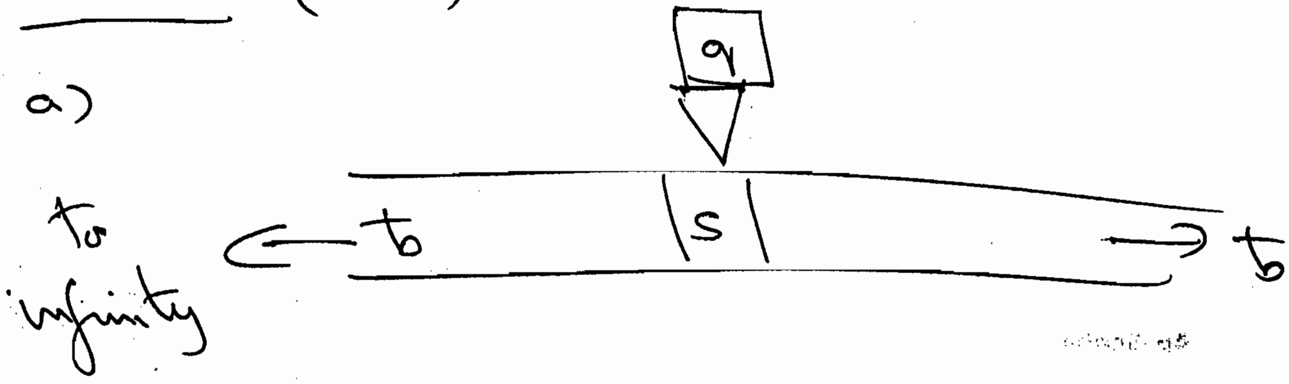
Solution (3, 10)

a)



A Turing machine is characterised by a finite state machine together with an infinite linear tape on which may be written the symbols of a finite alphabet S. Initially all but a finite number of the squares of the tape contain a special blank symbol. The machine is started in a specified initial state $q \in Q$, on a particular square of the tape. The action is deterministic and sequential; the process of computation depends solely on the current

Papers 3, 10 solution, ctd)

a) state $q \in Q$ and the symbol $s \in S$ on the current square: the machine has three deterministic response functions.

i) it enters a new state $Q(q,s)$;

ii) it overwrites the current symbol s with replacement symbol $R(q,s)$;

iii) it moves one square on the tape in direction $D(q,s)$ ($=$ L or R).

A possible value for the new state is the HALT state. The machine terminates, and the result of the computation is the final contents of the tape.

Papers 3, 10 Solution ctd)

b) A configuration of a Turing m/c records the current machine state $q \in Q$ together with a tape description. Typically we record the tape state by identifying the symbols $s \in S$ with digits $0, 1 \ldots (k-1)$ on scale k, ensuring that 0 represents b. We may now identify the tape state with a triple (s, m, n), where s is the current symbol value and integers m, n are the values of the left and right half-tapes in the natural representation in scale k.

c) To identify a Turing machine computation we must identify the quintuplet description

Papers 3, 10 Solution ctd)

and also the initial tape state. Assume given a pairing function $Z(x,y)$, say $[x,y]$. If we identify the states as say $0, 1, \ldots (n-1)$, we may let 0 be the initial state, 1 the unique HALTing state. Then each quintuplet may be written as (say) $[[q,s],[[q',s'],d]]$. The complete machine logic can be represented by a stack of quintuplets, coded by a single natural number e. The initial tape can be represented by $t = [s,[m,n]]$. Finally the pair code $c = [e,t]$ describes the complete computation.

Papers 3.10 Solution (ctd)

d) Suppose we could compute the maximum distance moved, say $l = l(c)$, as a function of the specification of the computation. The maximum number of possible configurations that the machine's tape could take is therefore bounded by k^{2l+1}, hence there is a total of at most $N . k^{2l+1} \times (2l+1)$ for the possible head positions! possible Turing machine configurations. Hence no halting computation could require more than $N . k^{2l+1}$ steps, since otherwise it must enter the same configuration twice, and therefore loop.

That would allow us to solve the HALTing problem by direct simulation.

hence $l(c)$ CANNOT be computable.