CS 154: Introduction to Automata and Complexity Theory
Spring 2009

Assignment 1 Solutions
April 14, 2009

Problem 1

(a) All strings that contain three consecutive 1's.

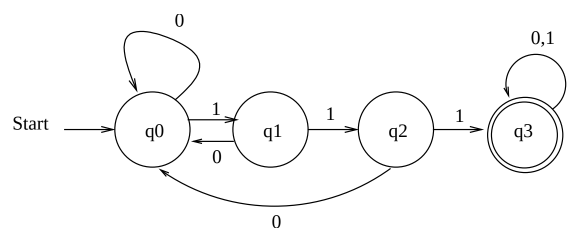

Figure 1: DFA for Problem 1(a)

(b) All strings that do not end with 00.

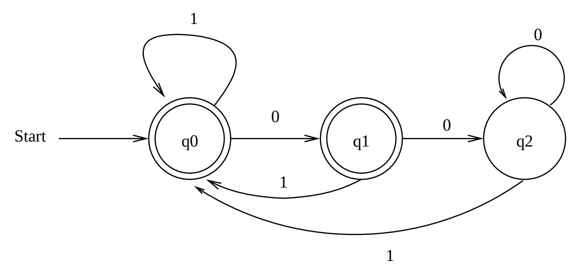

Figure 2: DFA for Problem 1(b)

Problem 2

(a) By induction on $|w|$, the length of w. For $|w| = 1$, we have:

$$\widehat{\delta}(q_f, w) = \delta(q_f, w) = \delta(q_0, w) = \widehat{\delta}(q_0, w)$$

Now, suppose that the statement is true for $|w| = k \geq 1$. For $|w| = k + 1$, we have $w = xa$ for some string x of length k and some symbol $a \in \Sigma$. Then, we have:

$$\begin{aligned}
\widehat{\delta}(q_f, w) &= \widehat{\delta}(q_f, xa) && \text{(by definition of } w\text{)} \\
&= \delta(\widehat{\delta}(q_f, x), a) && \text{(by definition of } \widehat{\delta}\text{)} \\
&= \delta(\widehat{\delta}(q_0, x), a) && \text{(by inductive hypothesis, since } |x| = k\text{)} \\
&= \widehat{\delta}(q_0, xa) && \text{(by definition of } \widehat{\delta}\text{)} \\
&= \widehat{\delta}(q_0, w) && \text{(by definition of } w\text{)}
\end{aligned}$$

This completes the proof. $\square$

(b) We shall use the following result (cf. Hopcroft et. al., Exercise 2.2.2):

$$\forall q \in Q, \forall x, y \in \Sigma^* : \qquad \widehat{\delta}(q, xy) = \widehat{\delta}(\widehat{\delta}(q, x), y) \tag{1}$$

Now, we shall proceed by induction on $k \geq 1$. The base case, i.e. $k = 1$, is trivial, since by hypothesis we have $w^1 = w \in L(M)$. Suppose that the statement is true for $k = m$. For $k = m + 1$, our goal is to show that $w^{m+1} \in L(M)$. Equivalently, we have to show that $\widehat{\delta}(q_0, w^{m+1}) = q_f$. Now, we compute:

$$\begin{aligned}
\widehat{\delta}(q_0, w^{m+1}) &= \widehat{\delta}(q_0, w^m w) && \\
&= \widehat{\delta}(\widehat{\delta}(q_0, w^m), w) && \text{(by (1))} \\
&= \widehat{\delta}(q_f, w) && \text{(by inductive hypothesis)} \\
&= \widehat{\delta}(q_0, w) && \text{(by the result of part (a))} \\
&= q_f && \text{(since } w \in L(M))
\end{aligned}$$

This completes the proof. $\square$

Problem 3

(a) The desired NFA is given as follows:

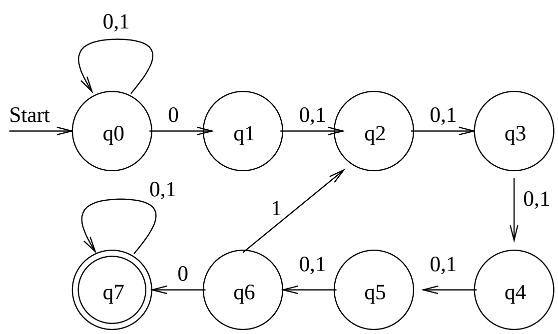

Figure 3: DFA for Problem 3(a)

(b) Suppose that the DFA has less than 2^5 states. Then, there exists a state q such that two distinct strings of length 5 will reach. Let these two strings be $a_0a_1a_2a_3a_4$ and $b_0b_1b_2b_3b_4$. Suppose that $a_0 \neq b_0$. Without loss of generality, we may assume that $a_0 = 0$. Then, the strings $a_0a_1a_2a_3a_4 10$ and $b_0b_1b_2b_3b_4 10$ will reach the same state, but the former would be accepted, while the latter rejected. This is a contradiction. Otherwise, the strings differ at the i–th position, where $i = 1, 2, 3$ or 4. Without loss, let $a_i = 0$ and $b_i = 1$. Then, the strings $a_0a_1a_2a_3a_4 11...0$ (concatenate $i + 1$ 1s after a_4 followed by one 0) and $b_0b_1b_2b_3b_4 11...0$ (concatenate $i + 1$ 1s after b_4 followed by one 0) will reach the same state, but the former is accepted, while the latter rejected. This is again a contradiction. Thus, any DFA must have at least 2^5 states. $\square$

Comment: The common mistake is to argue that the DFA must "remember" 5 positions in order to determine the membership of the string. While this is the general idea, it is not a proof.

Problem 4

The statement is false, as shown by the following NFA in Figure 5. Both machines accept the string 0.

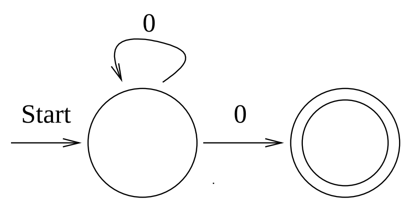

Figure 4: Counterexample for Problem 4

Problem 5

(a) Suppose that L is a regular language. Then, there exists an DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that $L(M) = L$. Consider the Total–NFA N defined by $N = (Q, \Sigma, \delta', q_0, F)$, where $\delta'(q, a) = \{\delta(q, a)\}$ for all $q \in Q$ and $a \in \Sigma$. Note that for each string $w \in \Sigma^*$, there is only one possible execution path in the DFA. Thus, the above construction gives $L(N) = L$ as desired. $\square$

(b) Let $N = (Q, \Sigma, \delta, q_0, F)$ be a Total–NFA. It suffices to show that N can be converted to an DFA $M = (Q_M, \Sigma, \delta_M, q_0, F_D)$ such that $L(N) = L(D)$. To do this, we can use the subset construction algorithm for converting an NFA into an DFA (cf. Hopcroft et. al., p. 61). All the steps are essentially the same, except that now we have F_D is the set of non–empty subsets S of Q such that $S \subset F$. $\square$