CSE 105: Homework Set 1

Joshua Wheeler

April 9, 2014

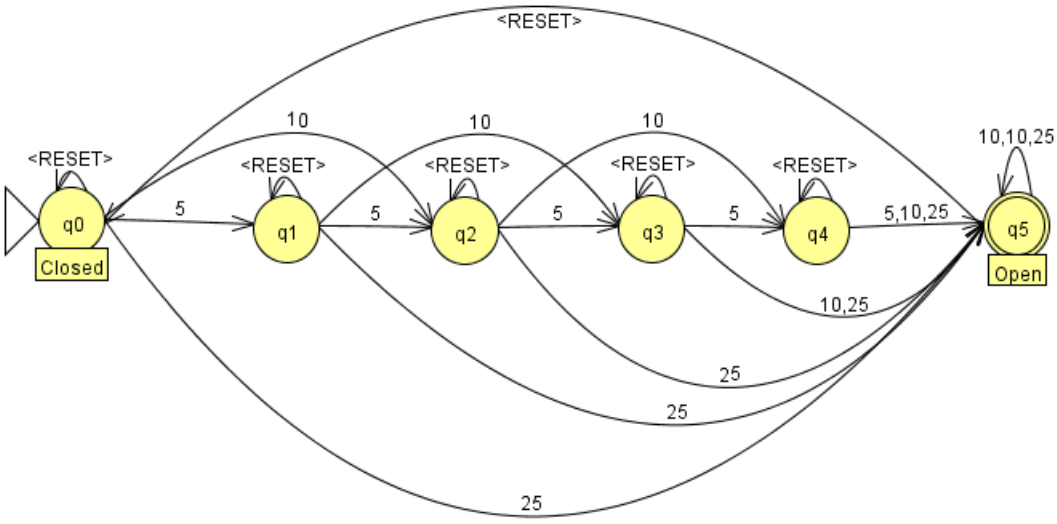
1. Toll Gate Controller DFA

   This DFA will have one state per possible input value. These states are when the machine totals are either 5, 10, 15, 20 or 25 cents. The state where the machine has 25 cents is the opened state. Each input transitions to the state representative of the sum of coins entered so long as the sum is less than or equal to 25. Otherwise, the state will transition to the open state for any input combination which sums to a value greater than or equal to 25. A RESET state is also added since the machine will need to know when a car passes through so it can close itself again and wait for the next car to enter its coins. A formal definition and image of this DFA follows:

$$M = \{Q, \Sigma, \delta, F, q_0\}$$
$$= \{\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{5, 10, 25, RESET\}, \delta, \{q_5\}, q_0\}$$

   Assume $\sigma_i \in \Sigma$

$$\delta(q_i, \sigma_i) \begin{cases} \delta(q_i, 5) = q_{(i+1)} & \text{if } 0 \leq q_i \leq 4 \\ \delta(q_i, 10) = q_{(i+2)} & \text{if } 0 \leq q_i \leq 3 \\ \delta(q_i, 5) = q_5 & \text{if } q_i = 5 \\ \delta(q_i, 10) = q_5 & \text{if } q_i \geq 4 \\ \delta(q_i, 25) = q_5 & \text{if } 0 \leq q_i \leq 5 \\ \delta(q_i, RESET) = q_i & \text{if } 0 \leq q_i \leq 4 \\ \delta(q_i, RESET) = q_0 & \text{if } q_i = 5 \end{cases}$$

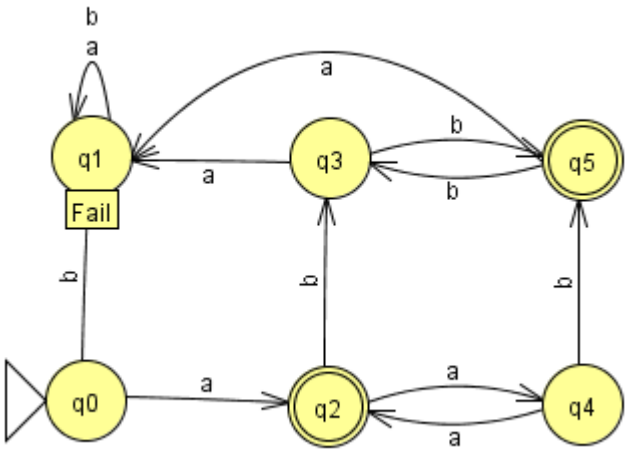


2. DFA Design of regular languages

2a.

$L_1 = \{a^i b^j | i, j \geq 0 \text{ and } i + j \text{ is an odd number}\}$ over the alphabet $\Sigma_1 = \{a, b\}$

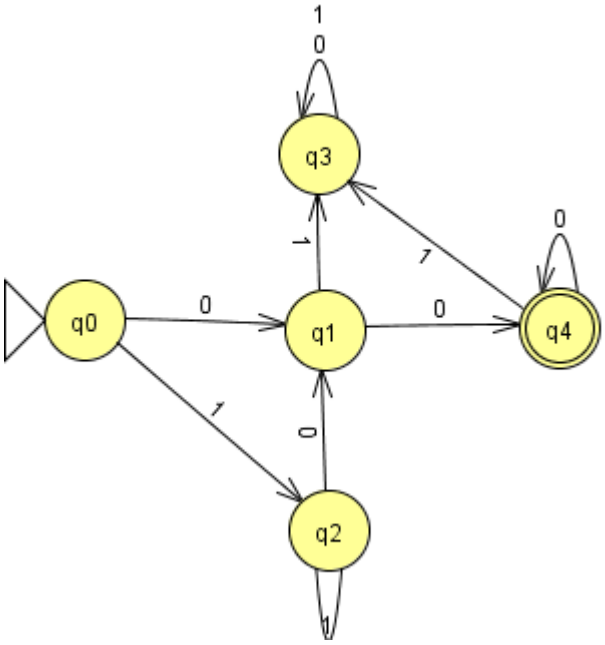
This expression is saying that whenever there are an odd number of inputs (i + j) we want to accept the string. The only extra caveat is that a string of a's must come before a string of b's. To accomplish this, six states are necessary. The first state is the initial state which indicates an empty string. The initial state is not a final state since i+j = 0 is not an odd number. There are two accepting states. One accepting state acts as a counter for an odd nunber of a's and the second accepting state recognizes and odd number of b's. If a b is recognized while the string is iterating through a's it switches the accepting state. If an a is recognized while iterating through b's, the string is invalid and the machine will fail (i.e. it will fail on abba). If the string starts with a b, it will also go to the fail state from the initial state.



2b.

$L_2 = \{w \in \{0,1\}^* | 00 \text{ is a substring of w but 01 is not}\}$ over the alphabet $\Sigma_1 = \{0,1\}$

This DFA is restricted to accepting a string of zero or more 1's followed by a string a two or more 0's. Whenever a 0 is followed by a 1, then that means the substring 01 exists and we never want to accept the string. Whenever a string of two or more zeros are recognized, as long as they are not followed by a 1, then we want to accept the string.



3. Closure Proof
   To see that prefixing the symbol a to the regular language over the alphabet $\{a, b\}$ is closed, define a machine M_1 such that $M_1 = \{Q, \Sigma, \delta, F, q_0\} = \{\{q_0\}, \{a, b\}, \delta, \{q_0\}, q_0\}$ where $\delta(q_0, a) = \delta(q_0, b) = q_0$
   Then $R(M_1) \subseteq \Sigma^* \implies \{l_1, l_2, \cdots, l_n\}$ is the language containing all possible words which may be formed by Σ
   if we prefix R with a then $aR(M_1) = \{al_1, al_2, \cdots, al_n\}$, since $al_i \in \Sigma^*$ then $aR \subseteq \Sigma^* \implies aR$ is also regular, so prefixing a onto R is a closed operation.

4. Modeling Arithmetic
   x and y will add to a power of two only if after the addition there is one 1 bit in the string. If the number of 1 bits in the added string exceeds one then the DFA should not recognize the sequence. Given this, for any x_i, y_i, the summation is accepted if x=0, y=1 or x=1, y=0 and the sequence x=y=1 has been recognized exactly once. If x=y=1 has been recognized, the DFA may only accept x=y=0 following the x=y=1 sequence. There is one flaw in this DFA that I did not have time to correct, and that is the case when x=y=1 has not been recognized and x=0, y=1 or x=1, y=0 has been recognized exactly once. Accomplishing this would require more states.

   The following DFA represents the above criterion: