CS 332: *Theory of Computation* Prof. Mark Bun
Boston University February 23, 2024

# Homework 4 – Due Thursday, February 29 at 11:59 PM

**Reminder** Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to the course staff if asked. You must also identify your collaborators and write "Collaborators: none" if you worked by yourself. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden. Collaboration is not allowed on problems marked "INDIVIDUAL."

1. (**INDIVIDUAL: Complement and Star**) For each of the following statements, give a proof or provide a counterexample.

   (a) For every language $L \subseteq \{0,1\}^*$, we have $(\overline{L})^* \neq \overline{L^*}$.

   (b) For every natural number $k$, if $L \subseteq \{0,1\}^*$ is recognized by a DFA with $k$ states, then there also exists a DFA with $k$ states recognizing $(\overline{L})^*$.

   Hint: Think about the language $A$ from Homework 2, Problem 6.

2. (**Non-regular languages**) Prove that the following languages are not regular. You may only use the distinguishing set method and the closure of the class of regular languages under union, intersection, complement, and reverse.

   (a) $L_1 = \{0^n 1^m \mid n, m \geq 0 \text{ and } n = m^2\}$.

   (b) $L_2 = \{0^n 1^m 0^n \mid m, n \geq 0\}$.

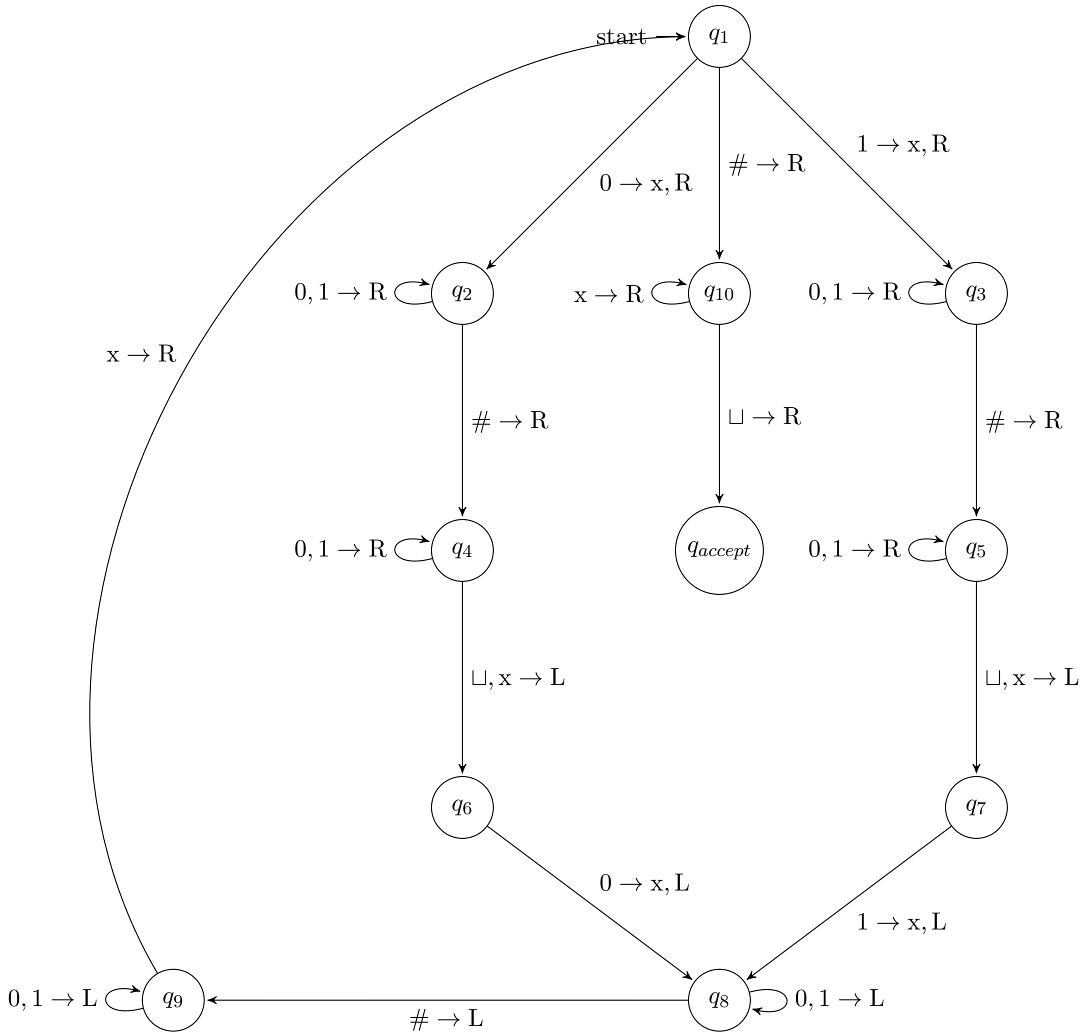
   (c) $L_3 = \left\{1^k y \mid y \in \left\{0,1\right\}^* \text{ and } |y| = k\right\}$.

   (d) $L_4 = \left\{a^i b^j c^k \mid i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k\right\}$.

   (e) Let $\Sigma = \{0, 1, \#\}$. Let $L_5 = \{x\#y\#z \mid x, y, z \text{ are nonnegative integers written in binary where } x + y = z\}$.

3. (**Low-Level to Implementation-Level**) The following page illustrates the state diagram of a Turing machine using input alphabet $\Sigma = \{0, 1, \#\}$ and tape alphabet $\Gamma = \{0, 1, \#, \mathrm{x}, \sqcup\}$.

   The notation "$a \to \mathrm{R}$" is shorthand for "$a \to a, \mathrm{R}$." The reject state and transitions to the reject state are not shown. Whenever the TM tries to read a character for which there is no explicit transition that means that the TM goes to the reject state. Use the convention that the head moves right in each of these transitions to the reject state.




(a) Give the sequences of configurations that this TM $M$ enters when given as input strings (i) 010#100, (ii) 10#01, and (iii) 0##0. Use the same representation for your configurations as we did in lecture 9.

(b) Give an implementation-level description of the Turing machine described by this state diagram. Hint: The machine is similar to Example 3.9 in Sipser.

(c) What is the language decided by $M$?

4. (**Implementation-Level to Low-Level**)

(a) Give a state diagram of a TM whose implementation-level description is below. The input alphabet of this TM is $\{a, b\}$ and the tape alphabet is $\{a, b, \mathrm{x}, \sqcup\}$.

**Input :** String $w$

1. Read the first symbol on the input string. If it is a blank symbol, *accept*. If it is an $a$, then erase this $a$ (i.e., replace it with a $\sqcup$), move the head right, and go to step 5. If it is a $b$, then erase this $b$, move the head right, and go on to the next step.
2. Continue moving the head right until a blank symbol is found. Replace this blank symbol with a $b$, move the head left, and go to step 6.
3. Move the head right until either an $a$ or a blank symbol is found. If it is a blank symbol, *accept*. If it is $a$ then cross out this $a$ (i.e., replace it with an x), move the head left, and go on to the next step.
4. Repeatedly move the head left until the blank symbol is found. After it is found move the head one cell to the right and go on to the next step.
5. Continue moving the head right until a $b$ or a blank symbol is found. If a blank symbol is found, *reject*. Otherwise, cross out the $b$ that was found, move the head one cell left, and go on to the next step.
6. Continue moving the head left until the blank symbol is found. When it is found, move the head right and go back to step 3.

(b) Give the sequences of configurations that your TM enters when given as input strings (i) *abbab*, and (ii) *baaab*.

(c) What language is decided by the TM from part (a)?

5. (**Bonus Problem**) The distinguishing set method doesn't tell us directly about the minimum number of states an <u>NFA</u> needs to recognize a language. And in general, we don't know anything quite as powerful as the Myhill-Nerode theorem for characterizing the minimum size of an NFA for language. Nevertheless, some techniques for proving NFA state size lower bounds are available, as this problem illustrates.

   A *fooling set* for a language $L$ is a set of pairs of strings $\{(x_i, y_i) \mid i = 1, \ldots, k\}$ such that:

   (a) For every $i = 1, \ldots, k$, we have $x_i y_i \in L$, and

   (b) For every $i \neq j$, at least one of the strings $x_i y_j$ or $x_j y_i$ is not in $L$.

   Show that if $L$ has a fooling set of size $k$ as described above, then every NFA recognizing $L$ requires at least $k$ states.