CSC236 Tutorial 9

Sample Solutions

1. Let L be the language $\{a, ab\}^{\circledast}$. List some of the elements of L and then define a predicate $P(s)$ such that for all strings s over the alphabet $\{a, b\}$, $P(s)$ is true iff s is in $\{a, ab\}^{\circledast}$.

   Solutions:

   $$\begin{aligned}\{a, ab\}^{\circledast} &= \{\epsilon\} \cup \{a, ab\} \cup \{aa, aab, aba, abab\} \cup \{aaa, \cdots\} \cup \cdots \\ &= \{\epsilon, a, ab, aa, aab, aba, abab, aaa, \cdots\}\end{aligned}$$

   $\{a, ab\}^{\circledast}$ is similar to $\{a, b\}^*$ (all strings composed of a's and b's), except that each b is preceded by an a.
   Let us define P as follows: $P(s)$: s is a string of a's and b's where each b is immediately preceded by an a.

   We should prove that $s \in \{a, ab\}^{\circledast}$ iff $P(s)$. The proof has two parts: $\Rightarrow$: It is easy to see that each string in $\{a, ab\}^{\circledast}$ satisfies P, that is, every b is immediately preceded by an a.
   $\Leftarrow$: Each string where each b is immediately preceded by an a can be broken up into pieces ab (one for each b in the string) with every other symbol being a. Hence, the string belongs to $\{a, ab\}^{\circledast}$.

2. Describe three different languages L over alphabet $\{a, b, c\}$ such that $L = L^{\circledast}$.

   - Let $L = \{\epsilon\}$.
     Then $L^{\circledast} = \{\epsilon\}$, and so $L = L^{\circledast}$.

   - Let $L = \{a, b, c\}^*$.
     By definition, $s \in L^{\circledast}$ iff $s = s_1 \circ s_2 \circ \cdots \circ s_k$ for some $s_1, \cdots, s_k \in L$.
     Each $s_i \in L$ is a sequence of a's, b's, c's, and hence s is a sequence of a's, b's, and c's.
     So, by definition $s \in L$.

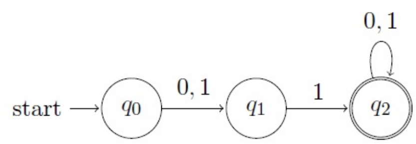
     $s \in L$ iff $s = s_1 \circ s_2 \circ \cdots \circ s_k$ for some $s_1, \cdots, s_k \in \{a, b, c\}$.
     On the other hand, $s \in L^{\circledast}$ because $\{a, b, c\} \in L^{\circledast}$.

   - Let $L = \{\epsilon, a, aa, aaa, \cdots\} = \{a\}^*$.
     Then $L = L^{\circledast}$ for similar reasons as in the previous case.

3. Give a DFSA for each language below.

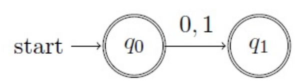
(a) $L_1 = \{s \in \{0,1\}^* : s$ contains at least 2 characters and $s's$ second character is $a1\}$.

Solution:



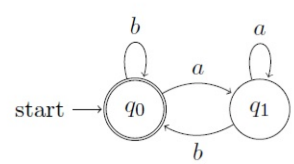
(b) $L_2 = \{s \in \{0,1\}^* : s$ contains fewer than 2 characters$\}$.

Solution: Note that the definition of L_2 implies that $L_2 = \{\epsilon, 0, 1\}$.



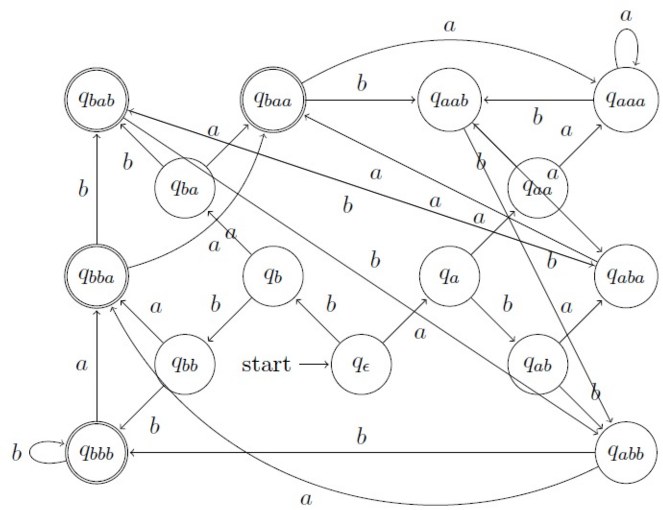
(c) $L_3 = \{s \in \{a,b\}^* :$ every a in s is eventually followed by $b\}$.
For example, $aaab \in L_3$ because there is a b that follows every a, even though it is not immediately after the first two as.

Solution: Based on the definition of L_3, $s \in L_3$ iff s does not end with an a.

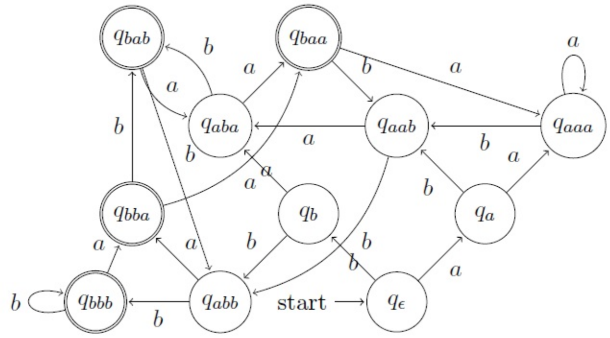


(d) $L_4 = \{s \in \{a,b\}^* :$ the third-last character of s is a $b\}$.

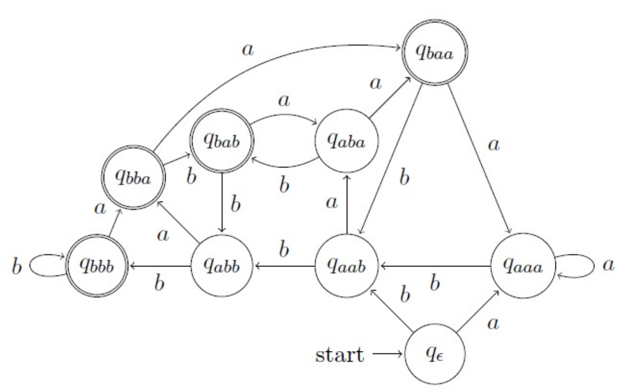
Solution: We need information about the third last character. Because of that, we need to keep track of the last three characters that we've seen. The straightforward way to do so is to label states by all the possible strings of length less than or equal to 3.
This way, we end up with 15 states: $\{q_\epsilon, q_a, q_b, q_{aa}, q_{ab}, q_{ba}, q_{bb}, q_{aaa}, q_{aab}, q_{aba}, q_{abb}, q_{baa}, q_{bab}, q_{bba}, q_{bbb}\}$.
q_ϵ is the initial state (representing that the string so far is empty) and each state q_{bxy} is accepting (representing that the last three characters are bxy).



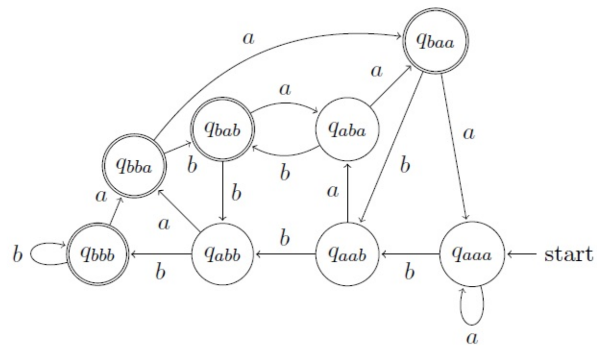
As you can see this graph is really complicated and hard to draw. There is a trick to simplify this a bit. Notice that the transitions from states q_{xy} are identical to the transitions from states q_{axy}, and these states are all rejecting. So we can remove the states q_{xy} and replace transitions going into them with transitions going into q_{axy}.



Similarly, states q_x behave exactly the same as states q_{aax}.



Finally, state q_ϵ behaves exactly the same as state q_{aaa}.



q_{baa}
q_{bab}
q_{aba}
q_{bba}
q_{bbb}
q_{abb}
q_{aab}
q_{aaa}
start