categorical if any two computable presentations of the algebra agree up to a *computable* isomorphism.

There has been a lot of work on computably categorical algebraic structures (LaRoche, Goncharov, Remmel, Nurtazin, Dobrica, Smith, and others).

In most of the "common" classes, computable categoricity is equal to being "computably homogeneous" after fixing finitely many elements. (It can be clarified using relative computable categoricity.)

Whence, in mathematical practice we usually have to deal with computable structures which are *not computably categorical*.

A more subtle notion is:

Definition (Goncharov)

The computable dimension of a countable algebra is the number of its computable presentations up to *computable* isomorphism.

In most of the familiar classes, the computable dimension is either 1 or ω, but there exist examples of computable dimension $n \geq 2$ (Goncharov).

Isomorphisms

When studying computable dimension, we again need to understand *non-computable* isomorphisms:

Isomorphisms

When studying computable dimension, we again need to understand *non-computable* isomorphisms:

Theorem (Goncharov)

If an algebra has two computable presentations that are not computably isomorphic, but isomorphic *relative to the Halting problem*, then its computable dimension is ω.

Isomorphisms

When studying computable dimension, we again need to understand *non-computable* isomorphisms:

Theorem (Goncharov)

If an algebra has two computable presentations that are not computably isomorphic, but isomorphic *relative to the Halting problem*, then its computable dimension is ω.

Theorem (Goncharov)

There exist an algebra of computable dimension 2 so that *the second iteration of the Halting problem* can decide an isomorphism between the two computable presentations.

In fact, all known examples of finite dimension > 1 have this property.

Isomorphisms

In computable algebra we often *construct* a presentation B of a computable structure A having some specific properties. The isomorphism $A \cong B$ is typically *not* computable.

Isomorphisms

In computable algebra we often *construct* a presentation B of a computable structure A having some specific properties. The isomorphism $A \cong B$ is typically *not* computable.

Isomorphisms

In computable algebra we often *construct* a presentation B of a computable structure A having some specific properties. The isomorphism $A \cong B$ is typically *not* computable.

We need to know more about non-computable isomorphisms between computable structures.

Categoricity relative to an oracle

Categoricity

We write $0'$ to denote the Halting problem, and $0^{(n)}$ stands for the $n'th$ iteration of the Halting problem.

Definition

A computable structure A is $0^{(n)}$-categorical if for every computable $B \cong A$ an *oracle* for $0^{(n)}$ can compute an isomorphism from A onto B.

Categoricity

We write $0'$ to denote the Halting problem, and $0^{(n)}$ stands for the $n'th$ iteration of the Halting problem.

Definition

A computable structure A is $0^{(n)}$-categorical if for every computable $B \cong A$ an *oracle* for $0^{(n)}$ can compute an isomorphism from A onto B.

In effective algebra, we use *priority constructions* of complexities at most $0'''$. Whence, typically $0'''$ can decide an isomorphism. We need to know more about the case $n \leq 3$.

Categoricity

We write $0'$ to denote the Halting problem, and $0^{(n)}$ stands for the $n'th$ iteration of the Halting problem.

Definition

A computable structure A is $0^{(n)}$-categorical if for every computable $B \cong A$ an *oracle* for $0^{(n)}$ can compute an isomorphism from A onto B.

In effective algebra, we use *priority constructions* of complexities at most $0'''$. Whence, typically $0'''$ can decide an isomorphism. We need to know more about the case $n \leq 3$.

We arrive at:

Problem

Describe $0^{(n)}$-categorical structures in familiar classes for *small* n.

Categoricity

Not much is known about $0^{(n)}$-categorical algebraic structures even in familiar classes.

Categoricity

Not much is known about $0^{(n)}$-categorical algebraic structures even in familiar classes.

- (Ash): A classification of $0^{(n)}$-categoricity for well-orderings.

Categoricity

Not much is known about $0^{(n)}$-categorical algebraic structures even in familiar classes.

- (Ash): A classification of $0^{(n)}$-categoricity for well-orderings.
- (McCoy): A partial information on $0'$ and $0''$-categorical Boolean algebras and linear orders.

Categoricity

Not much is known about $0^{(n)}$-categorical algebraic structures even in familiar classes.

- (Ash): A classification of $0^{(n)}$-categoricity for well-orderings.
- (McCoy): A partial information on $0'$ and $0''$-categorical Boolean algebras and linear orders.
- (Miller, R.): An analysis of $0^{(n)}$-categoricity of algebraic fields.

Categoricity

Not much is known about $0^{(n)}$-categorical algebraic structures even in familiar classes.

- (Ash): A classification of $0^{(n)}$-categoricity for well-orderings.
- (McCoy): A partial information on $0'$ and $0''$-categorical Boolean algebras and linear orders.
- (Miller, R.): An analysis of $0^{(n)}$-categoricity of algebraic fields.
- (Downey and M. 2011): A complete classification of $0'$- and $0''$-categorical homogeneous completely decomposable groups.

Categoricity

Not much is known about $0^{(n)}$-categorical algebraic structures even in familiar classes.

- (Ash): A classification of $0^{(n)}$-categoricity for well-orderings.
- (McCoy): A partial information on $0'$ and $0''$-categorical Boolean algebras and linear orders.
- (Miller, R.): An analysis of $0^{(n)}$-categoricity of algebraic fields.
- (Downey and M. 2011): A complete classification of $0'$- and $0''$-categorical homogeneous completely decomposable groups.
- (Calvert, Cenzer, Harizanov, and Morozov 2009): A partial information on $0'$-categoricity in equivalence structures and abelian p-groups of Ulm type 1.

Three new results on categoricity

Completely decomposable groups

Completely decomposable groups are abelian groups of the form

$$\bigoplus_i H_i,$$

where $H_i \leq \mathbb{Q}$. First studied by Baer ($\sim$1930), then by Kulikov and others.

Problem (Khisamiev $\sim$1998)

Classify computably presented completely decomposable groups.

The first result

Theorem (Downey and M., 2012)

Computable completely decomposable groups are $\mathbf{0}^{(4)}$-categorical, and this is sharp.

The first result

Theorem (Downey and M., 2012)

Computable completely decomposable groups are $0^{(4)}$-categorical, and this is sharp.

It requires some work to derive:

Corollary

The index set and the isomorphism problem for computable c.d. groups are Σ^0_7. (We do not know if it is sharp.)

The first result

Theorem (Downey and M., 2012)

Computable completely decomposable groups are $\mathbf{0}^{(4)}$-categorical, and this is sharp.

It requires some work to derive:

Corollary

The index set and the isomorphism problem for computable c.d. groups are Σ^0_7. (We do not know if it is sharp.)

Intuitively, it means that $\mathbf{0}^{(6)}$ can list all computable presentations of c.d. groups without repetitions (up to isomorphism).

The first result

Theorem (Downey and M., 2012)

Computable completely decomposable groups are $0^{(4)}$-categorical, and this is sharp.

It requires some work to derive:

Corollary

The index set and the isomorphism problem for computable c.d. groups are Σ^0_7. (We do not know if it is sharp.)

Intuitively, it means that $0^{(6)}$ can list all computable presentations of c.d. groups without repetitions (up to isomorphism).

So far, it is the best known solution to Khisamiev's problem.

Equivalence structures

An equivalence structure is a set with an equivalence relation on it.

Calvert, Cenzer, Harizanov, and Morozov (2009) observed that every computable equivalence structure is $0''$-categorical. They left open:

Problem

Which computable equivalence structures are $0'$-categorical?

Equivalence structures

An equivalence structure is a set with an equivalence relation on it.

Calvert, Cenzer, Harizanov, and Morozov (2009) observed that every computable equivalence structure is $0''$-categorical. They left open:

Problem

Which computable equivalence structures are $0'$-categorical?

Equivalence structures have simple algebraic invariants. But the invariants are not algorithmically simple, as it turns out.

The second result

Definition

For a set $X \subset \omega$, let $E(X)$ be an equivalence structure with ω-many infinite classes and exactly one class of size n for each $n \in X$.

Say that an infinite (necessarily c.e. in $0'$) set X is categorical if the computable $E(X)$ is $0'$-categorical.

The second result

Definition

For a set $X \subset \omega$, let $E(X)$ be an equivalence structure with ω-many infinite classes and exactly one class of size n for each $n \in X$.

Say that an infinite (necessarily c.e. in $0'$) set X is categorical if the computable $E(X)$ is $0'$-categorical.

The best we know is:

Theorem (Downey, M., Ng 2013)

For a c.e. Turing degree $\boldsymbol{a}$, the following are equivalent:

1. $\boldsymbol{a}$ is high.
2. There exists an infinite categorical set $X \leq_T \boldsymbol{a}$.

The third result

Can we at least reduce the general problem to the set case without repetitions?

The third result

Can we at least reduce the general problem to the set case without repetitions?

Given an equivalence structure E, remove repetitions of finite classes from E. Call the resulting E_0 the condensation of E.

The third result

Can we at least reduce the general problem to the set case without repetitions?

Given an equivalence structure E, remove repetitions of finite classes from E. Call the resulting E_0 the condensation of E.

Theorem (Downey, M., Ng 2013)

A computable eq. structure E is $0'$-categorical if, and only if, its condensation is $0'$-categorical.

The third result

Can we at least reduce the general problem to the set case without repetitions?

Given an equivalence structure E, remove repetitions of finite classes from E. Call the resulting E_0 the condensation of E.

Theorem (Downey, M., Ng 2013)

A computable eq. structure E is $0'$-categorical if, and only if, its condensation is $0'$-categorical.

The proof is an unusual “monster” priority argument (Lachlan) which is similar to the Golden Run (Nies) construction rather than to any usual degree-theoretic construction. That is, *the true path* is actually finite. Nothing of this sort has ever been seen in effective algebra.

Bibliography and thanks

1. Effectively categorical abelian groups, Downey and M., Journal of Algebra, Volume 373, Pages 223–248.
2. Computable Completely Decomposable Groups, Downey and M., to appear in *Trans. Amer. Math. Soc.*.
3. Equivalence Structures Categorical Relative to the Halting Problem, Downey, M., and Ng, in prep.

Bibliography and thanks

1. Effectively categorical abelian groups, Downey and M., Journal of Algebra, Volume 373, Pages 223–248.
2. Computable Completely Decomposable Groups, Downey and M., to appear in *Trans. Amer. Math. Soc.*.
3. Equivalence Structures Categorical Relative to the Halting Problem, Downey, M., and Ng, in prep.

Thanks!