Limitwise monotonic functions and classifications of structures

Alexander Melnikov

Singapore, 15 Sep 2017.
Happy Birthday, the Father-Node of Logic in New Zealand!
Introduction
Idea: Approach classification problems in computable algebra from the perspective of pure recursion theory (neither via definability nor via algebra).

The main tools: Limitwise monotonic approximations, priority arguments, and various tricks separating algebra from combinatorics.

Definition

A function \( f : \omega \to \omega \cup \{\infty\} \) is **limitwise monotonic** if there exists a (total) recursive \( g : \omega \times \omega \to \omega \) such that

\[
f(x) = \sup_{y} g(x, y),
\]

for all \( x \).

If we forbid \( \infty \) then it gives a special subclass of \( \Delta^0_2 \) functions.
Idea: Approach classification problems in computable algebra from the perspective of pure recursion theory (neither via definability nor via algebra).

The main tools: Limitwise monotonic approximations, priority arguments, and various tricks separating algebra from combinatorics.

Definition

A function \( f : \omega \to \omega \cup \{\infty\} \) is limitwise monotonic if there exists a (total) recursive \( g : \omega \times \omega \to \omega \) such that

\[
    f(x) = \sup_y g(x, y),
\]

for all \( x \).

If we forbid \( \infty \) then it gives a special subclass of \( \Delta^0_2 \) functions.
Idea: Approach classification problems in computable algebra from the perspective of pure recursion theory (neither via definability nor via algebra).

The main tools: Limitwise monotonic approximations, priority arguments, and various tricks separating algebra from combinatorics.

Definition

A function \( f : \omega \rightarrow \omega \cup \{\infty\} \) is limitwise monotonic if there exists a (total) recursive \( g : \omega \times \omega \rightarrow \omega \) such that

\[
f(x) = \sup_y g(x, y),
\]

for all \( x \).

If we forbid \( \infty \) then it gives a special subclass of \( \Delta^0_2 \) functions.
Why do we care?
Limitwise monotonic functions show up in computable:

1. equivalence structures;
2. linear orders ($\eta$-presentations, shuffle sums, initial segments etc.);
3. abelian groups;
4. models of $\aleph_1$-categorical structures
5. many other things that “grow”.

See a survey of Downey, Kach, Turetsky; see also my paper with Kalimullin and Khoussainov.

Limitwise monotonic functions are not very well understood.
Why do we care?
Limitwise monotonic functions show up in computable:
1. equivalence structures;
2. linear orders (η-presentations, shuffle sums, initial segments etc.);
3. abelian groups;
4. models of $\aleph_1$-categorical structures
5. many other things that “grow”.

See a survey of Downey, Kach, Turetsky; see also my paper with Kalimullin and Khoussainov.

Limitwise monotonic functions are not very well understood.
Why do we care?
Limitwise monotonic functions show up in computable:

1. equivalence structures;
2. linear orders ($\eta$-presentations, shuffle sums, initial segments etc.);
3. abelian groups;
4. models of $\aleph_1$-categorical structures
5. many other things that “grow”.

See a survey of Downey, Kach, Turetsky; see also my paper with Kalimullin and Khoussainov.

Limitwise monotonic functions are not very well understood.
A (naive) thought: There is not much to say about computable equivalence structures.

This class is useless and trivial.

A computable equivalence structure is essentially a limitwise monotonic function or an approximation of a $\Sigma^0_2$ multiset.
A (naive) thought: There is not much to say about computable equivalence structures.

This class is useless and trivial.

A computable equivalence structure is essentially a limitwise monotonic function or an approximation of a $\Sigma^0_2$ multiset.
Part 1: The problem of Khisamiev-Ash-Knight-Oates is hard
Countable abelian p-groups can be viewed as layers of equivalence structures (multisets) living on a tree.

1. A group $G$ is **reduced** if the tree is well-founded.
2. Iterate the Ulm derivative $G \rightarrow G'$ to form (essentially) equivalence structures $G/G'$.
3. We have the sequence $G_\alpha = G^{(\alpha)}/G^{(\alpha+1)}$ that terminates at $u(G)$, the **Ulm type** of the group.
4. The sequence of **Ulm factors** $G_\alpha = G^{(\alpha)}/G^{(\alpha+1)}$ fully describes the group (this fact is non-trivial).

Strictly speaking, the Ulm factors are direct sums of cyclic $p$-groups.
Countable abelian $p$-groups can be viewed as layers of equivalence structures (multisets) living on a tree.

1. A group $G$ is **reduced** if the tree is well-founded.
2. Iterate the Ulm derivative $G \rightarrow G'$ to form (essentially) equivalence structures $G/G'$.
3. We have the sequence $G_\alpha = G^{(\alpha)}/G^{(\alpha+1)}$ that terminates at $u(G)$, the **Ulm type** of the group.
4. The sequence of **Ulm factors** $G_\alpha = G^{(\alpha)}/G^{(\alpha+1)}$ fully describes the group (this fact is non-trivial).

Strictly speaking, the Ulm factors are direct sums of cyclic $p$-groups.
Countable abelian $p$-groups can be viewed as layers of equivalence structures (multisets) living on a tree.

1 A group $G$ is **reduced** if the tree is well-founded.
2 Iterate the Ulm derivative $G \rightarrow G'$ to form (essentially) equivalence structures $G/G'$.
3 We have the sequence $G_\alpha = G^{(\alpha)}/G^{(\alpha+1)}$ that terminates at $u(G)$, the **Ulm type** of the group.
4 The sequence of **Ulm factors** $G_\alpha = G^{(\alpha)}/G^{(\alpha+1)}$ fully describes the group (this fact is non-trivial).

Strictly speaking, the Ulm factors are direct sums of cyclic $p$-groups.
Theorem (Khisamiev; Ash-Knight-Oates)

For a reduced abelian $p$-group $G$ of finite Ulm type $m$, TFAE:

1. $G$ has a computable copy;
2. $G_0, G_1, \ldots, G_m$ have $\Delta^0_1, \Delta^0_3, \ldots, \Delta^0_{2m+1}$-copies, respectively.

Recall each $G_i$ is (essentially) a limitwise monotonic function.

Problem

What happens when the Ulm type of $G$ is $\omega$?
Theorem (Khisamiev; Ash-Knight-Oates)

For a reduced abelian $p$-group $G$ of finite Ulm type $m$, TFAE:

1. $G$ has a computable copy;
2. $G_0, G_1, \ldots, G_m$ have $\Delta^0_1, \Delta^0_3, \ldots, \Delta^0_{2m+1}$-copies, respectively.

Recall each $G_i$ is (essentially) a limitwise monotonic function.

Problem

What happens when the Ulm type of $G$ is $\omega$?
Theorem (Khisamiev; Ash-Knight-Oates)

For a reduced abelian $p$-group $G$ of finite Ulm type $m$, TFAE:

1. $G$ has a computable copy;
2. $G_0, G_1, \ldots, G_m$ have $\Delta^0_1, \Delta^0_3, \ldots, \Delta^0_{2m+1}$-copies, respectively.

Recall each $G_i$ is (essentially) a limitwise monotonic function.

Problem

What happens when the Ulm type of $G$ is $\omega$?
Theorem (Downey, M., and Ng)

(Essentially:) The case of Ulm type $\omega$ is hard.

We proved: Given a computable $G$, calculating the index of its $n^{th}$ $0^{(2n)}$-monotonic function requires $0^{(2n+3)}$.

If such a sequence is played by God, we must analyse an iterated $0'''$ in its full generality to either build a copy of $G$ or construct a counter-example.

Our proof is the first known example of an iterated $0'''$.

Have you noticed? This was all about equivalence structures.
Theorem (Downey, M., and Ng)

(Essentially:) The case of Ulm type $\omega$ is hard.

We proved: Given a computable $G$, calculating the index of its $n^{th}$ $0^{(2n)}$-monotonic function requires $0^{(2n+3)}$.

If such a sequence is played by God, we must analyse an iterated $0'''$ in its full generality to either build a copy of $G$ or construct a counter-example.

Our proof is the first known example of an iterated $0'''$.

Have you noticed? This was all about equivalence structures.
Theorem (Downey, M., and Ng)

(Essentially:) The case of Ulm type $\omega$ is hard.

We proved: Given a computable $G$, calculating the index of its $n^{th}$ $0^{(2n)}$-monotonic function requires $0^{(2n+3)}$.

If such a sequence is played by God, we must analyse an iterated $0'''$ in its full generality to either build a copy of $G$ or construct a counter-example.

Our proof is the first known example of an iterated $0'''$.

Have you noticed? This was all about equivalence structures.
Theorem (Downey, M., and Ng)

(Essentially:) The case of Ulm type $\omega$ is hard.

We proved: Given a computable $G$, calculating the index of its $n^{th}$ $0^{(2n)}$-monotonic function requires $0^{(2n+3)}$.

If such a sequence is played by God, we must analyse an iterated $0'''$ in its full generality to either build a copy of $G$ or construct a counter-example.

Our proof is the first known example of an iterated $0'''$.

Have you noticed? This was all about equivalence structures.
Theorem (Downey, M., and Ng)

(Essentially:) The case of Ulm type $\omega$ is hard.

We proved: Given a computable $G$, calculating the index of its $n^{th}$ $0^{(2n)}$-monotonic function requires $0^{(2n+3)}$.

If such a sequence is played by God, we must analyse an iterated $0'''$ in its full generality to either build a copy of $G$ or construct a counter-example.

Our proof is the first known example of an iterated $0'''$.

Have you noticed? This was all about equivalence structures.
Friedberg enumerations of structures
Suppose $\mathcal{K}$ is a class of (computable) algebraic structures.

**Definition**

A computable enumeration of structures in $\mathcal{K}$ is *Friedberg* if it is 1-1 up to isomorphism.

Very few classes admit a Friedberg enumeration.

References:

- Three theorems on recursive enumeration (Friedberg)
- Friedberg Numberings of Families of n-Computably Enumerable Sets (Goncharov, Lempp, Solomon)
- Structure and Anti-structure theorems (Goncharov and Knigh)
- Effective classification of computable structures (MillerR., Lange, and Steiner)
- Effectively closed sets and enumerations (Brodhead and Cenzer)
- Theory of numberings (A book by Ershov)
- The CSc Dissertation of Ospichev (in Russian)
Suppose $\mathcal{K}$ is a class of (computable) algebraic structures.

**Definition**

A computable enumeration of structures in $\mathcal{K}$ is *Friedberg* if it is 1-1 up to isomorphism.

Very few classes admit a Friedberg enumeration.

References:

- Three theorems on recursive enumeration (Friedberg)
- Friedberg Numberings of Families of n-Computably Enumerable Sets (Goncharov, Lempp, Solomon)
- Structure and Anti-structure theorems (Goncharov and Knight)
- Effective classification of computable structures (Miller R., Lange, and Steiner)
- Effectively closed sets and enumerations (Brodhead and Cenzer)
- Theory of numberings (A book by Ershov)
- The CSc Dissertation of Ospichev (in Russian)
Suppose $\mathcal{K}$ is a class of (computable) algebraic structures.

**Definition**

A computable enumeration of structures in $\mathcal{K}$ is *Friedberg* if it is 1-1 up to isomorphism.

Very few classes admit a Friedberg enumeration.

References:

- Three theorems on recursive enumeration (Friedberg)
- Friedberg Numberings of Families of n-Computably Enumerable Sets (Goncharov, Lempp, Solomon)
- Structure and Anti-structure theorems (Goncharov and Knigh)
- Effective classification of computable structures (MillerR., Lange, and Steiner)
- Effectively closed sets and enumerations (Brodhead and Cenzer)
- Theory of numberings (A book by Ershov)
- The CSc Dissertation of Ospichev (in Russian)
Question (Goncharov and Knight 2002)

Is there a Friedberg enumeration of the class of computable equivalence structures?

Goncharov and Knight conjectured that the answer is NO because the invariants are too complicated.
Question (Goncharov and Knight 2002)

Is there a Friedberg enumeration of the class of computable equivalence structures?

Goncharov and Knight conjectured that the answer is NO because the invariants are too complicated.
Guessing isomorphism $E \cong F$ between eq. structures is a $\Pi^0_4$-complete problem.

Compare this to c.e. sets where $W_e = W_j$ is $\Pi^0_2$.

There were earlier attempts by Goncharov and Knight, and by Miller R., Lange, and Steiner.

Theorem (Downey, M., Ng)
There exists a Friedberg enumeration of computable eq. structures.

This is a non-uniform $0'''$. 
Guessing isomorphism $E \cong F$ between eq. structures is a $\Pi^0_4$-complete problem.

Compare this to c.e. sets where $W_e = W_j$ is $\Pi^0_2$.

There were earlier attempts by Goncharov and Knight, and by Miller R., Lange, and Steiner.

**Theorem (Downey, M., Ng)**

There exists a Friedberg enumeration of computable eq. structures.

This is a non-uniform $0''''$. 
Guessing isomorphism $E \cong F$ between eq. structures is a $\Pi^0_4$-complete problem.

Compare this to c.e. sets where $W_e = W_f$ is $\Pi^0_2$.

There were earlier attempts by Goncharov and Knight, and by Miller R., Lange, and Steiner.

**Theorem (Downey, M., Ng)**

There exists a Friedberg enumeration of computable eq. structures.

This is a non-uniform $0''''$. 
Guessing isomorphism $E \cong F$ between eq. structures is a $\Pi^0_4$-complete problem.

Compare this to c.e. sets where $W_e = W_j$ is $\Pi^0_2$.

There were earlier attempts by Goncharov and Knight, and by Miller R., Lange, and Steiner.

**Theorem (Downey, M., Ng)**

There exists a Friedberg enumeration of computable eq. structures.

This is a non-uniform $0'''$. 

Alexander Melnikov

Limitwise monotonic functions and classifications of structures
It would be nice to extend the result to abelian $p$-groups.

We know that reduced abelian $p$-groups of a fixed finite Ulm type (observed by Goncharov and Knight).

Remarkably, if we drop “reduced” than such an enumeration exists:

**Theorem (with Ng)**

1. For each $m < \omega$, there exists a Friedberg enumeration of all computable abelian $p$-groups of Ulm type $\leq m$.
2. There exists a Friedberg enumeration of all computable abelian $p$-groups of finite Ulm type.

This are the first non-trivial and natural algebraic classes with a Friedberg enumeration. The proof is rather technical and uses a Friedberg enumeration of computable eq. structures.
It would be nice to extend the result to abelian $p$-groups.

We know that reduced abelian $p$-groups of a fixed finite Ulm type (observed by Goncharov and Knight).

Remarkably, if we drop “reduced” than such an enumeration exists:

**Theorem (with Ng)**

1. For each $m < \omega$, there exists a Friedberg enumeration of all computable abelian $p$-groups of Ulm type $\leq m$.
2. There exists a Friedberg enumeration of all computable abelian $p$-groups of finite Ulm type.

This are the first non-trivial and natural algebraic classes with a Friedberg enumeration. The proof is rather technical and uses a Friedberg enumeration of computable eq. structures.
It would be nice to extend the result to abelian $p$-groups.

We know that reduced abelian $p$-groups of a fixed finite Ulm type (observed by Goncharov and Knight).

Remarkably, if we drop “reduced” than such an enumeration exists:

**Theorem (with Ng)**

1. For each $m < \omega$, there exists a Friedberg enumeration of all computable abelian $p$-groups of Ulm type $\leq m$.
2. There exists a Friedberg enumeration of all computable abelian $p$-groups of finite Ulm type.

This are the first non-trivial and natural algebraic classes with a Friedberg enumeration. The proof is rather technical and uses a Friedberg enumeration of computable eq. structures.
It would be nice to extend the result to abelian $p$-groups.

We know that **reduced** abelian $p$-groups of a fixed finite Ulm type (observed by Goncharov and Knight).

Remarkably, if we drop “reduced” than such an enumeration exists:

**Theorem (with Ng)**

1. For each $m < \omega$, there exists a Friedberg enumeration of all computable abelian $p$-groups of Ulm type $\leq m$.
2. There exists a Friedberg enumeration of all computable abelian $p$-groups of finite Ulm type.

This are the first non-trivial and natural algebraic classes with a Friedberg enumeration. The proof is rather technical and uses a Friedberg enumeration of computable eq. structures.
A problem of Mal’cev
A structure is **computably categorical** if it has a unique computable copy, up to computable isomorphism.

**Problem (Maltsev, in the 1960-s)**

Describe computably categorical abelian groups.

We have nice satisfactory classifications for:

- *p*-groups (Smith, indep. Goncharov)
- torsion-free (Nurtazin)
- infinite rank (Goncharov)

Missing cases:

- torsion
- mixed of finite rank > 1
A structure is **computably categorical** if it has a unique computable copy, up to computable isomorphism.

**Problem (Maltsev, in the 1960-s)**
Describe computably categorical abelian groups.

We have nice satisfactory classifications for:

- $p$-groups (Smith, indep. Goncharov)
- torsion-free (Nurtazin)
- infinite rank (Goncharov)

Missing cases:
- torsion
- mixed of finite rank $> 1$
A structure is **computably categorical** if it has a unique computable copy, up to computable isomorphism.

**Problem (Maltsev, in the 1960-s)**

Describe computably categorical abelian groups.

We have nice satisfactory classifications for:

- $p$-groups (Smith, indep. Goncharov)
- torsion-free (Nurtazin)
- infinite rank (Goncharov)

Missing cases:

- torsion
- mixed of finite rank $> 1$
Case of study: Torsion abelian groups.

What would be considered a “good” classification of c.c. torsion abelian groups?

It is not hard to show:

**Fact**
There exist c.c. but not relatively c.c. torsion abelian groups.

Thus, there should not be any **algebraic description** of c.c. torsion groups.

We decided to look at the **index set**

\[ \{ i : M_i \text{ is a c.c. torsion abelian group} \} \]
Case of study: Torsion abelian groups.

What would be considered a “good” classification of c.c. torsion abelian groups?

It is not hard to show:

**Fact**
There exist c.c. but not relatively c.c. torsion abelian groups.

Thus, there should not be any *algebraic description* of c.c. torsion groups.

We decided to look at the *index set*

\[ \{ i : M_i \text{ is a c.c. torsion abelian group} \} . \]
Case of study: Torsion abelian groups.

What would be considered a “good” classification of c.c. torsion abelian groups?

It is not hard to show:

**Fact**

There exist c.c. but not relatively c.c. torsion abelian groups.

Thus, there should not be any algebraic description of c.c. torsion groups.

We decided to look at the index set

\[ \{ i : M_i \text{ is a c.c. torsion abelian group} \}. \]
Case of study: Torsion abelian groups.

What would be considered a “good” classification of c.c. torsion abelian groups?

It is not hard to show:

Fact

There exist c.c. but not relatively c.c. torsion abelian groups.

Thus, there should not be any \textit{algebraic description} of c.c. torsion groups.

We decided to look at the \textit{index set}

\[\{i : M_i \text{ is a c.c. torsion abelian group}\}\]
Case of study: Torsion abelian groups.

What would be considered a “good” classification of c.c. torsion abelian groups?

It is not hard to show:

**Fact**

There exist c.c. but not relatively c.c. torsion abelian groups.

Thus, there should not be any *algebraic description* of c.c. torsion groups.

We decided to look at the **index set**

\[ \{ i : M_i \text{ is a c.c. torsion abelian group} \} \]
The crude upper bound is $\Pi^1_1$.

Using known techniques it can be pushed down to $\Pi^0_5$.

**Theorem (M. and Ng)**

The index set

$$\{ i : M_i \text{ is a c.c. torsion abelian group} \}$$

is $\Pi^0_4$-complete.

- $\Pi^0_4$-harness of the index set is the easy(er) part.
- The proof relies on several subtle **algebraic reductions**.
- We use that a certain diagonalization attempt on **equivalence structures** must fail.
- **Computable equivalence structures** are in the (scary) combinatorial core of the proof.
The crude upper bound is $\Pi_1^1$.

Using known techniques it can be pushed down to $\Pi_5^0$.

**Theorem (M. and Ng)**

The index set

$$\{ i : M_i \text{ is a c.c. torsion abelian group} \}$$

is $\Pi_4^0$-complete.

- $\Pi_4^0$-harness of the index set is the easy(er) part.
- The proof relies on several subtle algebraic reductions.
- We use that a certain diagonalization attempt on equivalence structures must fail.
- Computable equivalence structures are in the (scary) combinatorial core of the proof.
The crude upper bound is $\Pi_1^1$.

Using known techniques it can be pushed down to $\Pi_5^0$.

**Theorem (M. and Ng)**

The index set

$$\{ i : M_i \text{ is a c.c. torsion abelian group} \}$$

is $\Pi_4^0$-complete.

- $\Pi_4^0$-harness of the index set is the easy(er) part.
- The proof relies on several subtle algebraic reductions.
- We use that a certain diagonalization attempt on equivalence structures must fail.
- Computable equivalence structures are in the (scary) combinatorial core of the proof.
The crude upper bound is $\Pi^1_1$.

Using known techniques it can be pushed down to $\Pi^0_5$.

**Theorem (M. and Ng)**

The index set

$$\{ i : M_i \text{ is a c.c. torsion abelian group} \}$$

is $\Pi^0_4$-complete.

- $\Pi^0_4$-harness of the index set is the easy(er) part.
- The proof relies on several subtle **algebraic reductions**.
- We use that a certain diagonalization attempt on equivalence structures must fail.
- Computable equivalence structures are in the (scary) combinatorial core of the proof.
The crude upper bound is $\Pi^1_1$.

Using known techniques it can be pushed down to $\Pi^0_5$.

**Theorem (M. and Ng)**

The index set

$$\{i : M_i \text{ is a c.c. torsion abelian group}\}$$

is $\Pi^0_4$-complete.

- $\Pi^0_4$-harness of the index set is the easy(er) part.
- The proof relies on several subtle **algebraic reductions**.
- We use that a certain diagonalization attempt on **equivalence structures** must fail.
- Computable equivalence structures are in the (scary) combinatorial core of the proof.
The crude upper bound is $\Pi_1^1$.

Using known techniques it can be pushed down to $\Pi_5^0$.

**Theorem (M. and Ng)**

The index set

$$\{ i : M_i \text{ is a c.c. torsion abelian group} \}$$

is $\Pi_4^0$-complete.

- $\Pi_4^0$-harness of the index set is the easy(er) part.
- The proof relies on several subtle **algebraic reductions**.
- We use that a certain diagonalization attempt on **equivalence structures** must fail.
- **Computable equivalence structures** are in the (scary) combinatorial core of the proof.
From computable groups to Polish groups
A **computable Polish group** is a computable Polish (metric) space equipped with computable group operations.

We consider Polish groups up to topological isomorphism.

Suppose $K$ is a natural class of Polish groups (e.g., connected compact groups).

Can we classify members of $K$?
Definition

A **computable Polish group** is a computable Polish (metric) space equipped with computable group operations.

We consider Polish groups up to topological isomorphism.

Suppose $K$ is a natural class of Polish groups (e.g., connected compact groups).

Can we classify members of $K$?
A computable Polish group is a computable Polish (metric) space equipped with computable group operations.

We consider Polish groups up to topological isomorphism.

Suppose $\mathcal{K}$ is a natural class of Polish groups (e.g., connected compact groups).

Can we classify members of $\mathcal{K}$?
Theorem (M.)

1. The index sets of **profinite** and of **connected compact** Polish groups are arithmetical.
2. The topological isomorphism problems for **profinite abelian groups** and for **connected compact abelian groups** are $\Sigma_1^1$-complete.

We can list all partial computable Polish groups: $G_0, G_1, G_2, \ldots$

- $\{i : G_i \text{ is a connected topological group}\}$ is Arithmetical.
- $\{(i, j) : G_i \cong G_j \text{ and } G_i, G_j \text{ are connected}\}$ is $\Sigma_1^1$-complete.

The result is uniform. It follows connected and profinite (abelian) groups are **unclassifiable**.
Theorem (M.)

1. The index sets of **profinite** and of **connected compact** Polish groups are arithmetical.
2. The topological isomorphism problems for **profinite abelian groups** and for **connected compact abelian** groups are $\Sigma^1_1$-complete.

We can list all partial computable Polish groups: $G_0, G_1, G_2, \ldots$

- $\{i : G_i \text{ is a connected topological group}\}$ is Arithmetical.
- $\{(i, j) : G_i \cong G_j \text{ and } G_i, G_j \text{ are connected}\}$ is $\Sigma^1_1$-complete.

The result is uniform. It follows connected and profinite (abelian) groups are **unclassifiable**.
Theorem (M.)

1. The index sets of **profinite** and of **connected compact** Polish groups are arithmetical.

2. The topological isomorphism problems for **profinite abelian groups** and for **connected compact abelian groups** are $\Sigma^1_1$-complete.

We can list all partial computable Polish groups: $G_0, G_1, G_2, \ldots$

- $\{i : G_i \text{ is a connected topological group}\}$ is Arithmetical.
- $\{(i, j) : G_i \cong G_j \text{ and } G_i, G_j \text{ are connected}\}$ is $\Sigma^1_1$-complete.

The result is uniform. It follows connected and profinite (abelian) groups are **unclassifiable**.
Theorem (M.)

1. The index sets of \textit{profinite} and of \textit{connected compact} Polish groups are arithmetical.

2. The topological isomorphism problems for \textit{profinite abelian groups} and for \textit{connected compact abelian} groups are $\Sigma^1_1$-complete.

The main tools of the proof include:

- Computable Polish space theory.
- Computable (discrete) abelian group theory (e.g., the old result of Dobrica on bases, the result of Downey and Montalban mentioned by Julia, etc.).
- Abstract harmonic analysis.
Theorem (M.)

1. The index sets of **profinite** and of **connected compact** Polish groups are arithmetical.

2. The topological isomorphism problems for **profinite abelian groups** and for **connected compact abelian groups** are \( \Sigma^1_1 \)-complete.

The main tools of the proof include:

- Computable Polish space theory.
- Computable (discrete) abelian group theory (e.g., the old result of Dobrica on bases, the result of Downey and Montalban mentioned by Julia, etc.).
- Abstract harmonic analysis.
Definition (Smith and La Roche, after Nerode)

A profinite group is *recursive* if it is the limit of a computable surjective inverse system of finite groups.

($\hat{G}$ stands for the Pontryagin dual of $G$.)

Theorem (M.)

Let $G$ be a countable torsion abelian group. Then

- $G$ is computable iff $\hat{G}$ is a recursive profinite group;
- $G$ is computably categorical iff $\hat{G}$ is computably categorical (as a recursive profinite group).

Corollary (follows from M. and Ng)

The index set of c.c. recursive profinite groups is $\Pi^0_4$-complete.

eq. structures $\rightarrow$ (discrete) abelian groups $\rightarrow$ Polish groups.
Definition (Smith and La Roche, after Nerode)

A profinite group is *recursive* if it is the limit of a computable surjective inverse system of finite groups.

($\hat{G}$ stands for the Pontryagin dual of $G$.)

Theorem (M.)

Let $G$ be a countable torsion abelian group. Then
- $G$ is computable iff $\hat{G}$ is a recursive profinite group;
- $G$ is computably categorical iff $\hat{G}$ is computably categorical (as a recursive profinite group).

Corollary (follows from M. and Ng)

The index set of c.c. recursive profinite groups is $\Pi^0_4$-complete.
Definition (Smith and La Roche, after Nerode)
A profinite group is *recursive* if it is the limit of a computable surjective inverse system of finite groups.

(\(\hat{G}\) stands for the Pontryagin dual of \(G\).)

Theorem (M.)
Let \(G\) be a countable torsion abelian group. Then
- \(G\) is computable iff \(\hat{G}\) is a recursive profinite group;
- \(G\) is computably categorical iff \(\hat{G}\) is computably categorical (as a recursive profinite group).

Corollary (follows from M. and Ng)
The index set of c.c. recursive profinite groups is \(\Pi^0_4\)-complete.

eq. structures \(\rightarrow\) (discrete) abelian groups \(\rightarrow\) Polish groups.
Thanks!