

CMSC 28100-1 / MATH 28100-1  
 Introduction to Complexity Theory  
 Fall 2017 – Homework 3  
 Solution

October 20, 2017

**Exercise 1** (HMU 8.2.2). Design Turing machines for the following languages.

- (a) The set of strings with an equal number of 0's and 1's.
- (b)  $\{a^n b^n c^n : n \geq 1\}$ .
- (c)  $\{w w^R : w \text{ is any string of 0's and 1's}\}$ , where  $w^R$  is the reverse of a string. For instance, we have  $10010^R = 01001$ .

*Solution.* For item (a), consider the Turing machine  $M$  with transition function  $\delta$  given by

	0	1	$B$	*
$q_0$	$(q_1, *, R)$	$(q_2, *, R)$	$(q_f, B, L)$	$(q_0, *, R)$
$q_1$	$(q_1, 0, R)$	$(q_3, *, L)$	–	$(q_1, *, R)$
$q_2$	$(q_3, *, L)$	$(q_2, 1, R)$	–	$(q_2, *, R)$
$q_3$	$(q_3, 0, L)$	$(q_3, 1, L)$	$(q_0, B, R)$	$(q_3, *, L)$
$q_f$	–	–	–	–

where  $q_0$  is the initial state and  $q_f$  is the accepting state.

Let us prove that  $M$  decides the language  $L$  of strings with an equal number of 0's and 1's. First, we clearly have that the empty string  $\epsilon$  is in  $L(M)$ .

Fix now an input  $x \in \{0, 1\}^*$  of length  $n > 0$  and consider the computation of  $M$  on input  $x$ .

First note that  $M$  only changes symbols in the tape from 0 to \* or from 1 to \*, and these happen only in states  $q_0$ ,  $q_1$  and  $q_2$ . This in particular implies that the tape of  $M$  during the computation for  $x$  is always a string of length  $n$  in  $\{0, 1, *\}^n$ .

Let us enumerate these non-blank positions in the tape  $1, \dots, n$  and let 0 and  $n + 1$  be the positions immediately before and after these respectively.

Note now that if  $M$  is in one of the state  $q_0$ ,  $q_1$  or  $q_2$ , the previous movement must have been  $R$ . On the other hand, if  $M$  is in state  $q_3$ , the previous movement must have been  $L$ .

For  $i \in \{0, 1, 2, 3\}$ , let  $P_i$  be the set of all positions of the head of  $M$  when it is in state  $q_0$ .

We now claim that

$$\begin{aligned} P_0 &\subseteq \{1, \dots, n + 1\}; & P_1 &\subseteq \{1, \dots, n + 1\}; \\ P_2 &\subseteq \{1, \dots, n + 1\}; & P_3 &\subseteq \{0, \dots, n\}. \end{aligned} \tag{1}$$

Suppose not and consider the first time in which the head of  $M$  violates the above (this is clearly not before the first step).

If  $M$  is in state  $q_0$ ,  $q_1$  or  $q_2$ , then the previous movement must have been  $R$ , so the current position must be  $n + 2$  (as this is the first time of violation of the property). This means that previous symbol read was the blank symbol  $B$  at position  $n + 1$ , so the previous state must have been  $q_3$ . But this contradicts the choice of the first moment when the property was violated as the position in  $q_3$  was  $n + 1$  in the previous step.

Suppose then that  $M$  is in state  $q_3$  in the first moment of violation. Then the previous movement must have been  $L$  and the previous state must have been one of  $q_1$ ,  $q_2$  or  $q_3$ . If it was one of the first two, we are done by the choice of the first moment of violation, so the previous state must have been  $q_3$ . However, this means that the previous symbol read must be non-blank, so the previous position must have been in  $\{1, \dots, n\}$ , which implies that the current position is in  $\{0, \dots, n\}$ , a contradiction.

Therefore (1) holds.

We now claim that  $M$  never keeps looping in the same state. Indeed, for  $M$  to remain in one of the states  $q_0$ ,  $q_1$  or  $q_2$ , it must keep reading non-blank symbols and keep moving right, so it must eventually reach a blank symbol, which would make it leave these states. On the other hand, for  $M$  to remain in  $q_3$ , it must keep reading non-blank symbols and keep moving left, so it must eventually reach a blank symbol, which would make it leave  $q_3$ .

Note now that every time  $M$  leaves one of the states  $q_0$ ,  $q_1$  or  $q_2$ , one 0 or 1 gets replaced with  $*$ . This means that either eventually  $M$  halts or all positions  $1, \dots, n$  become  $*$ . If the latter happens in any of  $q_0$ ,  $q_1$  or  $q_2$ , then  $M$  moves the head to the right until it reaches a blank symbol, then halts. If this happens in  $q_3$ , then  $M$  moves the head to the left until it reaches a blank symbol, then transitions to  $q_0$  and by the above it will halt later.

Therefore  $M$  always halts.

We claim now that when  $M$  transitions into  $q_0$  from a different state, then the head must be in position 1. Indeed, the only state other than  $q_0$  that transitions to  $q_0$  is  $q_3$ , and it can only do so when it reads  $B$  and since  $P_3 \subseteq \{0, \dots, n\}$ , it follows that the blank read must be of position 0 and  $M$  then goes  $q_0$  in position 1.

We claim now that when  $M$  transitions into  $q_1$  or  $q_2$  from a different state all non-blank symbols to the left of the head are  $*$ . Indeed, for  $i \in \{1, 2\}$ , the only state that transitions to  $q_i$  that is not  $q_i$  is  $q_0$ . Consider the last time before the current time in which  $M$  transitioned to  $q_0$  from another state. We know that the head must have been in position 1, but inspecting the transitions of  $q_0$ , we see that  $M$  will remain in  $q_0$  and moving to the right while it reads  $*$  and in the first occurrence of a non- $*$  symbol, it will transition to  $q_i$ , hence the claim follows.

Let  $|w|_a$  be the number of occurrences of the symbol  $a$  in the string  $w$  and let  $d(w) = |w|_1 - |w|_0$ .

We claim now the following.

- If  $M$  is in state  $q_0$  with  $w$  in its tape, then  $d(w) = d(x)$ .
- If  $M$  is in state  $q_1$  with  $w$  in its tape, then  $d(w) = d(x) + 1$ .
- If  $M$  is in state  $q_2$  with  $w$  in its tape, then  $d(w) = d(x) - 1$ .
- If  $M$  is in state  $q_3$  with  $w$  in its tape, then  $d(w) = d(x)$ .

Suppose not and consider the first time when the above is violated (this is clearly not before the first step). Since when  $M$  stays in the same state it does not replace any symbol, it follows that the previous state must have been different than the current one. But by our choice of the first time of violation, in the previous step the above was true and by inspecting the possible transitions that change state, we conclude that the above must hold for the current state (it is just a boring case analysis).

Let us now prove that if  $M$  accepts  $x$ , then  $x \in L$ . Note that for  $M$  to accept  $x$ , we must read  $B$  in  $q_0$ . Let  $w$  be the content of the tape at this moment.

The last time in which  $M$  transitioned into  $q_0$  from another state, we know that it was in position 1 and from the transitions of  $q_0$ , we know that  $M$  will scan all symbols of  $w$  and they must all be  $*$ , which implies

$$0 = d(w) = d(x),$$

hence  $x \in L$ .

Let us finally prove that if  $M$  rejects  $x$ , then  $x \notin L$ . Note that for  $M$  to reject  $x$ , we must read  $B$  in  $q_i$  for some  $i \in \{1, 2\}$ . Let  $w$  be the content of the tape at this moment.

The last time in which  $M$  transitioned into  $q_i$  from another state, we know that all symbols to the left of the head were  $*$  and from the transitions of  $q_i$ , it follows that  $M$  must read all other symbols in the tape, and they are all either  $i - 1$  or  $*$ , which means that we have

$$\begin{aligned} 0 &\leq |w|_{i-1} - |w|_{2-i} = (-1)^i d(w) = (-1)^i (d(x) - (-1)^i) \\ &= (-1)^i d(x) - 1, \end{aligned}$$

which implies  $d(x) \neq 0$ , hence  $x \notin L$ .

Therefore, since  $M$  always halts, it follows that  $M$  decides  $L$ .

For item (b), consider the Turing machine with transition function  $\delta$  given by

	$a$	$b$	$c$	$B$	$*$	$\#$	$\%$
$q_0$	$(q_1, *, R)$	—	—	—	—	$(q_6, \#, R)$	—
$q_1$	$(q_1, a, R)$	$(q_3, \#, R)$	—	—	—	$(q_2, \#, R)$	—
$q_2$	—	$(q_3, \#, R)$	—	—	—	$(q_2, \#, R)$	—
$q_3$	—	$(q_3, b, R)$	$(q_5, \%, L)$	—	—	—	—
$q_4$	—	—	$(q_5, \%, L)$	—	—	—	$(q_4, \%, R)$
$q_5$	$(q_5, a, L)$	$(q_5, b, L)$	$(q_5, c, L)$	—	$(q_0, *, R)$	$(q_5, \#, L)$	$(q_5, \%, L)$
$q_6$	—	—	—	$(q_f, B, R)$	$(q_6, *, R)$	$(q_6, \#, R)$	$(q_6, \%, R)$
$q_f$	—	—	—	—	—	—	—

where  $q_0$  is the initial state and  $q_f$  is the accepting state.

Let us prove that  $M$  decides the language  $L = \{a^k b^k c^k : k \geq 1\}$ .

First, it is clear that  $M$  rejects the empty string  $\epsilon$ .

Fix now an input  $x \in \{a, b, c\}^*$  of length  $n > 0$  and consider the computation of  $M$  on input  $x$ .

As in item (a), note that  $M$  only changes symbols from  $a, b$  or  $c$  to  $*$ ,  $\#$  or  $\%$ , which implies that the tape of  $M$  is always a string of length  $n$  in  $\{a, b, c, *, \#, \%\}^n$ , and as before, let us enumerate these positions as  $1, \dots, n$  and let  $0$  and  $n + 1$  be the positions immediately before and after respectively.

Note also that  $*$  can only replace an  $a$  and  $a$  can only be replaced by  $*$ . The same also holds for the pairs  $(\#, b)$  and  $(\%, c)$ . This means that at any time if the content of the tape is  $w$ , then we have

$$|x|_a = |w|_a + |w|_*; \quad |x|_b = |w|_b + |w|_{\#}; \quad |x|_c = |w|_c + |w|_{\%}. \quad (2)$$

We now claim that the following holds.

- If  $M$  is in state  $q_0$ , then the content to the left of the head is  $*^k$  for some  $k \geq 0$ .
- If  $M$  is in state  $q_1$ , then the content to the left of the head is  $*^{k_1} a^{k_2}$  for some  $k_1, k_2 \geq 0$ .

- If  $M$  is in state  $q_2$ , then the content to the left of the head is  $*^{k_1}a^{k_2}\#^{k_3}$  for some  $k_1, k_2, k_3 \geq 0$ .
- If  $M$  is in state  $q_3$ , then the content to the left of the head of  $M$  is  $*^{k_1}a^{k_2}\#^{k_3}b^{k_4}$  for some  $k_1, k_2, k_3, k_4 \geq 0$ .
- If  $M$  is in state  $q_4$ , then the content to the left of the head is  $*^{k_1}a^{k_2}\#^{k_3}b^{k_4}\%^{k_5}$  for some  $k_1, k_2, k_3, k_4, k_5 \geq 0$ .
- If  $M$  is in state  $q_5$ , then the content to the left of the head is  $*^{k_1}a^{k_2}\#^{k_3}b^{k_4}\%^{k_5}$  for some  $k_1, k_2, k_3, k_4, k_5 \geq 0$  such that  $k_i = 0$  implies  $k_{i+1} = 0$  for all  $i \in \{1, \dots, 4\}$ .
- If  $M$  is in state  $q_6$ , then the content to the left of the head is  $*^{k_1}\#^{k_2}\%^{k_3}$  for some  $k_1, k_2, k_3 \geq 0$  such that  $k_i = 0$  implies  $k_{i+1} = 0$  for all  $i \in \{1, 2\}$ .

Suppose not and consider the first time  $t$  in which the above is violated. By inspecting the table of transitions, one can conclude (after a boring case analysis) that time  $t - 1$  must have also violated the property.

With a proof analogous to the one in item (a), we get also that  $M$  cannot loop in the same state, which implies that  $M$  eventually halts by again a proof analogous to the one in item (a).

We claim now that whenever  $M$  is in  $q_0$  and the content of the tape is  $w$ , we have

$$|w|_* = |w|_{\#} = |w|_{\%}. \quad (3)$$

This is indeed true in the beginning and assuming inductively it was true the last time  $M$  was in  $q_0$ , it is easy to see that the next states until the occurrence of  $q_0$  must have been a sequence of  $q_1^{k_1}q_2^{k_2}q_3^{k_3}q_4^{k_4}q_5^{k_5}$  for  $k_1, k_2, k_3, k_4, k_5 \geq 0$  and inspecting the transition table, it follows that exactly one new of each  $*$ ,  $\#$  and  $\%$  must have been written, so the property remains true.

Finally, note that this means that if  $M$  is in state  $q_6$  and the content of the tape is  $w$ , then we must have (3) as well. Since when in  $q_6$  the machine  $M$  keeps reading symbols to the right and the part to left of the head always contains  $*^{k_1}\#^{k_2}\%^{k_3}$  for some  $k_1, k_2, k_3 \geq 0$ , it follows that  $M$  accepts  $x$  if and only if (3) and

$$|w|_a = |w|_b = |w|_c = 0.$$

Putting this together with (2), it follows that  $M$  accepts  $x$  if and only if  $x \in L$ , hence  $M$  decides  $L$  (as it always halts).

For item (c), consider the Turing machine with transition function  $\delta$  given by

	0	1	$B$
$q_0$	$(q_1, B, R)$	$(q_3, B, R)$	$(q_f, B, R)$
$q_1$	$(q_1, 0, R)$	$(q_1, 1, R)$	$(q_2, B, L)$
$q_2$	$(q_5, B, L)$	—	—
$q_3$	$(q_3, 0, R)$	$(q_3, 1, R)$	$(q_4, B, L)$
$q_4$	—	$(q_5, B, L)$	—
$q_5$	$(q_5, 0, L)$	$(q_5, 1, L)$	$(q_0, B, R)$
$q_f$	—	—	—

where  $q_0$  is the initial state and  $q_f$  is the accepting state.

Let us prove that  $M$  decides the language  $L = \{ww^R : w \in \{0, 1\}^*\}$ .

To prove this, we will first prove that for every  $x \in \{0, 1\}^*$ , we have  $x \in L$  if and only if one of the following holds.

- We have  $x = \epsilon$ .
- We have  $x = 0y0$  for  $y \in L$ .
- We have  $x = 1y1$  for  $y \in L$ .

The proof is by induction in the length  $n$  of  $x$ . For  $n = 0$ , this is trivial. Suppose then that  $n > 0$  and that the result holds for strings of smaller lengths.

If  $x \in L$ , then we have  $x = ww^R$ . Writing  $w = \sigma z$  for  $\sigma \in \{0, 1\}$ , we have  $x = \sigma z z^R \sigma$  and setting  $y = z z^R$ , we have  $y \in L$ . On the other hand, if  $x = \sigma y \sigma$  for some  $\sigma \in \{0, 1\}$  and  $y \in L$ , then we can write  $y = ww^R$ , which gives  $x = (\sigma w)(\sigma w)^R$ , hence  $x \in L$ .

By inspecting the transition table of  $M$ , we see that  $M$  can only erase symbols (i.e., replace them by blanks).

We claim that  $M$  can only erase symbols at the ends of the tape (i.e., the tape always contains a word in  $\{0, 1\}^*$  with infinitely many blanks to both sides, but *not* in the middle). Suppose not and consider the first moment when a symbol not at the end of the tape is erased. But this must have been erased in one of the states  $q_0, q_2$  or  $q_4$ . If it is  $q_0$ , then the previous state must have been  $q_5$  (it cannot be in the very start of the computation since the erasing is not at the end of the tape), so there must be a blank to the left of the head, which contradicts the fact that the erasing was not at the end of the tape. If it is  $q_2$  or  $q_4$ , then the previous state must have been either  $q_1$  or  $q_3$ , so there must be a blank to the right of the head, which contradicts the fact that the erasing was not at the end of the tape.

Analogously to the previous items, one can prove that the head of  $M$  is always at distance at most 1 from a non-blank symbol if one exists.

Note now that by inspecting the table, we see that the only possible progression of states is of the form

$$q_0 q_{i_1}^{k_1} q_{i_1+1} q_5^{m_1} q_0 q_{i_2}^{k_2} q_{i_2+1} q_5^{m_2} \cdots q_0 q_{i_t}^{k_t} q_{i_t+1} q_5^{m_t} \cdots,$$

for some  $i_1, i_2, \dots, i_t \in \{1, 3\}$  and  $k_1, \dots, k_t, m_1, \dots, m_t \geq 0$  and some  $t \geq 0$ . Furthermore each new time  $M$  goes through  $q_0$ , it has erased exactly two symbols. This immediately implies that  $M$  always halts.

Moreover, in each of the blocks  $q_0 q_{i_u}^{k_u} q_{i_u+1} q_5^{m_u}$ , the symbol that is erased on  $q_0$  at the left end of the tape is the same as the one erased in  $q_{i_u+1}$  at the right end of the tape.

Therefore, since  $M$  accepts  $x$  if and only if it erases the whole string  $x$ , i.e., if and only if  $x$  is of the form

$$\sigma_1 \sigma_2 \cdots \sigma_{k-1} \sigma_k \sigma_k \sigma_{k-1} \cdots \sigma_2 \sigma_1,$$

that is, if and only if  $x \in L$ .

Therefore  $M$  decides  $L$ . ◁

**Exercise 2** (HMU 8.2.3). Design a Turing machine that takes as input a number  $N$  in binary and adds 1 to it. To be precise, the tape initially contains a \$ followed by  $N$  in binary. The head is initially scanning the \$ in state  $q_0$ . Your Turing machine should halt with  $N + 1$  in binary on its tape, scanning the leftmost symbol of  $N + 1$ , in state  $q_f$ . You may destroy the \$ in creating  $N + 1$ , if necessary. For instance,  $q_0 \$10011 \mapsto \$q_f 10100$  and  $q_0 \$11111 \mapsto q_f 100000$ .

- Give the transitions of your Turing machine, and explain the purpose of each state.
- Show the sequence of instantaneous descriptions (IDs) of your Turing machine when given input \$111.

*Solution.* Consider the Turing machine with transition function  $\delta$  given by

	0	1	\$	B
$q_0$	$(q_0, 0, R)$	$(q_0, 1, R)$	$(q_0, \$, R)$	$(q_1, B, L)$
$q_1$	$(q_2, 1, L)$	$(q_1, 0, L)$	$(q_2, 1, L)$	—
$q_2$	$(q_2, 0, L)$	$(q_2, 1, L)$	$(q_f, \$, R)$	$(q_f, B, R)$
$q_f$	—	—	—	—

where  $q_0$  is the initial state and  $q_f$  is the accepting state.

The state  $q_0$  simply serves to move the head to the end of the input.

The state  $q_1$  means that the digit under the head needs to be incremented.

The state  $q_2$  means that the digit under the head does not need to be incremented (neither do all digits to the left), hence we only need to put the head in the leftmost digit, which is also the function of state  $q_2$ .

For input  $\$111$ , the sequence of instantaneous descriptions is

$$\begin{aligned}
 q_0\$111 &\mapsto \$q_0111 \mapsto \$1q_011 \mapsto \$11q_01 \mapsto \$111q_0B \\
 &\mapsto \$11q_11 \mapsto \$1q_110 \mapsto \$q_1100 \mapsto q_1\$000 \\
 &\mapsto q_2B1000 \mapsto q_f1000. \quad \triangleleft
 \end{aligned}$$

**Exercise 3** (HMU 8.2.5). Recall that the language  $L(M)$  defined by a Turing machine  $M$  is the set of input strings  $x$  such that  $M$  on input  $x$  halts in an accepting state. ( $M$  halts when there is no entry in the transition function for the current state symbol pair.)

Consider the Turing machine  $M$  with states  $\{q_0, q_1, q_2, q_f\}$ , input alphabet  $\{0, 1\}$ , tape alphabet  $\{0, 1, B\}$ , start state  $q_0$ , accepting state  $q_f$ , and transition function  $\delta$ . For each of the following transition functions  $\delta$ , *informally but clearly describe* the language  $L(M)$  if  $\delta$  consists of the following set of rules.

(a)  $\delta$  is given by

$$\delta(q_0, 0) = (q_1, 1, R); \quad \delta(q_1, 1) = (q_0, 0, R); \quad \delta(q_1, B) = (q_f, B, R).$$

(b)  $\delta$  is given by

$$\begin{aligned}
 \delta(q_0, 0) &= (q_0, B, R); & \delta(q_0, 1) &= (q_1, B, R); \\
 \delta(q_1, 1) &= (q_1, B, R); & \delta(q_1, B) &= (q_f, B, R).
 \end{aligned}$$

(c)  $\delta$  is given by

$$\begin{aligned}
 \delta(q_0, 0) &= (q_1, 1, R); & \delta(q_1, 1) &= (q_2, 0, L); \\
 \delta(q_2, 1) &= (q_0, 1, R); & \delta(q_1, B) &= (q_f, B, R).
 \end{aligned}$$

*Solution.* For item (a), we have

$$L(M) = \{0\} \cdot \{10\}^* = \{0(10)^n : n \geq 0\}.$$

For item (b), we have

$$L(M) = \{0\}^* \cdot \{1\}^+ = \{0^n 1^m : n \geq 0, m > 0\}.$$

For item (c), we have

$$L(M) = \{0\} \cdot \{1\}^* = \{01^n : n \geq 0\}. \quad \triangleleft$$

**Exercise 4** (HMU 8.4.3). *Informally but clearly describe* nondeterministic Turing machines (multiple tape, if you like) that accept the following languages. Try to take advantage of nondeterminism to avoid iteration and save time in the nondeterministic sense. That is, prefer to have your nondeterministic Turing machine nondeterministically branch a lot, while each branch is short.

- (a) The language of all strings of 0's and 1's that have some string of length 100 that repeats, not necessarily consecutively. Formally, this language is the set of strings of 0's and 1's of the form  $wxyxz$ , where  $|x| = 100$ , and  $w$ ,  $y$  and  $z$  are strings of arbitrary length (possibly zero).
- (b) The language of all strings of the form  $w_1\#w_2\#\dots\#w_n$  for any  $n$  such that each  $w_i$  is a string of 0's and 1's, and for some  $j$ ,  $w_j$  is the integer  $j$  in binary.
- (c) The language of all strings of the same form as (b), but for at least two values of  $j$ , we have  $w_j$  equal to  $j$  in binary.

*Solution.* For item (a), consider the following nondeterministic algorithm.

We nondeterministically write on the second tape a string  $s$  of  $\{0,1\}^{100}$ . Then we start reading the input tape  $z_1z_2\cdots z_n$  from left to right and nondeterministically choose a position  $k$  in it (if the input ends before we choose  $k$ , we reject). We then compare  $x_k\cdots x_{k+99}$  with  $s$ , if they are different, we reject, otherwise we continue reading the input from left to right starting at  $x_{k+99}$  and choose a position  $\ell$  in it (if the input ends before we choose  $\ell$ , we reject). We then compare  $x_\ell\cdots x_{\ell+99}$  with  $s$ , if they are equal we accept, otherwise we reject.

For item (b), consider the following nondeterministic algorithm.

We start by reading the input tape and counting the number  $k$  of occurrences of  $\#$ , writing  $k$  in the second tape. We then nondeterministically write a number  $j$  in  $\{1, \dots, k\}$  on the third tape (this can be done by nondeterministically writing a string of 0's and 1's of length at most the length of the binary representation of  $k$  and comparing if the resulting number  $j$  is non-zero and at most  $k$ , if it is not, we simply reject the input).

Finally, we read the input again from left to right and find the  $(j-1)$ -th occurrence of  $\#$  (using another tape to count) and we compare the string between the  $(j-1)$ -th and the  $j$ -th occurrence of  $\#$  with  $j$ , if they are equal we accept, otherwise we reject.

For item (c), we use the same algorithm as item (b), except that we choose nondeterministically two numbers  $j_1, j_2 \in \{1, \dots, k\}$ , compare them, if they are equal we reject, otherwise we proceed to the final part of comparing the string between the  $(j_i-1)$ -th and the  $j_i$ -th occurrence of  $\#$  with  $j_i$ . If for both  $i = 1$  and  $i = 2$  this comparison is true we accept, otherwise we reject. ◁