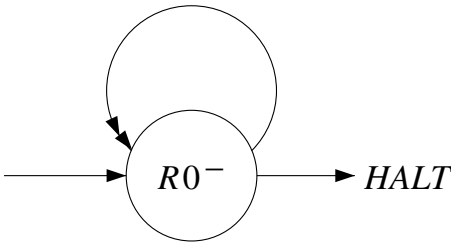
2009 Paper 6 Question 3

Computation Theory

(a) What is meant by a *state* (or *configuration*) of a register machine? [2 marks]

(b) A register machine program *Prog* is said to *loop at* $x \in \mathbb{N}$ if, when started with register $R1$ containing x and all other registers set to zero, the sequence of states *Prog* computes contains the same non-halted state at two different times.

(i) At which x does the following program loop?



[2 marks]

(ii) Show that if *Prog* loops at x, then the computation of *Prog* does not halt when started with register $R1$ containing x and all other registers set to zero. Is the converse true? [4 marks]

(iii) Consider the set $S = \{\langle e, x\rangle \mid Prog_e \text{ loops at } x\}$ of codes of pairs of numbers (e, x) such that the register machine program $Prog_e$ with index e loops at x. By adapting the usual proof of undecidability of the halting problem, or otherwise, show that S is an undecidable set of numbers. [Hint: if M were a register machine that decided membership of S, first consider replacing each *HALT* instruction (and each jump to a label with no instruction) with the program in part (i).] [12 marks]