The Halting Problem

DEFINITION : a register machine H decides the Halting Problem if, loading R1 with e and R2 with $[a_1, \ldots, a_n]$ (and all other registers with 0), the computation of H halts with R0 containing either 0 or 1 ; moreover R0 contains 1 when H halts if & only if the computation of the register machine program $Prog_e$ started with registers $R1, \ldots, Rn$ set to $a_1, \ldots, a_n$ (and all other registers set to 0) does halt.

THEOREM : $\boxed{\text{no such register machine H can exist.}}$

Proof :- suppose such an H exists and derive a contradiction ...

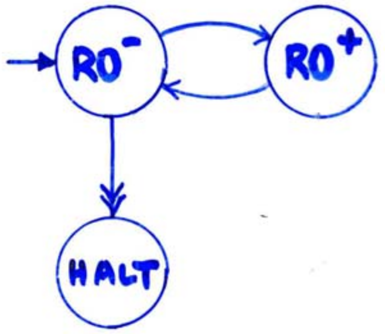
Let H' be obtained from H by replacing

START $\rightarrow$ by START $\rightarrow$ $\boxed{\text{copy } R1 \text{ to } Z}$ — $\boxed{\text{push } Z \text{ to } R2}$ $\rightarrow$

(where Z is a register not mentioned in H's program)

Let C be obtained from H' by replacing each HALT (& each jump to a label with no instruction)

by



Let $c \in \mathbb{N}$ be the index of C's program.

C started with $R1 = c$ eventually halts

iff

H' started with $R1 = c$ halts with $R0 = 0$

iff

H started with $R1 = c$ & $R2 = [c]$ halts with $R0 = 0$

iff

$Prog_c$ started with $R1 = c$ does not halt

iff

C started with $R1 = c$ does not halt

CONTRADICTION!

(to the assumption that such an H exists)

Recall:

DEFINITION:

$f \in Pfn(\mathbb{N}^n, \mathbb{N})$ is (register machine) computable if & only if there is a register machine M with at least $n+1$ registers, $R0, R1, R2, \ldots, Rn$ say, (and maybe some other registers as well) with the property that for all $(x_1, \ldots, x_n) \in \mathbb{N}^n$ and all $y \in \mathbb{N}$

$f(x_1, \ldots, x_n) = y$ if & only if the computation of M starting with $R1 = x_1, \ldots, Rn = x_n$, and all other registers $= 0$, halts with $R0 = y$.

Enumerating computable functions

For each $e \in \mathbb{N}$ let $\varphi_e \in Pfn(\mathbb{N}, \mathbb{N})$ be the partial function computed by $Prog_e$, i.e.

$\varphi_e(x) = y \overset{def}{\iff}$ the computation of $Prog_e$ started with $R1 = x$ (and all other registers zeroed) halts with $R0 = y$

Thus:

the function $e \mapsto \varphi_e$ maps $\mathbb{N}$ onto the collection of all computable partial functions from $\mathbb{N}$ to $\mathbb{N}$.

Not all partial functions are computable

Define $f \in Pfn(\mathbb{N}, \mathbb{N})$ by:

$$f(e) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \varphi_e(e)\uparrow \\ \text{undefined} & \text{if } \varphi_e(e)\downarrow \end{cases}$$

CLAIM: f is not computable.

PROOF: If f computable, then $f = \varphi_e$ for some e.

Then

- $\varphi_e(e)\uparrow \xRightarrow[f]{\text{def}^{\underline{n}}\text{ of}} f(e) = 0 \xRightarrow[e]{\text{def}^{\underline{n}}\text{ of}} \varphi_e(e) = 0 \Rightarrow \varphi_e(e)\downarrow$
- $\varphi_e(e)\downarrow \Longrightarrow f(e)\uparrow \Rightarrow \varphi_e(e)\uparrow$

contradiction!

(Un)decidable sets of numbers

A subset $S \subseteq \mathbb{N}$ is (register machine) decidable if & only if there is a register machine M with the property: for all $x \in \mathbb{N}$, M started with $R1 = x$ (and other registers zeroed) always halts with $R0$ containing either 0 or 1; moreover $R0 = 1$ when M halts if and only if $x \in S$.

Equivalently: S is decidable if & only if there is some e such that for all $x \in \mathbb{N}$

either $(\varphi_e(x) = 0 \ \& \ x \notin S)$ or $(\varphi_e(x) = 1 \ \& \ x \in S)$

S is called undecidable if no such M (or e) exists.

Some examples of undecidable sets of numbers

$S_1 \stackrel{\text{def}}{=} \{\langle e, a\rangle \mid \varphi_e(a)\downarrow\}$

i.e. one-argument version of Halting Problem

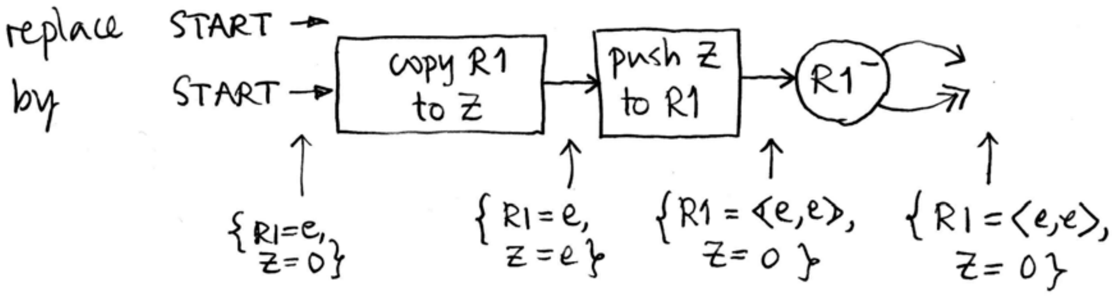
$S_2 \stackrel{\text{def}}{=} \{ e \mid \varphi_e(0)\downarrow\}$

i.e. $\nexists$ register machine to decide whether any program halts when supplied with input 0

$S_3 \stackrel{\text{def}}{=} \{ e \mid \varphi_e \text{ is a total function}\}$

i.e. $\nexists$ register machine to decide whether any program halts for <u>all</u> input data

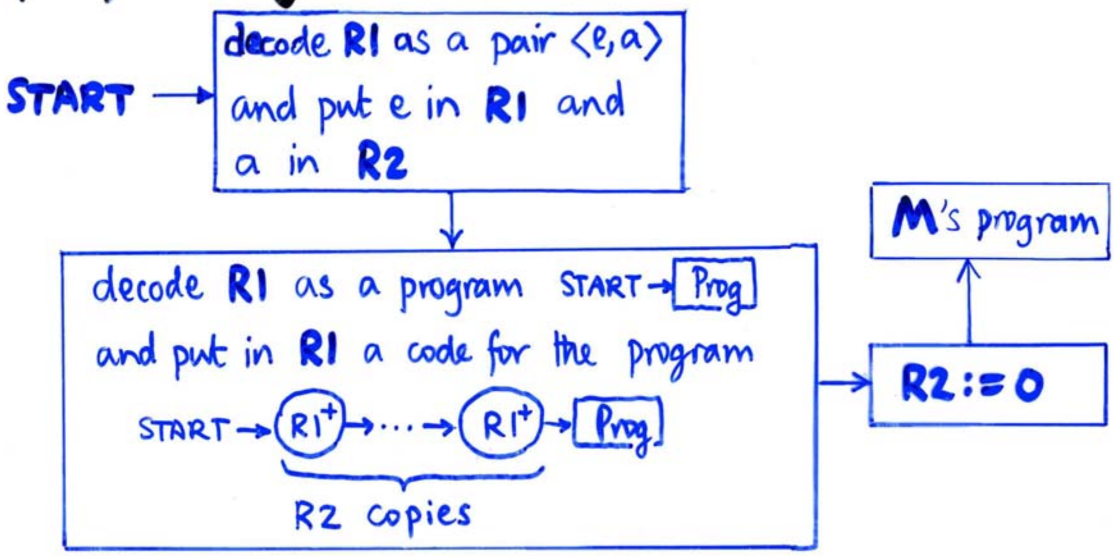
Ex.1. The proof that S_1 is undecidable is like the proof of the undecidability of the n-argument Halting Problem given above, except that now the modification of H to H' is:



(the rest of the argument is the same).

Ex.2. Undecidability of S_2 can be reduced to the undecidability of S_1:

If M were a register machine for deciding membership of S_2, then the register machine specified by



would decide membership of S_1. So no such M exists.

<u>Remark</u>. We can restate the proof of Ex.2 in terms of functions: it suffices to show that there is a function $f \in Fun(\mathbb{N}, \mathbb{N})$ satisfying

- f is computable
- for all $e, a \in \mathbb{N}$ $\quad \varphi_{f(\langle e,a \rangle)}(0) \equiv \varphi_e(a)$

meaning left hand side ↓ if & only if right hand side ↓ and in that case they are equal (see page 89)

and hence $\langle e,a \rangle \in S_1 \Leftrightarrow f(\langle e,a \rangle) \in S_2$.

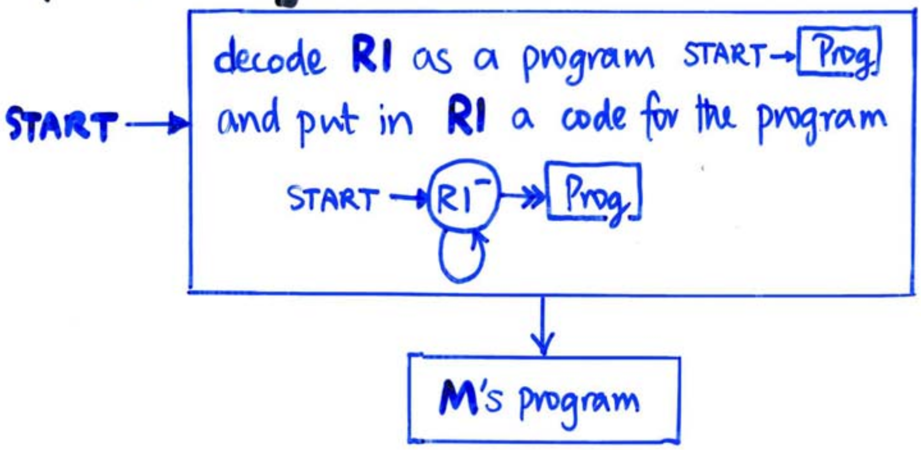
For in general we have for subsets $S_1, S_2 \subseteq \mathbb{N}$

$$\boxed{\begin{array}{l} S_2 \text{ decidable}, f \text{ computable } \& \ \forall x \in \mathbb{N}.\ x \in S_1 \Leftrightarrow f(x) \in S_2 \\ \Rightarrow S_1 \text{ decidable} \end{array}}$$

(Why?)

Ex.3. Undecidability of S_3 can be reduced to that of S_2:

If M were a register machine for deciding membership of S_3, then the register machine specified by



would decide membership of S_2. So no such M exists.