Bram Schuijff
CSCI 387, §S01/02
Homework 3
February 27th, 2025 by 10:30am

1. (a) Let M_0 be the name of our Turing machine. We give a description of its behavior below:

M_0 = "On input string w

1. Scan from left to right until an unmarked 1 is found and mark it. If a blank is encountered before a 1, go to step 5.

2. Scan from left to right until an unmarked 0 is found and mark it. If no unmarked 0s are found, reject.

3. Continuing scanning until another unmarked 0 is found and mark it. If no unmarked 0s are found, reject.

4. Return the head to the start of the tape. Go to step 1.

5. Scan from left to right. If an unmarked 0 is found, reject. Otherwise, accept."

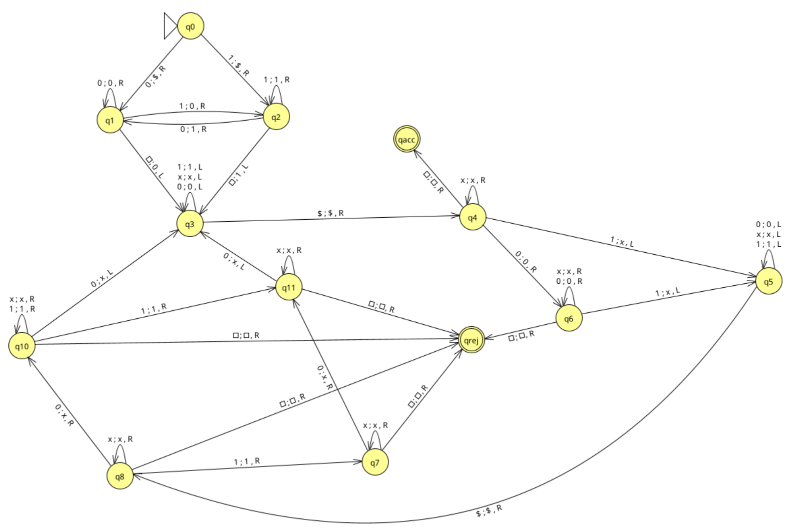

Figure 1. Turing Machine for M_0

(b)

(c) The computation history for the string $w = 01011$ is:

$$q_0 01011$$
$$\$q_1 1011$$
$$\$0q_2 011$$
$$\$01q_1 11$$
$$\$010q_2 11$$
$$\$0101q_2 1$$
$$\$01011q_2$$
$$\$0101q_3 1$$
$$\$010q_3 11$$
$$\$01q_3 011$$
$$\$0q_3 1011$$
$$\$q_3 01011$$
$$q_3\$01011$$
$$\$q_4 01011$$
$$\$0q_6 1011$$
$$\$0q_5 x011$$
$$\$q_5 0x011$$
$$q_5\$0x011$$
$$\$q_8 0x011$$
$$\$xq_{10}x011$$
$$\$xxq_{10}011$$
$$\$xxxq_3 11$$
$$\$xxq_3 x11$$
$$\$xq_3 xx11$$
$$\$q_3 xxx11$$
$$q_3\$xxx11$$
$$\$q_4 xxx11$$
$$\$xq_4 xx11$$
$$\$xxq_4 x11$$
$$\$xxxq_4 11$$
$$\$xxq_5 xx1$$
$$\$xq_5 xxx1$$
$$\$q_5 xxxx1$$
$$q_5\$xxxx1$$
$$\$q_8 xxxx1$$
$$\$xq_8 xxx1$$
$$\$xxq_8 xx1$$
$$\$xxxq_8 x1$$
$$\$xxxxq_8 1$$
$$\$xxxx1q_7$$
$$\$xxxx1q_{rej}$$

2. We demonstrated in class—and in Theorem 3.10 in the textbook—that for any positive integer $k \in \mathbb{Z}_{\geqslant 1}$, the k-headed Turing machine (i.e. the Turing machine with k tapes that can move independently) is just as powerful as the ordinary Turing machine. Let us take $k = 5$, F being the 5-in-1 Turing machine and S being the 5-headed Turing machine. We describe how to interpret a string with the 5-headed Turing machine in an equivalent manner to the 5-in-1 Turing machine:

   "On input string $w = w_1 ... w_n$:

   1. Whenever F writes symbols under its head, S writes the first symbol to its first tape, the second symbol to its second tape, and so on for each of the 5 tapes.
   2. Whenever F moves its head, S moves the heads of every one of its tape in the same manner.
   3. Whenever F reads a combination of 5 symbols, S will then clearly read all of the same 5 symbols in the same order because each head of S has been kept concurrent with the head of F."

   Therefore, the 5-in-1 Turing machine is just as powerful as the 5-headed Turing machine which is just as powerful as the ordinary Turing machine. Therefore, the 5-in-1 Turing machine has no more power.

3. Let $w_1 \in L_1, w_2 \in L_2$ be two strings from Turing-recognizable languages L_1, L_2. We want to show that $w_1 w_2 \in L_1 L_2$ is also Turing-recognizable. Let M_1, M_2 be Turing machines that recognize L_1, L_2 respectively. We demonstrate that the nondeterministic 2-headed Turing machine $M_1 M_2$ where the first tape is controlled by M_1 and the second tape is controlled by M_2 recognizes $L_1 L_2$. We assume that our input tape is the first tape and that the second tape starts blank. We construct $M_1 M_2$ as follows:

   "For an input string $w = w_1 w_2$

   1. Scan through the input tape until we read a blank.
   2. Nondeterministically select exactly one non-blank symbol, placing a mark on it.
   3. Copy all of the non-blank symbols past—but not including—the marked symbol onto the second tape, erasing them from the first tape as they are read.
   4. Remove the mark from the marked symbol and place the heads of both tapes at the beginning.
   5. Run M_1 and M_2 on the contents of the two tapes, rejecting if either of them rejects and accepting if and only if they both accept."

   This will nondeterministically select a delineating point between w_1 and w_2 and accept if and only if it gets it right, splitting $w_1 w_2$ into its two components and writing them to the separate tapes.

4. Construct a DFA D_1 such that $L(D_1) = \{w \mid w \text{ contains more than three 0s}\}$. This can be given by the equivalent regular expression $\Sigma^* 0 \Sigma^* 0 \Sigma^* 0 \Sigma^* 0 \Sigma^*$ and is therefore a regular language and has a DFA. We know from footnote 3 on page 46 of the textbook that regular languages are closed under intersection, hence there exists an algorithm to construct the DFA for the intersection of two regular languages—we might've also talked about this in class, I don't remember. For a pair of DFAs X, Y, let $X \cap Y$ be the DFA that accepts a string w if and only if $w \in L(X) \cap L(Y)$. We then construct a new set B such that

$B = \{\langle M \cap D_1 \rangle \mid \langle M \rangle \in A\}$. Because D_1 only accepts strings with more than three 0s and M does not accept any string containing more than three 0s, we can test M's membership in A by ensuring that $L(M \cap D_1) = \varnothing$. By Theorem 4.4 in the textbook, we construct a Turing machine that accepts M if $\langle M \cap D_1 \rangle \in E_{\mathrm{DFA}}$ and rejects otherwise. Thus, A is decidable.