

Lecture 27

Homework #27: 4.3.1 (a), 4.3.3

Misc: Let M be a Turing Machine with a two-way infinite tape. Define the relation \vdash formally, as was done for standard Turing Machines.

Today we consider extensions to the standard TM and will learn that these extensions do not add to the computational power of TMs.

Extension #1. Two-way infinite tape

$M = (K, \Sigma, \delta, s, H)$, all as before.

a configuration is: (q, wau)

w does not begin with $\#$
 u does not end with $\#$

note that this machine has no reason to consider the left-end-of-tape marker

Lemma. Let $M_1 = (K_1, \Sigma_1, \delta_1, s_1, H_1)$ be any TM with a two-way infinite tape.

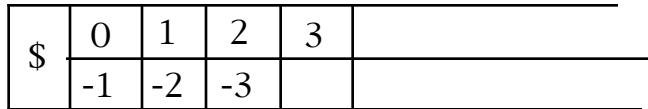
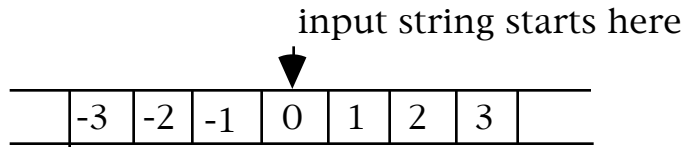
Then there is a standard TM $M_2 = (K_2, \Sigma_2, \delta_2, s_2, H_2)$ such that for any $w \in (\Sigma_1 - \{\#\})^*$, the following hold:

a) if M_1 halts on w , then so does M_2

and if $(s_1, \#w) \vdash^* (h, uav)$ [in M_1]
then $(s_2, >\#w) \vdash^* (h, >\#uav)$ [in M_2]

b) if M_1 does not halt on w , then M_2 does not halt on w .

Proof. "Fold" M_1 's tape and treat it like the tape of a standard TM.



How to carry out the idea:

- the alphabet of M_2 will contain every ordered pair (a,b) of symbols of Σ .

a = the symbol on the upper track
 b = the symbol on the lower track

- we're also going to include some other symbols:

let $\bar{\Sigma}_1 = \{\bar{a} : a \in \Sigma_1\}$

\$, the "turnaround" marker (different from $>$)

So $\Sigma_2 = \{\$\} \cup \Sigma_1 \cup \bar{\Sigma}_1 \cup (\Sigma_1 \times \Sigma_1) \cup (\Sigma_1 \times \bar{\Sigma}_1) \cup (\bar{\Sigma}_1 \times \Sigma_1)$

Operation of M_2 :

1. Set up the tape; remember, it is starting out in the "usual" configuration, which is single track.
2. Simulate M_1 on the "divided" tape.
3. If M_1 would halt, M_2 will halt, but only after it returns the tape to standard output configuration.

Step I. Need a subroutine TM to set up the tape. That is, we need to make it look like a double track tape.

[Note that we're assuming that there are no # in the input.]

Step II. Main part of the simulation of M_1 .

When simulating M_1 , M_2 needs to be able to distinguish between working on the upper track of the tape and working on the lower track.

We handle this in M_2 by having two states for every one state in M_1 .

For each state q in M_1 , M_2 will have states $\langle q, 1 \rangle$ and $\langle q, 2 \rangle$.

$\langle q, 1 \rangle$ means that the simulation is working on the upper track;

$\langle q, 2 \rangle$ means that the simulation is working on the lower track.

M_2 also needs states $\langle h, 1 \rangle$ and $\langle h, 2 \rangle$ for every $h \in H_1$. These will not act as real halt states. Instead, they will behave like “temporary” halt states.

We need to remember that when the simulation “halts”, M_2 will still have its tape formatted as if it is two-track. Before we can really halt, we need to return it to a one-track format. This is why $\langle h, 1 \rangle$ and $\langle h, 2 \rangle$ are only “temporary” halt states.

When M_2 enters one of these states, it needs to do one more important thing. It marks the symbol currently being scanned. We don’t want to lose track of this symbol during the process of converting to one-track format.

So:

$$K_2 = K_1 \times \{1, 2\}$$

$$s_2 = \langle s_1, 1 \rangle$$

δ_2 is defined as follows:

Category 1 transitions – those that simulate M_1 on the upper track

For all a_2 ,

$$\begin{aligned} \delta_2 (\langle q, 1 \rangle, (a_1, a_2)) = & (\langle p, 1 \rangle, \leftarrow), \text{ if } \delta_1 (q, a_1) = (p, \leftarrow) \\ & (\langle p, 1 \rangle, \rightarrow), \text{ if } \delta_1 (q, a_1) = (p, \rightarrow) \\ & (\langle p, 1 \rangle, \langle b, a_2 \rangle), \text{ if } \delta_1 (q, a_1) = (p, b) \end{aligned}$$

Category 2 transitions – those that simulate M_1 on the lower track

For all a_1 ,

$$\begin{aligned} \delta_2 (\langle q, 2 \rangle, (a_1, a_2)) = & (\langle p, 2 \rangle, \leftarrow), \text{ if } \delta_1 (q, a_2) = (p, \rightarrow) \\ & (\langle p, 2 \rangle, \rightarrow), \text{ if } \delta_1 (q, a_2) = (p, \leftarrow) \end{aligned}$$

$(\langle p, 2 \rangle, \langle a_1, b \rangle)$, if $\delta_1 (q, a_2) = (p, b)$

Category 3 transitions – those that change tracks

$$\delta_2 (\langle q, 1 \rangle, \$) = (\langle q, 2 \rangle, \rightarrow)$$

$$\delta_2 (\langle q, 2 \rangle, \$) = (\langle q, 1 \rangle, \rightarrow)$$

Category 4 transitions – those that change single-track blanks to double-track blanks

$$\delta_2 (\langle q, 1 \rangle, \#) = (\langle q, 1 \rangle, (\#, \#))$$

$$\delta_2 (\langle q, 2 \rangle, \#) = (\langle q, 2 \rangle, (\#, \#))$$

Category 5 transitions – those that “halt” and record the position of the tape head

$$\delta_2 (\langle h, 1 \rangle, (a_1, a_2)) = (\langle h, 1 \rangle, (\bar{a}_1, a_2))$$

$$\delta_2 (\langle h, 2 \rangle, (a_1, a_2)) = (\langle h, 2 \rangle, (a_1, \bar{a}_2))$$

Step III. Once the simulation is complete, we need to transform the tape back into a one-track representation. This is somewhat tricky, but not too bad. Details are left to the reader.

Theorem. Any function that is computed or language that is decided or semi-decided by a Turing Machine with a two-way infinite tape is also computed, decided, or semi-decided by a standard Turing Machine.