A Introduction to the Complexity of Sets

Nan Fang

Department of Philosophy, Peking University
2013.11.26

Partial computable functions

Definition

Let ϕ be a function with domain a subset of $\mathbb{N}^k$ and range a subset of $\mathbb{N}$. We say that ϕ is partial computable if there is a Turing program P with k input tapes such that $\phi(x_0, \ldots, x_{k-1}) = y$ iff P on inputs $x_0, \ldots, x_{k-1}$ outputs y. We say that ϕ is computable if ϕ is partial computable and the domain of ϕ is $\mathbb{N}^k$.

- Fix k and an effective listing of the Turing programs for k inputs. Let P_e^k be the program for k inputs given by the e-th program. Let Φ_e^k denote the partial computable function with k arguments given by P_e^k. If $\Phi = \Phi_e^k$ then e is called an index for Φ. Instead of Φ_e^1 we write Φ_e.
- (Parameter Theorem, Padding Lemma, Uniformity, Recursion Theorem, Recursion Theorem with Parameters)

Computably enumerable sets

Definition

We say a set $A \subseteq \mathbb{N}$ is *computably enumerable* (*c.e.*) if A is the domain of some partial computable function.

Definition

A is called computable if its characteristic function is computable; otherwise *A* is called incomputable.

Proposition

A is computable $\Leftrightarrow$ *A and* $\mathbb{N} - A$ *are c.e.*

- halting problem: $\emptyset' = \{e : e \in W_e\}$

Proposition

The set $\emptyset'$ *is c.e. but not computable*

Indices and approcimations

- We write $\Phi_{e,s}(x) = y$ if $e, x, y < s$ and the computation of program P_e on input x yields y in at most s computation steps.
- Let $D_0 = \emptyset$. If $n > 0$ has the form $2^{x_1} + 2^{x_2} + ... + 2^{x_r}$, where $x_1 < ... < x_r$, then let $D_n = x_1, ..., x_r$. We say that n is a strong index for D_n.
- A computable enumeration of a set A is an effective sequence $(A_s)_{s \in N}$ of finite sets such that $A_s \subseteq A_{s+1}$ for each s, and $A = \bigcup_s A_s$

Each c.e. set W_e has the computable enumeration $(W_{e,s})_{s \in N}$. Conversely, if A has a computable enumeration then A is c.e..

many-one reducibility

relative computational complexity of a set A is measured by comparing A to other sets via preorderings called reducibilities. many-one reducibility is one of the simplest examples of a reducibility.

Definition

X is many-one reducible to Y, denoted $X \leq_m Y$, if there is a computable function f such that $n \in X \leftrightarrow f(n) \in Y$ for all n.

Proposition

A is c.e. $\Leftrightarrow$ $A \leq_m \emptyset'$

Definition

A c.e. set C is called $r-$*complete* if $A \leq_m C$ for each c.e. set A.A c.e. set C is called $r-$*complete* if $A \leq_m C$ for each c.e. set A.

Turing reducibility

Many-one reducibility is too restricted to serve as an appropriate measure for the relative computational complexity of sets. So we introduced the Turing machine with an oracle tape.

We will extend the definition of the basic concepts to oracle Turing machines. Now we view the effective listing $(\Phi_e)_{e\in N}$ as a listing of partial functions depending on two arguments, the oracle set and the input.

Turing reducibility

Definition

A total function $f : N \mapsto N$ is called Turing reducible to Y, or computable in Y, if there is an e such that $f = \Phi_e^Y$. We denote this by $f \leq_T Y$. For a set A, we write $A \leq_T Y$ if the characteristic function of A is Turing reducible to Y.A total function $f : N \mapsto N$ is called Turing reducible to Y, or computable in Y, if there is an e such that $f = \Phi_e^Y$. We denote this by $f \leq_T Y$. For a set A, we write $A \leq_T Y$ if the characteristic function of A is Turing reducible to Y.

- $\leq_m$ and $\leq_T$ are preorderings of the subsets of N.
- A is c.e. in Y if $A = W_e^Y$ for some e.
- A is computable in $Y \Leftrightarrow A$ and $N - A$ are c.e. in Y.

Jump operator

Definition

We write $J^Y(e) \simeq \Phi_e^Y(e)$. The set $Y' = dom(J^Y)$ is the Turing jump of Y.

Proposition

A is c.e. in $Y \Leftrightarrow A \leq_m Y'$

Proposition

For each Y, the set Y' is c.e. in Y. And $Y <_T Y'$.

Jump operator

Definition

We define $Y^{(n)}$ inductively by $Y^{(0)} = Y$ and $Y^{(n+1)} = (Y^{(n)})'$. Thus $Y <_T Y^{(1)} <_T Y^{(2)} <_T \dots$.

Proposition

For each Y,Z, we hace $Y \leq_T Z \Leftrightarrow Y' \leq_m Z'$.

approximating the functionals Φ_e, and the use principle

Definition

We write $\Phi^Y_{e,s}(x) = y$ if $e, x, y < s$ and the computation of program P_e on input x yields y in at most s computation steps, with all oracle queries less than s. And we let $W^Y_{e,s} = dom(\Phi^Y_{e,s})$.

Proposition (use principle)

a terminating oracle computation only asks finitely many oracle questions. Hence $(\Phi^Y_{e,s})_{s\in N}$ *approximates* Φ^Y_e, *namely,*

$$\Phi^Y_e(x) = y \leftrightarrow \exists s \Phi^Y_{e,s} = y \tag{1}$$

approximating the functionals Φ_e, and the use principle

Definition

the use of $\Phi_e^Y(x)$, denoted $use\Phi_e^Y(x)$, is defined if $\Phi_e^Y(x)\downarrow$, in which case its value is 1+the largest oracle query asked during this computation (and 1 if no question is asked at all). Similarly, $use\Phi_{e,s}^Y(x)$ is 1+the largest oracle question asked up to stage s.

We write $\Phi_e^\sigma(x) = y$ if $\Phi_e^F(x)$ yields the output y, where $F = i < |\sigma| : \sigma(i) = 1$, and the use is at most $|\sigma|$, and $\Phi_e^\sigma(x)\uparrow$ if there is no such y. Then for each set Y,

$$\Phi_e^Y(x) = y \leftrightarrow \Phi_e^{Y\upharpoonright u}(x) = y, \tag{2}$$

where $u = use\Phi_e^Y(x)$.

weak truth-table reducibility

In the Turing reduction, the use function is unbounded and may grow very quickly. So when we bound the use function we can get stronger reducibilities.

Definition

A function $f : N \mapsto N$ is weak truth-table reducible to Y, denoted $f \leq_{wtt} Y$, if there is Φ_e and a computable bound r such that $f = \Phi_e^Y$ and $\forall n \, use\Phi_e^Y(n) \leq r(n)$. For a set A, we write $A \leq_{wtt} Y$ if the characteristic function of A is weak truth-table reducible to Y.

It may happen that $f \leq_{wtt} Y$ via Φ_e and r such that Φ_e^Z is not total function for some oracle $Z \neq Y$. So there is a stronger reducibility.

truth-table reducibility

Definition

A function $f : N \mapsto N$ is truth-table reducible to Y, denoted $f \leq_{tt} Y$, if there is Φ_e such that $f = \Phi_e^Y$ and Φ_e^Z is total for each oracle Z. For a set A, we write $A \leq_{tt} Y$ if the characteristic function of A is truth-table reducible to Y.

A truth table is a finite boolean combination of the atomic formulas "$n \in X$", for $n \in N$. Let $\sigma_{nn\in N}$ be an effective list of all truth tables.

truth-table reducibility

Proposition

$X \leq_{tt} Y \Leftrightarrow$ *there is a computale function f such that for all n,* $n \in X \leftrightarrow Y \vDash \sigma_{f(x)}$.

Proposition

$X \leq_{tt} \Rightarrow X \leq_{wtt} Y$.

Clearly, there are implications between our reducibilities: $\leq_m \Rightarrow \leq_{tt} \Rightarrow \leq_{wtt} \Rightarrow \leq_T$.

degree structures

As we already known that the previous redicibilities are preorderings. For a reducibility $\leq_r$ we can give a equivalence relation by $X \equiv_r Y \leftrightarrow X \leq_r Y \leq_r X$. The equivalence classes are called r-degrees. Then the r-degrees form a partial order denoted by D_r.

absolute computational complexity

A hierarchy of absolute computational complexity is abtained by considering $C^{(n)}$ within the Turing degrees, for $n \geq 0$. Note that $C^{(n)} \geq_T \emptyset^{(n)}$.

A lowness property of a set specifies a sense in which the set is computationally weak. Usually this means that it is not very useful as an oracle. While a highness property says that the set is computationally strong.

Sets that are low_n

Definition

Let $n \geq 0$. We say that C is low_n if $C^{(n)} \equiv_T \emptyset^{(n)}$. The low_1 sets are called low.

By the note above, for a set C to be low_n it suffices to require that $C^{(n)} \leq_T \emptyset^{(n)}$.
As $C^{(n)} \leq_T \emptyset^{(n)} \Leftrightarrow C^{(n+1)} \leq_m \emptyset^{(n+1)} \Rightarrow C^{(n+1)} \leq_T \emptyset^{(n+1)}$. We have the following hierarchy:

$$computable \subseteq low_1 \subseteq low_2 \subseteq ... \subseteq \{Z : Z \not\geq_T \emptyset'\} \tag{3}$$

Actually, it is a proper hierarchy.
(some other lowness properties: superlow, generalized $low_n(GL_n)$, computably dominated)

Sets that are $high_n$

Definition

Let $n \geq 0$. A set C is $high_n$ if $\emptyset^{(n+1)} \leq C^{(n)}$.

As $\emptyset^{(n+1)} \leq C^{(n)} \Rightarrow \emptyset^{(n+2)} \leq C^{(n+1)}$, we have $high_n \subseteq high_{n+1}$.
And we can easily show that for any $m, n \in N$,
$C \in low_n \Rightarrow C \notin high_m$.
Let $non - high_n$ denotes the complementary of $high_n$, we can refine the hierarchy above as following:

$$computable \subseteq low_1 \subseteq low_2 \subseteq ...$$
$$\subseteq non - high_2 \subseteq non - high_1 \subseteq \{Z : Z \not\geq_T \emptyset'\}$$

The arithmetical hierarchy

Definition

1. A is Σ_n^0 if there is a computable relation $R(x, y_1, ..., y_n)$ such that $x \in A \leftrightarrow \exists y_1 \forall y_2 ... Q y_n R(x, y_1, ..., y_n)$, where Q is $\exists$ if n is odd and $\forall$ if n is even.
2. A is Π_n^0 if there is a computable relation $R(x, y_1, ..., y_n)$ such that $x \in A \leftrightarrow \forall y_1 \exists y_2 ... Q y_n R(x, y_1, ..., y_n)$, where Q is $\forall$ if n is odd and $\exists$ if n is even.
3. A is Δ_n^0 if A is both Σ_n^0 and Π_n^0.

It is easy to see that A *is* $\Pi_n^0 \Leftrightarrow \overline{A}$ *is* Σ_n^0.

The arithmetical hierarchy

Proposition

- *A is* $\Sigma^0_1 \Leftrightarrow$ *A is c.e.*
- *A is* $\Sigma^0_n \Leftrightarrow$ *A is c.e. in* $\emptyset^{(n-1)}$
- *For* $n \geq 1$. *A is* $\Delta^0_n \Leftrightarrow A \leq_T \emptyset^{(n-1)}$.

There is a arithmetical hierarchy as following:

$$\begin{array}{ccccccccccc} & & \subset & \Sigma^0{}_1 & \subset & & \subset & \Sigma^0{}_2 & \subset & & \subset \\ \Delta^0{}_1 & & & & & \Delta^0{}_2 & & & & \Delta^0{}_3 & & \cdots \\ & & \subset & \Pi^0{}_1 & \subset & & \subset & \Pi^0{}_2 & \subset & & \subset \end{array}$$

The Limit Lemma

Theorem (Limit Lemma, Shoenfield)

For a set Z, we have Z is Δ^0_2 iff there is a computable binary function g such that, for all n,
(1) $lim_s g(n,s)$ exists (i.e. $|\{s : g(n,s) \neq g(n,s+1)\}| < \infty$), and
(2) $Z(n) = lim_s g(n,s)$.

Actually, we can view $g(n,s)_{s\in N}$ as a list of set $Z_s(n)$, then $Z(n) = lim_s Z_s(n)$. We say that $Z_{s\,s\in N}$ is a computable approximation of Z.

Difference hierarchy

For a Δ_2^0 set, every number n change finite many times in its computabel approximation, but not bounded. When we bound it, we can get an other hierarchy, that's the difference hierarchy.

Definition

A set Z is $\omega - c.e.$ if there is a computable approximation $Z_{s\,s\in N}$ of Z and a computable function b such that
$|\{s : Z_s(n) \neq Z_{s+1}(n)\}| \leq b(n)$ for each $n \in N$.

If the function can be chosen constant of value n, then we say Z is n-c.e..
Obviously, Z is 1-c.e. iff Z is c.e.. And the difference hierarchy is

$$computable \subset c.e. \subset 2-c.e. \subset 3-c.e. \subset ... \subset \omega-c.e. \subset \Delta_2^0. \quad (4)$$

This is also proper.

Post's problem

In 1944, Post observed that all computably enumerable problems known at the time were either computable or of Turing degree $\emptyset'$. He asked the following question.

- Post's Problem: Does there exist a c.e. set A such that $\emptyset <_T A <_T \emptyset'$.

It took 12 years to answer his question. Kleene and Post (1954) made a first step by building a pair of Truing incomparable Δ^0_2-sets, that is a pair of sets $Y, Z \leq_T \emptyset'$ such that $Y \not\leq_T Z$ and $Z \not\leq_T Y$. To do so they introduced the method of finite extentions. Post's question was finally answered in the affirmative by Friedberg(1957) and Muchnik (1956) independently. They built a pair of Turing incomparable sets that are also computably enumerable. Their proof technique is nowadays called the priority method with finite injury.

Post's problem

But the solution by Friedberg and Muchnik is far from being natural. It depends on the particular choice of a universal Turing program. In computability theory a natural class of sets should be closed under computable permutations. We want to find a natural Post property which is a property of c.e. sets which is satisfied by some incomputable set and implies Turing incompleteness. Several people have found some Post properties but all are not natural and depend on our particular version of the universal Turing program. Any reasonable solution W to Post's problem should be relativizable to an oracle set X, so one would expect that $X <_T W^X <_T X'$ for each X. If the solution does not depend on the choice of the universal program, it should also be degree invariant: if $X \equiv_T Y$, then $W^X \equiv_T W^Y$. The existence of such a degree invariant solution to Post's problem is a long-standing open question posed by Sacks (1963).

References

- Odifreddi. P. G. *Classical Recursion Theory*. North-Holland, 1989.
- Nies, Andre. *Computability and Randomness*. Oxford: Oxford UP, 2009.
- Downey, Ronald G., and Denis R. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer, 2010.