Computable structure theory and infinitary logic

Part 1: Presentations and computations

Matthew Harrison-Trainor
University of Illinois Chicago

ASL NA Meeting 2026 Tutorial Part 1

Computable structure theory is the study of the complexity of countable mathematical objects.

Computable structure theory is the study of the complexity of countable mathematical objects.

- countable objects mean countable $\mathcal{L}$-structures including graphs, groups, and linear orders.

Computable structure theory is the study of the complexity of countable mathematical objects.

- countable objects mean countable $\mathcal{L}$-structures including graphs, groups, and linear orders.
- complexity includes computational complexity, descriptive or syntactic complexity, and topological complexity.

Computable structure theory is the study of the complexity of countable mathematical objects.

- countable objects mean countable $\mathcal{L}$-structures including graphs, groups, and linear orders.
- complexity includes computational complexity, descriptive or syntactic complexity, and topological complexity.

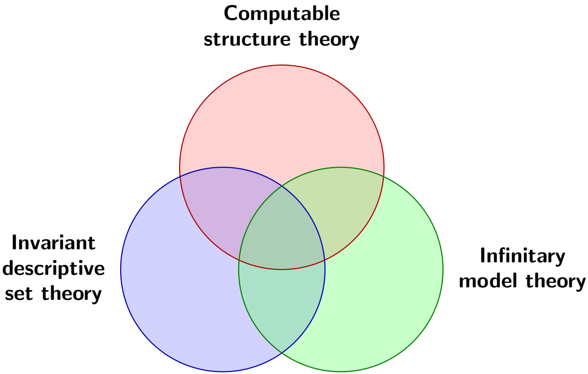
In mathematics, some proofs, constructions, characterizations, and classifications are more difficult than others.

Computable structure theory is the study of the complexity of countable mathematical objects.

- countable objects mean countable $\mathcal{L}$-structures including graphs, groups, and linear orders.
- complexity includes computational complexity, descriptive or syntactic complexity, and topological complexity.

In mathematics, some proofs, constructions, characterizations, and classifications are more difficult than others.

The goal of computable structure theory is to understand and measure this complexity.



Infinitary model theory: Model theory in the logic $\mathcal{L}_{\omega_1\omega}$ with countably infinite conjunctions and disjunctions.

Invariant descriptive set theory: Descriptive set theory of the action of S_∞ on the space of $\mathcal{L}$-structures.

There are three books on computable structure theory:

- Ash-Knight (2000): *Computable Structures and the Hyperarithmetical Hierarchy*.
- Montalbán (2021 and 2026): *Computable Structure Theory*, parts I and II.
- Downey-Melnikov (2026): *Computable Structure Theory: A Unified Approach*.

Outline

Outline

Part 1: Presentations and computations

Part 2: Infinitary logic and forcing

Part 3: Back-and-forth and the Scott analysis

Computability-theoretic background

A function $f : \mathbb{N}^m \to \mathbb{N}^n$ is computable if it can be computed by a program in your favourite programming language.

Computability-theoretic background

A function $f : \mathbb{N}^m \to \mathbb{N}^n$ is computable if it can be computed by a program in your favourite programming language.

A set $R \subseteq \mathbb{N}^n$ is computable if there is a program to compute membership in R, or equivalently, if the characteristic function of R is computable.

Computability-theoretic background

A function $f : \mathbb{N}^m \to \mathbb{N}^n$ is computable if it can be computed by a program in your favourite programming language.

A set $R \subseteq \mathbb{N}^n$ is computable if there is a program to compute membership in R, or equivalently, if the characteristic function of R is computable.

A function is computable if and only if its graph is computable.

Computability-theoretic background

A function $f : \mathbb{N}^m \to \mathbb{N}^n$ is computable if it can be computed by a program in your favourite programming language.

A set $R \subseteq \mathbb{N}^n$ is computable if there is a program to compute membership in R, or equivalently, if the characteristic function of R is computable.

A function is computable if and only if its graph is computable.

There are computable pairing functions, so we can computably identify $\mathbb{N}^n$ with $\mathbb{N}$.

Computability-theoretic background

An important non-computable set is the Halting problem

$$0' = K = \{e \in \mathbb{N} : \varphi_e(e) \text{ halts}\}.$$

$0'$ is computably enumerable which means that there is a computable list of the elements of $0'$.

Computability-theoretic background

An oracle is a set $A \subseteq \mathbb{N}$.

Computability-theoretic background

An oracle is a set $A \subseteq \mathbb{N}$.

An oracle program is a program that can also query an oracle.

Computability-theoretic background

An oracle is a set $A \subseteq \mathbb{N}$.

An oracle program is a program that can also query an oracle.

We say that A is B-computable if there is a program with oracle B computing A. Questions of membership in A can be computably decided with reference to questions of membership in B.

Computability-theoretic background

An oracle is a set $A \subseteq \mathbb{N}$.

An oracle program is a program that can also query an oracle.

We say that A is B-computable if there is a program with oracle B computing A. Questions of membership in A can be computably decided with reference to questions of membership in B.

We write $A \leq_T B$, "A is Turing-reducible to B", if A is B-computable.

Computability-theoretic background

An oracle is a set $A \subseteq \mathbb{N}$.

An oracle program is a program that can also query an oracle.

We say that A is B-computable if there is a program with oracle B computing A. Questions of membership in A can be computably decided with reference to questions of membership in B.

We write $A \leq_T B$, "A is Turing-reducible to B", if A is B-computable.

A Turing degree is an equivalence class of Turing equivalence

$$A \equiv_T B \iff A \leq_T B \text{ and } B \leq_T A.$$

Computability-theoretic background

The Turing degrees form a partial order.

Computability-theoretic background

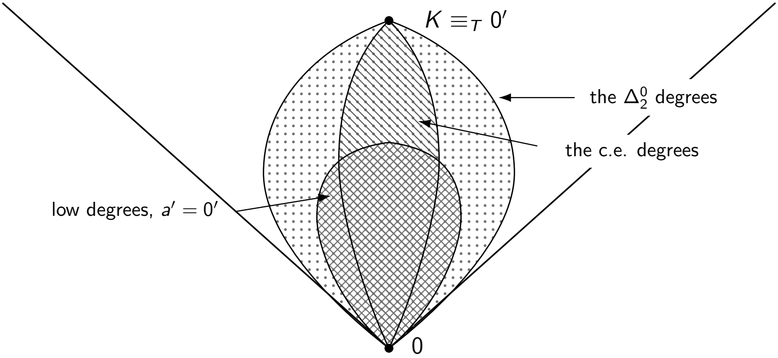
The Turing degrees form a partial order.

We will see more of this in the later parts of the tutorial; for now, it is enough to know that the Turing degrees are a very rich structure, e.g., it is unknown if they have a non-trivial automorphism.

Computability-theoretic background

The Turing degrees form a partial order.

We will see more of this in the later parts of the tutorial; for now, it is enough to know that the Turing degrees are a very rich structure, e.g., it is unknown if they have a non-trivial automorphism.



Background assumptions

Throughout this tutorial:

- All languages will be countable and computable.
- All structures will be countable.

Presentations of structures

In order to compute with a countable structure, we must have a representation of that structure.

Presentations of structures

In order to compute with a countable structure, we must have a representation of that structure.

Definition

A presentation (copy, representation, constructivization) of a structure $\mathcal{M}$ is an isomorphic structure $\mathcal{A}$ with domain $\mathbb{N}$.

Presentations of structures

In order to compute with a countable structure, we must have a representation of that structure.

Definition

A presentation (copy, representation, constructivization) of a structure $\mathcal{M}$ is an isomorphic structure $\mathcal{A}$ with domain $\mathbb{N}$.

We often conflate $\mathcal{A}$ with its atomic diagram

$$\mathrm{Diag}(\mathcal{A}) = \bigoplus_{f \in \mathcal{L}} f^{\mathcal{A}} \oplus \bigoplus_{R \in \mathcal{L}} R^{\mathcal{A}} \oplus \bigoplus_{c \in \mathcal{L}} \{c\}^{\mathcal{A}}.$$

Presentations of structures

In order to compute with a countable structure, we must have a representation of that structure.

Definition

A presentation (copy, representation, constructivization) of a structure $\mathcal{M}$ is an isomorphic structure $\mathcal{A}$ with domain $\mathbb{N}$.

We often conflate $\mathcal{A}$ with its atomic diagram

$$\mathrm{Diag}(\mathcal{A}) = \bigoplus_{f \in \mathcal{L}} f^{\mathcal{A}} \oplus \bigoplus_{R \in \mathcal{L}} R^{\mathcal{A}} \oplus \bigoplus_{c \in \mathcal{L}} \{c\}^{\mathcal{A}}.$$

We can identify $\mathrm{Diag}(\mathcal{A})$ with a subset of $\mathbb{N}$.

Presentations of structures

In order to compute with a countable structure, we must have a representation of that structure.

Definition

A presentation (copy, representation, constructivization) of a structure $\mathcal{M}$ is an isomorphic structure $\mathcal{A}$ with domain $\mathbb{N}$.

We often conflate $\mathcal{A}$ with its atomic diagram

$$\mathrm{Diag}(\mathcal{A}) = \bigoplus_{f \in \mathcal{L}} f^{\mathcal{A}} \oplus \bigoplus_{R \in \mathcal{L}} R^{\mathcal{A}} \oplus \bigoplus_{c \in \mathcal{L}} \{c\}^{\mathcal{A}}.$$

We can identify $\mathrm{Diag}(\mathcal{A})$ with a subset of $\mathbb{N}$.

$\mathcal{A}$ is computable if its atomic diagram is a computable set, that is, if all of the functions, relations, and constants of $\mathcal{A}$ are uniformly computable.

Presentations of structures

In order to compute with a countable structure, we must have a representation of that structure.

Definition

A presentation (copy, representation, constructivization) of a structure $\mathcal{M}$ is an isomorphic structure $\mathcal{A}$ with domain $\mathbb{N}$.

We often conflate $\mathcal{A}$ with its atomic diagram

$$\mathrm{Diag}(\mathcal{A}) = \bigoplus_{f \in \mathcal{L}} f^{\mathcal{A}} \oplus \bigoplus_{R \in \mathcal{L}} R^{\mathcal{A}} \oplus \bigoplus_{c \in \mathcal{L}} \{c\}^{\mathcal{A}}.$$

We can identify $\mathrm{Diag}(\mathcal{A})$ with a subset of $\mathbb{N}$.

$\mathcal{A}$ is computable if its atomic diagram is a computable set, that is, if all of the functions, relations, and constants of $\mathcal{A}$ are uniformly computable.

Throughout the tutorial, all structures will have domain $\mathbb{N}$.

Presentations of structures

Consider $(\mathbb{N}, <)$.

The standard copy of $(\mathbb{N}, <)$ is computable because the ordering on the natural numbers is computable.

Presentations of structures

Consider $(\mathbb{N}, <)$.

The standard copy of $(\mathbb{N}, <)$ is computable because the ordering on the natural numbers is computable.

There are also other computable copies. Let $0'$ be the Halting set, which is computably enumerable but not computable. Put

Presentations of structures

Consider $(\mathbb{N}, <)$.

The standard copy of $(\mathbb{N}, <)$ is computable because the ordering on the natural numbers is computable.

There are also other computable copies. Let $0'$ be the Halting set, which is computably enumerable but not computable. Put

- $2i \prec 2j$ for $i < j$;

Presentations of structures

Consider $(\mathbb{N}, <)$.

The standard copy of $(\mathbb{N}, <)$ is computable because the ordering on the natural numbers is computable.

There are also other computable copies. Let $0'$ be the Halting set, which is computably enumerable but not computable. Put

- $2i \prec 2j$ for $i < j$;
- $2i \prec 2k+1 \prec 2i+2$ when i is the kth element of $0'$.

Presentations of structures

Consider $(\mathbb{N}, <)$.

The standard copy of $(\mathbb{N}, <)$ is computable because the ordering on the natural numbers is computable.

There are also other computable copies. Let $0'$ be the Halting set, which is computably enumerable but not computable. Put

- $2i \prec 2j$ for $i < j$;
- $2i \prec 2k + 1 \prec 2i + 2$ when i is the kth element of $0'$.

Then $(\mathbb{N}, \prec) \cong (\mathbb{N}, <)$ is a computable copy.

Presentations of structures

Consider $(\mathbb{N}, <)$.

The standard copy of $(\mathbb{N}, <)$ is computable because the ordering on the natural numbers is computable.

There are also other computable copies. Let $0'$ be the Halting set, which is computably enumerable but not computable. Put

- $2i \prec 2j$ for $i < j$;
- $2i \prec 2k+1 \prec 2i+2$ when i is the kth element of $0'$.

Then $(\mathbb{N}, \prec) \cong (\mathbb{N}, <)$ is a computable copy.

If $0' = \{3, 0, 5, 2, \dots\}$,

$$0 \to 3 \to 2 \to 4 \to 7 \to 6 \to 1 \to 8 \to 10 \to 5 \to 12 \to \cdots$$

Presentations of structures

$$0 \to 3 \to 2 \to 4 \to 7 \to 6 \to 1 \to 8 \to 10 \to 5 \to 12 \to \cdots$$

In fact, $(\mathbb{N}, \prec)$ is a qualitatively different computable copy: The successor relation is not computable on $(\mathbb{N}, \prec)$.

Presentations of structures

$$0 \to 3 \to 2 \to 4 \to 7 \to 6 \to 1 \to 8 \to 10 \to 5 \to 12 \to \cdots$$

In fact, $(\mathbb{N}, \prec)$ is a qualitatively different computable copy: The successor relation is not computable on $(\mathbb{N}, \prec)$.

Proof.

For each i, $i \in 0'$ if and only if $2i + 2$ is not the $\prec$-successor of $2i$.

Presentations of structures

$$0 \to 3 \to 2 \to 4 \to 7 \to 6 \to 1 \to 8 \to 10 \to 5 \to 12 \to \cdots$$

In fact, $(\mathbb{N}, \prec)$ is a qualitatively different computable copy: The successor relation is not computable on $(\mathbb{N}, \prec)$.

Proof.

For each i, $i \in 0'$ if and only if $2i + 2$ is not the $\prec$-successor of $2i$.

$0' = \{0, 2, 3, 5, \ldots\}$.

Presentations of structures

$$0 \to 3 \to 2 \to 4 \to 7 \to 6 \to 1 \to 8 \to 10 \to 5 \to 12 \to \cdots$$

In fact, $(\mathbb{N}, \prec)$ is a qualitatively different computable copy: The successor relation is not computable on $(\mathbb{N}, \prec)$.

Proof.

For each i, $i \in 0'$ if and only if $2i + 2$ is not the $\prec$-successor of $2i$.

$0' = \{0, 2, 3, 5, \ldots\}$.

So if the successor relation on $(\mathbb{N}, \prec)$ were computable, then $0'$ would be computable. $\square$

Presentations of structures

$$0 \to 3 \to 2 \to 4 \to 7 \to 6 \to 1 \to 8 \to 10 \to 5 \to 12 \to \cdots$$

In fact, $(\mathbb{N}, \prec)$ is a qualitatively different computable copy: The successor relation is not computable on $(\mathbb{N}, \prec)$.

Proof.

For each i, $i \in 0'$ if and only if $2i + 2$ is not the $\prec$-successor of $2i$.

$0' = \{0, 2, 3, 5, \ldots\}$.

So if the successor relation on $(\mathbb{N}, \prec)$ were computable, then $0'$ would be computable. $\square$

In particular, $(\mathbb{N}, <)$ and $(\mathbb{N}, \prec)$ are not computably isomorphic.

Presentations of structures

There are also non-computable copies of $(\mathbb{N}, <)$. Given a set $A \subseteq \mathbb{N}$, let $(\mathbb{N}, \lessdot)$ be defined as follows:

Presentations of structures

There are also non-computable copies of $(\mathbb{N}, <)$. Given a set $A \subseteq \mathbb{N}$, let $(\mathbb{N}, \lessdot)$ be defined as follows:

- $2i$ and $2i + 1 \lessdot 2j$ and $2j + 1$ whenever $i < j$.

Presentations of structures

There are also non-computable copies of $(\mathbb{N}, <)$. Given a set $A \subseteq \mathbb{N}$, let $(\mathbb{N}, \lessdot)$ be defined as follows:

- $2i$ and $2i+1 \lessdot 2j$ and $2j+1$ whenever $i < j$.
- $2i \lessdot 2i+1$ if $i \in A$ and $2i+1 \lessdot 2i$ if $i \notin A$.

Presentations of structures

There are also non-computable copies of $(\mathbb{N}, <)$. Given a set $A \subseteq \mathbb{N}$, let $(\mathbb{N}, \lessdot)$ be defined as follows:

- $2i$ and $2i+1 \lessdot 2j$ and $2j+1$ whenever $i < j$.
- $2i \lessdot 2i+1$ if $i \in A$ and $2i+1 \lessdot 2i$ if $i \notin A$.

The atomic diagram of $(\mathbb{N}, \lessdot)$ is Turing equivalent to A.

Presentations of structures

There are also non-computable copies of $(\mathbb{N}, <)$. Given a set $A \subseteq \mathbb{N}$, let $(\mathbb{N}, \lessdot)$ be defined as follows:

- $2i$ and $2i+1 \lessdot 2j$ and $2j+1$ whenever $i < j$.
- $2i \lessdot 2i+1$ if $i \in A$ and $2i+1 \lessdot 2i$ if $i \notin A$.

The atomic diagram of $(\mathbb{N}, \lessdot)$ is Turing equivalent to A.

For example, if $0 \notin A$, $1 \in A$, and $2 \notin A$, then

$$\underbrace{1 \longrightarrow 0}_{0 \notin A} \longrightarrow \underbrace{2 \longrightarrow 3}_{1 \in A} \longrightarrow \underbrace{5 \longrightarrow 4}_{2 \notin A} \longrightarrow \cdots$$

Presentations of structures

It is the fact that a structure can have many different presentations that makes computable structure theory interesting.

Today, we will explore some of the basic ideas, objects, and types of questions in computable structure theory.

Computable categoricity

Definition

A structure $\mathcal{A}$ is computably categorical if any two computable copies of $\mathcal{A}$ are computably isomorphic.

Computable categoricity

Definition

A structure $\mathcal{A}$ is computably categorical if any two computable copies of $\mathcal{A}$ are computably isomorphic.

Example

$(\mathbb{N}, <)$ is not computably categorical.

Computable categoricity

Definition

A structure $\mathcal{A}$ is computably categorical if any two computable copies of $\mathcal{A}$ are computably isomorphic.

Example

$(\mathbb{N}, <)$ is not computably categorical.

Example

$(\mathbb{Z}, 0, +)$ is computably categorical.

Computable categoricity

Definition

A structure $\mathcal{A}$ is computably categorical if any two computable copies of $\mathcal{A}$ are computably isomorphic.

Example

$(\mathbb{N}, <)$ is not computably categorical.

Example

$(\mathbb{Z}, 0, +)$ is computably categorical.

Example

A $\mathbb{Q}$-vector space is computably categorical if and only if it has finite dimension.

Computable dimension

Definition

The computable dimension of $\mathcal{A}$ is the number of computable copies of $\mathcal{A}$ up to computable isomorphism.

Computable dimension

Definition

The computable dimension of $\mathcal{A}$ is the number of computable copies of $\mathcal{A}$ up to computable isomorphism.

Most structures you can think of will have computable dimension 1 or ∞, but one can construct other examples.

Computable dimension

Definition

The computable dimension of $\mathcal{A}$ is the number of computable copies of $\mathcal{A}$ up to computable isomorphism.

Most structures you can think of will have computable dimension 1 or ∞, but one can construct other examples.

Theorem (Goncharov)

For each $n \in \mathbb{N}$ there is a computable structure of computable dimension n.

Computable dimension

Let's consider the case of the $\mathbb{Q}$-vector spaces of infinite dimension.

Computable dimension

Let's consider the case of the $\mathbb{Q}$-vector spaces of infinite dimension.

There is a standard copy with a computable basis (and equivalently a computable linear dependence relation).

Computable dimension

Let's consider the case of the $\mathbb{Q}$-vector spaces of infinite dimension.

There is a standard copy with a computable basis (and equivalently a computable linear dependence relation).

There are also copies with no computable basis.

Computable dimension

Let's consider the case of the $\mathbb{Q}$-vector spaces of infinite dimension.

There is a standard copy with a computable basis (and equivalently a computable linear dependence relation).

There are also copies with no computable basis.

The construction of a basis can be carried out using the Halting problem as an oracle, so all such vector spaces have $0'$-computable bases and hence are $0'$-computably isomorphic.

Computable dimension

Let's consider the case of the $\mathbb{Q}$-vector spaces of infinite dimension.

There is a standard copy with a computable basis (and equivalently a computable linear dependence relation).

There are also copies with no computable basis.

The construction of a basis can be carried out using the Halting problem as an oracle, so all such vector spaces have $0'$-computable bases and hence are $0'$-computably isomorphic.

Theorem (Goncharov)

If $\mathcal{A} \cong \mathcal{B}$ are computable structures which are $0'$-computably isomorphic but not computably isomorphic, then their computable dimension is ∞.

Computable dimension

Question (Hirschfeldt, Khoussainov)

Is there a locally finite graph of computable dimension 2?

Computable categoricity

Definition
Let $X \subseteq \mathbb{N}$. A computable structure $\mathcal{A}$ is X-computably categorical if X computes an isomorphism between any two computable copies of $\mathcal{A}$.

Computable categoricity

Definition

Let $X \subseteq \mathbb{N}$. A computable structure $\mathcal{A}$ is X-computably categorical if X computes an isomorphism between any two computable copies of $\mathcal{A}$.

Example

$(\mathbb{N}, <)$ is $0'$-computably categorical. Given any two computable copies of $(\mathbb{N}, <)$, $0'$ can compute the adjacency relation on each:

$$\mathrm{Adj}(x, y) \text{ if there is no } z \text{ between } x \text{ and } y.$$

With the adjacency relation, we can compute the isomorphism.

Degrees of categoricity

Definition

$\mathcal{A}$ has degree of categoricity $\mathbf{d}$ if

- $\mathcal{A}$ is $\mathbf{d}$-computably categorical, and
- $\mathbf{d} \leq \mathbf{c}$ for every $\mathbf{c}$ such that $\mathcal{A}$ is $\mathbf{c}$-computably categorical.

Degrees of categoricity

Definition

$\mathcal{A}$ has degree of categoricity $\mathbf{d}$ if

- $\mathcal{A}$ is $\mathbf{d}$-computably categorical, and
- $\mathbf{d} \leq \mathbf{c}$ for every $\mathbf{c}$ such that $\mathcal{A}$ is $\mathbf{c}$-computably categorical.

Example

0 is the degree of categoricity of a computably categorical structure.

Degrees of categoricity

Definition

$\mathcal{A}$ has degree of categoricity $\mathbf{d}$ if

- $\mathcal{A}$ is $\mathbf{d}$-computably categorical, and
- $\mathbf{d} \leq \mathbf{c}$ for every $\mathbf{c}$ such that $\mathcal{A}$ is $\mathbf{c}$-computably categorical.

Example

0 is the degree of categoricity of a computably categorical structure.

Example

$0'$ is the degree of categoricity of $(\mathbb{N}, <)$ and of the infinite-dimensional $\mathbb{Q}$-vector space.

Degrees of categoricity

Definition

$\mathcal{A}$ has degree of categoricity $\mathbf{d}$ if

- $\mathcal{A}$ is $\mathbf{d}$-computably categorical, and
- $\mathbf{d} \leq \mathbf{c}$ for every $\mathbf{c}$ such that $\mathcal{A}$ is $\mathbf{c}$-computably categorical.

Example

0 is the degree of categoricity of a computably categorical structure.

Example

$0'$ is the degree of categoricity of $(\mathbb{N}, <)$ and of the infinite-dimensional $\mathbb{Q}$-vector space.

Question

Which degrees can be degrees of categoricity?

A set A is Δ^0_2 if it has a computable approximation $A(n, s)$ with finitely many mind changes:

$$A(n) = \lim_{s \to \infty} A(n, s).$$

A set A is Δ_2^0 if it has a computable approximation $A(n, s)$ with finitely many mind changes:

$$A(n) = \lim_{s \to \infty} A(n, s).$$

Which Δ_2^0 degrees are degrees of categoricity?

A set A is Δ^0_2 if it has a computable approximation $A(n, s)$ with finitely many mind changes:

$$A(n) = \lim_{s \to \infty} A(n, s).$$

Which Δ^0_2 degrees are degrees of categoricity?

Example

Every c.e. degree is a degree of categoricity.

A set A is c.e. if it has an approximation $A(n,s)$ which starts with $A(n,0)=0$ and then has at most one mind change $0 \to 1$.

Example

Every c.e. degree is a degree of categoricity. Let D be a c.e. set.

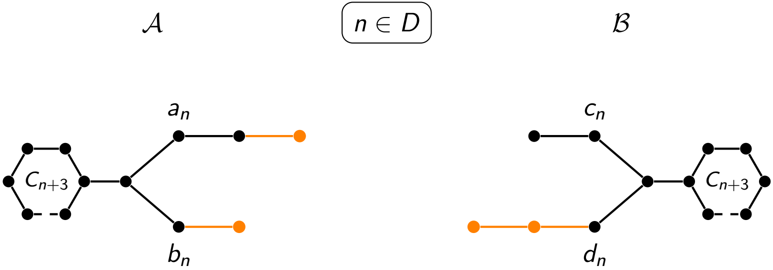
$\mathcal{A}$ $\qquad$ $\boxed{n \notin D}$ $\qquad$ $\mathcal{B}$

$$\boxed{f(a_n) = c_n, \qquad f(b_n) = d_n}$$

A set A is c.e. if it has an approximation $A(n, s)$ which starts with $A(n, 0) = 0$ and then has at most one mind change $0 \to 1$.

Example

Every c.e. degree is a degree of categoricity. Let D be a c.e. set.



$$\boxed{f(a_n) = d_n, \qquad f(b_n) = c_n}$$

A set A is 2-c.e. if it has an approximation $A(n, s)$ which starts with $A(n, 0) = 0$ and then has at most two mind changes $0 \to 1$ and then $1 \to 0$.

A set A is 2-c.e. if it has an approximation $A(n,s)$ which starts with $A(n,0)=0$ and then has at most two mind changes $0 \to 1$ and then $1 \to 0$.

Example (Fokina, Kalimullin, R. Miller)

Every 2-c.e. degree is a degree of categoricity.

A set A is 2-c.e. if it has an approximation $A(n,s)$ which starts with $A(n,0) = 0$ and then has at most two mind changes $0 \to 1$ and then $1 \to 0$.

Example (Fokina, Kalimullin, R. Miller)

Every 2-c.e. degree is a degree of categoricity.

Let $D = A - B$ be a c.e. set.

A set A is 2-c.e. if it has an approximation $A(n,s)$ which starts with $A(n,0)=0$ and then has at most two mind changes $0 \to 1$ and then $1 \to 0$.

Example (Fokina, Kalimullin, R. Miller)

Every 2-c.e. degree is a degree of categoricity.

Let $D = A - B$ be a c.e. set.

We'll build two isomorphic computable graphs $\mathcal{A}$ and $\mathcal{B}$ such that:

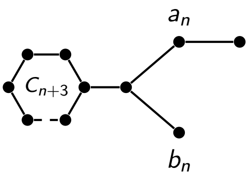
- any isomorphism between $\mathcal{A}$ and $\mathcal{B}$ computes D,
- for any other computable copy $\mathcal{C}$ of $\mathcal{A}$ and $\mathcal{B}$, D computes an isomorphism between $\mathcal{C}$ and $\mathcal{A}$.

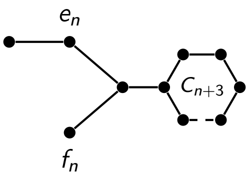
Degrees of categoricity

nth component of $\mathcal{A}$

$$\boxed{n \notin A \Rightarrow n \notin D}$$

nth component of $\mathcal{B}$

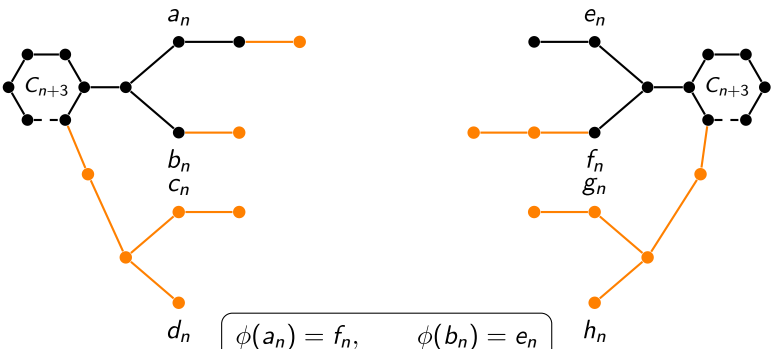




$$\boxed{\phi(a_n) = e_n, \qquad \phi(b_n) = f_n}$$

Degrees of categoricity

nth component of $\mathcal{A}$ $\quad\boxed{n \in A \quad n \notin B \Rightarrow n \in D}\quad$ nth component of $\mathcal{B}$



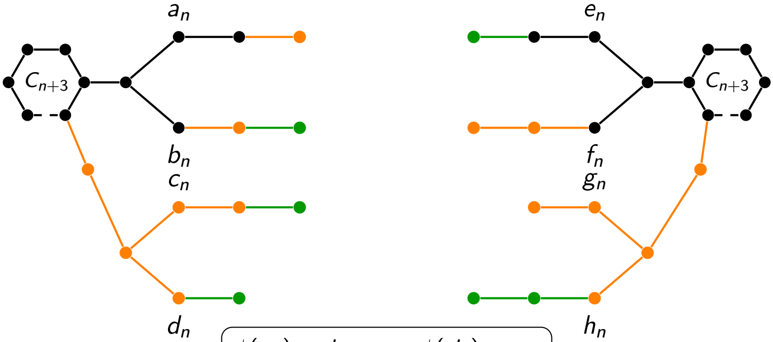
$$\boxed{\begin{array}{ll} \phi(a_n) = f_n, & \phi(b_n) = e_n \\ \phi(c_n) = g_n, & \phi(d_n) = h_n \end{array}}$$

Degrees of categoricity

nth component of $\mathcal{A}$

$\boxed{n \in A \quad n \in B \Rightarrow n \notin D}$

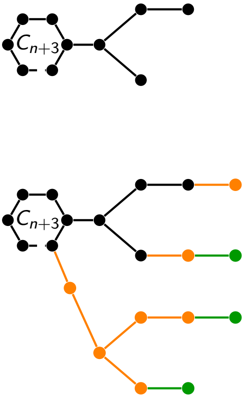
nth component of $\mathcal{B}$



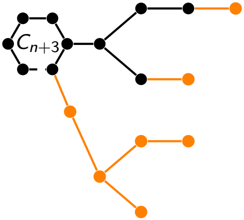
$\boxed{\phi(c_n) = h_n, \qquad \phi(d_n) = g_n}$

Degrees of categoricity

$\boxed{n \notin D}$



$\boxed{n \in D}$



Degrees of categoricity

Theorem (Csima, Ng)

Every Δ_2^0 degree is a degree of categoricity.

Degrees of categoricity

Theorem (Csima, Ng)

Every Δ^0_2 degree is a degree of categoricity.

Theorem (Anderson, Csima)

There is a Σ^0_2 degree which is not a degree of categoricity.

Degree spectra

Definition

The degree spectrum of $\mathcal{A}$ is

$$\mathrm{DgSp}(\mathcal{A}) = \{X \subseteq \mathbb{N} : X \text{ computes a copy of } \mathcal{A}\}.$$

Degree spectra

Definition

The degree spectrum of $\mathcal{A}$ is

$$\mathrm{DgSp}(\mathcal{A}) = \{X \subseteq \mathbb{N} : X \text{ computes a copy of } \mathcal{A}\}.$$

Theorem (Knight)

Except for automorphically trivial structures, if X computes a copy of $\mathcal{A}$, then there is a copy of $\mathcal{A}$ whose atomic diagram is Turing equivalent to X.

$$\mathrm{DgSp}(\mathcal{A}) = \{\deg_T(\mathcal{B}) : \mathcal{A} \cong \mathcal{B}\}.$$

Degree spectra

The degree spectrum of $\mathcal{A}$ measures the difficulty of building a copy of $\mathcal{A}$.

Degree spectra

The degree spectrum of $\mathcal{A}$ measures the difficulty of building a copy of $\mathcal{A}$.

Example

$(\mathbb{N}, <)$ has a computable copy, so its degree spectrum is everything.

Degree spectra

The degree spectrum of $\mathcal{A}$ measures the difficulty of building a copy of $\mathcal{A}$.

Example

$(\mathbb{N}, <)$ has a computable copy, so its degree spectrum is everything.

Example

For every set X, there is a structure $\mathcal{A}_X$ whose degree spectrum is

$$\{Y \subseteq \mathbb{N} : Y \geq_T X\}.$$

Degree spectra

Example

For every set X, there is a structure $\mathcal{A}_X$ whose degree spectrum is

$$\{Y \subseteq \mathbb{N} : Y \geq_T X\}.$$

Degree spectra

Example

For every set X, there is a structure $\mathcal{A}_X$ whose degree spectrum is

$$\{Y \subseteq \mathbb{N} : Y \geq_T X\}.$$

$\mathcal{A}_X$ is a graph which consists of a loop of length $2n+3$ for each $n \in X$ and $2n+4$ for each $n \notin X$.

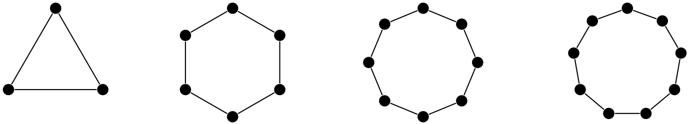
Degree spectra

Example

For every set X, there is a structure $\mathcal{A}_X$ whose degree spectrum is

$$\{Y \subseteq \mathbb{N} : Y \geq_T X\}.$$

$\mathcal{A}_X$ is a graph which consists of a loop of length $2n + 3$ for each $n \in X$ and $2n + 4$ for each $n \notin X$.



$2(0) + 3 = 3$ $\quad$ $2(1) + 4 = 6$ $\quad$ $2(2) + 4 = 8$ $\quad$ $2(3) + 3 = 9$

$0 \in X$ $\quad$ $1 \notin X$ $\quad$ $2 \notin X$ $\quad$ $3 \in X$

Degree spectra

Question

What are the possible degree spectra?

Degree spectra

Question

What are the possible degree spectra?

Theorem (Slaman, Wehner)

There is a structure $\mathcal{A}$ which has no computable copy, but which has a copy computable from each non-computable degree.

Thus the non-computable degrees are a degree spectrum:

$$\mathrm{DgSp}(\mathcal{A}) = \{X \subseteq \mathbb{N} : X \textit{ non-computable}\}.$$

Degree spectra

List out all of the computable structures as $(\mathcal{C}_e)_{e \in \mathbb{N}}$.

Degree spectra

List out all of the computable structures as $(\mathcal{C}_e)_{e \in \mathbb{N}}$.

Our language will have infinitely many unary relations (sorts) U_e. Each sort will contain a linear order.

Degree spectra

List out all of the computable structures as $(\mathcal{C}_e)_{e \in \mathbb{N}}$.

Our language will have infinitely many unary relations (sorts) U_e. Each sort will contain a linear order.

It is enough to make, for each e, a structure $\mathcal{A}_e \ncong U_e^{\mathcal{C}_e}$ such that for each non-computable set X there is an X-computable copy of $\mathcal{A}_e$ uniformly in X.

Degree spectra

List out all of the computable structures as $(\mathcal{C}_e)_{e\in\mathbb{N}}$.

Our language will have infinitely many unary relations (sorts) U_e. Each sort will contain a linear order.

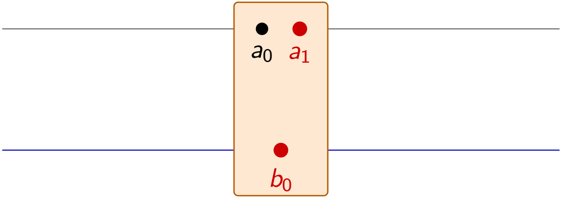
It is enough to make, for each e, a structure $\mathcal{A}_e \not\cong U_e^{\mathcal{C}_e}$ such that for each non-computable set X there is an X-computable copy of $\mathcal{A}_e$ uniformly in X.

Then our structure $\mathcal{A}$ has as its eth sort $\mathcal{A}_e$.

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \ncong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

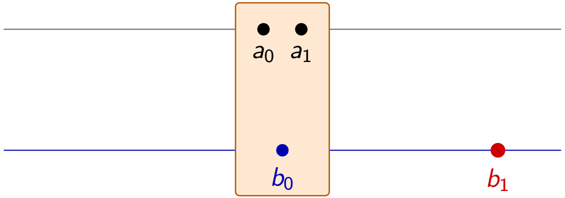


$X(0) = 0$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \not\cong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

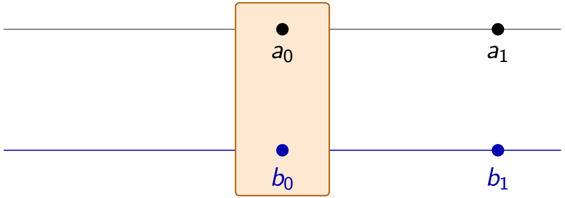


$X(0) = 0$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \ncong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

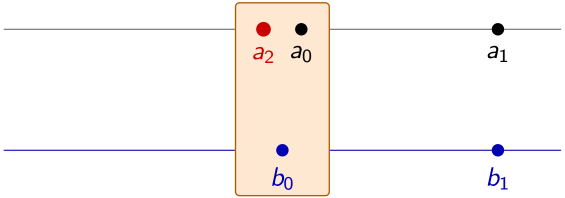


$$X(0) = 0$$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \not\cong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

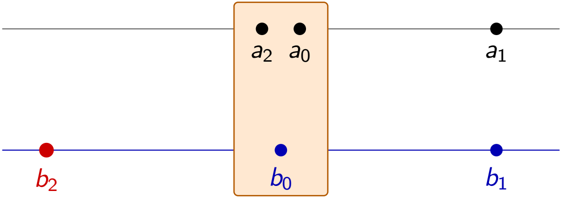


$$X(1) = 1$$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \ncong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

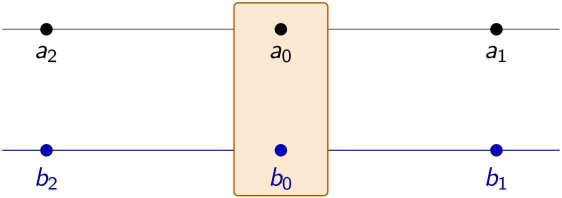


$X(1) = 1$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \not\cong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

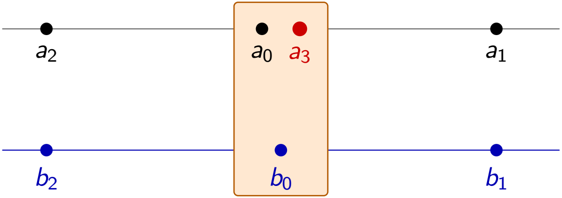


$X(1) = 1$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \not\cong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

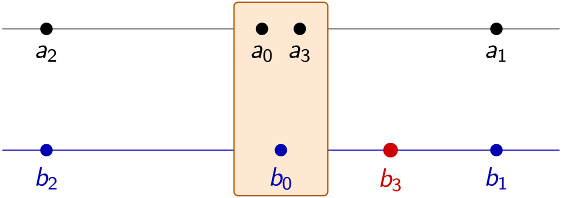


$X(2) = 0$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \ncong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

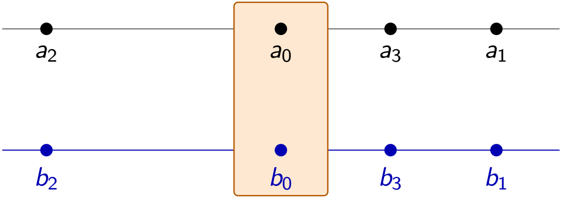


$X(2) = 0$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \ncong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

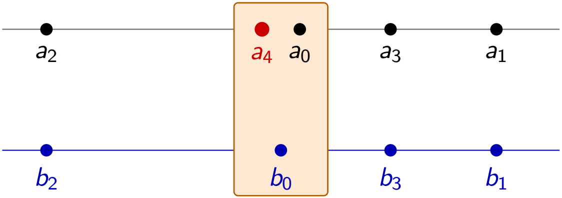


$X(2) = 0$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \ncong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

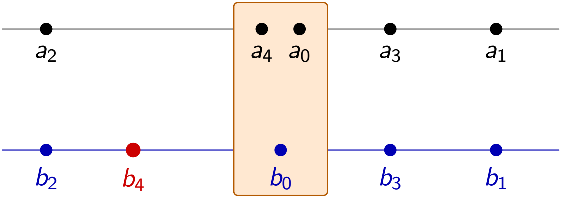


$X(3) = 1$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \ncong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

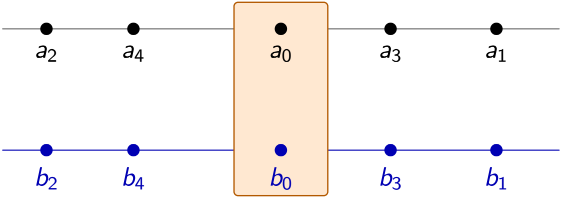


$X(3) = 1$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \not\cong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

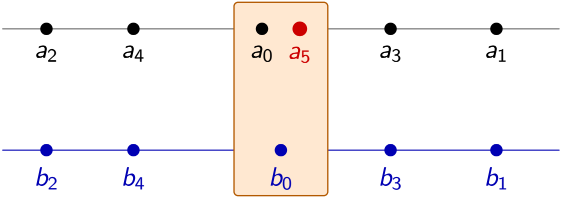


$$X(3) = 1$$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \not\cong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

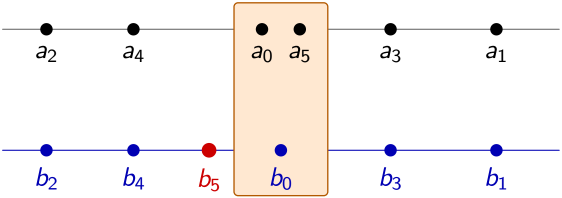


$X(4) = 0$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \ncong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

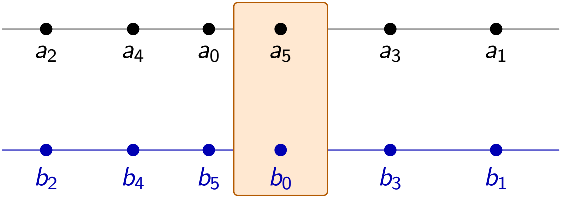


$X(4) = 0$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \ncong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

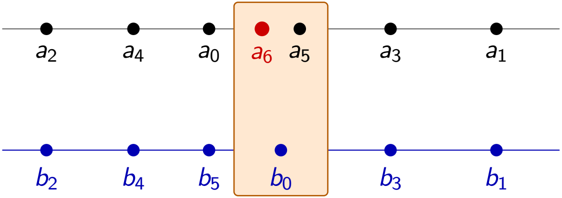


$X(4) = 0$

Degree spectra

Given a computable linear order $\mathcal{B}$ and non-computable set X, we build a linear order $\mathcal{A}^X$ such that:

- $\mathcal{A} \ncong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.



$X(5) = 1$

Degree spectra

What do we end up with?

Degree spectra

What do we end up with?

$\mathcal{A}^X$ is either:

Degree spectra

What do we end up with?

$\mathcal{A}^X$ is either:

- n, if $\mathcal{B}$ has $n-1$ elements,

Degree spectra

What do we end up with?

$\mathcal{A}^X$ is either:

- n, if $\mathcal{B}$ has $n-1$ elements,
- $n+\omega^*$, if $\mathcal{B}$ has n elements left of a_0,

Degree spectra

What do we end up with?

$\mathcal{A}^X$ is either:

- n, if $\mathcal{B}$ has $n-1$ elements,
- $n+\omega^*$, if $\mathcal{B}$ has n elements left of a_0,
- $\omega+n$, if $\mathcal{B}$ has n elements right of a_0,

Degree spectra

What do we end up with?

$\mathcal{A}^X$ is either:

- n, if $\mathcal{B}$ has $n-1$ elements,
- $n+\omega^*$, if $\mathcal{B}$ has n elements left of a_0,
- $\omega+n$, if $\mathcal{B}$ has n elements right of a_0,
- $\omega+\omega^*$, if $\mathcal{B}$ has infinitely many elements left and right of a_0.

Degree spectra

What do we end up with?

$\mathcal{A}^X$ is either:

- n, if $\mathcal{B}$ has $n-1$ elements,
- $n+\omega^*$, if $\mathcal{B}$ has n elements left of a_0,
- $\omega+n$, if $\mathcal{B}$ has n elements right of a_0,
- $\omega+\omega^*$, if $\mathcal{B}$ has infinitely many elements left and right of a_0.

This is what we wanted:

- $\mathcal{A} \ncong \mathcal{B}$,
- if X and Y are non-computable then $\mathcal{A}^X \cong \mathcal{A}^Y$.

Degree spectra

Question (Downey)

Is there a linear order with Slaman-Wehner degree spectrum?

[1] I think with my REU student Crittenden we can show the answer is yes.

Degree spectra

Question (Downey)

Is there a linear order with Slaman-Wehner degree spectrum?

Question (Downey, Goncharov)

Is there a torsion-free abelian group with Slaman-Wehner degree spectrum?[1]

[1] I think with my REU student Crittenden we can show the answer is yes.

Degree spectra

Question (Downey)

Is there a linear order with Slaman-Wehner degree spectrum?

Question (Downey, Goncharov)

Is there a torsion-free abelian group with Slaman-Wehner degree spectrum?[1]

Question (Harrison-Trainor)

Is every degree spectrum the degree spectrum of a simple rooted tree?

[1] I think with my REU student Crittenden we can show the answer is yes.

Degree spectra

Theorem (Kalimullin)

The non-Δ^0_2 degrees are a degree spectrum.

Degree spectra

Theorem (Kalimullin)

The non-Δ^0_2 degrees are a degree spectrum.

Theorem (Andrews, Cai, Kalimullin, Lempp, and J. Miller)

The non-Δ^0_3 degrees are not a degree spectrum.

Degree spectra

Theorem (Kalimullin)

The non-Δ^0_2 degrees are a degree spectrum.

Theorem (Andrews, Cai, Kalimullin, Lempp, and J. Miller)

The non-Δ^0_3 degrees are not a degree spectrum.

Theorem (Greenberg, Montalbán, Slaman)

The non-hyperarithmetic degrees are a degree spectrum.

Degree spectra

Theorem (Kalimullin)

The non-Δ^0_2 degrees are a degree spectrum.

Theorem (Andrews, Cai, Kalimullin, Lempp, and J. Miller)

The non-Δ^0_3 degrees are not a degree spectrum.

Theorem (Greenberg, Montalbán, Slaman)

The non-hyperarithmetic degrees are a degree spectrum.

Theorem (Harrison-Trainor)

The non-arithmetic degrees are a degree spectrum.

Degree spectra

X is low$_n$ if $X^{(n)} \equiv_T 0^{(n)}$.

Degree spectra

X is low$_n$ if $X^{(n)} \equiv_T 0^{(n)}$.

Theorem (Downey, Jockusch; Thurber; Knight, Stob)

If a Boolean algebra has a low$_4$ copy then it has a computable copy.

Degree spectra

X is low$_n$ if $X^{(n)} \equiv_T 0^{(n)}$.

Theorem (Downey, Jockusch; Thurber; Knight, Stob)

If a Boolean algebra has a low$_4$ copy then it has a computable copy.

Question

Is every low$_5$ Boolean algebra isomorphic to a computable one?

Degree spectra

X is low$_n$ if $X^{(n)} \equiv_T 0^{(n)}$.

Theorem (Downey, Jockusch; Thurber; Knight, Stob)

If a Boolean algebra has a low$_4$ copy then it has a computable copy.

Question

Is every low$_5$ Boolean algebra isomorphic to a computable one?

Harris and Montalbán: There is a change in behaviour starting at level 5.

Index sets

Let $\mathcal{A}_e$ be a listing of the (partial) computable $\mathcal{L}$-structures.

Index sets

Let $\mathcal{A}_e$ be a listing of the (partial) computable $\mathcal{L}$-structures. Given a property P, we can look at

$$\{i : \mathcal{A}_i \text{ is an } \mathcal{L}\text{-structure with property } P\}.$$

We consider the computability-theoretic difficulty of this set. If P has a good characterization, this set is close to computable; if the set is far from computable, P has no good characterization.

Index sets

Let $\mathcal{A}_e$ be a listing of the (partial) computable $\mathcal{L}$-structures. Given a property P, we can look at

$$\{i : \mathcal{A}_i \text{ is an } \mathcal{L}\text{-structure with property } P\}.$$

We consider the computability-theoretic difficulty of this set. If P has a good characterization, this set is close to computable; if the set is far from computable, P has no good characterization.

Similarly we consider the isomorphism problem for a class $\mathcal{C}$ of structures,

$$\{(i, j) : \mathcal{A}_i \text{ and } \mathcal{A}_j \text{ are in } \mathcal{C} \text{ and } \mathcal{A}_i \cong \mathcal{A}_j\}.$$

Index sets

Often this is the same as working with Borel complexity in $\mathrm{Mod}(\mathcal{L})$.

Index sets

Often this is the same as working with Borel complexity in Mod($\mathcal{L}$).

Cases where it is different include:

- there is a difference between looking at only computable copies and looking at all copies
- the class of objects we are considering is countable.

Index sets

Theorem (Downey, Kach, Lempp, Lewis-Pye, Montalbán, Turetsky)

The set

$$\{i : \mathcal{A}_i \text{ is computably categorical}\}$$

is Π^1_1*-complete.*

Thus there is no characterization of the computably categorical structures.

Index sets

Theorem (Harrison-Trainor)

The set

$$\{i : \mathcal{A}_i \text{ has a copy whose elementary diagram is computable}\}$$

is Σ^1_1-complete.

Thus there is no characterization of when a computable structure has a decidable copy with a computable elementary diagram.

Index sets

Let G_e be the eth finitely presented group.

Theorem (Lempp)

The set

$$\{e : G_e \text{ is torsion-free}\}$$

is Π^0_2-complete.

Index sets

In part 2 on Tuesday, we will introduce infinitary logic and see what it has to do with computation.

We will also learn about computability-theoretic forcing with countable structures.