Definability in degree structures

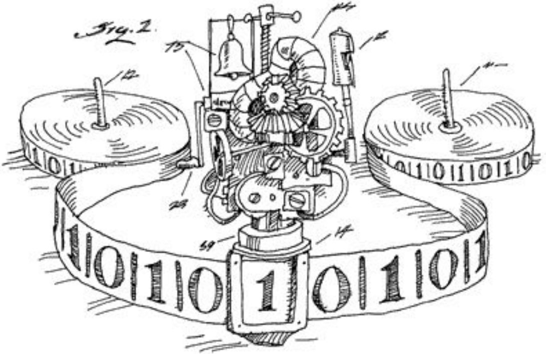
Mariya I. Soskova[1]

Sofia University

April 3, 2017

[1]Supported by the "Women in Science" program and Sofia University Science Fund.

Computable sets and functions



Definition (Turing, Church 1936)

A function $f : \mathbb{N} \to \mathbb{N}$ is *computable* if there is a **computer program** which takes as input a natural number n and outputs $f(n)$.

A set $A \subseteq \mathbb{N}$ is *computable* if its characteristic function is computable.

Computably enumerable sets

Definition

A set $A \subseteq \mathbb{N}$ is *computably enumerable* (c.e.) if it can be enumerated by a computer program.

Example (Davis, Matyasevich, Putnam, Robinson 1970)

A set $S \subseteq \mathbb{N}$ is Diophantine if $S = \{n \mid \exists \bar{m}(P(n, \bar{m}) = 0)\}$.

The Diophantine sets are exactly the c.e. sets.

Example (Novikov, Boone 1955)

The word problem for a finitely presented group is c.e.

Every c.e. set can be coded as the word problem for a finitely presented group.

Example (Hilbert's Entscheidungsproblem 1928)

The set of provable formulas in first order logic is c.e.

An incomputable c.e. set

A set A is computable if and only if both A and $\overline{A}$ are c.e.

We can code programs by natural numbers: P_e is the program with code e.

Definition

The *halting set* is $K = \{e \mid P_e \text{ halts on input } e\}$.

Theorem (Turing 1936)

The halting set K is not computable.

Proof.

If K were computable then $\overline{K}$ would be computable.

Let e be such that P_e halts on n if and only if $n \in \overline{K}$.

P_e halts on $e \Leftrightarrow e \in \overline{K} \Leftrightarrow P_e$ does not halt on e. □

Comparing the information content of sets

Consider a program that has access to an external database, an *oracle*. During its computation the program can ask the oracle membership questions: does n belong to you or not?

Definition (Post 1944)

$A \leq_T B$ if and only if there is a program that computes the elements of A using B as an oracle.

Example

Consider a Diophantine set $S = \{n \mid (\exists \bar{m}) P(n, \bar{m}) = 0\}$. Let $P_{e(n)}$ be the program that ignores its input and for a listing $\{\bar{m}_i\}_{i \in \mathbb{N}}$ of all tuples $\bar{m}_i$ calculates $P(n, \bar{m}_0), P(n, \bar{m}_1), \ldots$ until it sees that the result is 0 and then halts. Then $S \leq_T K$ by the program that on input n computes $e(n)$ and asks the oracle whether $e(n) \in K$.

Comparing the information content of sets

Definition (Friedberg and Rogers 1959)

$A \leq_e B$ if there is a program that transforms an enumeration of B to an enumeration of A.

The program is a c.e. table of axioms of the sort:

$$\textit{If } \{x_1, x_2, \dots, x_k\} \subseteq B \textit{ then } x \in A.$$

Example

Let $S = \{n \mid (\exists \bar{m}) P(n, \bar{m}) = 0\}$ again. $\overline{S} \leq_e \overline{K}$ by the program that consists of the axioms:

$$\textit{If } \{e(n)\} \subseteq \overline{K} \textit{ then } n \in \overline{S}.$$

Degree structures

Definition

1. $A \equiv B$ if and only if $A \leq B$ and $B \leq A$.
2. $\mathbf{d}(A) = \{B \mid A \equiv B\}$.
3. $\mathbf{d}(A) \leq \mathbf{d}(B)$ if an only if $A \leq B$.
4. Let $A \oplus B = \{2n \mid n \in A\} \cup \{2n+1 \mid n \in B\}$. Then $\mathbf{d}(A \oplus B) = \mathbf{d}(A) \vee \mathbf{d}(B)$.

And so we have two upper semi-lattices:

1. The Turing degrees $\mathcal{D}_T$ with least element $\mathbf{0}_T$ consisting of all computable sets.
2. The enumeration degrees $\mathcal{D}_e$ with least element $\mathbf{0}_e$ consisting of all c.e. sets.

The jump operation

The halting set with respect to A is the set
$K_A = \{e \mid P_e$ using oracle A halts on input e$\}$.

$A <_T K_A$.

Definition

The jump of a Turing degree is $\mathbf{d}_T(A)' = \mathbf{d}_T(K_A)$.

We can apply a similar construction to enumeration reducibility to obtain the enumeration jump.

We always have $\mathbf{a} < \mathbf{a}'$.

What connects $\mathcal{D}_T$ and $\mathcal{D}_e$

Proposition

$A \leq_T B \Leftrightarrow A \oplus \overline{A} \leq_e B \oplus \overline{B}$.

The embedding $\iota : \mathcal{D}_T \to \mathcal{D}_e$, defined by $\iota(\mathbf{d}_T(A)) = \mathbf{d}_e(A \oplus \overline{A})$, preserves the order, the least upper bound, and the jump operator.

$\mathcal{T} = \iota(\mathcal{D}_T)$ is the set of *total* enumeration degrees.

$$(\mathcal{D}_T, \leq_T, \mathbf{0}_T) \cong (\mathcal{T}, \leq_e, \mathbf{0}_e) \subseteq (\mathcal{D}_e, \leq_e, \mathbf{0}_e)$$

Properties of the degree structures

Similarities

1. Both $\mathcal{D}_T$ and $\mathcal{D}_e$ are uncountable structures with least element and no greatest element.
2. They have uncountable chains and antichains.
3. They are not lattices: there are pairs of degrees with no greatest lower bound.

Differences

1. (Spector 1956) In $\mathcal{D}_T$ there are *minimal degrees*, nonzero degrees $\mathbf{m}$ such that the interval $(\mathbf{0}_T, \mathbf{m})$ is empty.
2. (Gutteridge 1971) $\mathcal{D}_e$ is downwards dense.

Definability

Definition

A set B is definable in a structure $\mathcal{A}$ if there is a formula $\varphi(x)$ (in the language of the structure) such that $n \in B$ if and only if $\varphi(n)$ is true in $\mathcal{A}$.

This concept naturally extends to definable relations and functions.

Example

In the field of real numbers $\mathcal{R} = (\mathbb{R}, 0, 1, +, \times)$ the set of non-negative numbers is definable by the formula $\varphi(x) : (\exists y)(x = y \times y)$.

In arithmetic $\mathcal{N} = (\mathbb{N}, 0, 1, +, \times)$ the set of prime numbers is definable by the formula $\varphi(x) : x \neq 1 \wedge (\forall y)((\exists z)(y \times z = x) \rightarrow (y = 1 \vee y = x))$.

In the partial order of the Turing degrees $(\mathcal{D}_T, \leq_T, \mathbf{0}_T)$ the set of minimal degrees is definable by the formula $\varphi(x)$:
$x \neq \mathbf{0}_T \wedge (\forall y)(y \leq_T x \rightarrow (y = x \vee y = \mathbf{0}_T))$.

Arithmetic vs Degrees

Second order arithmetic $\mathcal{Z}_2$ is $\mathcal{N}$ with an additional sort for sets of natural numbers and a membership relation.

Classical results due to Kleene and Post show that the relations and functions $\leq_T$, $\leq_e$, $\oplus$, K_A are definable in second order arithmetic.

Do they translate to definable relations and functions in our degree structures?

Example

The function that maps A and B to $A \oplus B$ is definable in $\mathcal{Z}_2$ by $\varphi(X, Y, Z)$: $(\forall n)(n \in X \leftrightarrow n + n \in Z) \wedge (n \in Y \leftrightarrow n + n + 1 \in Z)$.

The function that maps $\mathbf{d}(A)$ and $\mathbf{d}(B)$ to $\mathbf{d}(A \oplus B)$ is definable by $\psi(x, y, z)$: $x \leq z \ \wedge \ y \leq z \wedge (\forall u)(x \leq u \ \wedge \ y \leq u \to z \leq u)$.

Rogers' questions from 1969

Question

Is the jump operator definable in $\mathcal{D}_T$?

If a structure $\mathcal{A}$ has an automorphism that maps an element in a set X to an element outside X then X is not definable.

Question

Are there any nontrivial automorphisms of $\mathcal{D}_T$ or $\mathcal{D}_e$?

Question

Is the copy of $\mathcal{D}_T$ in $\mathcal{D}_e$ (i.e. the total enumeration degrees) definable in $\mathcal{D}_e$?

Definability in the Turing degrees: Forcing + Coding

Paul Cohen in 1963 invented the method of forcing to prove that the continuum hypothesis is independent from ZFC.

The method gives a way to extend a model V of ZFC to a model $V[G]$ in which a generic object with predetermined properties has been added.

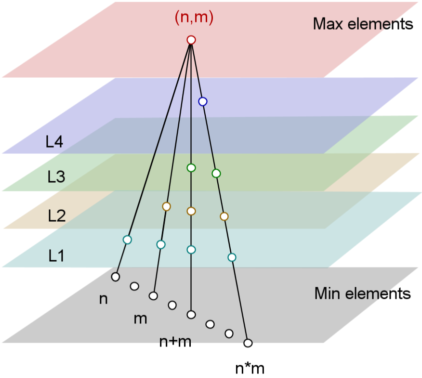
Slaman and Woodin attempted to use forcing to build a generic model $V[G]$ which adds a nontrivial automorphism for $\mathcal{D}_T$. They found that if $V[G]$ has such an automorphism, then so does V.

Theorem (Slaman, Woodin 1986)

There are at most countably many automorphisms of $\mathcal{D}_T$.

There is a single element $\mathbf{g} \leq_T \mathbf{0}_T^{(5)}$ that completely determines the behavior of every automorphism of $\mathcal{D}_T$.

Definability in the Turing degrees: Forcing + Coding



There is a way to represent a model of arithmetic in $\mathcal{D}_T$.

Start by translating arithmetic into a partial order.

Prove that this partial order can be embedded into $\mathcal{D}_T$.

Theorem (Slaman, Woodin 1986)

Every countable antichain $\mathcal{A}$ in $\mathcal{D}_T$ can be coded by three parameters $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$:

$\mathbf{x} \in \mathcal{A} \leftrightarrow \mathbf{x} \leq \mathbf{a} \wedge \mathbf{x} \neq (\mathbf{b} \vee \mathbf{x}) \wedge (\mathbf{c} \vee \mathbf{x}) \wedge \mathbf{x}$ is minimal with these properties.

Definability in the Turing degrees: Forcing + Coding

If $\mathcal{R}$ is definable in $\mathcal{D}_T$ then the set $S = \{X \mid \mathbf{d}_T(X) \in \mathcal{R}\}$ is:

1. definable in second order arithmetic;
2. invariant with respect to $\equiv_T$.

Lets call relations $\mathcal{R}$ on $\mathcal{D}_T$ that correspond to relations S in second order arithmetic of this sort *nice*.

Theorem (Slaman, Woodin 1986)

Every nice relation is definable in $\mathcal{D}_T$ if we allow the use of a parameter.

Every nice relation that is invariant under automorphisms is definable in $\mathcal{D}_T$.

$\mathcal{D}_T$ is rigid if and only if every nice relations is definable in $\mathcal{D}_T$.

The double jump is definable in $\mathcal{D}_T$.

Theorem (Slaman, Shore 1999)

The jump is definable in $\mathcal{D}_T$.

Definability of the enumeration jump

Definition (Kalimullin 2003)

A pair of sets A, B are called a $\mathcal{K}$-pair if there is a c.e. set W, such that $A \times B \subseteq W$ and $\overline{A} \times \overline{B} \subseteq \overline{W}$.

Example (Jockusch 1968)

A set A is a semi-computable if there is a computable selector function s_A:

1. $s_A(x, y) \in \{x, y\}$;
2. If $\{x, y\} \cap A \neq \emptyset$ then $s_A(x, y) \in A$.

The pair $\{A, \overline{A}\}$ is a $\mathcal{K}$-pair witnessed by $W = \{(m, n) \mid s_A(m, n) = m\}$.

Definability of the enumeration jump

Theorem (Kalimullin)

A pair of sets A, B is a $\mathcal{K}$-pair if and only if their enumeration degrees $\mathbf{a}$ and $\mathbf{b}$ satisfy:

$$\mathcal{K}(\mathbf{a}, \mathbf{b}) \leftrightharpoons (\forall \mathbf{x})((\mathbf{a} \vee \mathbf{x}) \wedge (\mathbf{b} \vee \mathbf{x}) = \mathbf{x}).$$

Theorem (Kalimullin)

$\mathbf{0}_e'$ is the largest degree which can be represented as the least upper bound of a triple $\mathbf{a}, \mathbf{b}, \mathbf{c}$, such that $\mathcal{K}(\mathbf{a}, \mathbf{b})$, $\mathcal{K}(\mathbf{b}, \mathbf{c})$ and $\mathcal{K}(\mathbf{c}, \mathbf{a})$.

Corollary (Kalimullin 2003)

The enumeration jump is first order definable in $\mathcal{D}_e$.

Maximal $\mathcal{K}$-pairs

Recall that a set A is semi-computable if is has a computable selector function.

Every set X is Turing equivalent to a semi-computable set L_X.

Every total degree contains $X \oplus \overline{X} \equiv_e L_X \oplus \overline{L_X}$.

So every total enumeration degree is the least upper bound of the elements of a semi-computable $\mathcal{K}$-pair.

Semi-computable $\mathcal{K}$-pairs are *maximal*.

Definition (Ganchev, S)

A $\mathcal{K}$-pair $\{\mathbf{a}, \mathbf{b}\}$ is maximal if for every $\mathcal{K}$-pair $\{\mathbf{c}, \mathbf{d}\}$ with $\mathbf{a} \leq \mathbf{c}$ and $\mathbf{b} \leq \mathbf{d}$, we have that $\mathbf{a} = \mathbf{c}$ and $\mathbf{b} = \mathbf{d}$.

A partial solution

Definition

$\mathcal{D}_e(\leq \mathbf{0}'_e)$ is the substructure of the enumeration degrees that consist of all enumeration degrees bounded by $\mathbf{0}'_e$.

$\mathcal{D}_e(\leq \mathbf{0}'_e)$ is countable and Cooper (1984) showed it is dense.

The members of degrees in $\mathcal{D}_e(\leq \mathbf{0}'_e)$ are easier to construct and handle using finite approximations.

Theorem (Ganchev, S 2009)

$\mathcal{K}$-pairs are first order definable in $\mathcal{D}_e(\leq \mathbf{0}'_e)$.

Theorem (Ganchev, S 2010)

In $\mathcal{D}_e(\leq \mathbf{0}'_e)$ maximal $\mathcal{K}$-pairs and semi-computable $\mathcal{K}$-pairs coincide.

The total degrees are definable in $\mathcal{D}_e(\leq \mathbf{0}'_e)$.

A conjecture

Conjecture (Ganchev and S 2010)

The maximal $\mathcal{K}$-pairs are the semi-computable $\mathcal{K}$-pairs.

Definition

A Turing degree $\mathbf{a}$ is *c.e. in* a Turing degree $\mathbf{x}$ if some $A \in \mathbf{a}$ is c.e. using as oracle some $X \in \mathbf{x}$.

A total degree $\mathbf{a}$ is *c.e. in* a total degree $\mathbf{x}$ if $\mathbf{a}$ is the image of a Turing degree that is c.e. in the pre-image of $\mathbf{x}$.

Proposition (Ganchev, S)

If the conjecture is true then the relation "$\mathbf{a}$ is c.e. in $\mathbf{x}$" for total degrees and $\mathbf{x} \neq \mathbf{0}_e$ is first order definable in $\mathcal{D}_e$.

The automorphisms of $\mathcal{D}_e$

Theorem (S 2012)

There are at most countably many automorphisms of $\mathcal{D}_e$.

There is a single element $\mathbf{g} \leq_e \mathbf{0}_e^{(8)}$ that completely determines the behavior of every automorphism of $\mathcal{D}_e$.

Every nice relation is definable in $\mathcal{D}_e$ if we allow the use of a parameter.

Every nice relation that is invariant under automorphisms is definable in $\mathcal{D}_e$.

$\mathcal{D}_e$ is rigid if and only if every nice relations is definable in $\mathcal{D}_e$.

Example

$\mathcal{T}$ is a nice relation, $\{A \mid \exists B(A \equiv_e B \oplus \overline{B})\}$ is definable in $\mathcal{Z}_2$.

So $\mathcal{T}$ is definable in $\mathcal{D}_e$ with one parameter.

Defining totallity in $\mathcal{D}_e$

Theorem (Cai, Ganchev, Lempp, Miller, S)

The maximal $\mathcal{K}$-pairs are the semi-computable $\mathcal{K}$-pairs.

Every semi-computable set is a left cut in some computable linear ordering of the natural numbers.

Let W be a c.e. set witnessing that a pair of sets $\{A, B\}$ forms a $\mathcal{K}$-pair. We build a semi-computable set C such that $A \leq_e C$ and $B \leq_e \overline{C}$.

1. The countable component: we use W to construct a computable linear ordering on the natural numbers.
2. The uncountable component: find an appropriate left cut in this ordering to define C.

Theorem (Cai, Ganchev, Lempp, Miller, S 2013)

The set of total enumeration degrees is first order definable in $\mathcal{D}_e$.

The relation "$\mathbf{a}$ is c.e. in $\mathbf{x}$" for total degrees is first order definable in $\mathcal{D}_e$.

The total degrees as an automorphism base

Theorem (Selman 1971)

$\mathbf{a} \leq_e \mathbf{b}$ if and only if $\{\mathbf{x} \in \mathcal{T} \mid \mathbf{a} \leq_e \mathbf{x}\} \supseteq \{\mathbf{x} \in \mathcal{T} \mid \mathbf{b} \leq_e \mathbf{x}\}$.

Corollary

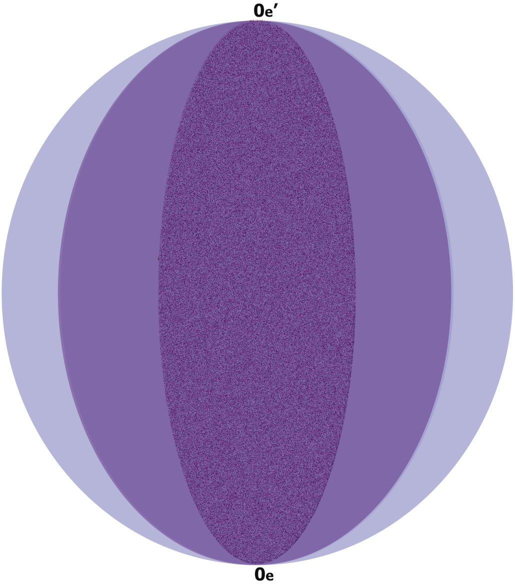
The total enumeration degrees form a definable automorphism base of the enumeration degrees.

- If $\mathcal{D}_e$ has a non-trivial automorphism then so does $\mathcal{D}_T$.
- The total degrees below $\mathbf{0}_e^{(5)}$ are an automorphism base of $\mathcal{D}_e$.

One final application in many steps

Theorem (Slaman, S)

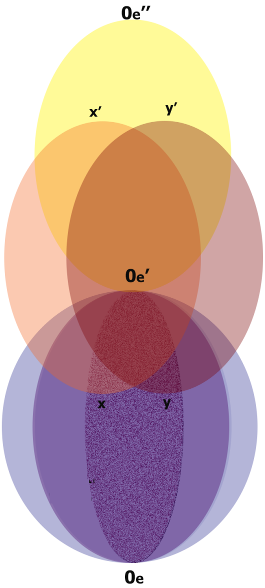
The position of every total enumeration degree in $\mathcal{D}_e(\leq \mathbf{0}_e')$ is completely determined with respect to the positions of all images of Turing degrees of c.e. sets.



One final application in many steps

Theorem (Slaman, S)

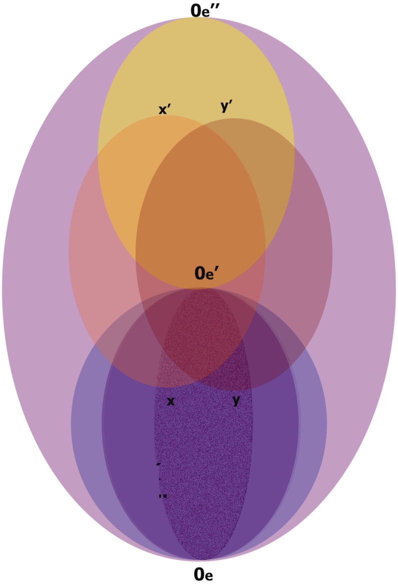
If $\mathbf{x}$ is total and below $\mathbf{0}'_e$ then the position of every total degree in the interval $[\mathbf{x}, \mathbf{x}']$ is completely determined by the total degrees below $\mathbf{0}'_e$.



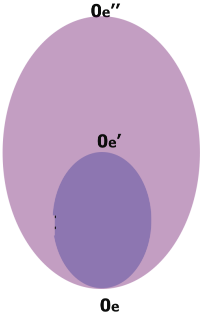
One final application in many steps

Theorem (Slaman, S)

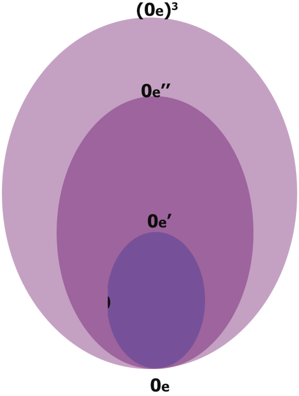
The position of every total degree below $\mathbf{0}_e''$ is completely determined with respect to the positions of all degrees in intervals $[\mathbf{x}, \mathbf{x}']$ with $\mathbf{x}$ total and below $\mathbf{0}_e'$.



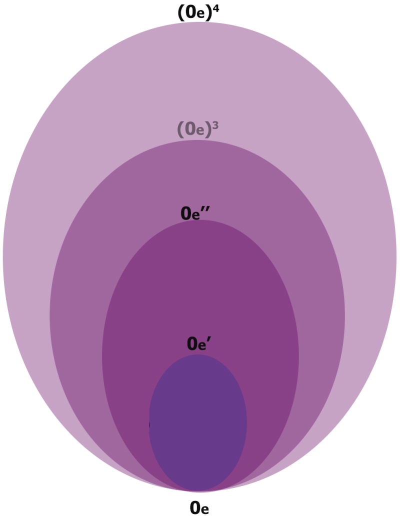
And now we iterate



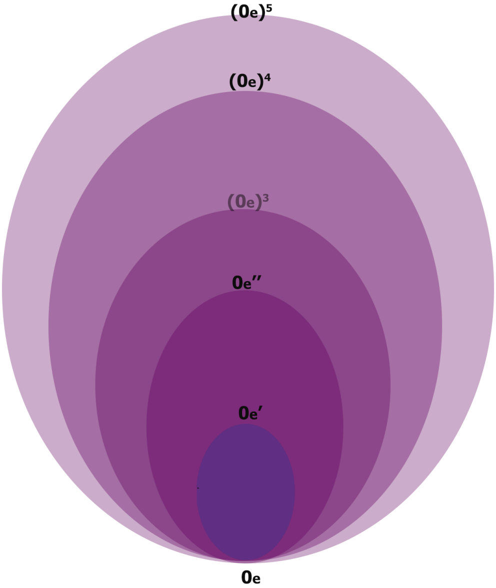
And now we iterate



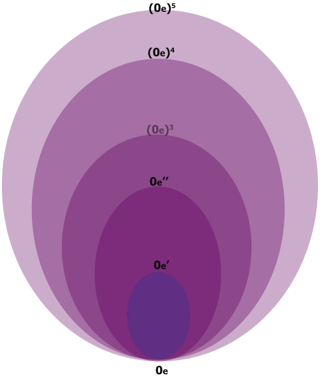
And now we iterate



And now we iterate



And now we iterate



Theorem (Slaman, S 2015)

If $\mathcal{D}_e$ has a nontrivial automorphism then so does the structure of the c.e. Turing degrees.

Thank you!