Computably enumerable sets

Recall the "halting problem" $\emptyset' = 0' = \{ e \in \mathbb{N} \mid \Phi_e(e)\downarrow \}$ is not computable

But there is a program which can "generate" every element in $\emptyset'$ given enough time.

To characterize the intuition of "effectively generable", we define:

Def set $A \subseteq \mathbb{N}$ is computably enumerable (c.e) iff
$$A = \mathrm{dom}(\Phi_e) \text{ for some } e \in \mathbb{N}$$

Fact $\emptyset'$ is c.e.

Proof Consider the program Φ_d s.t. $\Phi_d(e) \simeq \Phi_e(e)$

Fact Set A is computable (i.e. χ_A is computable) iff A and $\bar{A}$ ($=\mathbb{N}\setminus A$) are both c.e.

Proof ($\Rightarrow$) Let Φ_{e_A} be s.t. on input n, it computes $\chi_A(n)$, if $\chi_A(n)=1$ it halts and output 1, o.w. it doesn't halt

Let $\Phi_{e_{\bar{A}}}$ be ... $=0$ ...

then $A = \mathrm{dom}\,\Phi_{e_A}$, $\bar{A} = \mathrm{dom}\,\Phi_{e_{\bar{A}}}$

($\Leftarrow$) Assume $A = \mathrm{dom}\,\Phi_{e_1}$, $\bar{A} = \mathrm{dom}\,\Phi_{e_2}$

Consider the program Φ_e: with input n, it runs $\Phi_{e_1}(n)$, $\Phi_{e_2}(n)$ in turn, once $\Phi_{e_1}(n)\downarrow$, it outputs 1, or if $\Phi_{e_2}(n)\downarrow$ finally, it outputs 0 $\square$

Convention Define $W_e = \mathrm{dom}\,\Phi_e = \{ n \mid \Phi_e(n)\downarrow \}$

$$W_{e,s} = W_e[s] = \{ n \leq s \mid \Phi_{e,s}(n)\downarrow \}$$

(we can safely assume that $\Phi_{e,s}(n)\downarrow$ only when $s \geq n$)

Note $W_0, W_1, \dots$ is an effective enumeration of all c.e. sets.

If $A = W_e$, we say e is an index for the c.e. set A

Note Every c.e. set have infinitely many indices

Lemma Let $A \subseteq \mathbb{N}$. TFAE (the following are equivalent):

1) A is c.e.
2) $A = \mathrm{ran}\,\Phi_e$ for some $e \in \mathbb{N}$
3) There is a computable enumeration of A, i.e. there is a (total) computable function $g: \mathbb{N} \to \mathbb{N}$ s.t. for each $s \in \mathbb{N}$

$0 \cdots 0\, 1\, 0 \cdots 0\, 1\, 0 \cdots \; 0\, 1$ (x_i, x_1, x_j)

$g(s)$ codes a finite set A_s [$g(s) = 2^{x_1} + \dots + 2^{x_n}$, where $A_s = \{x_1, \dots, x_n\}$] — in this case we also write $g(s) = A_s$

$g(s) = 0$ if $A_s = \emptyset$

where $A_s \subseteq A_{s+1}$ for all $s \in \mathbb{N}$, and $A = \bigcup_s A_s$

where $A_s \subseteq A_{s+1}$ for all $s \in \mathbb{N}$, and $A = \bigcup_s A_s$

(we can further require every $|A_{s+1} \setminus A_s| \le 1$)

Proof 1) $\Rightarrow$ 3) Assume $A = W_e = \mathrm{dom}\,\Phi_e$

The function $f: \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ $f(e,s)$ = the code of $W_{e,s}$ is computable

Let $A_s = W_{e,s}$, and $g(s) = f(e,s)$

To ensure $\left|\{n \mid \Phi_{e,s+1}(n)\downarrow\} \setminus \{n \mid \Phi_{e,s}(n)\downarrow\}\right| \le 1$

Consider the modified program $\Phi_{p(e)}$: At stage $s+1$, it checks all $n < s+1$ if $n = \min\{m \in W_{e,s+1} \mid m \notin W_{p(e),s}\}$, then $\Phi_{p(e),s+1}(n)\downarrow$

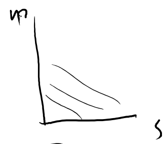
3) $\Rightarrow$ 2) Let g be a computable enumeration of A, i.e. $g(s) = A_s$

$A_s \subseteq A_{s+1}$
$A = \bigcup_s A_s$

Consider the program Φ_d: with input $s \in \mathbb{N}$, it computes A_{s+1} and A_s if $|A_{s+1} \setminus A_s| \neq \phi$, output the unique element in $A_{s+1} \setminus A_s$

2) $\Rightarrow$ 1) Let $A = \mathrm{ran}\,\Phi_d$

Let Φ_e be the program: with input n, it runs $\Phi_{d,s}(m)$ for every pair (m,s), once $\Phi_{d,s}(m)\downarrow = n$, it halts and output 0



□

Recall We say $f_0, f_1, \cdots$ are uniformly partial computable, if there is a partial computable f s.t. $f(n,m) = f_n(m)$ ...

Def ① We say $A_0, A_1, \cdots$ are uniformly c.e. if $A_n = \mathrm{dom}\, f_n$ for where $f_0, f_1, \cdots$ are uniformly partial computable functions

② We say $A_0, A_1, \cdots$ are uniformly computable if $\chi_{A_0}, \chi_{A_1}, \cdots$ are uniformly partial computable

Fact ① $A_0, A_1, \cdots$ are uniformly c.e. iff ② there is a computable (total function) $g: \mathbb{N} \to \mathbb{N}$ s.t. $A_n = W_{g(n)}$ iff ③ there is a c.e. set A s.t. $A_n = \{x \mid (n,x) \in A\}$

Proof Exe

Fact $A_0, A_1, \cdots$ are uniformly computable iff both $A_0, A_1, \cdots$ and $\overline{A_0}, \overline{A_1}, \cdots$ are uniformly c.e.

Proof Exe

A set $A \subseteq \mathbb{N}$ is an index set iff whenever $e \in A$ and $\Phi_e = \Phi_d$, then $d \in A$

Example: ϕ, $\mathbb{N}$, $\{e \mid \Phi_e \text{ is total}\}$ are index sets

Intuitively: An index set codes a problem on program whose answer does not depend on the choice of among the equivalent programs.

Theorem (Rice)

An index set A is computable iff $A = \mathbb{N}$ or $A = \phi$

Proof: Assume to contradiction that index set A is computable, $A \neq \phi$ and $A \neq \mathbb{N}$.

We show ϕ' is also computable (and it is a contradiction)

To decide if $\Phi_x(x)\downarrow$,

Fix e s.t. $\mathrm{dom}\,\Phi_e = \phi$. w.l.o.g. (without loss of generality)

Assume $e \notin A$. Pick $i \in A$ [this is possible, because $A \neq \mathbb{N}$ and $A \neq \phi$]

Consider the program $\theta(x,y)$:

$$\theta(x,y) = \begin{cases} \Phi_i(y) & \Phi_x(x)\downarrow \\ \uparrow & \Phi_x(x)\uparrow \end{cases}$$

By s-n-m theorem, there is a computable $g: \mathbb{N} \to \mathbb{N}$ s.t.

$$\Phi_{g(x)}(y) = \theta(x,y)$$

Now, if $\Phi_x(x)\downarrow$, then $\Phi_{g(x)} \simeq \Phi_i$, so $g(x) \in A$

if $\Phi_x(x)\uparrow$ then $\Phi_{g(x)} \simeq \Phi_e$ (the program never halts), so $g(x) \notin A$

So to see if $\Phi_x(x)\downarrow$, we just compute if $g(x) \in A$ $\square$

Corollary: $\{(e,d) \mid \Phi_e \simeq \Phi_d\}$ is not computable

Recursion Theorem (Fixed Point Theorem)

Theorem (Kleene)

Let $f: \mathbb{N} \to \mathbb{N}$ be a total computable function. Then there is a fixed point for f, i.e. there is $e \in \mathbb{N}$ s.t. $\Phi_e \simeq \Phi_{f(e)}$

Furthermore, a fixed point of f can be effectively computed from an index of f, i.e. there is a total function $g: \mathbb{N} \to \mathbb{N}$ if $f \simeq \Phi_d$, then $g(d)$ is a fixed point for f, i.e.

$$\Phi_{g(d)} \simeq \Phi_{f \circ g(d)}$$

Proof: Fix $f = \Phi_d$. Consider the following program

$$\theta(x,y) = \begin{cases} \Phi_{\Phi_x(x)}(y) & \text{if } \Phi_x(x)\downarrow \\ \uparrow & \text{if } \Phi_x(x)\uparrow \end{cases}$$

By s-n-m, there is a total computable function $\Phi_p: \mathbb{N} \to \mathbb{N}$ s.t.

$$\Phi_{\Phi_p(x)}(y) = \theta(x,y) \quad \text{for all } x, y.$$

Let i be s.t. $\Phi_i = f \circ \Phi_p = \Phi_d \circ \Phi_p$ (i can be effectively obtained from d and p)

Then Φ_i is also total

Let $e = \Phi_p(i)$, we show e is the fixed point. For each y,

$$\Phi_e(y) = \Phi_{\Phi_p(i)}(y) = \theta(i,y) \underset{\text{since } \Phi_i(i)\downarrow}{=} \Phi_{\Phi_i(i)}(y) = \Phi_{f \circ \Phi_p(i)}(y) = \Phi_{f(e)}(y) \quad \square$$

Lemma (Fixed point theorem, parameter version)

Assume $f : \mathbb{N}^2 \to \mathbb{N}$ is partial computable. There is a total computable function $h : \mathbb{N} \to \mathbb{N}$ s.t. for any $n \in \mathbb{N}$ if $f(h(n), n)\downarrow$, then

$$\Phi_{h(n)} = \Phi_{f(h(n), n)}$$

Furthermore, an index of h can be effectively obtain from any index of f

Exe [Hint Let $\Theta(x,y,z) = \begin{cases} \Phi_{\Phi_x(x,y)}(z) & \text{if } \Phi_x(x,y)\downarrow \\ \uparrow & \text{o.w.} \end{cases}$

Let $\Phi_i(x,y) = f(d(x,y), y)$ where $\Phi_{d(x,y)}(z) = \Theta(x,y,z)$

Let $h(y) = d(i,y)$]

Application of Fixed point theorem

Fact $\emptyset'$ is not an index set

Proof Let f be a computable function s.t. $\Phi_{f(n)}(n)\downarrow$ and $\Phi_{f(n)}(m)\uparrow$ for all $n \neq m$ [$f(n)$ codes the program to check if the input is n, if so it halts, o.w. not halt]

Let n be a fixed point for f, i.e. $\Phi_{f(n)} \simeq \Phi_n$

Let $m \neq n$ be another index for n, i.e. $\Phi_n \simeq \Phi_m$

For $\emptyset'$ to be index set, we need n, m both in $\emptyset'$ or both out.

But since $\Phi_{f(n)} = \Phi_m$, so $\Phi_n(n) = \Phi_{f(n)}(n)\downarrow$, $\Phi_m(m) = \Phi_{f(n)}(m)\uparrow$ □