Looking for a ruler to measure complexity

Antonio Montalbán

U.C. Berkeley

May 2022
Topos Institute
Berkeley, California

Computability Theory

Computability Theory

Objective: Study the complexity of countably infinite mathematical objects.

Computability Theory

Objective: Study the complexity of countably infinite mathematical objects.

The tools come from:

- For finite objects: Computer Science.
- For countably infinite objects: Computability Theory.
- For uncountable objects: Set Theory.

Computability Theory

Objective: Study the complexity of countably infinite mathematical objects.

The tools come from:

- For finite objects: Computer Science.
- For countably infinite objects: Computability Theory.
- For uncountable objects: Set Theory.

Definition: A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable* if there is a computer program that, on input n, calculates $f(n)$.

Computability Theory

Objective: Study the complexity of countably infinite mathematical objects.

The tools come from:

- For finite objects: Computer Science.
- For countably infinite objects: Computability Theory.
- For uncountable objects: Set Theory.

Definition: A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable*
if there is a computer program that, on input n, calculates $f(n)$.

Definition: A set $A \subseteq \mathbb{N}$ is *computable*
if its characteristic funciton $\chi_A\colon \mathbb{N} \to \{0, 1\}$ is computable.

Computability Theory

Objective: Study the complexity of countably infinite mathematical objects.

The tools come from:
- For finite objects: Computer Science.
- For countably infinite objects: Computability Theory.
- For uncountable objects: Set Theory.

Definition: A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable* if there is a computer program that, on input n, calculates $f(n)$.

Definition: A set $A \subseteq \mathbb{N}$ is *computable* if its characteristic funciton $\chi_A\colon \mathbb{N} \to \{0,1\}$ is computable.

First idea: Computable is easy —vs— Not computable is hard

Computability Theory

Objective: Study the complexity of countably infinite mathematical objects.

The tools come from:

- For finite objects: Computer Science.
- For countably infinite objects: Computability Theory.
- For uncountable objects: Set Theory.

Definition: A function $f : \mathbb{N} \to \mathbb{N}$ is *computable* if there is a computer program that, on input n, calculates $f(n)$.

Definition: A set $A \subseteq \mathbb{N}$ is *computable* if its characteristic funciton $\chi_A : \mathbb{N} \to \{0, 1\}$ is computable.

First idea: Computable is easy —vs— Not computable is hard

(Every finite object can be coded by a natural number.)

Examples: not computable sets

HT: *10^{th} Hilbert's problem:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

Examples: not computable sets

HT: *10^{th} Hilbert's problem:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *Halting problem:* The set of programs that do **not** run forever.

Examples: not computable sets

HT: *10^{th} Hilbert's problem:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *Halting problem:* The set of programs that do **not** run forever.

TF: Finite presentations $((a_1, ..., a_k), (R_1, ..., R_\ell))$ of *torsion free* groups.

Examples: not computable sets

HT: *10^{th} Hilbert's problem:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *Halting problem:* The set of programs that do **not** run forever.

TF: Finite presentations $((a_1, ..., a_k), (R_1, ..., R_\ell))$ of *torsion free* groups.

COF: Polynomials in $\mathbb{Z}[x, y_0, y_1, y_2, ...]$ with integer roots for almost every $x \in \mathbb{N}$.

Examples: not computable sets

HT: *10^{th} Hilbert's problem:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *Halting problem:* The set of programs that do **not** run forever.

TF: Finite presentations $((a_1, ..., a_k), (R_1, ..., R_\ell))$ of *torsion free* groups.

COF: Polynomials in $\mathbb{Z}[x, y_0, y_1, y_2, ...]$ with integer roots for almost every $x \in \mathbb{N}$.

TA: *True* first-order sentences about $(\mathbb{N}; 0, 1, +, \times)$

Examples: not computable sets

HT: *10^{th} Hilbert's problem:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *Halting problem:* The set of programs that do **not** run forever.

TF: Finite presentations $((a_1, ..., a_k), (R_1, ..., R_\ell))$ of *torsion free* groups.

COF: Polynomials in $\mathbb{Z}[x, y_0, y_1, y_2, ...]$ with integer roots for <u>almost</u> every $x \in \mathbb{N}$.

TA: *True* first-order sentences about $(\mathbb{N}; 0, 1, +, \times)$

WF: Programs $\mathtt{p}$ for which there is a $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}(a_{i+1}) = a_i$.

Summary of the talk

Summary of the talk

Objetive: Find a ruler to measure complexity

K **HT** **TF** **COF** **TA** **WF**.

Summary of the talk

Objetive: Find a ruler to measure complexity

K **HT** **TF** **COF** **TA** **WF**.

(1) We define an order $\leq_T$ in $\mathcal{P}(\mathbb{N})$ to compare the complexity of sets.

Summary of the talk

Objetive: Find a ruler to measure complexity

$$\mathbf{K} \qquad \mathbf{HT} \qquad \mathbf{TF} \qquad \mathbf{COF} \qquad \mathbf{TA} \qquad \mathbf{WF}.$$

(1) We define an order $\leq_T$ in $\mathcal{P}(\mathbb{N})$ to compare the complexity of sets.

(2) Properties of $(\mathcal{P}(\mathbb{N}): \leq_T)$

Summary of the talk

Objetive: Find a ruler to measure complexity

K **HT** **TF** **COF** **TA** **WF**.

(1) We define an order $\leq_T$ in $\mathcal{P}(\mathbb{N})$ to compare the complexity of sets.

(2) Properties of ($\mathcal{P}(\mathbb{N})$: $\leq_T$)

chaos!

Summary of the talk

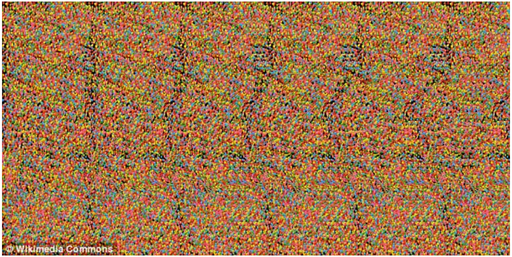
Objetive: Find a ruler to measure complexity

K **HT** **TF** **COF** **TA** **WF**.

(1) We define an order $\leq_T$ in $\mathcal{P}(\mathbb{N})$ to compare the complexity of sets.

(2) Properties of $(\mathcal{P}(\mathbb{N}) \colon \leq_T)$

chaos!



(3) The structure underneath the chaos: Martin's conjecture

Relative computability

Idea: To define an order relation $\leq_T$ such that:

Relative computability

Idea: To define an order relation $\leq_T$ such that:

$B \leq_T A$ if A contains enough information to calculate B (A is complicated as B)

Relative computability

Idea: To define an order relation $\leq_T$ such that:

$B \leq_T A$ if A contains enough information to calculate B (A is complicated as B)

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A* if there exists a program that calculates f using χ_A as primitive function.

($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of A)

Relative computability

Idea: To define an order relation $\leq_T$ such that:

$B \leq_T A$ if A contains enough information to calculate B (A is complicated as B)

Definition: Let $A \subseteq \mathbb{N}$. A function $f : \mathbb{N} \to \mathbb{N}$ is *computable in A* if there exists a program that calculates f using χ_A as primitive function.

($\chi_A : \mathbb{N} \to \{0, 1\}$ is the characteristic function of A)

We write $f \leq_T A$.

Relative computability

Idea: To define an order relation $\leq_T$ such that:

$B \leq_T A$ if A contains enough information to calculate B (A is complicated as B)

Definition: Let $A \subseteq \mathbb{N}$. A function $f \colon \mathbb{N} \to \mathbb{N}$ is *computable in A*
if there exists a program that calculates f using χ_A as primitive function.

($\chi_A \colon \mathbb{N} \to \{0, 1\}$ is the characteristic function of A)

We write $f \leq_T A$.

Example: $\mathbf{HT} \leq_T \mathbf{K}$

HT: *Hilbert's 10^{th}:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *The halting problem:* The set of programs that do **not** run forever.

Relative computability

Idea: To define an order relation $\leq_T$ such that:

$B \leq_T A$ if A contains enough information to calculate B (A is complicated as B)

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A* if there exists a program that calculates f using χ_A as primitive function.

($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of A)

We write $f \leq_T A$.

Example: $\mathbf{HT} \leq_T \mathbf{K}$

HT: *Hilbert's 10th:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *The halting problem:* The set of programs that do **not** run forever.

Proof:

Relative computability

Idea: To define an order relation $\leq_T$ such that:

$B \leq_T A$ if A contains enough information to calculate B (A is complicated as B)

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A*
if there exists a program that calculates f using χ_A as primitive function.

($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of A)

We write $f \leq_T A$.

Example: $\mathbf{HT} \leq_T \mathbf{K}$

HT: *Hilbert's 10^{th}:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *The halting problem:* The set of programs that do **not** run forever.

Proof: Given a polynomial $p(x_1, ..., x_k) \in \mathbb{Z}[x_1, x_2, ...]$

Relative computability

Idea: To define an order relation $\leq_T$ such that:

$B \leq_T A$ if A contains enough information to calculate B (A is complicated as B)

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A* if there exists a program that calculates f using χ_A as primitive function.

($\chi_A\colon \mathbb{N} \to \{0, 1\}$ is the characteristic function of A)

We write $f \leq_T A$.

Example: $\mathbf{HT} \leq_T \mathbf{K}$

HT: *Hilbert's 10th:* Polynomials in $\mathbb{Z}[x_1, x_2, \ldots]$ with integer roots.

K: *The halting problem:* The set of programs that do **not** run forever.

Proof: Given a polynomial $p(x_1, \ldots, x_k) \in \mathbb{Z}[x_1, x_2, \ldots]$ write a program $\mathtt{q}_p$ that enumerates all k-tuples $(a_1, \ldots, a_k) \in \mathbb{Z}^k$ and checks if they are roots of $p(x_1, \ldots, x_k)$.

Relative computability

Idea: To define an order relation $\leq_T$ such that:

$B \leq_T A$ if A contains enough information to calculate B (A is complicated as B)

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A* if there exists a program that calculates f using χ_A as primitive function.

($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of A)

We write $f \leq_T A$.

Example: $\mathbf{HT} \leq_T \mathbf{K}$

HT: *Hilbert's* 10^{th}*:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *The halting problem:* The set of programs that do **not** run forever.

Proof: Given a polynomial $p(x_1, ..., x_k) \in \mathbb{Z}[x_1, x_2, ...]$ write a program q_p that enumerates all k-tuples $(a_1, ..., a_k) \in \mathbb{Z}^k$ and checks if they are roots of $p(x_1, ..., x_k)$. If it finds one, the program stops.

Relative computability

Idea: To define an order relation $\leq_T$ such that:

$B \leq_T A$ if A contains enough information to calculate B (A is complicated as B)

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A* if there exists a program that calculates f using χ_A as primitive function.

($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of A)

We write $f \leq_T A$.

Example: $\mathbf{HT} \leq_T \mathbf{K}$

HT: *Hilbert's 10^{th}:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *The halting problem:* The set of programs that do **not** run forever.

Proof: Given a polynomial $p(x_1, ..., x_k) \in \mathbb{Z}[x_1, x_2, ...]$ write a program q_p that enumerates all k-tuples $(a_1, ..., a_k) \in \mathbb{Z}^k$ and checks if they are roots of $p(x_1, ..., x_k)$. If it finds one, the program stops. To know if $p(x_1, ..., x_k)$ has integer roots, ask $\mathbf{K}$ if this program q_p ever halts or not.

Examples

$$0 <_T \mathbf{HT} \equiv_T \mathbf{K} <_T \mathbf{TF} <_T \mathbf{COF} <_T \mathbf{TA} <_T \mathbf{WF}.$$

HT: *10^{th} Hilbert's problem:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

Examples

$$0 <_T \mathbf{HT} \equiv_T \mathbf{K} <_T \mathbf{TF} <_T \mathbf{COF} <_T \mathbf{TA} <_T \mathbf{WF}.$$

HT: *10^{th} Hilbert's problem:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *Halting problem:* The set of programs that do **not** run forever.

Examples

$$0 <_T \mathbf{HT} \equiv_T \mathbf{K} <_T \mathbf{TF} <_T \mathbf{COF} <_T \mathbf{TA} <_T \mathbf{WF}.$$

HT: *10th Hilbert's problem:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *Halting problem:* The set of programs that do **not** run forever.

TF: Finite presentations $((a_1, ..., a_k), (R_1, ..., R_\ell))$ of *torsion free* groups.

Examples

$$0 <_T \mathbf{HT} \equiv_T \mathbf{K} <_T \mathbf{TF} <_T \mathbf{COF} <_T \mathbf{TA} <_T \mathbf{WF}.$$

HT: *10th Hilbert's problem:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *Halting problem:* The set of programs that do **not** run forever.

TF: Finite presentations $((a_1, ..., a_k), (R_1, ..., R_\ell))$ of *torsion free* groups.

COF: Polynomials in $\mathbb{Z}[x, y_0, y_1, y_2, ...]$ with integer roots for <u>almost</u> every $x \in \mathbb{N}$.

Examples

$$0 <_T \mathbf{HT} \equiv_T \mathbf{K} <_T \mathbf{TF} <_T \mathbf{COF} <_T \mathbf{TA} <_T \mathbf{WF}.$$

HT: *10^{th} Hilbert's problem:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *Halting problem:* The set of programs that do **not** run forever.

TF: Finite presentations $((a_1, ..., a_k), (R_1, ..., R_\ell))$ of *torsion free* groups.

COF: Polynomials in $\mathbb{Z}[x, y_0, y_1, y_2, ...]$ with integer roots for almost every $x \in \mathbb{N}$.

TA: *True* first-order sentences about $(\mathbb{N}; 0, 1, +, \times)$

Examples

$$0 <_T \mathbf{HT} \equiv_T \mathbf{K} <_T \mathbf{TF} <_T \mathbf{COF} <_T \mathbf{TA} <_T \mathbf{WF}.$$

HT: *10^{th} Hilbert's problem:* Polynomials in $\mathbb{Z}[x_1, x_2, ...]$ with integer roots.

K: *Halting problem:* The set of programs that do **not** run forever.

TF: Finite presentations $((a_1, ..., a_k), (R_1, ..., R_\ell))$ of *torsion free* groups.

COF: Polynomials in $\mathbb{Z}[x, y_0, y_1, y_2, ...]$ with integer roots for almost every $x \in \mathbb{N}$.

TA: *True* first-order sentences about $(\mathbb{N}; 0, 1, +, \times)$

WF: Programs $\mathtt{p}$ for which there is a $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}(a_{i+1}) = a_i$.

Turing degrees

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A* if there exists a program that calculates f using χ_A as a primitive function.

($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of A)

We write $f \leq_T A$.

Turing degrees

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A*
if there exists a program that calculates f using χ_A as a primitive function.
($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of$A$)
We write $f \leq_T A$.

Definition: A and B are *Turing-equivalent* ($A \equiv_T B$) if $\mathcal{A} \leq_T B$ and $B \leq_T A$.

Turing degrees

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A*
if there exists a program that calculates f using χ_A as a primitive function.

($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of A)

We write $f \leq_T A$.

Definition: A and B are *Turing-equivalent* ($A \equiv_T B$) if $\mathcal{A} \leq_T B$ and $B \leq_T A$.

A *Turing degrees* is a $\equiv_T$-equivalence class.

Turing degrees

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A*
if there exists a program that calculates f using χ_A as a primitive function.

($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of$A$)

We write $f \leq_T A$.

Definition: A and B are *Turing-equivalent* ($A \equiv_T B$) if $\mathcal{A} \leq_T B$ and $B \leq_T A$.

A *Turing degrees* is a $\equiv_T$-equivalence class.

$\leq_T$ is a partial order in $\mathcal{P}(\mathbb{N})/\equiv_T$. (transitive and anti-symmetic)

Turing degrees

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A*
if there exists a program that calculates f using χ_A as a primitive function.
($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of$A$)

We write $f \leq_T A$.

Definition: A and B are *Turing-equivalent* ($A \equiv_T B$) if $\mathcal{A} \leq_T B$ and $B \leq_T A$.

A *Turing degrees* is a $\equiv_T$-equivalence class.

$\leq_T$ is a partial order in $\mathcal{P}(\mathbb{N})/\equiv_T$. (transitive and anti-symmetic)

Basic observations:

Turing degrees

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A* if there exists a program that calculates f using χ_A as a primitive function.

($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of$A$)

We write $f \leq_T A$.

Definition: A and B are *Turing-equivalent* ($A \equiv_T B$) if $\mathcal{A} \leq_T B$ and $B \leq_T A$.

A *Turing degrees* is a $\equiv_T$-equivalence class.

$\leq_T$ is a partial order in $\mathcal{P}(\mathbb{N})/\equiv_T$. (transitive and anti-symmetic)

Basic observations:

1. A and $\mathbb{N} \setminus A$ are Turing-equivalent.

Turing degrees

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A*
if there exists a program that calculates f using χ_A as a primitive function.
($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of$A$)
We write $f \leq_T A$.

Definition: A and B are *Turing-equivalent* ($A \equiv_T B$) if $\mathcal{A} \leq_T B$ and $B \leq_T A$.

A *Turing degrees* is a $\equiv_T$-equivalence class.

$\leq_T$ is a partial order in $\mathcal{P}(\mathbb{N})/\equiv_T$. (transitive and anti-symmetic)

Basic observations:

1. A and $\mathbb{N} \setminus A$ are Turing-equivalent.
2. if A is computable, $A \leq_T B$ for every $B \subseteq \mathbb{N}$.

Turing degrees

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A* if there exists a program that calculates f using χ_A as a primitive function.

($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of A)

We write $f \leq_T A$.

Definition: A and B are *Turing-equivalent* ($A \equiv_T B$) if $\mathcal{A} \leq_T B$ and $B \leq_T A$.

A *Turing degrees* is a $\equiv_T$-equivalence class.

$\leq_T$ is a partial order in $\mathcal{P}(\mathbb{N})/\equiv_T$. (transitive and anti-symmetic)

Basic observations:

1. A and $\mathbb{N} \setminus A$ are Turing-equivalent.
2. if A is computable, $A \leq_T B$ for every $B \subseteq \mathbb{N}$.
3. Given B, the set $\{A \subseteq \mathbb{N} : A \leq_T B\}$ is countable.

Turing degrees

Definition: Let $A \subseteq \mathbb{N}$. A function $f\colon \mathbb{N} \to \mathbb{N}$ is *computable in A*
if there exists a program that calculates f using χ_A as a primitive function.

($\chi_A\colon \mathbb{N} \to \{0,1\}$ is the characteristic function of A)

We write $f \leq_T A$.

Definition: A and B are *Turing-equivalent* ($A \equiv_T B$) if $\mathcal{A} \leq_T B$ and $B \leq_T A$.

A *Turing degrees* is a $\equiv_T$-equivalence class.

$\leq_T$ is a partial order in $\mathcal{P}(\mathbb{N})/\equiv_T$. (transitive and anti-symmetic)

Basic observations:

1. A and $\mathbb{N} \setminus A$ are Turing-equivalent.
2. if A is computable, $A \leq_T B$ for every $B \subseteq \mathbb{N}$.
3. Given B, the set $\{A \subseteq \mathbb{N} : A \leq_T B\}$ is countable.
4. $\mathcal{P}(\mathbb{N})/\equiv_T$ is uncountable.

Chaos

Until now, all examples are linearly ordered...

Chaos

Until now, all examples are linearly ordered...

BUT...

Chaos

Until now, all examples are linearly ordered...

BUT...

Theorem: [Kleene, Post] There are $A, B \subseteq \mathbb{N}$ such that $A \not\leq_T B$ and $B \not\leq_T A$.

Chaos

Until now, all examples are linearly ordered...

BUT...

Theorem: [Kleene, Post] There are $A, B \subseteq \mathbb{N}$ such that $A \not\leq_T B$ and $B \not\leq_T A$.
Even worst, there are uncountable $\leq_T$- antichains.

Chaos

Until now, all examples are linearly ordered...

BUT...

Theorem: [Kleene, Post] There are $A, B \subseteq \mathbb{N}$ such that $A \not\leq_T B$ and $B \not\leq_T A$.
Even worst, there are uncountable $\leq_T$- antichains.

Theorem: [Kleene, Post][Lachlan–Shore, Nerode]
Every countable partial ordering can be embedded in the Turing degrees.

Chaos

Until now, all examples are linearly ordered...

BUT...

Theorem: [Kleene, Post] There are $A, B \subseteq \mathbb{N}$ such that $A \not\leq_T B$ and $B \not\leq_T A$.

Even worst, there are uncountable $\leq_T$- antichains.

Theorem: [Kleene, Post][Lachlan–Shore, Nerode]
Every countable partial ordering can be embedded in the Turing degrees.

Even worst, it can be embedded below K.

Chaos

Until now, all examples are linearly ordered...

BUT...

Theorem: [Kleene, Post] There are $A, B \subseteq \mathbb{N}$ such that $A \not\leq_T B$ and $B \not\leq_T A$.

Even worst, there are uncountable $\leq_T$- antichains.

Theorem: [Kleene, Post][Lachlan–Shore, Nerode]
Every countable partial ordering can be embedded in the Turing degrees.

Even worst, it can be embedded below K.

There are many more results showing that $(\mathcal{P}(\mathbb{N}); \leq_T)$ is extremely complex.

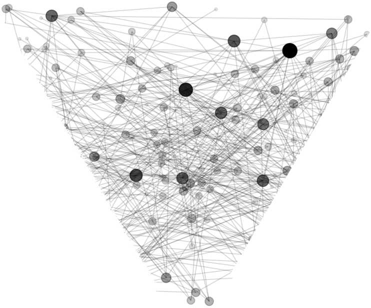
Natural examples

$0 <_T K \equiv_T WP \equiv_T HTP <_T TF <_T COF <_T TA <_T WF.$

$\leq_T$

⇐ Natural examples

All the sets ⇒



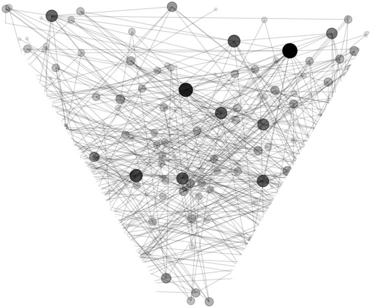
Natural examples

$0 <_T K \equiv_T WP \equiv_T HTP <_T TF <_T COF <_T TA <_T WF.$

$\leq_T$

$\Leftarrow$ Natural examples

All the sets $\Rightarrow$



Mathematically, we don't have a definition of "natural example".

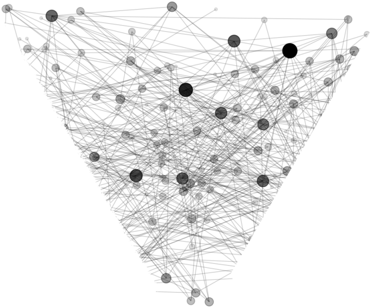
Natural examples

$0 <_T K \equiv_T WP \equiv_T HTP <_T TF <_T COF <_T TA <_T WF.$

$\leq_T$

$\Leftarrow$ Natural examples

All the sets $\Rightarrow$



Mathematically, we don't have a definition of "natural example".

We know that
all natural examples are:

- constructible
- relativizable

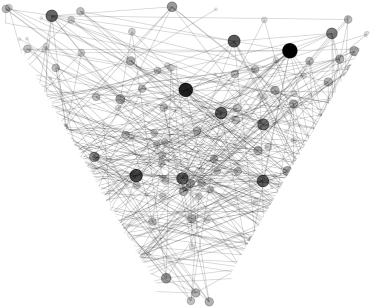
Natural examples

$0 <_T K \equiv_T WP \equiv_T HTP <_T TF <_T COF <_T TA <_T WF.$

$\leq_T$

$\Leftarrow$ Natural examples

All the sets $\Rightarrow$



Mathematically, we don't have a definition of "natural example".

We know that
all natural examples are:

- constructible
- relativizable

Plan: relativizable examples $\Longleftrightarrow$ functions $\mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ $\equiv_T$-invariants

$\Updownarrow$ $\Updownarrow$

ordered by $\leq_T$ $\Longleftrightarrow$ Ordered by $\leq_T^{\nabla}$

Relativization

Observación:
The functions that are computable in A behave "like" the computable functions.

Relativization

Observación:
The functions that are computable in A behave "like" the computable functions.

Relativize a construction, a theorem, or a proof to a set A
means to change the notion of computable for that of computable in A.

Relativization

Observación:
The functions that are computable in A behave "like" the computable functions.

Relativize a construction, a theorem, or a proof to a set A
means to change the notion of computable for that of computable in A.

Examples:

Relativization

Observación:
The functions that are computable in A behave "like" the computable functions.

Relativize a construction, a theorem, or a proof to a set A
means to change the notion of computable for that of computable in A.

Examples:

K $=$ The set of programas `p` that eventually halt.

Relativization

Observación:
The functions that are computable in A behave "like" the computable functions.

Relativize a construction, a theorem, or a proof to a set A
means to change the notion of computable for that of computable in A.

Examples:

$\mathbf{K}$ = The set of programas $\mathtt{p}$ that eventually halt.
$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

Relativization

Observación:
The functions that are computable in A behave "like" the computable functions.

Relativize a construction, a theorem, or a proof to a set A
means to change the notion of computable for that of computable in A.

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

FIN = The set of programas $\mathtt{p}(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x) \text{ halts}\}$ is finite.

Relativization

Observación:
The functions that are computable in A behave "like" the computable functions.

Relativize a construction, a theorem, or a proof to a set A
means to change the notion of computable for that of computable in A.

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

FIN = The set of programas $\mathtt{p}(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x) \text{ halts}\}$ is finite. $(\equiv_T \mathbf{TF})$

Relativization

Observación:
The functions that are computable in A behave "like" the computable functions.

Relativize a construction, a theorem, or a proof to a set A
means to change the notion of computable for that of computable in A.

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

Relativization

Observación:
The functions that are computable in A behave "like" the computable functions.

Relativize a construction, a theorem, or a proof to a set A
means to change the notion of computable for that of computable in A.

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

Relativization

Observación:
The functions that are computable in A behave "like" the computable functions.

Relativize a construction, a theorem, or a proof to a set A
means to change the notion of computable for that of computable in A.

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

Relativization

Observación:
The functions that are computable in A behave "like" the computable functions.

Relativize a construction, a theorem, or a proof to a set A
means to change the notion of computable for that of computable in A.

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

Relativization

Observación:
The functions that are computable in A behave "like" the computable functions.

Relativize a construction, a theorem, or a proof to a set A
means to change the notion of computable for that of computable in A.

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.
$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A$ halts$\}$ is finite.
$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.
$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$
$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

Relativization

Relativize a construction, a theorem, or a proof to a set A
means to change the notion of computable for that of computable in A.

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

Relativization

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

Relativization

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

$$0 <_T \mathbf{K} <_T \mathbf{FIN} <_T \mathbf{COF} <_T \cdots <_T \mathbf{TA} <_T \cdots <_T \mathbf{WF}$$

Relativization

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

$$A <_T \mathbf{K}^A <_T \mathbf{FIN}^A <_T \mathbf{COF}^A <_T \cdots <_T \mathbf{TA}^A <_T \cdots <_T \mathbf{WF}^A$$

Relativization

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

$$A <_T \mathbf{K}^A <_T \mathbf{FIN}^A <_T \mathbf{COF}^A <_T \cdots <_T \mathbf{TA}^A <_T \cdots <_T \mathbf{WF}^A$$

Definition: The function $A \mapsto \mathbf{K}^A : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is called the *Turing jump*.

Relativization

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

$$A \ <_T \ \mathbf{K}^A \ <_T \ \mathbf{FIN}^A \ <_T \ \mathbf{COF}^A \ <_T \cdots <_T \ \mathbf{TA}^A \ <_T \cdots <_T \ \mathbf{WF}^A$$

Definition: The function $A \mapsto \mathbf{K}^A : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is called the *Turing jump*.

- For every A, $\quad A <_T \mathbf{K}^A$.

Relativization

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

$$A \;<_T\; \mathbf{K}^A \;<_T\; \mathbf{FIN}^A \;<_T\; \mathbf{COF}^A \;<_T \cdots <_T\; \mathbf{TA}^A \;<_T \cdots <_T\; \mathbf{WF}^A$$

Definition: The function $A \mapsto \mathbf{K}^A : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is called the *Turing jump*.

- For every A, $\quad A <_T \mathbf{K}^A$.
- if $A \equiv_T B$, $\quad \mathbf{K}^A \equiv_T \mathbf{K}^B$

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

$$A <_T \mathbf{K}^A <_T \mathbf{FIN}^A <_T \mathbf{COF}^A <_T \cdots <_T \mathbf{TA}^A <_T \cdots <_T \mathbf{WF}^A$$

Definition: The function $A \mapsto \mathbf{K}^A : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is called the *Turing jump*.

- For every A, $A <_T \mathbf{K}^A$.
- if $A \equiv_T B$, $\mathbf{K}^A \equiv_T \mathbf{K}^B$
- $\mathbf{K}^{\mathbf{K}} \equiv_T$

Relativization

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

$$A <_T \mathbf{K}^A <_T \mathbf{FIN}^A <_T \mathbf{COF}^A <_T \cdots <_T \mathbf{TA}^A <_T \cdots <_T \mathbf{WF}^A$$

Definition: The function $A \mapsto \mathbf{K}^A : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is called the *Turing jump*.

- For every A, $A <_T \mathbf{K}^A$.
- if $A \equiv_T B$, $\mathbf{K}^A \equiv_T \mathbf{K}^B$
- $\mathbf{K}^{\mathbf{K}} \equiv_T \mathbf{FIN}$

Relativization

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

$$A <_T \mathbf{K}^A <_T \mathbf{FIN}^A <_T \mathbf{COF}^A <_T \cdots <_T \mathbf{TA}^A <_T \cdots <_T \mathbf{WF}^A$$

Definition: The function $A \mapsto \mathbf{K}^A : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is called the *Turing jump*.

- For every A, $A <_T \mathbf{K}^A$.
- if $A \equiv_T B$, $\mathbf{K}^A \equiv_T \mathbf{K}^B$
- $\mathbf{K}^{\mathbf{K}} \equiv_T \mathbf{FIN}$
- $\mathbf{K}^{\mathbf{FIN}} \equiv_T$

Relativization

Examples:

$\mathbf{K}^A$ = The set of programas $\mathtt{p}^A$ that use the funciton χ_A and eventually stop.

$\mathbf{FIN}^A$ = The set of programas $\mathtt{p}^A(x)$ such that $\{x \in \mathbb{N} : \mathtt{p}(x)^A \text{ halts}\}$ is finite.

$\mathbf{COF}^A$ = The set of programas $\mathtt{p}^A(x)$ that halts for almost every $x \in \mathbb{N}$.

$\mathbf{TA}^A$: True first order sentence about $(\mathbb{N}; A, 0, 1, +, \times)$

$\mathbf{WF}^A$: Programs $\mathtt{p}^A$ for which there exists $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{N}$ such that $\mathtt{p}^A(a_{i+1}) = a_i$.

$$A <_T \mathbf{K}^A <_T \mathbf{FIN}^A <_T \mathbf{COF}^A <_T \cdots <_T \mathbf{TA}^A <_T \cdots <_T \mathbf{WF}^A$$

Definition: The function $A \mapsto \mathbf{K}^A : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is called the *Turing jump*.

- For every A, $A <_T \mathbf{K}^A$.
- if $A \equiv_T B$, $\mathbf{K}^A \equiv_T \mathbf{K}^B$
- $\mathbf{K}^{\mathbf{K}} \equiv_T \mathbf{FIN}$
- $\mathbf{K}^{\mathbf{FIN}} \equiv_T \mathbf{COF}$

Example of relativization in algebra

(all rings are countable, commutative, and with unity)

Theorem: Every ring $(D; 0, 1, +, \times)$ has a maximal ideal.

Example of relativization in algebra

(all rings are countable, commutative, and with unity)

Theorem: Every ring $(D; 0, 1, +, \times)$ has a maximal ideal.

Question ¿How hard is it to find it?

Example of relativization in algebra

(all rings are countable, commutative, and with unity)

Theorem: Every ring $(D; 0, 1, +, \times)$ has a maximal ideal.

Question ¿How hard is it to find it?

Given $A \in \mathcal{P}(A)$, let $\mathbf{MI}(A)$ be the least Turing degree such that

every ring computable in A has a maximal ideal computable in $\mathbf{MI}(A)$

Example of relativization in algebra

(all rings are countable, commutative, and with unity)

Theorem: Every ring $(D; 0, 1, +, \times)$ has a maximal ideal.

Question ¿How hard is it to find it?

Given $A \in \mathcal{P}(A)$, let $\mathbf{MI}(A)$ be the least Turing degree such that

every ring computable in A has a maximal ideal computable in $\mathbf{MI}(A)$

Theorem: [Friedman, Simpson, Smith '85] For every A, $\mathbf{MI}(A) \equiv_T \mathbf{K}^A$.

Example of relativization in algebra

(all rings are countable, commutative, and with unity)

Theorem: Every ring $(D; 0, 1, +, \times)$ has a maximal ideal.

Question ¿How hard is it to find it?

Given $A \in \mathcal{P}(A)$, let $\mathbf{MI}(A)$ be the least Turing degree such that

every ring computable in A has a maximal ideal computable in $\mathbf{MI}(A)$

Theorem: [Friedman, Simpson, Smith '85] For every A, $\mathbf{MI}(A) \equiv_T \mathbf{K}^A$.

The Jacobson ideal is the intersection of all the maximal ideals.

Example of relativization in algebra

(all rings are countable, commutative, and with unity)

Theorem: Every ring $(D; 0, 1, +, \times)$ has a maximal ideal.

Question ¿How hard is it to find it?

Given $A \in \mathcal{P}(A)$, let $\mathbf{MI}(A)$ be the least Turing degree such that

every ring computable in A has a maximal ideal computable in $\mathbf{MI}(A)$

Theorem: [Friedman, Simpson, Smith '85] For every A, $\mathbf{MI}(A) \equiv_T \mathbf{K}^A$.

The Jacobson ideal is the intersection of all the maximal ideals.

Given $A \in \mathcal{P}(A)$, let $\mathbf{JI}(A)$ be the least Turing degree such that

for every ring computable in A, its Jacobson ideal is computable in $\mathbf{JI}(A)$

Example of relativization in algebra

(all rings are countable, commutative, and with unity)

Theorem: Every ring $(D; 0, 1, +, \times)$ has a maximal ideal.

Question ¿How hard is it to find it?

Given $A \in \mathcal{P}(A)$, let $\mathbf{MI}(A)$ be the least Turing degree such that

every ring computable in A has a maximal ideal computable in $\mathbf{MI}(A)$

Theorem: [Friedman, Simpson, Smith '85] For every A, $\mathbf{MI}(A) \equiv_T \mathbf{K}^A$.

The Jacobson ideal is the intersection of all the maximal ideals.

Given $A \in \mathcal{P}(A)$, let $\mathbf{JI}(A)$ be the least Turing degree such that

for every ring computable in A, its Jacobson ideal is computable in $\mathbf{JI}(A)$

Theorem: [Downey, Lempp, Mileti '07] For every A, $\mathbf{JI}(A) \equiv_T \mathbf{K}^{\mathbf{K}^A}$.

Invariant functions

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Invariant functions

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Examples: $\mathbf{K}^{\odot}$, $\mathbf{FIN}^{\odot}$, $\mathbf{COF}^{\odot}$, $\mathbf{TA}^{\odot}$, $\mathbf{WF}^{\odot}$, $\mathbf{MI}(\odot)$, $\mathbf{JI}(\odot)$,....

Invariant functions

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Examples: $\mathbf{K}^{\odot}$, $\mathbf{FIN}^{\odot}$, $\mathbf{COF}^{\odot}$, $\mathbf{TA}^{\odot}$, $\mathbf{WF}^{\odot}$, $\mathbf{MI}(\odot)$, $\mathbf{JI}(\odot)$,....

Empirical observation: Natural Turing degrees induce $\equiv_T$ -invariant functions.

Invariant functions

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Examples: $\mathbf{K}^{\odot}$, $\mathbf{FIN}^{\odot}$, $\mathbf{COF}^{\odot}$, $\mathbf{TA}^{\odot}$, $\mathbf{WF}^{\odot}$, $\mathbf{MI}(\odot)$, $\mathbf{JI}(\odot)$,....

Empirical observation: Natural Turing degrees induce $\equiv_T$ -invariant functions.

Definition: Given $F, G\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$, we say that $F \leq_T G$ if,

for every B, $\quad F(B) \leq_T G(B)$.

Invariant functions

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Examples: $\mathbf{K}^{\odot}$, $\mathbf{FIN}^{\odot}$, $\mathbf{COF}^{\odot}$, $\mathbf{TA}^{\odot}$, $\mathbf{WF}^{\odot}$, $\mathbf{MI}(\odot)$, $\mathbf{JI}(\odot)$,....

Empirical observation: Natural Turing degrees induce $\equiv_T$ -invariant functions.

Definition: Given $F, G\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$, we say that $F \leq_T G$ if,

for every B, $\quad F(B) \leq_T G(B)$.

We know that for every B

$$0 <_T \mathbf{K}^B \equiv_T \mathbf{MI}(B) <_T \mathbf{FIN}^B \equiv_T \mathbf{JI}(B) <_T \mathbf{COF}^B <_T \mathbf{TA}^B <_T \mathbf{WF}^B$$

Invariant functions

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Examples: $\mathbf{K}^{\odot}$, $\mathbf{FIN}^{\odot}$, $\mathbf{COF}^{\odot}$, $\mathbf{TA}^{\odot}$, $\mathbf{WF}^{\odot}$, $\mathbf{MI}(\odot)$, $\mathbf{JI}(\odot)$,....

Empirical observation: Natural Turing degrees induce $\equiv_T$ -invariant functions.

Definition: Given $F, G\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$, we say that $F \leq_T G$ if,

for every B, $F(B) \leq_T G(B)$.

We know that for every B

$$0 <_T \mathbf{K}^B \equiv_T \mathbf{MI}(B) <_T \mathbf{FIN}^B \equiv_T \mathbf{JI}(B) <_T \mathbf{COF}^B <_T \mathbf{TA}^B <_T \mathbf{WF}^B$$

thus, if we look at the corresponding functions:

$$0 <_T \mathbf{K}^{\odot} \equiv_T \mathbf{MI}(\odot) <_T \mathbf{FIN}^{\odot} \equiv_T \mathbf{JI}(\odot) <_T \mathbf{COF}^{\odot} <_T \mathbf{TA}^{\odot} <_T \mathbf{WF}^{\odot}$$

Invariant functions

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Examples: $\mathbf{K}^{\odot}$, $\mathbf{FIN}^{\odot}$, $\mathbf{COF}^{\odot}$, $\mathbf{TA}^{\odot}$, $\mathbf{WF}^{\odot}$, $\mathbf{MI}(\odot)$, $\mathbf{JI}(\odot)$,....

Empirical observation: Natural Turing degrees induce $\equiv_T$ -invariant functions.

Definition: Given $F, G\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$, we say that $F \leq_T G$ if,

for every B, $F(B) \leq_T G(B)$.

We know that for every B

$$0 <_T \mathbf{K}^B \equiv_T \mathbf{MI}(B) <_T \mathbf{FIN}^B \equiv_T \mathbf{JI}(B) <_T \mathbf{COF}^B <_T \mathbf{TA}^B <_T \mathbf{WF}^B$$

thus, if we look at the corresponding functions:

$$0 <_T \mathbf{K}^{\odot} \equiv_T \mathbf{MI}(\odot) <_T \mathbf{FIN}^{\odot} \equiv_T \mathbf{JI}(\odot) <_T \mathbf{COF}^{\odot} <_T \mathbf{TA}^{\odot} <_T \mathbf{WF}^{\odot}$$

Problem: There are $\equiv_T$ -invariant functions of all shapes and colors.

Invariant functions

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Examples: $\mathbf{K}^{\odot}$, $\mathbf{FIN}^{\odot}$, $\mathbf{COF}^{\odot}$, $\mathbf{TA}^{\odot}$, $\mathbf{WF}^{\odot}$, $\mathbf{MI}(\odot)$, $\mathbf{JI}(\odot)$,....

Empirical observation: Natural Turing degrees induce $\equiv_T$ -invariant functions.

Definition: Given $F, G\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$, we say that $F \leq_T G$ if,

for every B, $\quad F(B) \leq_T G(B)$.

We know that for every B

$$0 <_T \mathbf{K}^B \equiv_T \mathbf{MI}(B) <_T \mathbf{FIN}^B \equiv_T \mathbf{JI}(B) <_T \mathbf{COF}^B <_T \mathbf{TA}^B <_T \mathbf{WF}^B$$

thus, if we look at the corresponding functions:

$$0 <_T \mathbf{K}^{\odot} \equiv_T \mathbf{MI}(\odot) <_T \mathbf{FIN}^{\odot} \equiv_T \mathbf{JI}(\odot) <_T \mathbf{COF}^{\odot} <_T \mathbf{TA}^{\odot} <_T \mathbf{WF}^{\odot}$$

Problem: There are $\equiv_T$ -invariant functions of all shapes and colors.

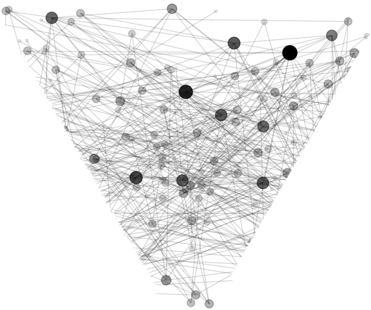
...but not if we compare them at the “limit”.

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.



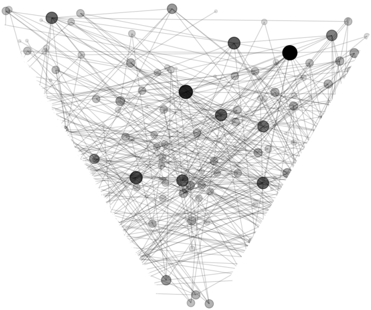
F is increasing if $F(A) \geq_T A$ for every A.

$\equiv_T$-invariants increasing
Functions

$\Leftarrow$ ordered by $\leq_T$

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.



F is increasing if $F(A) \geq_T A$ for every A.

$\equiv_T$-invariants increasing Functions

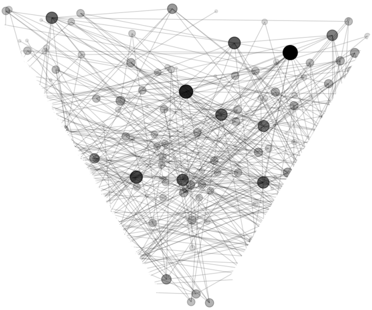
$\Leftarrow$ ordered by $\leq_T$

ordered by $\leq_T^{\triangledown}$ $\Rightarrow$

$0 <_T K \equiv_T WP \equiv_T HTP <_T TF <_T COF <_T TA <_T WF.$

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.



F is increasing if $F(A) \geq_T A$ for every A.

$\equiv_T$-invariants increasing
Functions

$\Leftarrow$ ordered by $\leq_T$

ordered by $\leq_T^\nabla$ $\Rightarrow$

$0 <_T K \equiv_T WP \equiv_T HTP <_T TF <_T COF <_T TA <_T WF.$

Definition: $F \leq_T^\nabla G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Definition: $F \leq_T^{\triangledown} G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Definition: $F \leq_T^{\nabla} G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Definition: $F \leq_T^{\triangledown} G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.

Let's use $\mathbf{K}^F$ to call the function $A \mapsto \mathbf{K}^{F(A)}$, the Turing jump of F.

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Definition: $F \leq_T^{\nabla} G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.

$$\mathbf{K}^F \quad \geq_T^{\nabla} \quad F$$

Let's use $\mathbf{K}^F$ to call the function $A \mapsto \mathbf{K}^{F(A)}$, the Turing jump of F.
Recall that $F <_T \mathbf{K}^F$.

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Definition: $F \leq_T^{\triangledown} G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.

$$\mathbf{K}^F \; \mathbin{\rotatebox{90}{$\geq_T^{\triangledown}$}} \; F$$

Let's use $\mathbf{K}^F$ to call the function $A \mapsto \mathbf{K}^{F(A)}$, the Turing jump of F.
Recall that $F <_T \mathbf{K}^F$.

Conjectura of Martin: (ZF+AD+DC)

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Definition: $F \leq_T^{\nabla} G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.

$\mathbf{K}^F$

$\geq_T^{\nabla}$

F

Let's use $\mathbf{K}^F$ to call the function $A \mapsto \mathbf{K}^{F(A)}$, the Turing jump of F.
Recall that $F <_T \mathbf{K}^F$.

Conjectura of Martin: (ZF+AD+DC)

1. Every $\equiv_T$-invariant function is $\equiv_T^{\nabla}$-equivalent to one that is constant or increasing.

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Definition: $F \leq_T^{\nabla} G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.

$\mathbf{K}^F$

$\geq_T^{\nabla}$

F

Let's use $\mathbf{K}^F$ to call the function $A \mapsto \mathbf{K}^{F(A)}$, the Turing jump of F.
Recall that $F <_T \mathbf{K}^F$.

Conjectura of Martin: (ZF+AD+DC)

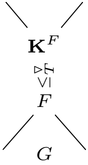
1. Every $\equiv_T$-invariant function is $\equiv_T^{\nabla}$-equivalent to one that is constant or increasing.
2. if F, G are $\equiv_T$-invariant increasing functions $\implies$ $G \leq_T^{\nabla} F$ o $\mathbf{K}^F \leq_T^{\nabla} G$.

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Definition: $F \leq_T^{\triangledown} G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.



Let's use $\mathbf{K}^F$ to call the function $A \mapsto \mathbf{K}^{F(A)}$, the Turing jump of F.
Recall that $F <_T \mathbf{K}^F$.

Conjectura of Martin: (ZF+AD+DC)

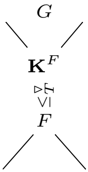
1. Every $\equiv_T$-invariant function is $\equiv_T^{\triangledown}$-equivalent to one that is constant or increasing.
2. if F, G are $\equiv_T$-invariant increasing functions $\implies$ $G \leq_T^{\triangledown} F$ o $\mathbf{K}^F \leq_T^{\triangledown} G$.

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Definition: $F \leq_T^{\nabla} G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.



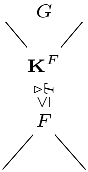
Let's use $\mathbf{K}^F$ to call the function $A \mapsto \mathbf{K}^{F(A)}$, the Turing jump of F.
Recall that $F <_T \mathbf{K}^F$.

Conjectura of Martin: (ZF+AD+DC)

1. Every $\equiv_T$-invariant function is $\equiv_T^{\nabla}$-equivalent to one that is constant or increasing.
2. if F, G are $\equiv_T$-invariant increasing functions $\implies$ $G \leq_T^{\nabla} F$ o $\mathbf{K}^F \leq_T^{\nabla} G$.

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.



Definition: $F \leq_T^{\nabla} G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.

Let's use $\mathbf{K}^F$ to call the function $A \mapsto \mathbf{K}^{F(A)}$, the Turing jump of F.
Recall that $F <_T \mathbf{K}^F$.

Conjectura of Martin: (ZF+AD+DC)

1. Every $\equiv_T$-invariant function is $\equiv_T^{\nabla}$-equivalent to one that is constant or increasing.
2. if F, G are $\equiv_T$-invariant increasing functions $\implies$ $G \leq_T^{\nabla} F$ o $\mathbf{K}^F \leq_T^{\nabla} G$.

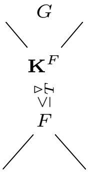
Thm:[Steel 82][Slaman-Steel 88] It's true for the uniformly $\equiv_T$-invariant functions.

Martin's conjecture

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Definition: $F \leq_T^{\nabla} G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.



Let's use $\mathbf{K}^F$ to call the function $A \mapsto \mathbf{K}^{F(A)}$, the Turing jump of F.
Recall that $F <_T \mathbf{K}^F$.

Conjectura of Martin: (ZF+AD+DC)

1. Every $\equiv_T$-invariant function is $\equiv_T^{\nabla}$-equivalent to one that is constant or increasing.
2. if F, G are $\equiv_T$-invariant increasing functions $\implies$ $G \leq_T^{\nabla} F$ o $\mathbf{K}^F \leq_T^{\nabla} G$.

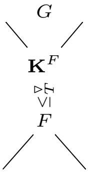
Thm:[Steel 82][Slaman-Steel 88] It's true for the uniformly $\equiv_T$-invariant functions.

Thm:[Kihara-Montalbán 18]: Connect natural many-one degrees and Wadge degrees.

Definition: $F\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ is $\equiv_T$ -invariant if $A \equiv_T B \implies F(A) \equiv_T F(B)$.

Definition: $F \leq_T^{\nabla} G$ if there exists A such that,

for every $B \geq_T A$, $F(B) \leq_T G(B)$.



Let's use $\mathbf{K}^F$ to call the function $A \mapsto \mathbf{K}^{F(A)}$, the Turing jump of F.
Recall that $F <_T \mathbf{K}^F$.

Conjectura of Martin: (ZF+AD+DC)

1. Every $\equiv_T$-invariant function is $\equiv_T^{\nabla}$-equivalent to one that is constant or increasing.
2. if F, G are $\equiv_T$-invariant increasing functions $\implies$ $G \leq_T^{\nabla} F$ o $\mathbf{K}^F \leq_T^{\nabla} G$.

Thm:[Steel 82][Slaman-Steel 88] It's true for the uniformly $\equiv_T$-invariant functions.

Thm:[Kihara-Montalbán 18]: Connect natural many-one degrees and Wadge degrees.

The conjecture is still open for the general case.

Wadge degrees

Consider the *Baire Space*: $\mathbb{N}^\mathbb{N} = \{f \colon \mathbb{N} \to \mathbb{N}\}$ with the product topology.

Wadge degrees

Consider the *Baire Space*: $\mathbb{N}^\mathbb{N} = \{f \colon \mathbb{N} \to \mathbb{N}\}$ with the product topology.

Obs: $\mathbb{N}^\mathbb{N}$ is homeomorphic to $\mathbb{R}^+ \setminus \mathbb{Q}$ via $f \mapsto f(0) + \frac{1}{1+f(1)+\frac{1}{1+f(2)+\cdots}}$

Wadge degrees

Consider the *Baire Space*: $\mathbb{N}^{\mathbb{N}} = \{f \colon \mathbb{N} \to \mathbb{N}\}$ with the product topology.

Obs: $\mathbb{N}^{\mathbb{N}}$ is homeomorphic to $\mathbb{R}^{+} \setminus \mathbb{Q}$ via $f \mapsto f(0) + \frac{1}{1+f(1)+\frac{1}{1+f(2)+\cdots}}$

Definition: For $A, B \subseteq \mathbb{N}^{\mathbb{N}}$, A is *Wadge reducible to* B, $A \leq_w B$ if there is a continuous $f \colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ s.t. $(\forall X \in 2^{\mathbb{N}})$, $X \in A \iff f(X) \in B$.

Wadge degrees

Consider the *Baire Space*: $\mathbb{N}^{\mathbb{N}} = \{f\colon \mathbb{N} \to \mathbb{N}\}$ with the product topology.

Obs: $\mathbb{N}^{\mathbb{N}}$ is homeomorphic to $\mathbb{R}^+ \setminus \mathbb{Q}$ via $f \mapsto f(0) + \frac{1}{1+f(1)+\frac{1}{1+f(2)+\cdots}}$

Definition: For $A, B \subseteq \mathbb{N}^{\mathbb{N}}$, A is *Wadge reducible to* B, $A \leq_w B$ if there is a continuous $f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ s.t. $(\forall X \in 2^{\mathbb{N}})$, $X \in A \iff f(X) \in B$.

Theorem: [Wadge 83](AD) The Wadge degrees are almost linearly ordered:

- For every $A, B \subseteq \mathbb{N}^{\mathbb{N}}$, either $A \leq_w B$ or $B \leq_w A^c$.
- For every $A, B \subseteq \mathbb{N}^{\mathbb{N}}$, if $A <_w B$, then $A <_w B^c$.

Theorem: (AD) [Martin] The Wadge degrees are well founded.

Natural many-one degrees $\iff$ Wadge degrees

Natural many-one degrees $\iff$ Wadge degrees

Definition: A set $A \subseteq \mathbb{N}$ is *many-one reducible* to $B \subseteq \mathbb{N}$ ($A \leq_m B$), if there is a computable $f\colon \mathbb{N} \to \mathbb{N}$ such that $n \in A \iff f(n) \in B \quad (\forall n)$.

Natural many-one degrees $\Longleftrightarrow$ Wadge degrees

Definition: A set $A \subseteq \mathbb{N}$ is *many-one reducible* to $B \subseteq \mathbb{N}$ ($A \leq_m B$), if there is a computable $f\colon \mathbb{N} \to \mathbb{N}$ such that $n \in A \iff f(n) \in B \quad (\forall n)$.

Definition: A function $f\colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ is *$(\equiv_T, \equiv_m)$-uniformly invariant (UI)* if

$X \equiv_T Y \Longrightarrow f(X) \equiv_m f(Y)$ and

Natural many-one degrees $\iff$ Wadge degrees

Definition: A set $A \subseteq \mathbb{N}$ is *many-one reducible* to $B \subseteq \mathbb{N}$ ($A \leq_m B$),
if there is a computable $f \colon \mathbb{N} \to \mathbb{N}$ such that $n \in A \iff f(n) \in B \quad (\forall n)$.

Definition: A function $f \colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ is *$(\equiv_T, \equiv_m)$-uniformly invariant (UI)* if

$$X \equiv_T Y \Longrightarrow f(X) \equiv_m f(Y) \qquad \text{and}$$

there is $u \colon \mathbb{N}^2 \to \mathbb{N}^2$, s.t., if $X \equiv_T Y$ via Φ_i and Φ_j, then $f(X) \equiv_m f(Y)$ via $\Phi_{u_0(i,j)}$ and $\Phi_{u_1(i,j)}$.

Natural many-one degrees $\iff$ Wadge degrees

Definition: A set $A \subseteq \mathbb{N}$ is *many-one reducible* to $B \subseteq \mathbb{N}$ ($A \leq_m B$), if there is a computable $f \colon \mathbb{N} \to \mathbb{N}$ such that $n \in A \iff f(n) \in B \quad (\forall n)$.

Definition: A function $f \colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ is *$(\equiv_T, \equiv_m)$-uniformly invariant (UI)* if

$X \equiv_T Y \implies f(X) \equiv_m f(Y)$ and

there is $u \colon \mathbb{N}^2 \to \mathbb{N}^2$, s.t., if $X \equiv_T Y$ via Φ_i and Φ_j, then $f(X) \equiv_m f(Y)$ via $\Phi_{u_0(i,j)}$ and $\Phi_{u_1(i,j)}$.

Def: For $A, B \subseteq \mathbb{N}$, A is *many-one reducible^Z to* B, written $A \leq_m^Z B$, if there is a Z-computable $f \colon \mathbb{N} \to \mathbb{N}$ s.t. $(\forall x \in \mathbb{N})$, $x \in A \iff f(x) \in B$.

Natural many-one degrees $\iff$ Wadge degrees

Definition: A set $A \subseteq \mathbb{N}$ is *many-one reducible* to $B \subseteq \mathbb{N}$ ($A \leq_m B$), if there is a computable $f\colon \mathbb{N} \to \mathbb{N}$ such that $n \in A \iff f(n) \in B \quad (\forall n)$.

Definition: A function $f\colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ is *$(\equiv_T, \equiv_m)$-uniformly invariant (UI)* if

$$X \equiv_T Y \Longrightarrow f(X) \equiv_m f(Y) \qquad \text{and}$$

there is $u\colon \mathbb{N}^2 \to \mathbb{N}^2$, s.t., if $X \equiv_T Y$ via Φ_i and Φ_j, then $f(X) \equiv_m f(Y)$ via $\Phi_{u_0(i,j)}$ and $\Phi_{u_1(i,j)}$.

Def: For $A, B \subseteq \mathbb{N}$, A is *many-one reducible^Z to* B, written $A \leq_m^Z B$, if there is a Z-computable $f\colon \mathbb{N} \to \mathbb{N}$ s.t. $(\forall x \in \mathbb{N})$, $x \in A \iff f(x) \in B$.

Def: $f \leq_{\mathbf{m}}^{\triangledown} g$ if $(\exists C \in 2^{\mathbb{N}})$ such that $f(X) \leq_m^C g(X)$ for every $X \geq_T C$.

Natural many-one degrees $\iff$ Wadge degrees

Definition: A set $A \subseteq \mathbb{N}$ is *many-one reducible* to $B \subseteq \mathbb{N}$ ($A \leq_m B$), if there is a computable $f\colon \mathbb{N} \to \mathbb{N}$ such that $n \in A \iff f(n) \in B \quad (\forall n)$.

Definition: A function $f\colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ is *$(\equiv_T, \equiv_m)$-uniformly invariant (UI)* if

$X \equiv_T Y \implies f(X) \equiv_m f(Y)$ and

there is $u\colon \mathbb{N}^2 \to \mathbb{N}^2$, s.t., if $X \equiv_T Y$ via Φ_i and Φ_j, then $f(X) \equiv_m f(Y)$ via $\Phi_{u_0(i,j)}$ and $\Phi_{u_1(i,j)}$.

Def: For $A, B \subseteq \mathbb{N}$, A is *many-one reducible^Z to* B, written $A \leq_m^Z B$, if there is a Z-computable $f\colon \mathbb{N} \to \mathbb{N}$ s.t. $(\forall x \in \mathbb{N})$, $x \in A \iff f(x) \in B$.

Def: $f \leq_{\mathbf{m}}^{\nabla} g$ if $(\exists C \in 2^{\mathbb{N}})$ such that $f(X) \leq_m^C g(X)$ for every $X \geq_T C$.

Theorem: [Kihara, M.] There is a one-to-one correspondence between $(\equiv_T, \equiv_m)$-UI functions ordered by $\leq_{\mathbf{m}}^{\nabla}$ and $\mathcal{P}(2^{\mathbb{N}})$ ordered by Wadge reducibility.