Assignment 3

Rui Qiu (c3qiurui), Yilin Xie (c4xieyil), Yan Zeng (c3zengya)

March 26, 2015

Proof: Assume the DFA we created in part(a) is called M.

CLAIM: No smaller DFA can compute the same language, i.e., M is a minimal DFA for $L(M)$.

SUPPOSE: For the sake of contradiction, suppose that there exists a smaller DFA $M' = \{Q', \Sigma', \delta', s'_0, F'\}$ for L such that $|Q'| \leqslant 2$.

Consider the following strings:

$$\begin{aligned} x_1 &= 0 \\ x_2 &= 1 \\ x_3 &= 2 \\ x_4 &= 3 \\ x_5 &= 4 \\ x_6 &= 5 \\ x_7 &= 6 \\ x_8 &= 7 \\ x_9 &= 8 \\ x_{10} &= 9 \end{aligned}$$

These strings are chosen such that the computation of these strings takes us into each of the three states in M. Observe this:

$$\begin{aligned} \hat{\delta}(s_0, 0) &= s_0 \\ \hat{\delta}(s_0, 1) &= s_1 \\ \hat{\delta}(s_0, 2) &= s_2 \\ \hat{\delta}(s_0, 3) &= s_0 \\ \hat{\delta}(s_0, 4) &= s_1 \\ \hat{\delta}(s_0, 5) &= s_2 \\ \hat{\delta}(s_0, 6) &= s_0 \\ \hat{\delta}(s_0, 7) &= s_1 \\ \hat{\delta}(s_0, 8) &= s_2 \\ \hat{\delta}(s_0, 9) &= s_0 \end{aligned}$$

By specifying s_0, s_1, s_2, we can reduce the number of cases into 3 major cases:

1. s_0: The string w is a multiple of 3.

2. s_1: The string w is 1 modulo 3.

3. s_2: The string w is 2 modulo 3.

Therefore, the simplified version is:

$$\hat{\delta}(s_0, t_0) = s_0$$
$$\hat{\delta}(s_0, t_1) = s_1$$
$$\hat{\delta}(s_0, t_2) = s_2$$

where $t_0 \equiv 0 \pmod 3, t_1 \equiv 1 \pmod 3, t_2 \equiv 2 \pmod 3, t_i \in \{0, 1, 2, ..., 9\}$.

By the pigeonhole principle, two of these computations $\hat{\delta}$ on strings x_1 to x_3 must yield the same state in M' since the number of states is smaller than 3. Therefore, we must show for each pair of computations $(\hat{\delta}(s_0, t_i), \hat{\delta}(s_0, t_j))$ that:

$$\hat{\delta}(s_0, t_i) \neq \hat{\delta}(s_0, t_j), \text{ where } i, j \in \{0, 1, 2\}, i \neq j.$$

There are $\binom{3}{2}$ cases we must show contradict our assumption. One of the cases is illustrated below:

CASE 1: Show contradiction for t_0 and t_1. We start with the following statement:

$$\hat{\delta}(s_0, t_0) = \hat{\delta}(s_0, t_1)$$

Note that by the defintion of $\hat{\delta}$ we can add the same string to both sides without affecting the equality. For example, we add 3 to both sides and get::

$$\hat{\delta}(s_0, t_0 3) = \hat{\delta}(s_0, t_1 3)$$

However, our language should accept the string $t_0 3$ but not the string $t_1 3$, since the latter is not a multiple of 3. Therefore $\hat{\delta}(s_0, t_0 3) \in F$ but $\hat{\delta}(s_0, t_1 3) \notin F$. Thus these two computations can not result in the same state, giving us a contradiction.

By proving each of the cases results in a contradiction, we prove that our DFA is indeed minimal, i.e., there is no smaller DFA can compute the language.