CSCI 1010 - Fall'26 Theory of Computation Lorenzo De Stefani

HW1

Due: n/a

Reminder: Submit your assignment on Gradescope by the due date. Submissions must be typeset. Each page should include work for only one problem (i.e., make a new page/new pages for each problem). See the course syllabus for the late policy.

While collaboration is encouraged, please remember not to take away notes from any labs or collaboration sessions. Your work should be your own. Use of other third-party resources is strictly forbidden.

Please monitor Ed discussion, as we will post clarifications of questions there.

Problem 1

Prove that for any integer $k > 0$, there is a language L such that

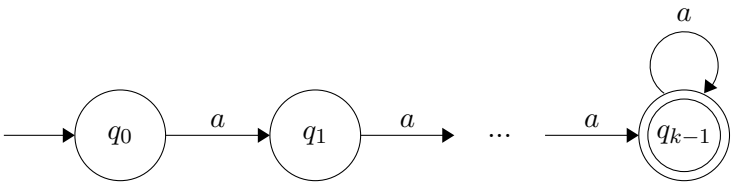
(a) L can be recognized by a DFA with k states

(b) No DFA with less than k states recognizes L

Solution: Consider the language $L = \{w \mid w \text{ has length at least } k-1\}$ over alphabet $\{a\}$.

To show part (a), we first find a DFA that recognizes L that has exactly k states.

Consider the following DFA, D:



D has exactly K states, $q_0, \ldots, q_{k-1}$. If $w \notin L$, there must be less than $k-1$ symbols. Then, D must terminate in a state other than q_{k-1} and reject w since q_{k-1} is the only accepting state. If $w \in L$, then w has length at

least $k-1$. Then, D will reach q_{k-1} at some point and remain there since all transitions from q_{k-1} stay in q_{k-1}. This means that D will accept w. Therefore, $L(D) = L$.

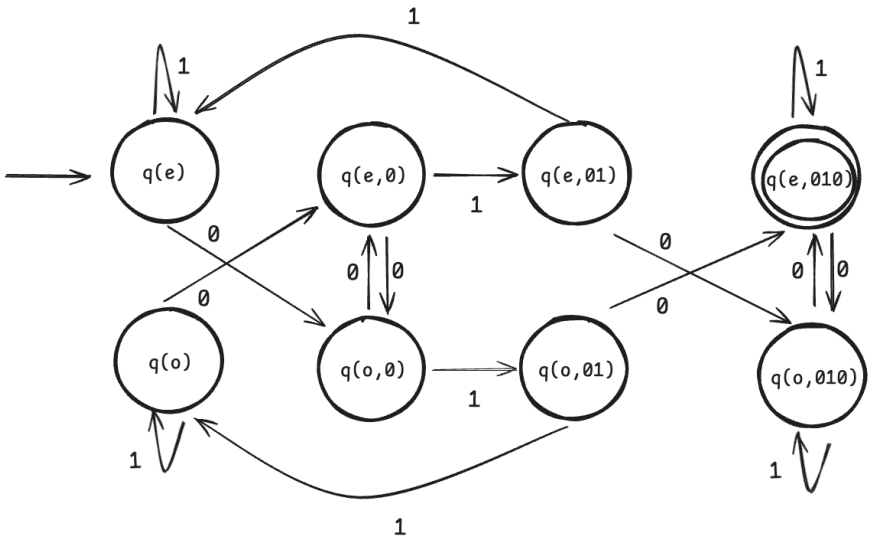
We will show part (b) by contradiction. Assume that such a DFA, D, exists such that $L(D) = L$ and D has $k-1$ or less states. Consider the string w, which is a repeated $k-1$ times. w has length $k-1$ so $w \in L$ and is therefore accepted by D. D has less than $k-1$ states so by the pigeonhole principle some state must be visited at least twice. This means that D contains a loop and therefore some substring of w can be removed and the resulting string will still be accepted by D, meaning that $w \in L$. However, such a string would have length less than $k-1$ and so is not in L, a contradiction.

Problem 2

Design a DFA that recognizes the following language over the alphabet $\{0, 1\}$. Provide a proof of correctness.

$\{w|\ w$ contains an even number of 0s and the substring 010$\}$

Solution: design a DFA that recognizes $L = \{w \mid w$ *contains an even number of 0s and the substring 010*$\}$ over alphabet $\{0, 1\}$.



Proof of Correctness. We will show that $L(M) = L$.

Assume $w \in L$. Then w contains an even number of 0s and the substring 010. As M processes w, it tracks both whether the 0 count is even or odd (e or o) and whether the substring 010 has occurred yet. Since w contains 010, at some point M will reach state $q_{e,010}$ or $q_{o,010}$. Since w has an even number of 0s total, M will end in $q_{e,010}$ (an accepting state) after processing all of w.

On the other hand, assume $w \in L(M)$. Then M accepts w, meaning w leads M to an accepting state. The only accepting state is $q_{e,010}$, which is reachable only once the substring 010 has been seen and if the count of 0s so far is even. Thus, $w \in L$.

Problem 3

(a) Let $A \subseteq \Sigma^*$ be a regular language, and let

$$M = (Q, \Sigma, \delta, q_0, F)$$

be a DFA recognizing A.

Define a new DFA

$$M' = (Q, \Sigma, \delta, q_0, Q \setminus F).$$

What language does M' recognize? Prove your answer.

What general conclusion can you draw from this construction about regular languages?

(b) Consider the language

$L = \{w \in \{a, b\}^* \mid w$ has an odd number of b's and contains the substring $ab\}$.

Write L as the intersection

$$L = L_1 \cap L_2$$

of two simpler languages. Clearly define L_1 and L_2.

(c) Construct a DFA recognizing L_1 and a DFA recognizing L_2.

For each DFA, prove that it recognizes the intended language.

(d) Assume that you have already proved that regular languages are closed under union, that is, the union of two regular languages is regular. Argue that L is regular.

Solution:

(a) Since

$$M' = (Q, \Sigma, \delta, q_0, Q \setminus F),$$

the machines M and M' have exactly the same states, transitions, and start state. Thus, on every input w, they finish in the same state.

The only difference is that accepting and non-accepting states have been swapped. Therefore, M' accepts w if and only if M rejects w. Since $L(M) = A$,

$$L(M') = \{w \in \Sigma^* : w \notin A\} = \overline{A}.$$

Hence, whenever A is regular, $\overline{A}$ is also regular. Therefore, regular languages are closed under complement.

(b) Define

$$L_1 = \{w \in \{a, b\}^* : w \text{ has an odd number of } b\text{'s}\}$$

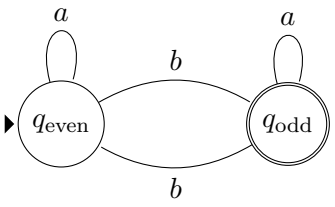
and

$$L_2 = \{w \in \{a, b\}^* : w \text{ contains the substring } ab\}.$$

A string belongs to L exactly when it belongs to both L_1 and L_2. Hence

$$L = L_1 \cap L_2.$$

(c) A DFA recognizing L_1 is:

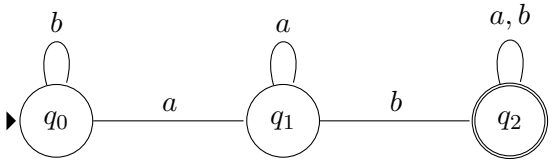


The state q_{even} represents that the portion of the input read so far contains an even number of b's, while q_{odd} represents that it contains an odd number of b's.

Reading a does not change the number of b's, while reading b changes its parity. Since the machine starts in q_{even} and accepts exactly in

q_{odd}, it accepts precisely the strings containing an odd number of b's. Thus it recognizes L_1.

A DFA recognizing L_2 is:



The state q_0 represents that the substring ab has not yet been seen and the most recently read symbol is not a. The state q_1 represents that ab has not yet been seen, but the most recently read symbol is a. Finally, q_2 represents that the substring ab has already occurred.

From q_1, reading b completes the substring ab, so the machine moves to q_2. Once q_2 has been reached, the substring has already occurred, so all remaining input leaves the machine in q_2.

Therefore, the DFA accepts exactly those strings containing the substring ab, and hence recognizes L_2.

(d) By part (c), both L_1 and L_2 are regular.

By part (a), regular languages are closed under complement. Therefore,

$$\overline{L_1} \qquad \text{and} \qquad \overline{L_2}$$

are regular.

Since regular languages are closed under union,

$$\overline{L_1} \cup \overline{L_2}$$

is regular. Applying closure under complement again,

$$\overline{\overline{L_1} \cup \overline{L_2}}$$

is also regular.

By De Morgan's law,

$$\overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2.$$

Since

$$L = L_1 \cap L_2,$$

it follows that

$$\boxed{L \text{ is regular.}}$$