Computable structure theory and Polish group actions.

Antonio Montalbán

U.C. Berkeley

April 2015
Singapore

(Joint work with Alexander Melnikov)

A generalization

What is the connection between the following two theorems?

A generalization

What is the connection between the following two theorems?

(A) **Thm** [M. 14]

A structure is
uniformly computably categorical on a cone $\iff$ it has a Π_2^{in} Scott Sentence.

A generalization

What is the connection between the following two theorems?

(A) **Thm** [M. 14]

A structure is
uniformly computably categorical on a cone $\iff$ it has a Π_2^{in} Scott Sentence.

(B)**Thm** [Effros 65]

Let $\mathcal{G}$ be a Polish group acting continuously on a Polish space $\mathcal{X}$, and let x be a point in $\mathcal{X}$.
The map $g \mapsto g \cdot x \colon \mathcal{G} \to \mathcal{X}$ is open $\iff$ the orbit of x is G_δ.

A generalization

What is the connection between the following two theorems?

(A) **Thm** [M. 14]

A structure is
uniformly computably categorical on a cone $\iff$ it has a Π_2^{in} Scott Sentence.

(B)**Thm** [Effros 65]

Let $\mathcal{G}$ be a Polish group acting continuously on a Polish space $\mathcal{X}$, and let x be a point in $\mathcal{X}$.
The map $g \mapsto g \cdot x \colon \mathcal{G} \to \mathcal{X}$ is open $\iff$ the orbit of x is G_δ.

Answer: (A) is a particular case of (B).

The idea

Results from computable structure theory can be generalized to the setting of Polish groups acting on Polish spaces.

The idea

Results from computable structure theory can be generalized to the setting of Polish groups acting on Polish spaces.

The idea of looking at countable structures in the setting of Polish groups actions existed in descriptive set theory. [Becker, Gao, Hjorth, Kechris,...]

The idea

Results from computable structure theory can be generalized to the setting of Polish groups acting on Polish spaces.

The idea of looking at countable structures in the setting of Polish groups actions existed in descriptive set theory. [Becker, Gao, Hjorth, Kechris,...]

We analyze the following theorems:

The idea

Results from computable structure theory can be generalized to the setting of Polish groups acting on Polish spaces.

The idea of looking at countable structures in the setting of Polish groups actions existed in descriptive set theory. [Becker, Gao, Hjorth, Kechris,...]

We analyze the following theorems:

1. [M. 14] Characterization of uniform computable categoricity on a cone.
2. [M. 14] Characterization of computable categoricity on a cone.
3. [McCoy 02] Proper finite dimension does not relativize.
4. [Knight et al. 90's] No degree spectrum is the union of two cones.
5. [Goncharov 80's] Δ^0_2- but not Δ^0_1-isomorphic structures have ∞ dim.

Part 1:

Background on Polish group actions.

The space of structures

The space of structures

Fix a computable vocabulary L.

The space of structures

Fix a computable vocabulary L.

Definition

Let $Mod(L)$ be the set of all L-structures with domain ω.

The space of structures

Fix a computable vocabulary L.

Definition

Let $Mod(L)$ be the set of all L-structures with domain ω.
We give $Mod(L)$ the topology generated by the basic open sets

$$[\varphi] = \{\mathcal{A} \in Mod(L) : \mathcal{A} \models \varphi\}$$

where φ is an atomic $(L \cup \text{Constants}_{\mathbb{N}})$-sentence and $\text{Constants}_{\mathbb{N}} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}, ...\}$.

The space of structures

Fix a computable vocabulary L.

Definition

Let *$Mod(L)$* be the set of all L-structures with domain ω.
We give $Mod(L)$ the topology generated by the basic open sets

$$[\varphi] = \{\mathcal{A} \in Mod(L) : \mathcal{A} \models \varphi\}$$

where φ is an atomic $(L \cup \text{Constants}_{\mathbb{N}})$-sentence and $\text{Constants}_{\mathbb{N}} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}, ...\}$.

Equivalentely:

Let $D\colon Mod(L) \to 2^\omega$ map $\mathcal{A} \in Mod(L)$ to its *atomic diagram* $D(\mathcal{A}) \in 2^\omega$.

The space of structures

Fix a computable vocabulary L.

Definition

Let *Mod*(L) be the set of all L-structures with domain ω.
We give $Mod(L)$ the topology generated by the basic open sets

$$[\varphi] = \{\mathcal{A} \in Mod(L) : \mathcal{A} \models \varphi\}$$

where φ is an atomic $(L \cup \text{Constants}_{\mathbb{N}})$-sentence and $\text{Constants}_{\mathbb{N}} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}, ...\}$.

Equivalentely:

Let $D\colon Mod(L) \to 2^\omega$ map $\mathcal{A} \in Mod(L)$ to its *atomic diagram* $D(\mathcal{A}) \in 2^\omega$.
The topology of $Mod(L)$ is so that $Mod(L)$ is homeomorphic to its image.

$Mod(L)$ is an effective Polish space

Definition

A topological space $\mathcal{X}$ is *Polish* if

$Mod(L)$ is an effective Polish space

Definition

A topological space $\mathcal{X}$ is *Polish* if

- It has a countable dense subset $\{x_0, x_1, x_2, \ldots\}$, and

$Mod(L)$ is an effective Polish space

Definition

A topological space $\mathcal{X}$ is *Polish* if

- It has a countable dense subset $\{x_0, x_1, x_2, ...\}$, and
- it admits a complete metric $d \colon X \times X \to \mathbb{R}^{\geq 0}$.

$Mod(L)$ is an effective Polish space

Definition

A topological space $\mathcal{X}$ is *Polish* if

- It has a countable dense subset $\{x_0, x_1, x_2, ...\}$, and
- it admits a complete metric $d \colon X \times X \to \mathbb{R}^{\geq 0}$.

$\mathcal{X}$ is *effectively Polish* if also d is computable on $\{x_0, x_1 ...\}$,

$Mod(L)$ is an effective Polish space

Definition

A topological space $\mathcal{X}$ is *Polish* if

- It has a countable dense subset $\{x_0, x_1, x_2, ...\}$, and
- it admits a complete metric $d \colon X \times X \to \mathbb{R}^{\geq 0}$.

$\mathcal{X}$ is *effectively Polish* if also d is computable on $\{x_0, x_1 ...\}$,

i.e., the questions $d(x_i, x_j) < q$ and $d(x_i, x_j) \leq q$ are decidable.

$Mod(L)$ is an effective Polish space

Definition

A topological space $\mathcal{X}$ is *Polish* if

- It has a countable dense subset $\{x_0, x_1, x_2, ...\}$, and
- it admits a complete metric $d \colon X \times X \to \mathbb{R}^{\geq 0}$.

$\mathcal{X}$ is *effectively Polish* if also d is computable on $\{x_0, x_1 ...\}$,

i.e., the questions $d(x_i, x_j) < q$ and $d(x_i, x_j) \leq q$ are decidable.

Obs: For a computable vocabulary L, $Mod(L)$ is a effectively Polish.

$Mod(L)$ is an effective Polish space

Definition

A topological space $\mathcal{X}$ is *Polish* if

- It has a countable dense subset $\{x_0, x_1, x_2, ...\}$, and
- it admits a complete metric $d\colon X \times X \to \mathbb{R}^{\geq 0}$.

$\mathcal{X}$ is *effectively Polish* if also d is computable on $\{x_0, x_1 ...\}$,

i.e., the questions $d(x_i, x_j) < q$ and $d(x_i, x_j) \leq q$ are decidable.

Obs: For a computable vocabulary L, $Mod(L)$ is a effectively Polish.

We represent *points* in $\mathcal{X}$ by fast Cauchy sequences from $\{x_0, x_1 ...\}$.

$Mod(L)$ is an effective Polish space

Definition

A topological space $\mathcal{X}$ is *Polish* if

- It has a countable dense subset $\{x_0, x_1, x_2, \dots\}$, and
- it admits a complete metric $d\colon X \times X \to \mathbb{R}^{\geq 0}$.

$\mathcal{X}$ is *effectively Polish* if also d is computable on $\{x_0, x_1 \dots\}$,

i.e., the questions $d(x_i, x_j) < q$ and $d(x_i, x_j) \leq q$ are decidable.

Obs: For a computable vocabulary L, $Mod(L)$ is a effectively Polish.

We represent *points* in $\mathcal{X}$ by fast Cauchy sequences from $\{x_0, x_1 \dots\}$.
Def: A *point is computable* if the sequence is computable and fast approaching.

$Mod(L)$ is an effective Polish space

Definition

A topological space $\mathcal{X}$ is *Polish* if

- It has a countable dense subset $\{x_0, x_1, x_2, ...\}$, and
- it admits a complete metric $d\colon X \times X \to \mathbb{R}^{\geq 0}$.

$\mathcal{X}$ is *effectively Polish* if also d is computable on $\{x_0, x_1 ...\}$,

i.e., the questions $d(x_i, x_j) < q$ and $d(x_i, x_j) \leq q$ are decidable.

Obs: For a computable vocabulary L, $Mod(L)$ is a effectively Polish.

We represent *points* in $\mathcal{X}$ by fast Cauchy sequences from $\{x_0, x_1 ...\}$.
Def: A *point is computable* if the sequence is computable and fast approaching.

Fact: $F\colon \mathcal{X} \to \mathcal{Y}$ is continuous $\iff$ it is *computable* relative to some oracle.

The permutation group of ω

Definition

Let S_∞ be the permutation group of ω. (I.e., the group of all bijections $\omega \to \omega$.)

The permutation group of ω

Definition

Let S_∞ be the permutation group of ω. (I.e., the group of all bijections $\omega \to \omega$.)

With the topology inherited from ω^ω, S_∞ is an *effective Polish group*,

The permutation group of ω

Definition

Let S_∞ be the permutation group of ω. (I.e., the group of all bijections $\omega \to \omega$.)

With the topology inherited from ω^ω, S_∞ is an *effective Polish group*, i.e., it's an effective Polish space where the group operations are computable.

The permutation group of ω

Definition

Let S_∞ be the permutation group of ω. (I.e., the group of all bijections $\omega \to \omega$.)

With the topology inherited from ω^ω, S_∞ is an *effective Polish group*, i.e., it's an effective Polish space where the group operations are computable.

Definition

$\mathcal{S}_\infty$ *acts* on $Mod(L)$ in an obvious way.

The permutation group of ω

Definition

Let S_∞ be the permutation group of ω. (I.e., the group of all bijections $\omega \to \omega$.)

With the topology inherited from ω^ω, S_∞ is an *effective Polish group*,
i.e., it's an effective Polish space where the group operations are computable.

Definition

$\mathcal{S}_\infty$ *acts* on $Mod(L)$ in an obvious way.
For $\mathcal{A} \in Mod(L)$, $f \in S_\infty$, $f \cdot \mathcal{A}$ is the structure $\mathcal{B}$ such that

$$(n_1, ..., n_k) \in R^{\mathcal{A}} \iff (f(n_1), ..., f(n_k)) \in R^{\mathcal{B}}.$$

The permutation group of ω

Definition

Let S_∞ be the permutation group of ω. (I.e., the group of all bijections $\omega \to \omega$.)

With the topology inherited from ω^ω, S_∞ is an *effective Polish group*, i.e., it's an effective Polish space where the group operations are computable.

Definition

$\mathcal{S}_\infty$ *acts* on $Mod(L)$ in an obvious way.

For $\mathcal{A} \in Mod(L)$, $f \in S_\infty$, $f \cdot \mathcal{A}$ is the structure $\mathcal{B}$ such that

$$(n_1, ..., n_k) \in R^{\mathcal{A}} \iff (f(n_1), ..., f(n_k)) \in R^{\mathcal{B}}.$$

Obs: This action, $: S_\infty \times Mod(L) \to Mod(L)$, is computable.

Effective Polish group actions

Throughout the rest of the talk

- $\mathcal{G}$ is an effective Polish group,
- $\mathcal{X}$ is an effective Polish space, and
- $\mathcal{G}$ acts on $\mathcal{X}$ computably.

Effective Polish group actions

Throughout the rest of the talk

- $\mathcal{G}$ is an effective Polish group,
- $\mathcal{X}$ is an effective Polish space, and
- $\mathcal{G}$ acts on $\mathcal{X}$ computably.

Definition

For $x, y \in \mathcal{X}$, we let $x \equiv y \iff (\exists g \in \mathcal{G})\ g \cdot x = y$.

Effective Polish group actions

Throughout the rest of the talk

- $\mathcal{G}$ is an effective Polish group,
- $\mathcal{X}$ is an effective Polish space, and
- $\mathcal{G}$ acts on $\mathcal{X}$ computably.

Definition

For $x, y \in \mathcal{X}$, we let $x \equiv y \iff (\exists g \in \mathcal{G})\ g \cdot x = y$.
We let the *$\mathcal{G}$-orbit of x* be $\{y \in \mathcal{X} : y \equiv x\}$

Effective Polish group actions

Throughout the rest of the talk

- $\mathcal{G}$ is an effective Polish group,
- $\mathcal{X}$ is an effective Polish space, and
- $\mathcal{G}$ acts on $\mathcal{X}$ computably.

Definition

For $x, y \in \mathcal{X}$, we let $x \equiv y \iff (\exists g \in \mathcal{G})\ g \cdot x = y$.
We let the *$\mathcal{G}$-orbit of x* be $\{y \in \mathcal{X} : y \equiv x\} = \mathcal{G} \cdot x$.

Effective Polish group actions

Throughout the rest of the talk

- $\mathcal{G}$ is an effective Polish group,
- $\mathcal{X}$ is an effective Polish space, and
- $\mathcal{G}$ acts on $\mathcal{X}$ computably.

Definition

For $x, y \in \mathcal{X}$, we let $x \equiv y \iff (\exists g \in \mathcal{G})\ g \cdot x = y$.
We let the *$\mathcal{G}$-orbit of x* be $\{y \in \mathcal{X} : y \equiv x\} = \mathcal{G} \cdot x$.

Note: In the case of S_∞ acting on $Mod(L)$, $\mathcal{A} \equiv \mathcal{B} \iff \mathcal{A} \cong \mathcal{B}$.

Other examples of computable Polish group actions

The following are examples of computable Polish group actions:

- GL_n acting on $\mathbb{R}^n$.
- Any computable Polish group acting on itself by congugation.
- $Hom^+[0,1]$ acting on $C[0,1]$ by right composition (using sup norm).

Part 2:

Theorems from computable structure theory.

Computable categoricity

Theorem ((2) [Scott 65; Lopez-Escobar 65; Goncharov 75; M. 14]**)**

For a structure $\mathcal{A}$, the following are equivalent:

1. *The set $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $\boldsymbol{\Sigma}^0_3$.*
2. *$\mathcal{A}$ is computably categorical on a cone.*
3. *$\mathcal{A}$ has a Scott family of $\exists$-formulas with parameters.*
4. *$\mathcal{A}$ has a Σ^{in}_3 Scott sentence.*

Computable categoricity

Definition A structure $\mathcal{A}$ is *computably categorical (c.c.)* if

every computable $\mathcal{B}$ isomorphic to $\mathcal{A}$ is computably isomorphic to $\mathcal{A}$.

Computable categoricity

Definition A structure $\mathcal{A}$ is *computably categorical (c.c.)* if

every computable $\mathcal{B}$ isomorphic to $\mathcal{A}$ is computably isomorphic to $\mathcal{A}$.

- A linear ordering is c.c. $\iff$ it has finitely many adjacencies [Dzgoev, Goncharov 80].

Computable categoricity

Definition A structure $\mathcal{A}$ is *computably categorical (c.c.)* if

every computable $\mathcal{B}$ isomorphic to $\mathcal{A}$ is computably isomorphic to $\mathcal{A}$.

- A linear ordering is c.c. $\iff$ it has finitely many adjacencies [Dzgoev, Goncharov 80].
- A Boolean algebras is c.c. $\iff$ it has finitely many atoms [Goncharov][La Roche 78].

Computable categoricity

Definition A structure $\mathcal{A}$ is *computably categorical (c.c.)* if

every computable $\mathcal{B}$ isomorphic to $\mathcal{A}$ is computably isomorphic to $\mathcal{A}$.

- A linear ordering is c.c. $\iff$ it has finitely many adjacencies [Dzgoev, Goncharov 80].
- A Boolean algebras is c.c. $\iff$ it has finitely many atoms [Goncharov][La Roche 78].
- A ordered abelian group is c.c. $\iff$ it has finite rank [Goncharov, Lempp, and Solomon 03].

Computable categoricity

Definition A structure $\mathcal{A}$ is *computably categorical (c.c.)* if

every computable $\mathcal{B}$ isomorphic to $\mathcal{A}$ is computably isomorphic to $\mathcal{A}$.

- A linear ordering is c.c. $\iff$ it has finitely many adjacencies [Dzgoev, Goncharov 80].
- A Boolean algebras is c.c. $\iff$ it has finitely many atoms [Goncharov][La Roche 78].
- A ordered abelian group is c.c. $\iff$ it has finite rank [Goncharov, Lempp, and Solomon 03].
- A tree of finite height is c.c. $\iff$ it is of finite type [Lempp, McCoy, R. Miller, Solomon 05].

Computable categoricity

Definition A structure $\mathcal{A}$ is *computably categorical (c.c.)* if

every computable $\mathcal{B}$ isomorphic to $\mathcal{A}$ is computably isomorphic to $\mathcal{A}$.

- A linear ordering is c.c. $\iff$ it has finitely many adjacencies [Dzgoev, Goncharov 80].
- A Boolean algebras is c.c. $\iff$ it has finitely many atoms [Goncharov][La Roche 78].
- A ordered abelian group is c.c. $\iff$ it has finite rank [Goncharov, Lempp, and Solomon 03].
- A tree of finite height is c.c. $\iff$ it is of finite type [Lempp, McCoy, R. Miller, Solomon 05].
- A torsion-free abelian group is c.c. $\iff$ it has finite rank [Nurtazin 74].

Computable categoricity

Definition A structure $\mathcal{A}$ is *computably categorical (c.c.)* if

every computable $\mathcal{B}$ isomorphic to $\mathcal{A}$ is computably isomorphic to $\mathcal{A}$.

- A linear ordering is c.c. $\iff$ it has finitely many adjacencies [Dzgoev, Goncharov 80].
- A Boolean algebras is c.c. $\iff$ it has finitely many atoms [Goncharov][La Roche 78].
- A ordered abelian group is c.c. $\iff$ it has finite rank [Goncharov, Lempp, and Solomon 03].
- A tree of finite height is c.c. $\iff$ it is of finite type [Lempp, McCoy, R. Miller, Solomon 05].
- A torsion-free abelian group is c.c. $\iff$ it has finite rank [Nurtazin 74].
- A computable p-group is c.c. $\iff$ it can be written in one of the following forms: (i) $(\mathbb{Z}(p^{\infty}))^{\ell} \oplus G$ for $\ell \in \omega \cup \{\infty\}$ and G finite, or (ii) $(\mathbb{Z}(p^{\infty}))^{n} \oplus (\mathbb{Z}_{p^k})^{\infty} \oplus G$ where G is finite, and $n, k \in \omega$ [Goncharov 80][Smith81].

Computable categoricity

Definition A structure $\mathcal{A}$ is *computably categorical (c.c.)* if

every computable $\mathcal{B}$ isomorphic to $\mathcal{A}$ is computably isomorphic to $\mathcal{A}$.

- A linear ordering is c.c. $\iff$ it has finitely many adjacencies [Dzgoev, Goncharov 80].
- A Boolean algebras is c.c. $\iff$ it has finitely many atoms [Goncharov][La Roche 78].
- A ordered abelian group is c.c. $\iff$ it has finite rank [Goncharov, Lempp, and Solomon 03].
- A tree of finite height is c.c. $\iff$ it is of finite type [Lempp, McCoy, R. Miller, Solomon 05].
- A torsion-free abelian group is c.c. $\iff$ it has finite rank [Nurtazin 74].
- A computable p-group is c.c. $\iff$ it can be written in one of the following forms: (i) $(\mathbb{Z}(p^\infty))^\ell \oplus G$ for $\ell \in \omega \cup \{\infty\}$ and G finite, or (ii) $(\mathbb{Z}(p^\infty))^n \oplus (\mathbb{Z}_{p^k})^\infty \oplus G$ where G is finite, and $n, k \in \omega$ [Goncharov 80][Smith81].

Theorem ([Downey, Kach, Lempp, Lewis, Montalbán, Turetsky 12])

There is no nice characterization of computably categorical structures.

Computable categoricity

Definition A structure $\mathcal{A}$ is *computably categorical (c.c.)* if

every computable $\mathcal{B}$ isomorphic to $\mathcal{A}$ is computably isomorphic to $\mathcal{A}$.

- A linear ordering is c.c. $\iff$ it has finitely many adjacencies [Dzgoev, Goncharov 80].
- A Boolean algebras is c.c. $\iff$ it has finitely many atoms [Goncharov][La Roche 78].
- A ordered abelian group is c.c. $\iff$ it has finite rank [Goncharov, Lempp, and Solomon 03].
- A tree of finite height is c.c. $\iff$ it is of finite type [Lempp, McCoy, R. Miller, Solomon 05].
- A torsion-free abelian group is c.c. $\iff$ it has finite rank [Nurtazin 74].
- A computable p-group is c.c. $\iff$ it can be written in one of the following forms: (i) $(\mathbb{Z}(p^\infty))^\ell \oplus G$ for $\ell \in \omega \cup \{\infty\}$ and G finite, or (ii) $(\mathbb{Z}(p^\infty))^n \oplus (\mathbb{Z}_{p^k})^\infty \oplus G$ where G is finite, and $n, k \in \omega$ [Goncharov 80][Smith81].

Theorem ([Downey, Kach, Lempp, Lewis, Montalbán, Turetsky 12])

There is no nice characterization of computably categorical structures.
The set of indices of computably categorical structures is Π^1_1-complete.

Computable categoricity

Definition A structure $\mathcal{A}$ is *computably categorical (c.c.)* if

every computable $\mathcal{B}$ isomorphic to $\mathcal{A}$ is computably isomorphic to $\mathcal{A}$.

- A linear ordering is c.c. $\iff$ it has finitely many adjacencies [Dzgoev, Goncharov 80].
- A Boolean algebras is c.c. $\iff$ it has finitely many atoms [Goncharov][La Roche 78].
- A ordered abelian group is c.c. $\iff$ it has finite rank [Goncharov, Lempp, and Solomon 03].
- A tree of finite height is c.c. $\iff$ it is of finite type [Lempp, McCoy, R. Miller, Solomon 05].
- A torsion-free abelian group is c.c. $\iff$ it has finite rank [Nurtazin 74].
- A computable p-group is c.c. $\iff$ it can be written in one of the following forms: (i) $(\mathbb{Z}(p^\infty))^\ell \oplus G$ for $\ell \in \omega \cup \{\infty\}$ and G finite, or (ii) $(\mathbb{Z}(p^\infty))^n \oplus (\mathbb{Z}_{p^k})^\infty \oplus G$ where G is finite, and $n, k \in \omega$ [Goncharov 80][Smith81].

Theorem ([Downey, Kach, Lempp, Lewis, Montalbán, Turetsky 12])

There is no nice characterization of computably categorical structures.
The set of indices of computably categorical structures is Π^1_1-complete.

Nice characterizations exist if we relativize to all oracles on a cone.

Computable categoricity

Definition A structure $\mathcal{A}$ is *computably categorical (c.c.) on a cone* if,
there is a $C \in 2^\omega$ such that for very $Z \geq_T C$,
every Z-computable $\mathcal{B}$ isomorphic to $\mathcal{A}$ is Z-computably isomorphic to $\mathcal{A}$.

- A linear ordering is c.c. $\iff$ it has finitely many adjacencies [Dzgoev, Goncharov 80].
- A Boolean algebras is c.c. $\iff$ it has finitely many atoms [Goncharov][La Roche 78].
- A ordered abelian group is c.c. $\iff$ it has finite rank [Goncharov, Lempp, and Solomon 03].
- A tree of finite height is c.c. $\iff$ it is of finite type [Lempp, McCoy, R. Miller, Solomon 05].
- A torsion-free abelian group is c.c. $\iff$ it has finite rank [Nurtazin 74].
- A computable p-group is c.c. $\iff$ it can be written in one of the following forms: (i) $(\mathbb{Z}(p^\infty))^\ell \oplus G$ for $\ell \in \omega \cup \{\infty\}$ and G finite, or (ii) $(\mathbb{Z}(p^\infty))^n \oplus (\mathbb{Z}_{p^k})^\infty \oplus G$ where G is finite, and $n, k \in \omega$ [Goncharov 80][Smith81].

Theorem ([Downey, Kach, Lempp, Lewis, Montalbán, Turetsky 12])

There is no nice characterization of computably categorical structures.
The set of indices of computably categorical structures is Π^1_1-complete.

Nice characterizations exist if we relativize to all oracles on a cone.

Categoricity on group actions

Recall: A computable structure $\mathcal{A} \in Mod(L)$ is *computably categorical* if for every computable $\mathcal{B} \cong \mathcal{A}$, there is a computable $g \in S_\infty$ with $g \cdot \mathcal{A} = \mathcal{B}$.

Categoricity on group actions

Recall: A computable structure $\mathcal{A} \in Mod(L)$ is *computably categorical* if for every computable $\mathcal{B} \cong \mathcal{A}$, there is a computable $g \in S_\infty$ with $g{\cdot}\mathcal{A} = \mathcal{B}$.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Categoricity on group actions

Recall: A computable structure $\mathcal{A} \in Mod(L)$ is *computably categorical* if for every computable $\mathcal{B} \cong \mathcal{A}$, there is a computable $g \in S_\infty$ with $g \cdot \mathcal{A} = \mathcal{B}$.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Definition: A point $x \in \mathcal{X}$ is *computably categorical* if for every computable $y \equiv x$, there is a computable $g \in \mathcal{G}$ with $g \cdot x = y$.

Categoricity on group actions

Recall: A computable structure $\mathcal{A} \in Mod(L)$ is *computably categorical* if for every computable $\mathcal{B} \cong \mathcal{A}$, there is a computable $g \in S_\infty$ with $g \cdot \mathcal{A} = \mathcal{B}$.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Definition: A point $x \in \mathcal{X}$ is *computably categorical* if for every computable $y \equiv x$, there is a computable $g \in \mathcal{G}$ with $g \cdot x = y$.

Example: Work out $Hom^+[0,1]$ acting on $C[0,1]$ by right composition.

Categoricity on group actions

Recall: A computable structure $\mathcal{A} \in Mod(L)$ is *computably categorical* if for every computable $\mathcal{B} \cong \mathcal{A}$, there is a computable $g \in S_\infty$ with $g \cdot \mathcal{A} = \mathcal{B}$.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Definition: A point $x \in \mathcal{X}$ is *computably categorical* if for every computable $y \equiv x$, there is a computable $g \in \mathcal{G}$ with $g \cdot x = y$.

Example: Work out $Hom^+[0,1]$ acting on $C[0,1]$ by right composition.

- x^2 and $\sin(4\pi x)$ are computably categorical.

Categoricity on group actions

Recall: A computable structure $\mathcal{A} \in Mod(L)$ is *computably categorical* if for every computable $\mathcal{B} \cong \mathcal{A}$, there is a computable $g \in S_\infty$ with $g \cdot \mathcal{A} = \mathcal{B}$.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Definition: A point $x \in \mathcal{X}$ is *computably categorical* if for every computable $y \equiv x$, there is a computable $g \in \mathcal{G}$ with $g \cdot x = y$.

Example: Work out $Hom^+[0,1]$ acting on $C[0,1]$ by right composition.

- x^2 and $\sin(4\pi x)$ are computably categorical.
- One can build computable functions that are not.

Categoricity on group actions

Recall: A computable structure $\mathcal{A} \in Mod(L)$ is *computably categorical* if for every computable $\mathcal{B} \cong \mathcal{A}$, there is a computable $g \in S_\infty$ with $g \cdot \mathcal{A} = \mathcal{B}$.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Definition: A point $x \in \mathcal{X}$ is *computably categorical* if for every computable $y \equiv x$, there is a computable $g \in \mathcal{G}$ with $g \cdot x = y$.

Example: Work out $Hom^+[0,1]$ acting on $C[0,1]$ by right composition.

- x^2 and $\sin(4\pi x)$ are computably categorical.
- One can build computable functions that are not.

Definition: A point $x \in \mathcal{X}$ is *computably categorical on a cone* if there is a $C \in 2^\omega$ such that for very $Z \geq_T C$, for every Z-computable $y \equiv x$, there is a Z-computable $g \in \mathcal{G}$ with $g \cdot x = y$.

Back to Theorem 2

Theorem ((2) [Lopez-Escobar 65; Scott 65; Goncharov 75; M. 14])

For a structure $\mathcal{A}$, the following are equivalent:

1. *$\mathcal{A}$ is computably categorical on a cone.*
2. *$\mathcal{A}$ has a Scott family of $\exists$-formulas with parameters.*
3. *$\mathcal{A}$ has a Σ_3^{in} Scott sentence.*
4. *The set $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $\mathbf{\Sigma_3^0}$.*

Back to Theorem 2

Theorem ((2) [Lopez-Escobar 65; Scott 65; Goncharov 75; M. 14])

For a structure $\mathcal{A}$, the following are equivalent:

1. *$\mathcal{A}$ is computably categorical on a cone.*
2. *$\mathcal{A}$ has a Scott family of $\exists$-formulas with parameters.*
3. *$\mathcal{A}$ has a Σ_3^{in} Scott sentence.*
4. *The set $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $\mathbf{\Sigma_3^0}$.*

(2) A Scott family is a set of formulas defining the automorphism orbits of all the tuples.

Back to Theorem 2

Theorem ((2) [Lopez-Escobar 65; Scott 65; Goncharov 75; M. 14])

For a structure $\mathcal{A}$, the following are equivalent:

1. *$\mathcal{A}$ is computably categorical on a cone.*
2. *$\mathcal{A}$ has a Scott family of $\exists$-formulas with parameters.*
3. *$\mathcal{A}$ has a Σ_3^{in} Scott sentence.*
4. *The set $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $\mathbf{\Sigma_3^0}$.*

(2) A Scott family is a set of formulas defining the automorphism orbits of all the tuples.
(1) $\iff$ (2) is due to [Goncharov 75].

Back to Theorem 2

Theorem ((2) [Lopez-Escobar 65; Scott 65; Goncharov 75; M. 14])

For a structure $\mathcal{A}$, the following are equivalent:

1. *$\mathcal{A}$ is computably categorical on a cone.*
2. *$\mathcal{A}$ has a Scott family of $\exists$-formulas with parameters.*
3. *$\mathcal{A}$ has a Σ_3^{in} Scott sentence.*
4. *The set $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $\mathbf{\Sigma_3^0}$.*

(2) A Scott family is a set of formulas defining the automorphism orbits of all the tuples.
(1) $\iff$ (2) is due to [Goncharov 75].
(3) A Scott sentence one that determines $\mathcal{A}$ up to isomorphism, within $Mod(L)$.

Back to Theorem 2

Theorem ((2) [Lopez-Escobar 65; Scott 65; Goncharov 75; M. 14])

For a structure $\mathcal{A}$, the following are equivalent:

1. *$\mathcal{A}$ is computably categorical on a cone.*
2. *$\mathcal{A}$ has a Scott family of $\exists$-formulas with parameters.*
3. *$\mathcal{A}$ has a Σ_3^{in} Scott sentence.*
4. *The set $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $\mathbf{\Sigma_3^0}$.*

(2) A Scott family is a set of formulas defining the automorphism orbits of all the tuples.
(1) $\iff$ (2) is due to [Goncharov 75].
(3) A Scott sentence one that determines $\mathcal{A}$ up to isomorphism, within $Mod(L)$.
(2) $\implies$ (3) is due to [Scott 65]

Back to Theorem 2

Theorem ((2) [Lopez-Escobar 65; Scott 65; Goncharov 75; M. 14])

For a structure $\mathcal{A}$, the following are equivalent:

1. *$\mathcal{A}$ is computably categorical on a cone.*
2. *$\mathcal{A}$ has a Scott family of $\exists$-formulas with parameters.*
3. *$\mathcal{A}$ has a Σ_3^{in} Scott sentence.*
4. *The set $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $\mathbf{\Sigma_3^0}$.*

(2) A Scott family is a set of formulas defining the automorphism orbits of all the tuples.
(1) $\iff$ (2) is due to [Goncharov 75].
(3) A Scott sentence one that determines $\mathcal{A}$ up to isomorphism, within $Mod(L)$.
(2) $\implies$ (3) is due to [Scott 65]
(2) $\impliedby$ (3) [M. 14] uses sharp version of the type-omitting theorem for $L_{\omega_1,\omega}$.

Back to Theorem 2

Theorem ((2) [Lopez-Escobar 65; Scott 65; Goncharov 75; M. 14])

For a structure $\mathcal{A}$, the following are equivalent:

1. *$\mathcal{A}$ is computably categorical on a cone.*
2. *$\mathcal{A}$ has a Scott family of $\exists$-formulas with parameters.*
3. *$\mathcal{A}$ has a Σ_3^{in} Scott sentence.*
4. *The set $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $\mathbf{\Sigma_3^0}$.*

(2) A Scott family is a set of formulas defining the automorphism orbits of all the tuples.
(1) $\iff$ (2) is due to [Goncharov 75].
(3) A Scott sentence one that determines $\mathcal{A}$ up to isomorphism, within $Mod(L)$.
(2) $\implies$ (3) is due to [Scott 65]
(2) $\impliedby$ (3) [M. 14] uses sharp version of the type-omitting theorem for $L_{\omega_1,\omega}$.
(3) $\iff$ (4) is due to [Lopez-Escobar 65].

Back to Theorem 2

Theorem ((2) [Lopez-Escobar 65; Scott 65; Goncharov 75; M. 14])

For a structure $\mathcal{A}$, the following are equivalent:

1. *$\mathcal{A}$ is computably categorical on a cone.*
2. *$\mathcal{A}$ has a Scott family of $\exists$-formulas with parameters.*
3. *$\mathcal{A}$ has a Σ_3^{in} Scott sentence.*
4. *The set $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $\mathbf{\Sigma^0_3}$.*

(2) A Scott family is a set of formulas defining the automorphism orbits of all the tuples.
(1) $\iff$ (2) is due to [Goncharov 75].
(3) A Scott sentence one that determines $\mathcal{A}$ up to isomorphism, within $Mod(L)$.
(2) $\implies$ (3) is due to [Scott 65]
(2) $\impliedby$ (3) [M. 14] uses sharp version of the type-omitting theorem for $L_{\omega_1,\omega}$.
(3) $\iff$ (4) is due to [Lopez-Escobar 65].

Question: If we have a computable Polish action of $\mathcal{G}$ on $\mathcal{X}$, do we have that

$$x \in \mathcal{X} \text{ is computably categorical on a cone} \iff \text{its orbit is } \mathbf{\Sigma^0_3}.$$

A simpler question

Theorem ((1) [Lopez-Escobar 65; Scott 65; Goncharov 75; Ventsov 93; M. 14])

For a structure $\mathcal{A}$, the following are equivalent:

1. *$\mathcal{A}$ is uniformly computably categorical on a cone.*
2. *$\mathcal{A}$ has a Scott family without parameters.*
3. *$\mathcal{A}$ has a Π^{in}_2 Scott sentence.*
4. *The set $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $\mathbf{\Pi^0_2}$.*

A simpler question

Theorem ((1) [Lopez-Escobar 65; Scott 65; Goncharov 75; Ventsov 93; M. 14]**)**

For a structure $\mathcal{A}$, the following are equivalent:

1. *$\mathcal{A}$ is uniformly computably categorical on a cone.*
2. *$\mathcal{A}$ has a Scott family without parameters.*
3. *$\mathcal{A}$ has a Π^{in}_2 Scott sentence.*
4. *The set $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $\mathbf{\Pi^0_2}$.*

Definition: A point $x \in \mathcal{X}$ is *uniformly computably categorical* if there is a computable operator Φ that,

given a fast Cauchy sequence for $y \equiv x$, outputs $g \in \mathcal{G}$ with $g{\cdot}x = y$.

A simpler question

Theorem ((1) [Lopez-Escobar 65; Scott 65; Goncharov 75; Ventsov 93; M. 14])

For a structure $\mathcal{A}$, the following are equivalent:

1. *$\mathcal{A}$ is uniformly computably categorical on a cone.*
2. *$\mathcal{A}$ has a Scott family without parameters.*
3. *$\mathcal{A}$ has a Π^{in}_2 Scott sentence.*
4. *The set $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $\mathbf{\Pi^0_2}$.*

Definition: A point $x \in \mathcal{X}$ is *uniformly computably categorical* if there is a computable operator Φ that,

given a fast Cauchy sequence for $y \equiv x$, outputs $g \in \mathcal{G}$ with $g \cdot x = y$.

Question: If we have a computable Polish action of $\mathcal{G}$ on $\mathcal{X}$, do we have that $x \in \mathcal{X}$ is uniformly computably categorical on a cone $\iff$ its orbit is G_δ.

Uniformly computable categoricity

Definition: A point $x \in \mathcal{X}$ is *uniformly computably categorical* if there is a computable operator Φ that,

given a fast Cauchy sequence for $y \equiv x$, outputs $g \in \mathcal{G}$ with $g \cdot x = y$.

Uniformly computable categoricity

Definition: A point $x \in \mathcal{X}$ is *uniformly computably categorical* if there is a computable operator Φ that,
given a fast Cauchy sequence for $y \equiv x$, outputs $g \in \mathcal{G}$ with $g{\cdot}x = y$.

Lemma $x \in \mathcal{X}$ is uniformly computably categorical $\iff$
the map $g \mapsto g \cdot x \colon \mathcal{G} \to \mathcal{X}$ is effectively open.

Uniformly computable categoricity

Definition: A point $x \in \mathcal{X}$ is *uniformly computably categorical* if there is a computable operator Φ that,

given a fast Cauchy sequence for $y \equiv x$, outputs $g \in \mathcal{G}$ with $g{\cdot}x = y$.

Lemma $x \in \mathcal{X}$ is uniformly computably categorical on a cone $\iff$ the map $g \mapsto g \cdot x \colon \mathcal{G} \to \mathcal{X}$ is open.

Uniformly computable categoricity

Definition: A point $x \in \mathcal{X}$ is *uniformly computably categorical* if there is a computable operator Φ that,
given a fast Cauchy sequence for $y \equiv x$, outputs $g \in \mathcal{G}$ with $g{\cdot}x = y$.

Lemma $x \in \mathcal{X}$ is uniformly computably categorical on a cone $\iff$
the map $g \mapsto g \cdot x \colon \mathcal{G} \to \mathcal{X}$ is open.

Theorem [Effross 65] For a point $x \in \mathcal{X}$, TFAE:

1. The $\mathcal{G}$-orbit of x is G_δ.
2. the map $f \mapsto f \cdot x \colon G \to \mathcal{X}$ is open.

Uniformly computable categoricity

Definition: A point $x \in \mathcal{X}$ is *uniformly computably categorical* if there is a computable operator Φ that,
given a fast Cauchy sequence for $y \equiv x$, outputs $g \in \mathcal{G}$ with $g \cdot x = y$.

Lemma $x \in \mathcal{X}$ is uniformly computably categorical on a cone $\iff$ the map $g \mapsto g \cdot x \colon \mathcal{G} \to \mathcal{X}$ is open.

Theorem [Effross 65] For a point $x \in \mathcal{X}$, TFAE:

1. The $\mathcal{G}$-orbit of x is G_δ.
2. the map $f \mapsto f \cdot x \colon G \to \mathcal{X}$ is open.

Corollary (1) For a structure $\mathcal{A}$, TFAE:

1. $\mathcal{A}$ has a Π_2^{in} Scott sentence.
2. $\mathcal{A}$ is uniformly computably categorical on a cone.

Back to Theorem 2

Theorem (2) [L-E 65; G. 75; M. 14] For a structure $\mathcal{A}$, TFAE:

1. $\mathcal{A}$ is computably categorical on a cone.
2. $\mathcal{A}$ has a Σ_3^{in} Scott sentence.

Back to Theorem 2

Theorem (2) [L-E 65; G. 75; M. 14] For a structure $\mathcal{A}$, TFAE:

1. $\mathcal{A}$ is computably categorical on a cone.
2. $\mathcal{A}$ has a Σ_3^{in} Scott sentence.

Recall: $x \in X$ is computably categorical $\iff$
for computable $y \equiv x$ there is computable $g \in G$ with $g \cdot x = y$.

Back to Theorem 2

Theorem (2) [L-E 65; G. 75; M. 14] For a structure $\mathcal{A}$, TFAE:

1. $\mathcal{A}$ is computably categorical on a cone.
2. $\mathcal{A}$ has a Σ_3^{in} Scott sentence.

Recall: $x \in X$ is computably categorical $\iff$
for computable $y \equiv x$ there is computable $g \in G$ with $g \cdot x = y$.

Recall: [Lopez-Escobar 65]: $\mathcal{A}$ has a Σ_3^{in} Scott sentence $\iff$ $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $F_{\sigma\delta}$.

Back to Theorem 2

Theorem (2) [L-E 65; G. 75; M. 14] For a structure $\mathcal{A}$, TFAE:

1. $\mathcal{A}$ is computably categorical on a cone.
2. $\mathcal{A}$ has a Σ_3^{in} Scott sentence.

Recall: $x \in X$ is computably categorical $\iff$
for computable $y \equiv x$ there is computable $g \in G$ with $g \cdot x = y$.

Recall: [Lopez-Escobar 65]: $\mathcal{A}$ has a Σ_3^{in} Scott sentence $\iff$ $\{\mathcal{B} \in Mod(L) : \mathcal{B} \cong \mathcal{A}\}$ is $F_{\sigma\delta}$.

Theorem [Melnikov, M.] For a point $x \in \mathcal{X}$, TFAE:

1. x is computably categorical on a cone.
2. The $\mathcal{G}$-orbit of x is $G_{\delta\sigma}$.

Theorem 3 – Knight's group 90's

Recall: Given a structure $\mathcal{A}$:

$$DgSp(\mathcal{A}) = \{Z \in 2^\omega : Z \text{ computes a copy of} \mathcal{A}\}.$$

Theorem 3 – Knight's group 90's

Recall: Given a structure $\mathcal{A}$:

$$DgSp(\mathcal{A}) = \{Z \in 2^\omega : Z \text{ computes a copy of} \mathcal{A}\}.$$

Theorem Each of the following is a degree spectra of some structure:

- upper cones: $\{Z \in 2^\omega : Z \geq_T C\}$ for some $C \in 2^\omega$ [Van der Waerden 30]

Theorem 3 – Knight's group 90's

Recall: Given a structure $\mathcal{A}$:

$$DgSp(\mathcal{A}) = \{Z \in 2^\omega : Z \text{ computes a copy of} \mathcal{A}\}.$$

Theorem Each of the following is a degree spectra of some structure:

- upper cones: $\{Z \in 2^\omega : Z \geq_T C\}$ for some $C \in 2^\omega$ [Van der Waerden 30]
- non-zero degrees: $\{Z \in 2^\omega : Z \not\equiv_T \emptyset\}$ [Slaman 98; Wehner 98]

Theorem 3 – Knight's group 90's

Recall: Given a structure $\mathcal{A}$:

$$DgSp(\mathcal{A}) = \{Z \in 2^\omega : Z \text{ computes a copy of} \mathcal{A}\}.$$

Theorem Each of the following is a degree spectra of some structure:

- upper cones: $\{Z \in 2^\omega : Z \geq_T C\}$ for some $C \in 2^\omega$ [Van der Waerden 30]
- non-zero degrees: $\{Z \in 2^\omega : Z \not\equiv_T \emptyset\}$ [Slaman 98; Wehner 98]
- non-Δ^0_2 degrees: $\{Z \in 2^\omega : Z \not\leq_T 0'\}$ [Kalimullin 08]

Theorem 3 – Knight's group 90's

Recall: Given a structure $\mathcal{A}$:

$$DgSp(\mathcal{A}) = \{Z \in 2^\omega : Z \text{ computes a copy of} \mathcal{A}\}.$$

Theorem Each of the following is a degree spectra of some structure:

- upper cones: $\{Z \in 2^\omega : Z \geq_T C\}$ for some $C \in 2^\omega$ [Van der Waerden 30]
- non-zero degrees: $\{Z \in 2^\omega : Z \not\equiv_T \emptyset\}$ [Slaman 98; Wehner 98]
- non-Δ^0_2 degrees: $\{Z \in 2^\omega : Z \not\leq_T 0'\}$ [Kalimullin 08]
- the hyperimmune degrees [Csima, Kalimullin 10]

Theorem 3 – Knight's group 90's

Recall: Given a structure $\mathcal{A}$:

$$DgSp(\mathcal{A}) = \{Z \in 2^\omega : Z \text{ computes a copy of} \mathcal{A}\}.$$

Theorem Each of the following is a degree spectra of some structure:

- upper cones: $\{Z \in 2^\omega : Z \geq_T C\}$ for some $C \in 2^\omega$ [Van der Waerden 30]
- non-zero degrees: $\{Z \in 2^\omega : Z \not\equiv_T \emptyset\}$ [Slaman 98; Wehner 98]
- non-Δ^0_2 degrees: $\{Z \in 2^\omega : Z \not\leq_T 0'\}$ [Kalimullin 08]
- the hyperimmune degrees [Csima, Kalimullin 10]
- non-hyp-degrees: $\{Z \in 2^\omega : Z \notin hyp\}$ [Greenberg, Montalbán, Slaman 12]

Theorem 3 – Knight's group 90's

Recall: Given a structure $\mathcal{A}$:

$$DgSp(\mathcal{A}) = \{Z \in 2^\omega : Z \text{ computes a copy of} \mathcal{A}\}.$$

Theorem Each of the following is a degree spectra of some structure:

- upper cones: $\{Z \in 2^\omega : Z \geq_T C\}$ for some $C \in 2^\omega$ [Van der Waerden 30]
- non-zero degrees: $\{Z \in 2^\omega : Z \not\equiv_T \emptyset\}$ [Slaman 98; Wehner 98]
- non-Δ^0_2 degrees: $\{Z \in 2^\omega : Z \not\leq_T 0'\}$ [Kalimullin 08]
- the hyperimmune degrees [Csima, Kalimullin 10]
- non-hyp-degrees: $\{Z \in 2^\omega : Z \notin hyp\}$ [Greenberg, Montalbán, Slaman 12]
- ...

Theorem 3 – Knight's group 90's

Recall: Given a structure $\mathcal{A}$:

$$DgSp(\mathcal{A}) = \{Z \in 2^\omega : Z \text{ computes a copy of} \mathcal{A}\}.$$

Theorem Each of the following is a degree spectra of some structure:

- upper cones: $\{Z \in 2^\omega : Z \geq_T C\}$ for some $C \in 2^\omega$ [Van der Waerden 30]
- non-zero degrees: $\{Z \in 2^\omega : Z \not\equiv_T \emptyset\}$ [Slaman 98; Wehner 98]
- non-Δ^0_2 degrees: $\{Z \in 2^\omega : Z \not\leq_T 0'\}$ [Kalimullin 08]
- the hyperimmune degrees [Csima, Kalimullin 10]
- non-hyp-degrees: $\{Z \in 2^\omega : Z \notin hyp\}$ [Greenberg, Montalbán, Slaman 12]
- ...

Theorem [Knight et al. 90's] The degree spectrum of a structure is never a non-trivial union of countably many upper cones.

Theorem 3 – Knight's group 90's

Recall: Given a structure $\mathcal{A}$:

$$DgSp(\mathcal{A}) = \{Z \in 2^{\omega} : Z \text{ computes a copy of} \mathcal{A}\}.$$

Theorem [Knight et al. 90's] The degree spectrum of a structure is never a non-trivial union of countably many upper cones.

Theorem 3 – Knight's group 90's

Recall: Given a structure $\mathcal{A}$:

$$DgSp(\mathcal{A}) = \{Z \in 2^\omega : Z \text{ computes a copy of} \mathcal{A}\}.$$

Theorem [Knight et al. 90's] The degree spectrum of a structure is never a non-trivial union of countably many upper cones.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Definition For $x \in \mathcal{X}$ we define the *degree spectrum of X* be

$$DgSp_{\mathcal{G}}(x) = \{Z \in 2^\omega : Z \text{ computes a point } y \equiv x\}.$$

Theorem 3 – Knight's group 90's

Recall: Given a structure $\mathcal{A}$:

$$DgSp(\mathcal{A}) = \{Z \in 2^{\omega} : Z \text{ computes a copy of} \mathcal{A}\}.$$

Theorem [Knight et al. 90's] The degree spectrum of a structure is never a non-trivial union of countably many upper cones.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Definition For $x \in \mathcal{X}$ we define the *degree spectrum of X* be

$$DgSp_{\mathcal{G}}(x) = \{Z \in 2^{\omega} : Z \text{ computes a point } y \equiv x\}.$$

In the general setting of Polish group actions:

Theorem [Melnikov, M.] The degree spectrum of a point is never a non-trivial union of two upper cones.

Theorem 4 – Computable dimension.

Definition A structure $\mathcal{A}$ has *computable dimension n* if

Theorem 4 – Computable dimension.

Definition A structure $\mathcal{A}$ has *computable dimension n* if
the set $\{\mathcal{B} \cong \mathcal{A} : \mathcal{B} \text{ computable}\}$ splits into n $\cong^c$-equivalence classes,
where $\mathcal{B} \cong^c \mathcal{C}$ if there is a computable isomorphism between them.

Theorem 4 – Computable dimension.

Definition A structure $\mathcal{A}$ has *computable dimension n* if
the set $\{\mathcal{B} \cong \mathcal{A} : \mathcal{B} \text{ computable}\}$ splits into n $\cong^c$-equivalence classes,
where $\mathcal{B} \cong^c \mathcal{C}$ if there is a computable isomorphism between them.

Theorem [Goncharov 80] For every $n \in \{1, 2, 3, ..., \infty\}$,
there is a computable structure with computable dimension n.

Theorem 4 – Computable dimension.

Definition A structure $\mathcal{A}$ has *computable dimension n* if the set $\{\mathcal{B} \cong \mathcal{A} : \mathcal{B} \text{ computable}\}$ splits into n $\cong^c$-equivalence classes,
where $\mathcal{B} \cong^c \mathcal{C}$ if there is a computable isomorphism between them.

Theorem [Goncharov 80] For every $n \in \{1, 2, 3, ..., \infty\}$,
there is a computable structure with computable dimension n.

Theorem Any structure in the following classes has computable dimension either 1 or ω:

- Boolean Algebras [Goncharov 73]
- Linear Ordering [Remmel 81][Goncharov and Dzgoev 80]
- Real algebraically closed fields [Nurtazin [1974]]
- Archimedean ordered group [Goncharov, Lempp and Solomon 2000]

Proper finite computable dimension doesn't relativize

Theorem [McCoy 02] If a structure has finite dimension on a cone it is computably categorical on a cone.

Proper finite computable dimension doesn't relativize

Theorem [McCoy 02] If a structure has finite dimension on a cone it is computably categorical on a cone.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Proper finite computable dimension doesn't relativize

Theorem [McCoy 02] If a structure has finite dimension on a cone it is computably categorical on a cone.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Definition A computable point x has *computable dimension n* if

Proper finite computable dimension doesn't relativize

Theorem [McCoy 02] If a structure has finite dimension on a cone
it is computably categorical on a cone.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Definition A computable point x has *computable dimension n* if the set $\{y \equiv x : y \text{ computable}\}$ splits into n $\equiv^c$-orbits,
where $z \equiv^c w$ if there is a computable $g \in G$ with $g \cdot z = w$.

Proper finite computable dimension doesn't relativize

Theorem [McCoy 02] If a structure has finite dimension on a cone
it is computably categorical on a cone.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Definition A computable point x has *computable dimension n* if
the set $\{y \equiv x : y \text{ computable}\}$ splits into n $\equiv^c$-orbits,
where $z \equiv^c w$ if there is a computable $g \in G$ with $g \cdot z = w$.

In the general setting of Polish group actions:

Theorem [Melnikov, M.] If a point $x \in \mathcal{X}$ has finite dimension on a cone
it is computably categorical on a cone.

Proper finite computable dimension doesn't relativize

Theorem [McCoy 02] If a structure has finite dimension on a cone
it is computably categorical on a cone.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Definition A computable point x has *computable dimension n* if the set $\{y \equiv x : y \text{ computable}\}$ splits into n $\equiv^c$-orbits,
where $z \equiv^c w$ if there is a computable $g \in G$ with $g \cdot z = w$.

In the general setting of Polish group actions:

Theorem [Melnikov, M.] If a point $x \in \mathcal{X}$ has finite dimension on a cone
it is computably categorical on a cone.

Proof: Show that if a structure has finite dimension on a cone, its orbits is $\mathbf{\Sigma}^0_3$.

Theorem 5 – Goncharov

Theorem ([Goncharov 80's])

If a computable structure has two computable copies which are Δ^0_2-isomorphic but not computably isomorphic, then the structure has infinite computable dimension.

Theorem 5 – Goncharov

Theorem ([Goncharov 80's])

If a computable structure has two computable copies which are Δ^0_2-isomorphic but not computably isomorphic, then the structure has infinite computable dimension.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Theorem 5 – Goncharov

Theorem ([Goncharov 80's])

If a computable structure has two computable copies which are Δ^0_2-isomorphic but not computably isomorphic, then the structure has infinite computable dimension.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Def: y and z are NH-equivalent if there is *non-high, c.e.* $g \in \mathcal{G}$ with $g \cdot y = z$.

Theorem 5 – Goncharov

Theorem ([Goncharov 80's])

If a computable structure has two computable copies which are Δ^0_2-isomorphic but not computably isomorphic, then the structure has infinite computable dimension.

Recall that we have an computable Polish group action of $\mathcal{G}$ on $\mathcal{X}$.

Def: y and z are NH-equivalent if there is *non-high, c.e.* $g \in \mathcal{G}$ with $g \cdot y = z$.

Theorem ([Melnikov, M.])

If in the orbit of a point there are two computable points which are NH-equivalent but not computably equivalent, then the point has infinite computable dimension.