AMP

CST IB/II(G)/Dip. Computation Theory

2002, Paper 3 (10), question 6 (10)

(a) A register machine program is a finite list

$$\begin{aligned} L0 &: body_0 \\ L1 &: body_1 \\ &\vdots \\ Ln &: body_n \end{aligned}$$

$(n \geq 0)$ of label : body pairs, where each $body_i$ is of one of three forms

1 HALT or $R^+ \to Li$ or $R^- \to Li, Lj$

where R ranges over registers $R0, R1, R2, \ldots$

To (de)code these programs we make use of the following bijections:

1 $\langle -,- \rangle : \mathbb{N} \times \mathbb{N} \cong \mathbb{N}$

$$(x,y) \mapsto 2^x(2y+1) - 1$$

1 $\ulcorner - \urcorner : \mathbb{N}^* \cong \mathbb{N}$

$$\begin{cases} nil \longmapsto 0 \\ x :: xs \longmapsto \langle x, \ulcorner xs \urcorner \rangle + 1 \end{cases}$$

Thus each $e \in \mathbb{N}$ can be decoded, using the inverse bijection to $\ulcorner - \urcorner$, as a unique list of numbers $[x_0, x_1, \ldots, x_n]$; and each x_i can be decoded as a unique instruction body as follows:

3 $body(x) =$ if $x = 0$ then HALT

else $\{x > 0\}$ let $\langle y, z \rangle = x - 1$ in

if y even then let $\begin{cases} i = y/2 \\ j = z \end{cases}$ in

$$Ri^{+} \to Lj$$

else{y odd} let $\begin{cases} i = (y-1)/2 \\ \langle j, k \rangle = z \end{cases}$ in

$$Ri^{-} \to Lj, Lk$$

(6) We take $Prog_e = \{ L_0 : body(x_0), \ldots, L_n : body(x_n) \}$.

(b) A register machine H decides the halting problem if, on loading register R1 with a number
1 e and register R2 with the code $[a_1, \ldots, a_n]$ of a list of numbers (and setting all other registers to 0), the computation of H halts
1 with register R0 containing either 0 or 1; moreover R0 contains 1 on halting if & only if the computation of the register machine
2 with program $Prog_e$ (as in part (a)) started with registers $R1, \ldots, Rn$ set to $a_1, \ldots, a_n$ (and
(4) all others zeroed) does halt.

(c) Suppose H as in (b) exists & derive a
1 contradiction.

Let H' be derived from H by replacing

START → with

1 START → $\boxed{\text{copy R1 to Z}}$ → $\boxed{\text{push Z to R2}}$ →

where :

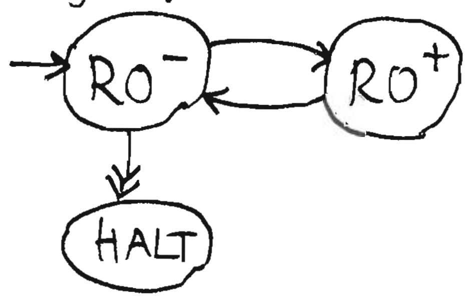
– Z is a fresh register not mentioned in H

1 – $\rightarrow\boxed{\begin{array}{c}\text{copy R1}\\ \text{to Z}\end{array}}\rightarrow$ is a register machine program carrying out the assignment $Z := R1$

1 – $\rightarrow\boxed{\begin{array}{c}\text{push Z}\\ \text{to R2}\end{array}}\rightarrow$ is a register machine program carrying out the assignments $\left\{\begin{array}{l} R2 := \langle Z, R2\rangle + 1 \\ \quad\;\; := 0 \end{array}\right.$

(where $\langle -,- \rangle$ is as in (a)).

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



1 whose effect is to HALT if $R0 = 0$ and to go into an infinite loop otherwise.

Then C started with $R1 = c$ eventually halts

iff H' started with $R1 = c$ eventually HALTS with $R0 = 0$

iff H started with $R1 = c$ & $R2 = [c] \; (= \langle 0, c \rangle + 1)$ eventually halts with $R0 = 0$

iff $Prog_c$ started with $R1 = c$ does not halt (by assumption on H)

4 iff C started with $R1 = c$ does not halt

10 — contradiction!

To which parts of the lecture course does this question refer?

(a) Bookwork from lecture 3

(b) Bookwork from lecture 5

(c) Same as part (b).