Reductions

2016年10月8日 15:23

Informal Claim The only way we can prove a set is not computable is essentially by diagonal argument.

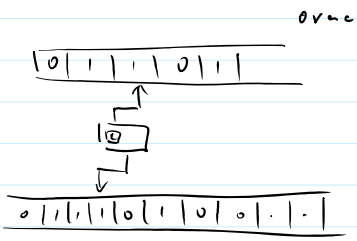
In general, if we want to prove a set B is not computable

we assume to the contradiction that B is computable, and we prove that if B is computable, then some set A is also computable, but A is not computable by a diagonal argument

In such a prove we actually show that, A is "reducible" to B

Formally

Oracle (Turing) Machine / Program



You can imagin an oracle as a hard disk with possibly infinite data stored in it.

Note. the set of all oracle machine / Program is effectively enumerable:

$\Phi_0, \Phi_1, \cdots$

Note also, a oracle program coded by e and an oracle A, determines a partial function (written Φ_e^A), i.e. $\Phi_e^A(n)\uparrow$ if the oracle program e with oracle A and input n will not halt, and $\Phi_e^A(n)\downarrow = m$ if the oracle program e with oracle A and input n halts in finit steps and outputs m.

The meanig of $\Phi_{e,s}^A(n)\uparrow$ or $\Phi_{e,s}^A(n)\downarrow$ is also obvious.

Note. if $\Phi_e^A(n)\downarrow$, i.e there is $s\in\mathbb{N}$ s.t. $\Phi_{e,s}^A(n)\downarrow$, the program Φ_e only makes finitely many queries to A.

Def We define the use of an oracle computation $\Phi_e^A(n)$ (written $\varphi_e^A(n)$, or use $\Phi_e^A(n)$) to be $x+1$, where x is the largest number s.t. the value of $A(x)$ is queried during the computation.
If $\Phi_e^A(n)\downarrow$ and no such number is queried, then $\varphi_e^A(n)=0$
If $\Phi_e^A(n)\uparrow$, then set $\varphi_e^A(n)=\infty$

We can also make sense of $\varphi_{e,s}^A(n)$ (or use $\Phi_{e,s}^A(n)$).
We may assume that $\varphi_{e,s}^A(n)<s$.

We can also use finite strigs as oracle. Let $\tau\in 2^{<\omega}$, by Φ_e^τ, we denote the partial function: $\Phi_e^\tau(n)\uparrow$ if the program does not halt or attempts to make any queries beyond the length of τ ($\geq|\tau|$); and $\Phi_e^\tau(n)\downarrow=m$ if it halts and only queries on numbers $<|\tau|$.

Def We say set A is Turing reducible to B (written $A\leq_T B$) if there is an oracle machine / Program whose oracle tape codes B

computes A, i.e. $\chi_A = \Phi_e^B$ for some $e \in \mathbb{N}$

Example $\emptyset' \leq_T Tot$, $\emptyset \not\equiv_T \emptyset'$

Def $A \equiv_T B$ iff $A \leq_T B$ and $B \leq_T A$

Exe $\emptyset' = \{e \mid \Phi_e(e)\downarrow\} \equiv_T \{\langle e, n\rangle \mid \Phi_e(n)\downarrow\}$

Note $\equiv_T$ is an equivalent relation on $\mathcal{P}(\mathbb{N})$, i.e. transitive, reflective and symmetric

For $A \subseteq \mathbb{N}$, define $\deg(A) = \{B \subseteq \mathbb{N} \mid B \equiv_T A\}$. We call $\deg(A)$ a Turing degree

Define $\mathcal{D}_T = \{\deg(A) \mid A \subseteq \mathbb{N}\}$ to be the class of all Turing degrees

Def For Turing degrees $\underset{\sim}{a}$, $\underset{\sim}{b}$, we say $\underset{\sim}{a} \leq_T \underset{\sim}{b}$ iff there are

$A \in \underset{\sim}{a}$, $B \in \underset{\sim}{b}$ $A \leq_T B$

Exe $\leq_T$ is well-defined

To understand the structure $(\mathcal{D}_T, \leq_T)$ is the central subject of classic computability theory.

Some simple facts about $(\mathcal{D}_T, \leq_T)$

1) $\leq_T$ is a partial order (transitive, reflexive and $a \leq_T b$, $b \leq_T a \Rightarrow a = b$)

2) For any $\underset{\sim}{a}, \underset{\sim}{b} \in \mathcal{D}_T$, there is a least upper bound (so $(\mathcal{D}_T, \leq_T)$ is a upper semi-lattice)

For $A, B \subseteq \mathbb{N}$, define the information sum (A join B):

$$A \oplus B = \{2n \mid n \in A\} \cup \{2n+1 \mid n \in B\}$$

clearly $A \leq_T A \oplus B$, $B \leq_T A \oplus B$, and for each set C, if $A \leq_T C$ and $B \leq_T C$, then $A \oplus B \leq_T C$.

For $\underset{\sim}{a}, \underset{\sim}{b} \in \mathcal{D}_T$, define $\underset{\sim}{a} \vee \underset{\sim}{b} = \deg(A \oplus B)$ for some $A \in \underset{\sim}{a}$, $B \in \underset{\sim}{b}$

Note. If $A_1 \equiv_T A_2$, $B_1 \equiv_T B_2$, then $A_1 \oplus B_1 \equiv_T A_2 \oplus B_2$, so $\underset{\sim}{a} \vee \underset{\sim}{b}$ is well-defined.

3) $(\mathcal{D}_T, \leq_T)$ has a least element $\underset{\sim}{0} = \deg(\emptyset)$

Since if A is computable, that $A \leq_T B$ for any $B \subseteq \mathbb{N}$.

4) $(\mathcal{D}_T, \leq_T)$ has no maximum element.

Def For any set A, define the (Turing) Jump of A (written A') to be

$$A' = \{ e \mid \Phi_e^A(e) \downarrow \}$$

i.e. the halting problem relative to A

Exe $\quad A' \not\equiv_T A$ for any A

Next time, we will switch to randomness, we will switch back to classic computability theory if we have to.