A classification of the natural Many-one degrees.

Antonio Montalbán

U.C. Berkeley

June, 2017
Nagoya, Japan

Joint work with Takayuki Kihara.

Example — Natural objects in proof theory

Example — Natural objects in proof theory

Empirical fact: *Natural* theories are well-ordered by consistency strength.

Example — Natural objects in proof theory

Empirical fact: *Natural* theories are well-ordered by consistency strength.

A theory is a set of axioms,

Example — Natural objects in proof theory

Empirical fact: *Natural* theories are well-ordered by consistency strength.

A theory is a set of axioms, like

- Peano Arithmetic.
- Second order Number theory.
- Zermelo Fraenkel set theory.
- Large cardinal assumptions.

Example — Natural objects in proof theory

Empirical fact: *Natural* theories are well-ordered by consistency strength.

A theory is a set of axioms, like

- Peano Arithmetic.
- Second order Number theory.
- Zermelo Fraenkel set theory.
- Large cardinal assumptions.

Def: For theories T and T', let $T \ll T'$ if T' can prove the consistency of T.

Example — Natural objects in proof theory

Empirical fact: *Natural* theories are well-ordered by consistency strength.

A theory is a set of axioms, like

- Peano Arithmetic.
- Second order Number theory.
- Zermelo Fraenkel set theory.
- Large cardinal assumptions.

Def: For theories T and T', let $T \ll T'$ if T' can prove the consistency of T.

Recall Gödel's theorem that $T \not\ll T$.

Example — Natural objects in proof theory

Empirical fact: *Natural* theories are well-ordered by consistency strength.

A theory is a set of axioms, like

- Peano Arithmetic.
- Second order Number theory.
- Zermelo Fraenkel set theory.
- Large cardinal assumptions.

Def: For theories T and T', let $T \ll T'$ if T' can prove the consistency of T.

Recall Gödel's theorem that $T \not\ll T$.

One can build theories T and T' incomparable under $\ll$.

Example — Natural objects in proof theory

Empirical fact: *Natural* theories are well-ordered by consistency strength.

A theory is a set of axioms, like

- Peano Arithmetic.
- Second order Number theory.
- Zermelo Fraenkel set theory.
- Large cardinal assumptions.

Def: For theories T and T', let $T \ll T'$ if T' can prove the consistency of T.

Recall Gödel's theorem that $T \not\ll T$.

One can build theories T and T' incomparable under $\ll$.

We don't really understand why the empirical fact holds.

Example — Natural objects in proof theory

Empirical fact: *Natural* theories are well-ordered by consistency strength.

A theory is a set of axioms, like

- Peano Arithmetic.
- Second order Number theory.
- Zermelo Fraenkel set theory.
- Large cardinal assumptions.

Def: For theories T and T', let $T \ll T'$ if T' can prove the consistency of T.

Recall Gödel's theorem that $T \not\ll T$.

One can build theories T and T' incomparable under $\ll$.

We don't really understand why the empirical fact holds.

In this talk we will study a similar phenomenon in Computability Theory.

Table of contents

Objective: Classify the natural many-one degrees.

Table of contents

Objective: Classify the natural many-one degrees.

Computability Theory

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

Computability Theory

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

A more formal definition:

The class of *partial computable functions* $\mathbb{N}^n \rightharpoonup \mathbb{N}$ is the

- closure of the projection and successor functions,
- under composition, recursion, and minimalization.

Examples of non-computable sets

Examples of non-computable sets

The word problem:

Examples of non-computable sets

The word problem: Consider the groups that can be constructed with a finite set of generators and a finite set of relations between the generators.

Examples of non-computable sets

The word problem: Consider the groups that can be constructed with a finite set of generators and a finite set of relations between the generators. The set of pairs (set-of-generators, relations) that generate the **trivial** group is **not** computable.

Examples of non-computable sets

The word problem: Consider the groups that can be constructed with a finite set of generators and a finite set of relations between the generators. The set of pairs (set-of-generators, relations) that generate the **trivial** group is **not** computable.

Simply connected manifolds: The set of finite triangulations of **simply connected** manifolds is **not** computable.

Examples of non-computable sets

The word problem: Consider the groups that can be constructed with a finite set of generators and a finite set of relations between the generators. The set of pairs (set-of-generators, relations) that generate the **trivial** group is **not** computable.

Simply connected manifolds: The set of finite triangulations of **simply connected** manifolds is **not** computable.

Hilbert's 10th problem: The set of polynomials in $\mathbb{Z}[x_1, x_2, ...]$ that have integer roots is **not** computable.

Examples of non-computable sets

The word problem: Consider the groups that can be constructed with a finite set of generators and a finite set of relations between the generators. The set of pairs (set-of-generators, relations) that generate the **trivial** group is **not** computable.

Simply connected manifolds: The set of finite triangulations of **simply connected** manifolds is **not** computable.

Hilbert's 10th problem: The set of polynomials in $\mathbb{Z}[x_1, x_2, ...]$ that have integer roots is **not** computable.

The Halting problem: The set of programs that **halt**, and don't run for ever, is **not** computable.

Proving non-computability

Proving non-computability

Enumerate the computer programs alphabetically as $\Phi_0, \Phi_1, \Phi_2, \ldots\ldots$

Def: *The halting problem* is the set $K = \{\langle e, n\rangle \in \mathbb{N}^2 : \Phi_e(n) \text{ halts }\}$.

Theorem: K is not computable:

Proving non-computability

Enumerate the computer programs alphabetically as $\Phi_0, \Phi_1, \Phi_2, \ldots\ldots$

Def: *The halting problem* is the set $K = \{\langle e, n\rangle \in \mathbb{N}^2 : \Phi_e(n) \text{ halts }\}$.

Theorem: K is not computable:
Proof: *Diagonalization*:

Proving non-computability

Enumerate the computer programs alphabetically as $\Phi_0, \Phi_1, \Phi_2, \ldots\ldots$

Def: *The halting problem* is the set $K = \{\langle e, n\rangle \in \mathbb{N}^2 : \Phi_e(n) \text{ halts } \}$.

Theorem: K is not computable:
Proof: *Diagonalization*:
If K were computable, write the program: $P(n)$ halts $\iff$ the $\langle n, n\rangle \notin K$.

Proving non-computability

Enumerate the computer programs alphabetically as $\Phi_0, \Phi_1, \Phi_2, \dots\dots$

Def: *The halting problem* is the set $K = \{\langle e, n\rangle \in \mathbb{N}^2 : \Phi_e(n) \text{ halts }\}$.

Theorem: K is not computable:
Proof: *Diagonalization*:
If K were computable, write the program: $P(n)$ halts $\iff$ the $\langle n, n\rangle \notin K$.
That program must be Φ_e for some e.

Proving non-computability

Enumerate the computer programs alphabetically as $\Phi_0, \Phi_1, \Phi_2, \ldots\ldots$

Def: *The halting problem* is the set $K = \{\langle e, n\rangle \in \mathbb{N}^2 : \Phi_e(n) \text{ halts }\}$.

Theorem: K is not computable:
Proof: *Diagonalization*:
If K were computable, write the program: $P(n)$ halts $\iff$ the $\langle n, n\rangle \notin K$.
That program must be Φ_e for some e.
Then $\langle e, e\rangle \notin K \iff P(e)$ halts

Proving non-computability

Enumerate the computer programs alphabetically as $\Phi_0, \Phi_1, \Phi_2, \ldots\ldots$

Def: *The halting problem* is the set $K = \{\langle e, n\rangle \in \mathbb{N}^2 : \Phi_e(n) \text{ halts }\}$.

Theorem: K is not computable:
Proof: *Diagonalization*:
If K were computable, write the program: $P(n)$ halts $\iff$ the $\langle n, n\rangle \notin K$.
That program must be Φ_e for some e.
Then $\langle e, e\rangle \notin K \iff P(e)$ halts $\iff \Phi_e(e)$ halts

Proving non-computability

Enumerate the computer programs alphabetically as $\Phi_0, \Phi_1, \Phi_2, \ldots\ldots$

Def: *The halting problem* is the set $K = \{\langle e, n\rangle \in \mathbb{N}^2 : \Phi_e(n) \text{ halts }\}$.

Theorem: K is not computable:
Proof: *Diagonalization*:
If K were computable, write the program: $P(n)$ halts $\iff$ the $\langle n, n\rangle \notin K$.
That program must be Φ_e for some e.
Then $\langle e, e\rangle \notin K \iff P(e)$ halts $\iff \Phi_e(e)$ halts $\iff \langle e, e\rangle \in K$.

Proving non-computability

Enumerate the computer programs alphabetically as $\Phi_0, \Phi_1, \Phi_2, \ldots\ldots$

Def: *The halting problem* is the set $K = \{\langle e, n\rangle \in \mathbb{N}^2 : \Phi_e(n) \text{ halts } \}$.

Theorem: K is not computable:
Proof: *Diagonalization*:
If K were computable, write the program: $P(n)$ halts $\iff$ the $\langle n, n\rangle \notin K$.
That program must be Φ_e for some e.
Then $\langle e, e\rangle \notin K \iff P(e)$ halts $\iff \Phi_e(e)$ halts $\iff \langle e, e\rangle \in K$.

Theorem: The word problem is not computable:

Proving non-computability

Enumerate the computer programs alphabetically as $\Phi_0, \Phi_1, \Phi_2, \ldots\ldots$

Def: *The halting problem* is the set $K = \{\langle e, n\rangle \in \mathbb{N}^2 : \Phi_e(n) \text{ halts }\}$.

Theorem: K is not computable:
Proof: *Diagonalization*:
If K were computable, write the program: $P(n)$ halts $\iff$ the $\langle n, n\rangle \notin K$.
That program must be Φ_e for some e.
Then $\langle e, e\rangle \notin K \iff P(e)$ halts $\iff \Phi_e(e)$ halts $\iff \langle e, e\rangle \in K$.

Theorem: The word problem is not computable:
Proof: *Reduce the Halting problem to the word problem*:

Proving non-computability

Enumerate the computer programs alphabetically as $\Phi_0, \Phi_1, \Phi_2, \ldots\ldots$

Def: *The halting problem* is the set $K = \{\langle e, n\rangle \in \mathbb{N}^2 : \Phi_e(n) \text{ halts }\}$.

Theorem: K is not computable:
Proof: *Diagonalization*:
If K were computable, write the program: $P(n)$ halts $\iff$ the $\langle n, n\rangle \notin K$.
That program must be Φ_e for some e.
Then $\langle e, e\rangle \notin K \iff P(e)$ halts $\iff \Phi_e(e)$ halts $\iff \langle e, e\rangle \in K$.

Theorem: The word problem is not computable:
Proof: *Reduce the Halting problem to the word problem*:
Define a computable function $f : \mathbb{N}^2 \to \{$ set of finite group presentations $\}$
such that $(e, n) \in K \iff$ the group with presentation $f(e, n)$ is trivial.

Proving non-computability

Enumerate the computer programs alphabetically as $\Phi_0, \Phi_1, \Phi_2, \ldots\ldots$

Def: *The halting problem* is the set $K = \{\langle e, n\rangle \in \mathbb{N}^2 : \Phi_e(n) \text{ halts } \}$.

Theorem: K is not computable:
Proof: *Diagonalization*:
If K were computable, write the program: $P(n)$ halts $\iff$ the $\langle n, n\rangle \notin K$.
That program must be Φ_e for some e.
Then $\langle e, e\rangle \notin K \iff P(e)$ halts $\iff \Phi_e(e)$ halts $\iff \langle e, e\rangle \in K$.

Theorem: The word problem is not computable:
Proof: *Reduce the Halting problem to the word problem*:
Define a computable function $f : \mathbb{N}^2 \to \{$ set of finite group presentations $\}$
such that $(e, n) \in K \iff$ the group with presentation $f(e, n)$ is trivial.
Then, if the word problem were computable, so would be K.

Many-one reducibility

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 $n \in A \iff f(n) \in B \quad (\forall n)$.

Many-one reducibility

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 and B are *many-one equivalent* ($A \equiv_m B$), if $\mathcal{A} \leq_m B$ and $B \leq_m A$.

Many-one reducibility

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 and B are *many-one equivalent* ($A \equiv_m B$), if $\mathcal{A} \leq_m B$ and $B \leq_m A$.
An *m-degree* is a $\equiv_m$-equivalence class.

Many-one reducibility

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 $f : \mathbb{N} \to \mathbb{N}$ such that $n \in A \iff f(n) \in B \quad (\forall n)$.

Definition:
A and B are *many-one equivalent* ($A \equiv_m B$), if $\mathcal{A} \leq_m B$ and $B \leq_m A$.
An *m-degree* is a $\equiv_m$-equivalence class.

Lemma:

1. $\emptyset \leq_m B$ for every $B \subseteq \mathbb{N}$, unless $B = \mathbb{N}$.
2. $\mathbb{N} \leq_m B$ for every $B \subseteq \mathbb{N}$, unless $B = \emptyset$.
3. If A is computable, then $A \leq_m B$ for every $B \subseteq \mathbb{N}$ unless $B = \emptyset, \mathbb{N}$.
4. If B is computable and $A \leq_m B$, then A is computable too.
5. Given B, the set $\{A \subseteq \mathbb{N} : A \leq_m B\}$ is countable.

Many-one reducibility — Natural Examples

The following are $\equiv_m$-equivalent:

K, the Halting problem.

$\equiv_m$ The Word problem.

$\equiv_m$ The simply-connected problem.

$\equiv_m$ Hilbert's 10th problem.

Many-one reducibility — Natural Examples

The following are $\equiv_m$-equivalent:

K, the Halting problem.

$\equiv_m$ The Word problem.

$\equiv_m$ The simply-connected problem.

$\equiv_m$ Hilbert's 10th problem.

The following are in strictly increasing $\leq_m$-order:

K, the Halting problem

Many-one reducibility — Natural Examples

The following are $\equiv_m$-equivalent:

K, the Halting problem.

$\equiv_m$ The Word problem.

$\equiv_m$ The simply-connected problem.

$\equiv_m$ Hilbert's 10th problem.

The following are in strictly increasing $\leq_m$-order:

K, the Halting problem

$<_m$ $\{\ p(x, \bar{y}) \in \mathbb{Z}[x, y_1, y_2, ...]$ with integers solutions for exactly one $x\}$

Many-one reducibility — Natural Examples

The following are $\equiv_m$-equivalent:

K, the Halting problem.

$\equiv_m$ The Word problem.

$\equiv_m$ The simply-connected problem.

$\equiv_m$ Hilbert's 10th problem.

The following are in strictly increasing $\leq_m$-order:

K, the Halting problem

$<_m$ $\{\ p(x, \bar{y}) \in \mathbb{Z}[x, y_1, y_2, ...]$ with integers solutions for exactly one $x\}$

$<_m$ { finite presentations of groups that are torsion-free }

Many-one reducibility — Natural Examples

The following are $\equiv_m$-equivalent:

K, the Halting problem.
$\equiv_m$ The Word problem.
$\equiv_m$ The simply-connected problem.
$\equiv_m$ Hilbert's 10th problem.

The following are in strictly increasing $\leq_m$-order:

K, the Halting problem
$<_m$ $\{ \ p(x, \bar{y}) \in \mathbb{Z}[x, y_1, y_2, ...]$ with integers solutions for exactly one $x\}$
$<_m$ $\{$ finite presentations of groups that are torsion-free $\}$
$<_m$ $\{$ true 1st-order sentences about arithmetic $\}$*

Many-one reducibility — Natural Examples

The following are $\equiv_m$-equivalent:

K, the Halting problem.
$\equiv_m$ The Word problem.
$\equiv_m$ The simply-connected problem.
$\equiv_m$ Hilbert's 10th problem.

The following are in strictly increasing $\leq_m$-order:

K, the Halting problem
$<_m$ $\{\ p(x, \bar{y}) \in \mathbb{Z}[x, y_1, y_2, ...]$ with integers solutions for exactly one $x\}$
$<_m$ $\{$ finite presentations of groups that are torsion-free $\}$
$<_m$ $\{$ true 1st-order sentences about arithmetic $\}$*
$<_m$ $\{$ programs computing linear orderings that contain a copy of $\mathbb{Q}\}$

Many-one reducibility — Natural Examples

The following are $\equiv_m$-equivalent:

K, the Halting problem.
$\equiv_m$ The Word problem.
$\equiv_m$ The simply-connected problem.
$\equiv_m$ Hilbert's 10th problem.

The following are in strictly increasing $\leq_m$-order:

K, the Halting problem
$<_m$ $\{ \ p(x, \bar{y}) \in \mathbb{Z}[x, y_1, y_2, ...]$ with integers solutions for exactly one $x\}$
$<_m$ { finite presentations of groups that are torsion-free }
$<_m$ { true 1st-order sentences about arithmetic }*
$<_m$ $\{$ programs computing linear orderings that contain a copy of $\mathbb{Q}\}$
$<_m$ { true 2nd-order sentences about arithmetic }*

Many-one reducibility — Natural Examples

The following are $\equiv_m$-equivalent:

K, the Halting problem.
$\equiv_m$ The Word problem.
$\equiv_m$ The simply-connected problem.
$\equiv_m$ Hilbert's 10th problem.

The following are in strictly increasing $\leq_m$-order:

K, the Halting problem
$<_m$ { $p(x, \bar{y}) \in \mathbb{Z}[x, y_1, y_2, ...]$ with integers solutions for exactly one x}
$<_m$ { finite presentations of groups that are torsion-free }
$<_m$ { true 1st-order sentences about arithmetic }*
$<_m$ { programs computing linear orderings that contain a copy of $\mathbb{Q}$}
$<_m$ { true 2nd-order sentences about arithmetic }*

All these sets are **not** $\equiv_m$-equivalent to their complements, except *.

Computably enumerable sets

Definition: A set A is *computably enumerable (c.e.)* if
$A = \{f(0), f(1), f(2), ...\}$ for some function $f : \mathbb{N} \to \mathbb{N}$.

Computably enumerable sets

Definition: A set A is *computably enumerable (c.e.)* if
$$A = \{f(0), f(1), f(2), ...\} \text{ for some function } f : \mathbb{N} \to \mathbb{N}.$$

Equivalently: A is c.e. $\iff$
A is of the form $\{x : \exists y\ \langle x, y\rangle \in P\}$ where P is a computable.

Computably enumerable sets

Definition: A set A is *computably enumerable (c.e.)* if
$$A = \{f(0), f(1), f(2), ...\} \text{ for some function } f : \mathbb{N} \to \mathbb{N}.$$

Equivalently: A is c.e. $\iff$
A is of the form $\{x : \exists y \ \langle x, y \rangle \in P\}$ where P is a computable.

The following are c.e.:

- The Halting problem.
- The Word problem.
- The simply-connected problem.
- Hilbert's 10th problem.

Computably enumerable sets

Definition: A set A is *computably enumerable (c.e.)* if
$$A = \{f(0), f(1), f(2), ...\} \text{ for some function } f : \mathbb{N} \to \mathbb{N}.$$

Equivalently: A is c.e. $\iff$
A is of the form $\{x : \exists y \ \langle x, y\rangle \in P\}$ where P is a computable.

Computably enumerable sets

Definition: A set A is *computably enumerable (c.e.)* if
$A = \{f(0), f(1), f(2), ...\}$ for some function $f : \mathbb{N} \to \mathbb{N}$.

Equivalently: A is c.e. $\iff$
A is of the form $\{x : \exists y \ \langle x, y \rangle \in P\}$ where P is a computable.

Observation: If A is c.e. and $B \leq_m A$, then B is c.e. too.

Computably enumerable sets

Definition: A set A is *computably enumerable (c.e.)* if
$A = \{f(0), f(1), f(2), ...\}$ for some function $f : \mathbb{N} \to \mathbb{N}$.

Equivalently: A is c.e. $\iff$
A is of the form $\{x : \exists y \; \langle x, y \rangle \in P\}$ where P is a computable.

Observation: If A is c.e. and $B \leq_m A$, then B is c.e. too.

Definition: A set A is *c.e.-complete* if it is c.e.
and for every c.e. set B, $B \leq_m A$.

Computably enumerable sets

Definition: A set A is *computably enumerable (c.e.)* if
$A = \{f(0), f(1), f(2), ...\}$ for some function $f : \mathbb{N} \to \mathbb{N}$.

Equivalently: A is c.e. $\iff$
A is of the form $\{x : \exists y \ \langle x, y\rangle \in P\}$ where P is a computable.

Observation: If A is c.e. and $B \leq_m A$, then B is c.e. too.

Definition: A set A is *c.e.-complete* if it is c.e.
and for every c.e. set B, $B \leq_m A$.

The examples before were all c.e.-complete

A side note — NP complete sets

NP-complete sets are the analogous of c.e.-complete sets,
but for computable functions that run in polynomial time.

A side note — NP complete sets

NP-complete sets are the analogous of c.e.-complete sets,
but for computable functions that run in polynomial time.

Definition: A set is A is *NP* if it is of the form

$$\{x \in 2^* : (\exists y \in 2^*) \ |y| < |x|^n \ \& \ \langle x, y\rangle \in R\}$$

where $n \in \mathbb{N}$ and $R \subseteq 2^* \times 2^*$ is a computable in polynomial time.

A side note — NP complete sets

NP-complete sets are the analogous of c.e.-complete sets,
but for computable functions that run in polynomial time.

Definition: A set is A is *NP* if it is of the form

$$\{x \in 2^* : (\exists y \in 2^*)\ |y| < |x|^n \ \&\ \langle x, y\rangle \in R\}$$

where $n \in \mathbb{N}$ and $R \subseteq 2^* \times 2^*$ is a computable in polynomial time.

Examples: The following are c.e.:

- Satisfiability for propositional formulas.
- Hamiltonian path problem.
- Traveler salesman problem.
- Graph coloring problem.

A side note — NP complete sets

NP-complete sets are the analogous of c.e.-complete sets,
but for computable functions that run in polynomial time.

Definition: A set is A is *NP* if it is of the form

$$\{x \in 2^* : (\exists y \in 2^*)\ |y| < |x|^n \ \&\ \langle x, y\rangle \in R\}$$

where $n \in \mathbb{N}$ and $R \subseteq 2^* \times 2^*$ is a computable in polynomial time.

Definition

A set A is *polynomial-time reducible* to B ($A \leq_m^P B$) if there is poly-time computable $f \colon 2^* \to 2^*$ such that $\sigma \in A \iff f(\sigma) \in B$ $(\forall \sigma \in 2^*)$

A side note — NP complete sets

NP-complete sets are the analogous of c.e.-complete sets,
but for computable functions that run in polynomial time.

Definition: A set is A is *NP* if it is of the form

$$\{x \in 2^* : (\exists y \in 2^*)\ |y| < |x|^n \ \&\ \langle x, y\rangle \in R\}$$

where $n \in \mathbb{N}$ and $R \subseteq 2^* \times 2^*$ is a computable in polynomial time.

Definition

A set A is *polynomial-time reducible* to B ($A \leq_m^P B$) if there is poly-time computable $f : 2^* \to 2^*$ such that $\sigma \in A \iff f(\sigma) \in B$ $(\forall \sigma \in 2^*)$

Definition: A set A is *NP-complete* if it is NP and
for every NP set B, $B \leq_m^P A$.

A side note — NP complete sets

NP-complete sets are the analogous of c.e.-complete sets,
but for computable functions that run in polynomial time.

Definition: A set is A is *NP* if it is of the form
$$\{x \in 2^* : (\exists y \in 2^*)\ |y| < |x|^n \ \&\ \langle x, y\rangle \in R\}$$
where $n \in \mathbb{N}$ and $R \subseteq 2^* \times 2^*$ is a computable in polynomial time.

Definition

A set A is *polynomial-time reducible* to B ($A \leq_m^P B$) if there is poly-time computable $f\colon 2^* \to 2^*$ such that $\sigma \in A \iff f(\sigma) \in B$ $(\forall \sigma \in 2^*)$

Definition: A set A is *NP-complete* if it is NP and
for every NP set B, $B \leq_m^P A$.

The examples above are NP-complete

Back to many-one degrees — d-c.e. sets

Definition: A set A is *d-c.e.* if $A = B \setminus C$ where B and C are c.e.

Back to many-one degrees — d-c.e. sets

Definition: A set A is *d-c.e.* if $A = B \setminus C$ where B and C are c.e.

Example: The following is d-c.e.:

- $\{\ p(x, \bar{y}) \in \mathbb{Z}[x, y_1, y_2, \ldots]$ with integers solutions for exactly one $x\}$.

Back to many-one degrees — d-c.e. sets

Definition: A set A is *d-c.e.* if $A = B \setminus C$ where B and C are c.e.

Observation: If A is d-c.e. and $B \leq_m A$, then B is d-c.e. too.

Back to many-one degrees — d-c.e. sets

Definition: A set A is *d-c.e.* if $A = B \setminus C$ where B and C are c.e.

Observation: If A is d-c.e. and $B \leq_m A$, then B is d-c.e. too.

Observation: There is a d-c.e. set that is not c.e.

Back to many-one degrees — d-c.e. sets

Definition: A set A is *d-c.e.* if $A = B \setminus C$ where B and C are c.e.

Observation: If A is d-c.e. and $B \leq_m A$, then B is d-c.e. too.

Observation: There is a d-c.e. set that is not c.e.

Definition: A set A is *d-c.e.-complete* if it is d-c.e.
and for every d-c.e. set B, $B \leq_m A$.

Back to many-one degrees — d-c.e. sets

Definition: A set A is *d-c.e.* if $A = B \setminus C$ where B and C are c.e.

Observation: If A is d-c.e. and $B \leq_m A$, then B is d-c.e. too.

Observation: There is a d-c.e. set that is not c.e.

Definition: A set A is *d-c.e.-complete* if it is d-c.e.
and for every d-c.e. set B, $B \leq_m A$.

The example above is d-c.e.-complete

Back to many-one degrees — d-c.e. sets

Definition: A set A is *d-c.e.* if $A = B \setminus C$ where B and C are c.e.

Observation: If A is d-c.e. and $B \leq_m A$, then B is d-c.e. too.

Observation: There is a d-c.e. set that is not c.e.

Definition: A set A is *d-c.e.-complete* if it is d-c.e. and for every d-c.e. set B, $B \leq_m A$.

The example above is d-c.e.-complete

We can continue on and define n-c.e. for $n \in \mathbb{N}$.

Π^0_2

Definition: A set A is Π^0_2 if it is of the form $\{z \in \mathbb{N} : (\forall x)(\exists y) \ \langle x, y, z\rangle \in R\}$ where $R \subseteq \mathbb{N}^3$ is computable.

Π^0_2

Definition: A set A is Π^0_2 if it is of the form
$\{z \in \mathbb{N} : (\forall x)(\exists y)\ \ \langle x, y, z\rangle \in R\}$ where $R \subseteq \mathbb{N}^3$ is computable.

Example:
The set of finite presentations (*generators*, *relations*) for torsion groups:
For every word w, and number n, if $w^n \sim e$
there exists a sequence of applications to the relations, such that $w \sim e$.

Π^0_2

Definition: A set A is Π^0_2 if it is of the form
$\{z \in \mathbb{N} : (\forall x)(\exists y)\ \langle x, y, z\rangle \in R\}$ where $R \subseteq \mathbb{N}^3$ is computable.

Example:
The set of finite presentations (*generators*, *relations*) for torsion groups:
For every word w, and number n, if $w^n \sim e$
there exists a sequence of applications to the relations, such that $w \sim e$.

Definition: A set A is Π^0_2-*complete* if it is Π^0_2
and for every Π^0_2 set B, $B \leq_m A$.

Π^0_2

Definition: A set A is Π^0_2 if it is of the form
$\{z \in \mathbb{N} : (\forall x)(\exists y)\ \ \langle x, y, z\rangle \in R\}$ where $R \subseteq \mathbb{N}^3$ is computable.

Example:
The set of finite presentations (*generators*, *relations*) for torsion groups:
For every word w, and number n, if $w^n \sim e$
there exists a sequence of applications to the relations, such that $w \sim e$.

Definition: A set A is Π^0_2-*complete* if it is Π^0_2
and for every Π^0_2 set B, $B \leq_m A$.

The example above is Π^0_2-complete

Π^0_2

Definition: A set A is Π^0_2 if it is of the form $\{z \in \mathbb{N} : (\forall x)(\exists y)\ \langle x, y, z\rangle \in R\}$ where $R \subseteq \mathbb{N}^3$ is computable.

Example:
The set of finite presentations (*generators*, *relations*) for torsion groups:
For every word w, and number n, if $w^n \sim e$
there exists a sequence of applications to the relations, such that $w \sim e$.

Definition: A set A is Π^0_2-*complete* if it is Π^0_2 and for every Π^0_2 set B, $B \leq_m A$.

The example above is Π^0_2-complete

We can continue on and define Π^0_n for $n \in \mathbb{N}$.

Chaos

So far, the examples are linearly ordered,

Chaos

So far, the examples are linearly ordered, except for complements. BB

Chaos

So far, the examples are linearly ordered, except for complements. BB

We know **no** natural example strictly $\leq_m$-between $\{0\}$ and K.

Chaos

So far, the examples are linearly ordered, except for complements. BB

We know **no** natural example strictly $\leq_m$-between $\{0\}$ and K.

HOWEVER

Chaos

So far, the examples are linearly ordered, except for complements. BB

We know **no** natural example strictly $\leq_m$-between $\{0\}$ and K.

HOWEVER

Theorem: [Kleene, Post] There are continuum size $\leq_m$-antichains of m-degrees.

Chaos

So far, the examples are linearly ordered, except for complements. BB

We know **no** natural example strictly $\leq_m$-between $\{0\}$ and K.

HOWEVER

Theorem: [Kleene, Post] There are continuum size $\leq_m$-antichains of m-degrees.

Theorem: [Kleene, Post][Lachlan–Shore, Nerode] Every countable partial ordering embeds into the many-one degrees.

Chaos

So far, the examples are linearly ordered, except for complements. BB

We know **no** natural example strictly $\leq_m$-between $\{0\}$ and K.

HOWEVER

Theorem: [Kleene, Post] There are continuum size $\leq_m$-antichains of m-degrees.

Theorem: [Kleene, Post][Lachlan–Shore, Nerode] Every countable partial ordering embeds into the many-one degrees.

Furthermore, it can even be embedded below the degree of K,

Chaos

So far, the examples are linearly ordered, except for complements. BB

We know **no** natural example strictly $\leq_m$-between $\{0\}$ and K.

HOWEVER

Theorem: [Kleene, Post] There are continuum size $\leq_m$-antichains of m-degrees.

Theorem: [Kleene, Post][Lachlan–Shore, Nerode] Every countable partial ordering embeds into the many-one degrees.

Furthermore, it can even be embedded below the degree of K,

or between $\leq_m$-between $K \oplus \bar{K}$ and $K \times \bar{K}$,

Chaos

So far, the examples are linearly ordered, except for complements. BB

We know **no** natural example strictly $\leq_m$-between $\{0\}$ and K.

HOWEVER

Theorem: [Kleene, Post] There are continuum size $\leq_m$-antichains of m-degrees.

Theorem: [Kleene, Post][Lachlan–Shore, Nerode] Every countable partial ordering embeds into the many-one degrees.

Furthermore, it can even be embedded below the degree of K,

or between $\leq_m$-between $K \oplus \bar{K}$ and $K \times \bar{K}$, or as an initial segment.

Chaos

So far, the examples are linearly ordered, except for complements. BB

We know **no** natural example strictly $\leq_m$-between $\{0\}$ and K.

HOWEVER

Theorem: [Kleene, Post] There are continuum size $\leq_m$-antichains of m-degrees.

Theorem: [Kleene, Post][Lachlan–Shore, Nerode] Every countable partial ordering embeds into the many-one degrees.
Furthermore, it can even be embedded below the degree of K,
or between $\leq_m$-between $K \oplus \bar{K}$ and $K \times \bar{K}$, or as an initial segment.

Theorem: [Shore, Nerode] The 1st-order theory of the poset of the m-degrees
is 1-1 equivalent to
The 2nd-order theory of $(\mathbb{N}; 0, 1, +, \times)$.

Natural vs arbitrary m-degrees

On one side:

The natural examples of m-degrees are *frew* and *nicely ordered*.

Natural vs arbitrary m-degrees

On one side:

The natural examples of m-degrees are *frew* and *nicely ordered*.

On the other side:

The structure of all the m-degrees is very *complex* and *badly behaved*.

Natural vs arbitrary m-degrees

On one side:

The natural examples of m-degrees are *frew* and *nicely ordered*.

On the other side:
The structure of all the m-degrees is very *complex* and *badly behaved*.

Can we explain this?

Natural vs arbitrary m-degrees

On one side:

The natural examples of m-degrees are *frew* and *nicely ordered*.

On the other side:
The structure of all the m-degrees is very *complex* and *badly behaved*.

Can we explain this?

Can we characterize the many-one degrees that have names?

Wadge degrees

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

Wadge degrees

Consider the *Baire Space*: $\mathbb{N}^{\mathbb{N}} = \{f : \mathbb{N} \to \mathbb{N}\}$ 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*: $\mathbb{N}^\mathbb{N} = \{f : \mathbb{N} \to \mathbb{N}\}$ 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 $f : \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ s.t. $(\forall X \in 2^\mathbb{N})$, $X \in A \iff f(X) \in B$.

Wadge degrees

Consider the *Baire Space*: $\mathbb{N}^{\mathbb{N}} = \{f : \mathbb{N} \to \mathbb{N}\}$ 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 $f : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ 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, Monk] The Wadge degrees are well founded.

The answer — informally

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. $(\forall X \in 2^{\mathbb{N}})$, $X \in A \iff f(X) \in B$.

The answer — informally

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. $(\forall X \in 2^{\mathbb{N}})$, $X \in A \iff f(X) \in B$.

All Wadge degrees have *names*. BB

The answer — informally

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. $(\forall X \in 2^\mathbb{N})$, $X \in A \iff f(X) \in B$.

All Wadge degrees have *names*. BB

[Kihra, Montalbán] There is a one-to-one correspondence between the natural m-degrees and the Wadge degrees. .

The answer — informally

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. $(\forall X \in 2^{\mathbb{N}})$, $X \in A \iff f(X) \in B$.

All Wadge degrees have *names*. BB

[Kihra, Montalbán] There is a one-to-one correspondence between
the natural m-degrees and the Wadge degrees. .

Relativization

Definition: Let $X \in 2^{\mathbb{N}}$. A function $f : \mathbb{N} \to \mathbb{N}$ is *X-computable* if there exists a computer program that calculates f using the characteristic function of X as a primitive.

Relativization

Definition: Let $X \in 2^{\mathbb{N}}$. A function $f : \mathbb{N} \to \mathbb{N}$ is *X-computable* if there exists a computer program that calculates f using the characteristic function of X as a primitive.
We write $f \leq_T X$.

Relativization

Definition: Let $X \in 2^{\mathbb{N}}$. A function $f : \mathbb{N} \to \mathbb{N}$ is *X-computable* if there exists a computer program that calculates f using the characteristic function of X as a primitive.
We write $f \leq_T X$.

A more formal definition:

The class of *partial X-computable functions* $\mathbb{N}^n \rightharpoonup \mathbb{N}$ is the

- closure of the projection, successor functions, and χ_X,
- under composition, recursion, and minimalization.

Relativization

Definition: Let $X \in 2^{\mathbb{N}}$. A function $f : \mathbb{N} \to \mathbb{N}$ is *X-computable* if there exists a computer program that calculates f using the characteristic function of X as a primitive.
We write $f \leq_T X$.

A more formal definition:

The class of *partial X-computable functions* $\mathbb{N}^n \rightharpoonup \mathbb{N}$ is the

- closure of the projection, successor functions, and χ_X,
- under composition, recursion, and minimalization.

Def: Given a notion P, we use P^X to denote *P relative to X*, obtained by replacing computable by X-computable within P.

Examples of relativization

Def: Given a notion P, we use P^X to denote *P relative to X*,
obtained by replacing computable by X-computable within P.

Examples of relativization

Def: Given a notion P, we use P^X to denote *P relative to X*,
obtained by replacing computable by X-computable within P.

- $\Phi_0^X, \Phi_1^X, \ldots$ are the X-computable programs.

Examples of relativization

Def: Given a notion P, we use P^X to denote *P relative to X*,
obtained by replacing computable by X-computable within P.

- $\Phi_0^X, \Phi_1^X, \ldots$ are the X-computable programs.
- $K^X = \{\langle e, n\rangle : \Phi_e(n) \text{ halts } \}$.

Examples of relativization

Def: Given a notion P, we use P^X to denote *P relative to X*,
obtained by replacing computable by X-computable within P.

- $\Phi_0^X, \Phi_1^X, \ldots$ are the X-computable programs.
- $K^X = \{\langle e, n\rangle : \Phi_e(n) \text{ halts }\}$.
- K^X is not X-computable.

Examples of relativization

Def: Given a notion P, we use P^X to denote *P relative to X*, obtained by replacing computable by X-computable within P.

- $\Phi_0^X, \Phi_1^X, \ldots$ are the X-computable programs.
- $K^X = \{\langle e, n\rangle : \Phi_e(n) \text{ halts }\}$.
- K^X is not X-computable.
- K^X is X-c.e.

Examples of relativization

Def: Given a notion P, we use P^X to denote *P relative to X*,
obtained by replacing computable by X-computable within P.

- $\Phi_0^X, \Phi_1^X, \ldots$ are the X-computable programs.
- $K^X = \{\langle e, n\rangle : \Phi_e(n) \text{ halts }\}$.
- K^X is not X-computable.
- K^X is X-c.e.-complete.

Examples of relativization

Def: Given a notion P, we use P^X to denote *P relative to X*,
obtained by replacing computable by X-computable within P.

- $\Phi_0^X, \Phi_1^X, \ldots$ are the X-computable programs.
- $K^X = \{\langle e, n\rangle : \Phi_e(n) \text{ halts } \}$.
- K^X is not X-computable.
- K^X is X-c.e.-complete.
- $K^X \times \bar{K}^X$ is X-d.c.e.-complete.

Empirical Fact: If P is a natural property, then:

$$P \iff (\forall X)\ P^X$$

Examples of relativization

Def: Given a notion P, we use P^X to denote *P relative to X*, obtained by replacing computable by X-computable within P.

- $\Phi_0^X, \Phi_1^X, \ldots$ are the X-computable programs.
- $K^X = \{\langle e, n \rangle : \Phi_e(n) \text{ halts } \}$.
- K^X is not X-computable.
- K^X is X-c.e.-complete.
- $K^X \times \bar{K}^X$ is X-d.c.e.-complete.

Empirical Fact: If P is a natural property, then:

$$P \iff (\forall X)\, P^X$$

Def: A *cone* is a set of the form $\{X \in 2^{\mathbb{N}} : X \geq_T Y\}$ for some $Y \in 2^{\mathbb{N}}$.

Examples of relativization

Def: Given a notion P, we use P^X to denote *P relative to X*,
obtained by replacing computable by X-computable within P.

- $\Phi_0^X, \Phi_1^X, \ldots$ are the X-computable programs.
- $K^X = \{\langle e, n\rangle : \Phi_e(n) \text{ halts }\}$.
- K^X is not X-computable.
- K^X is X-c.e.-complete.
- $K^X \times \bar{K}^X$ is X-d.c.e.-complete.

Empirical Fact: If P is a natural property, then:

$$P \iff (\forall X)\ P^X$$

Def: A *cone* is a set of the form $\{X \in 2^{\mathbb{N}} : X \geq_T Y\}$ for some $Y \in 2^{\mathbb{N}}$.
By *P on a cone* we mean $\{X : P^X \text{ holds}\}$ contains a cone.

Examples of relativization

Def: Given a notion P, we use P^X to denote *P relative to X*,
obtained by replacing computable by X-computable within P.

- $\Phi_0^X, \Phi_1^X, \ldots$ are the X-computable programs.
- $K^X = \{\langle e, n\rangle : \Phi_e(n) \text{ halts }\}$.
- K^X is not X-computable.
- K^X is X-c.e.-complete.
- $K^X \times \bar{K}^X$ is X-d.c.e.-complete.

Empirical Fact: If P is a natural property, then:

$$P \iff (\forall X)\ P^X \iff P \text{ on a cone.}$$

Def: A *cone* is a set of the form $\{X \in 2^{\mathbb{N}} : X \geq_T Y\}$ for some $Y \in 2^{\mathbb{N}}$.
By *P on a cone* we mean $\{X : P^X \text{ holds}\}$ contains a cone.

Back to degrees with names

Suppose **s** is a m-degree with a name.

Back to degrees with names

Suppose **s** is a m-degree with a name.

If P is a natural property, one would expect that

$$\mathbf{s} \text{ satisfies } P \iff \mathbf{s}^X \text{ satisfies } P^X \ (\forall X \in 2^{\mathbb{N}}).$$

Back to degrees with names

Suppose **s** is a m-degree with a name.

If P is a natural property, one would expect that

$$\mathbf{s} \text{ satisfies } P \iff \mathbf{s}^X \text{ satisfies } P^X \ (\forall X \in 2^{\mathbb{N}}).$$

Disclaimer: Not completely true though.
Ex: if S is the word problem, or Hilbert's 10'th problem, S^X doesn't make sense.

Back to degrees with names

Suppose **s** is a m-degree with a name.

If P is a natural property, one would expect that

$$\mathbf{s} \text{ satisfies } P \iff \mathbf{s}^X \text{ satisfies } P^X \ (\forall X \in 2^{\mathbb{N}}).$$

Disclaimer: Not completely true though.
Ex: if S is the word problem, or Hilbert's 10'th problem, S^X doesn't make sense.

Natural, relativizable, m-degrees **s**

usually give way to

to a function $X \mapsto S^X : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ such that $X \equiv_T Y \implies S^X \equiv_m S^Y$.

where $X \equiv_T Y$ iff X is Y-computable and Y is X-computable.

Natural many-one degrees $\iff$ Wadge degrees

Natural many-one degrees $\iff$ Wadge degrees

Def: 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

Def: 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 $\Phi_{u_1(i,j)}$.

Natural many-one degrees $\iff$ Wadge degrees

Def: 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

Def: 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 $X \geq_T C$.

Natural many-one degrees $\iff$ Wadge degrees

Def: 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.

The version for $(\equiv_T, \equiv_T)$-invariant is known as Martin's conjecture, and the uniform case was proved by Slaman and Steel in [Steel 82][Slaman, Steel 88]