Problem Set 7

Noella James

04/24/2017

collaborators: none

Problem 8-1: A decideable halting problem

Reduction

Let's assume that P is represented by a Turing machine M that has i states. Thus, a transition for a symbol q in state k where $0 \leq k \leq i$ can be represented as
$(q, s_k) \rightarrow (q', s_{k'}, ACTION)$
where k' is another state and ACTION is one of left, right, or state

We do a reduction by multiplying the number of transitions and state by n where n is the length of the input

Thus, a state on the left hand side of the transition table would be converted as follows

$(q, s_k) \rightarrow (q, s_{k_1}) \ldots (q, s_{k_n})$

Thus we end up creating n transitions for each transition

Lets assume a transition that doesn't result in an ACCEPT state.

Let the original transition be.
$(q, s_k) \rightarrow (q', s_{k'}, ACTION)$

We transform this transition into n transitions as follows:

$(q, s_{k_0}) \rightarrow (q', s_{k'_1}, ACTION)$
$(q, s_{k_1}) \rightarrow (q', s_{k'_2}, ACTION)$
$(q, s_{k_j}) \rightarrow (q', s_{k'_{j+1}}, ACTION)$

$(q, s_{k_{n-1}}) \rightarrow (q', REJECT, n/a)$

Thus the transition rule is converted into one that REJECTS and halts after reading n symbols

Let's assume that the following transition goes into ACCEPT state
$(q, s_k) \rightarrow (q', ACCEPT, ACTION)$

We can convert the rule as
$(q, s_{k_0}) \rightarrow (q', s_{k_1}, ACTION)$
$(q, s_{k_1}) \rightarrow (q', s_{k_2}, ACTION)$
$(q, s_{k_j}) \rightarrow (q', s_{k_{j+1}}, ACTION)$
$(q, s_{k_{n-1}}) \rightarrow (q', ACCEPT, n/a)$

Thus the transition rule is converted into one that ACCEPTS and halts after reading n symbols.

Why this works:

In the proof provided that the "halting problem is undecidable", the turing machine did not have any limits on the number of transitions. That is the Turing machine is a universal Turing machine. By limiting the input to size n, the reduction reduces the number of transition the Turing machine can take. Thus, the Turing machine we're using post reduction has limited capabilities and the number of transitions it can take is limited to n. No such limit on the number of transitions exists in the original proof. Thus we do not break or go against the original proof. This is equivalent to basically stopping a program after it has run for a predetermined period of time.