Self-referential algorithms - New proof that $Halt_0$ is not decidable

---

Recall $Halt_0 = \{ \ulcorner M \urcorner : M(\epsilon) \text{ halts} \}$

Former proof that $Halt_0$ not decidable.

Started with M assumed to decide $Halt_0$

$$\boxed{\text{Formed } M' \text{ s.t. } M'(\epsilon) \text{ acts like } M(\ulcorner M \urcorner)}$$

Gave a many-one reduction from $Halt_{self}$ to $Halt_0$

<u>Now</u> (Self-Reference)

We'll form D_M such that

$D_M(\epsilon)$ acts the same as $M(\ulcorner D_M \urcorner)$

<u>Encourage</u> google "self-printing programs" or "quines"

Thm Let M be an algorithm that takes one input.

$\Rightarrow$ There is an algorithm D_M such that

"Diagonal" D_M acts the same as $M(\ulcorner D_M \urcorner)$

Proof: Recall f', f

$f'(f(\ulcorner M_1 \urcorner), \ulcorner M_2 \urcorner)$ = Gödel number of algorithm that runs $M_2(\ulcorner M_1 \urcorner)$.

$\langle \ulcorner M_1 \urcorner, \ulcorner M_2 \urcorner \rangle \longmapsto f'(f(\ulcorner M_1 \urcorner), \ulcorner M_2 \urcorner)$

Computable

$f(\ulcorner M_1 \urcorner)$ = G.n. of an algorithm that ~~and~~ outputs $\ulcorner M_1 \urcorner$.

$f(w)$ = G.n. of an algorithm for constant function w

$f'(\ulcorner M_3 \urcorner, \ulcorner M_2 \urcorner)$ - G.n. of an algorithm that, on input w,

(1) Runs $M_3(w)$

(2) Runs M_2 on the output of M_3

Proof of Theorem

Form algorithm E_M s.t.

$$E_M(w) \text{ runs } M(\underbrace{f'(f(w), w)})$$

$$E_M(\ulcorner N \urcorner) \text{ runs } M(\underbrace{f'(f(\ulcorner N \urcorner), \ulcorner N \urcorner)})$$

G.n. of algorithm that runs N on $\ulcorner N \urcorner$.

Define D_M to the algorithm with Gödel number

$$\ulcorner D_M \urcorner = f'(f(\ulcorner E_M \urcorner), \ulcorner E_M \urcorner)$$

So D_M ignores its input and runs

E_M on $\ulcorner E_M \urcorner$

i.e. runs M on $f'(f(\ulcorner E_M \urcorner), \ulcorner E_M \urcorner)$

i.e runs M on $\ulcorner D_M \urcorner$.

$f(11) :-$ 

$f'(f(\ulcorner E_M \urcorner), \ulcorner E_M \urcorner) =$ G.n. of algorithm that runs E_M on $\ulcorner E_M \urcorner$.

$f(w) =$ "main(){ printf("w\n"); }"

substitute in w

$\underbrace{\text{"main() \{ printf("}}$ $\circ$ $\underbrace{w}$ $\circ$ $\underbrace{\text{"\\n"); \}"}}$

Theorem Halto is undecidable

Pf Assume N decides Halto

Define algorithm M by

Algorithm M

Input w

Runs N on w

If $N(w)$ rejects, then halt.

If $N(w)$ accepts, then enter an infinite loop and never halt.

$M(w)$ halts iff $N(w)$ rejects $\Rightarrow$

Form D_M

$D_M(\epsilon)$ ~~accepts~~ halts iff $M(\ulcorner D_M \urcorner)$ ~~rejects~~ halts iff $N(\ulcorner D_M \urcorner)$ rejects

iff D_M does not halt.

(since N decides Halto)

Contradiction / QED.

Rice's Theorem

"Cannot decide what a program does by looking at its source code"

<u>Definition</u> A nontrivial property of c.e. sets is a collection P of c.e. sets such that $P \neq \emptyset$ and does not contain all c.e. sets

<u>Thm</u> Let P be as above. Then

$$X := \{ \ulcorner M \urcorner : L^e(M) \in P \}$$

is undecidable. Where $L^e(M)$ = set enumerated by M.

<u>Example</u> P is the set of c.e. sets R s.t.

(a) R is nonempty

(b) ϵ is in R

(or) (c) Every member of R is even.

(d) R is finite

<u>Proof</u> Suppose X is decidable by M

Let M_1, M_2 be s.t. $L^e(M_1) \in P$ and $L^e(M_2) \notin P$.

These exist since P is nontrivial

<u>Algorithm N</u>

<u>Input</u> w

Run $M(w)$

If $M(w)$ accepts,

Run M_2 - use its outputs

If $M(w)$ rejects,

Run M_1 - use its outputs.

<u>Form D_N</u> (Diagonal Theorem)

D_N enumerates $L^e(M_2)$ if $M(\ulcorner D_N \urcorner)$ accepts

So $L^e(D_N) \notin P$ if $M(\ulcorner D_N \urcorner)$ accepts

D_N enumerates $L^e(M_1)$ if $M(\ulcorner D_N \urcorner)$ reject

$L^e(D_N) \in P$ if $M(\ulcorner D_N \urcorner)$ rejects

Contradiction by choice of M.

Contradiction is since M decides X

So $M(\ulcorner N_2 \urcorner)$ accept iff $L^e(N_2) \in P$

So $M(\ulcorner D_N \urcorner)$ accepts iff $L^e(D_N) \in P$.

#

Can use a shorthand notation $M(N)$ to denote $M(\ulcorner N \urcorner)$

In PDF - Read about computably inseparable c.e. sets

Independent reading