

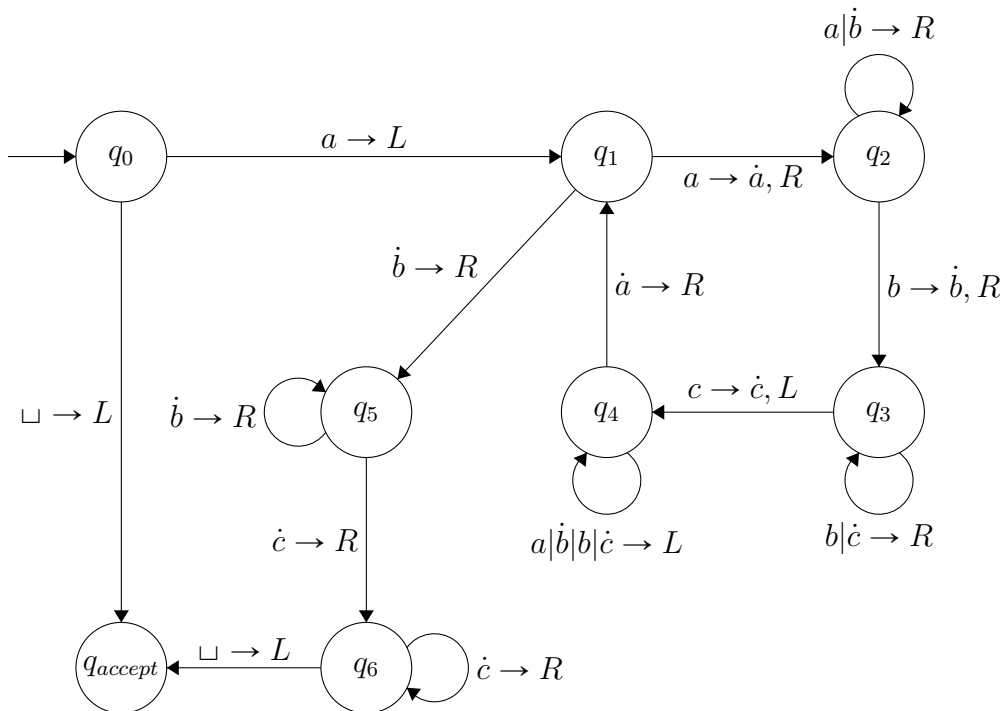
# CSC 341 - Fall 2022

## Problem Set 6 Solutions

**Problem 1.** Draw a TM  $M$  which decides the language  $A = \{a^n b^n c^n \mid n \geq 0\}$ , where the input alphabet  $\Sigma = \{a, b, c\}$ .

**Solution 1.** The principle of this Turing machine is to sweep right while checking for an  $a$ , a  $b$ , and a  $c$  in that order without breaking the formatting. We mark the ones we find until we run out of one type. When we do, if we did not run out of  $a$ 's first, then we reject. If we did run out of  $a$ 's first, we need only check that we're also out of  $b$ 's and  $c$ 's.

For the sake of clarity, all missing transitions in the Turing machine below go to the rejecting state.



**Problem 2.** Give an algorithm for a TM  $M$  which decides the language

$$C = \{\langle D, \omega \rangle \mid D \text{ is a DFA and } D \text{ accepts } \omega\}.$$

Note that  $\langle D, \omega \rangle$  is some reversible encoding of  $D$  and  $\omega$  into a string. The precise details of how this encoding is computed is irrelevant to the algorithm you give. The key point is you have access to each parameter.

**Solution 2.** Since a DFA is not entirely compatible with Turing machines (it doesn't know its own input, for example), we need to give a little more information than just use

the universal Turing machine  $U$  to simulate  $D$  on  $\omega$ . We will give a slightly more verbose answer than is strictly necessary for the sake of completeness.

Define the Turing machine  $M$  to be

$M =$  On input  $\nu$

1. If  $\nu \neq \langle D, \omega \rangle$ , then reject
2. Let  $D = (Q, \Sigma, \delta, q_0, F)$
3. Let  $\omega = \omega_1 \dots \omega_{|\omega|}$
4. Let  $q = q_0$
5. For  $i = 1$  to  $|\omega|$ 
  - (a) Compute  $\delta(q, \omega_i) = q'$
  - (b) Assign  $q = q'$
6. Accept if  $q \in F$  and reject otherwise

Each step of  $M$  is computable and is guaranteed to complete, so  $M$  halts. Moreover,  $M$  exactly simulates  $D$  on  $\omega$ , so  $L(M) = C$ .

**Problem 3.** To prove that enumerators are equivalent to Turing machines, we created an enumerator  $E$  given a TM  $M$  such that  $L(E) = L(M)$ . The algorithm for this is given below.

$E =$  "Ignore input,

1. For  $i = 1$  to  $\infty$ 
  - (a) For  $j = 1$  to  $i$ 
    - i. Run  $M$  on input  $\omega_j$  for  $i$  steps ( $\omega_j$  is the  $j^{\text{th}}$  element of  $\Sigma^*$ )
    - ii. If  $M$  accepts, print  $\omega_j$ "

Explain why the following algorithm for  $E$  does not work.

$E =$  "Ignore input,

1. For  $i = 1$  to  $\infty$ 
  - (a) Run  $M$  on input  $\omega_i$  ( $\omega_i$  is the  $i^{\text{th}}$  element of  $\Sigma^*$ )
  - (b) If  $M$  accepts, print  $\omega_i$ "

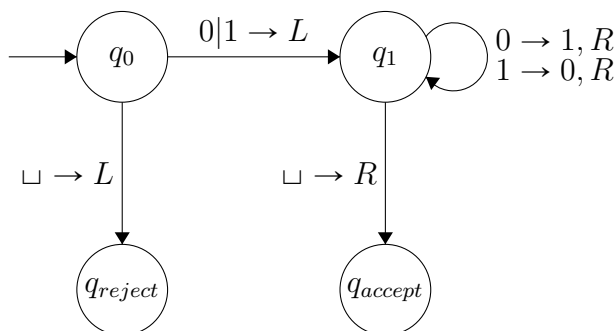
**Solution 3.** The trouble with the second algorithm is that  $M$  is not guaranteed to halt on any particular string. Suppose  $M$  loops forever on string  $\omega_i$  but there is some  $\omega_j$  with  $j > i$  that  $M$  accepts (it is trivial to show such an  $M$  exists). Then  $E$  will never output  $\omega_j$ .

**Problem 4.** For a string  $\omega \in \{0, 1\}^+$ , the *one's compliment* of  $\omega$ , written  $\bar{\omega}$ , is obtained by inverting all of its bits (flipping 0 to 1 and 1 to 0).

Draw a TM which, on input  $\omega \in \{0, 1\}^+$ , reaches some accepting configuration with exactly  $\bar{\omega}$  on the tape. If  $M$ 's input is  $\epsilon$ , it should enter a rejecting configuration with nothing on the tape.

In other words, give a TM which converts its input  $\omega$  into  $\bar{\omega}$  and accepts (or rejects if  $\omega = \epsilon$ ). The head may point anywhere on the tape when it halts.

**Solution 4.** As there's no requirement for the head's position upon termination, for this TM, we need only greedily change 0's into 1's and vice versa.



**Problem 5.** Draw a TM  $M$  which, on input  $\omega \in \{0, 1\}^+$ , reaches the accepting configuration  $q_a\bar{\omega}$ . If  $M$ 's input is  $\epsilon$ , it should enter the rejecting configuration  $q_r\epsilon$ .

In other words,  $M$  converts its input  $\omega$  into  $\bar{\omega}$  and accepts with the head pointing to the first cell (i.e. the head is as far left as possible). If  $M$ 's input is  $\epsilon$ , it instead rejects with the head pointing to the first cell on the tape (a blank cell).

**Solution 5.** This solution is the same as the prior problem's, except we must reset the head to the first position after we finish. This can easily be achieved by marking the first cell position to rewind to.

