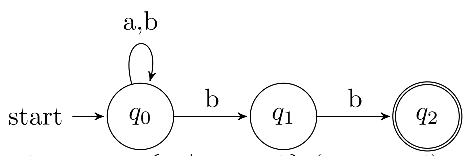
Theory of Computation Assignment no. 3

Goktug Saatcioglu

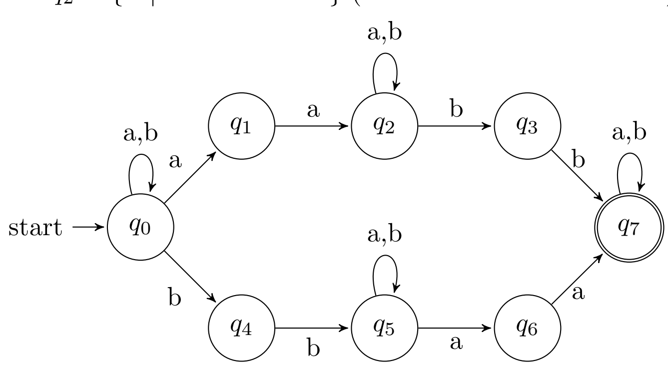
(1) Consider two copies of $DFA\ M$ that recognizes the langauge A. From these two copies, say M_1 and M_2, we can consruct an NFA M' that recognizes $omission(A)$. Begin by making all accepting states in M_1 non-accepting and keep the initial state of M_1 as the initial state of M'. The accepting states of M' are now going to be the accepting states of M_2. Then, for every transition $\delta(p_1, a) = q_1$, where $a \in \Sigma$, in M_1 there is an equivalent state q_2 such that $\delta(p_2, a) = q_2$ where p_2 is a copy of the state p_1 in M_2 and q_2 is a copy of the state q_1 in M_2. We can construct an NFA that recognizes $omission(A)$ by adding a λ-tranisiton for each $\delta(p_1, a) = q_1$ from p_1 to its corresponding q_2 in M_2. If there are no transitions into q_0 in M_1 then remove the corresponding q_0 in $M2$ and otherwise convert the q_0 in M_2 to a non-accepting non-initial state. M' recognizes $omission(A)$ since the shift from M_1 to M_2 can only happen once, meaning only one letter a can be removed from w and still be able to recognize u which is an omission of w.

(2) a. (ii)



Where $q_0 = \{w \mid w \in \Sigma^*\}$ (any word), $q_1 = \{w \mid w \text{ ends with } b\}$ (all words that end with b) and $q_2 = \{w \mid w \text{ ends with } bb\}$ (all words that end with bb).

(iii)



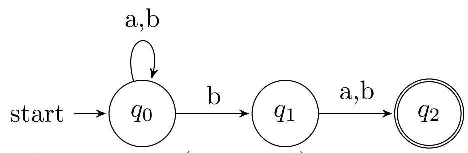
Where $q_0 = \{w \mid w \in \Sigma^*\}$ (any word), $q_1 = \{w \mid w \in \Sigma_a^*\}$ (all words that has a as a substring), $q_2 = \{w \mid w \in \Sigma_{aa}^*\}$ (all words that has aa as a substring), $q_3 = \{w \mid w \in \Sigma_{aa}^* \wedge w \in \Sigma_b^*\}$ (all words that have aa and b as substrings), $q_4 = \{w|w \in \Sigma_b^*\}$ (all words that has b as a substring), $q_5 = \{w \mid w \in \Sigma_{bb}^*\}$ (all words that has bb as a substring), $q_6 = \{w \mid w \in \Sigma_{bb}^* \wedge w \in \Sigma_a^*\}$ (all words that have bb and a as substrings), and $q_7 = \{w \mid w \in \Sigma_{aa}^* \wedge w \in \Sigma_{bb}^*\}$ (all words that have both aa and bb as substrings).

b. If $L \in REG$ then there $\exists$ a DFA that identifies L. Let $M = (\Sigma, Q, q_0, F, \delta)$ be the DFA that identifies L. Then the DFA $M' = (\Sigma, Q, q_0, Q \setminus F, \delta)$ identifies L^c. That is, M' is a DFA where the accepting and non-accepting states of M are switched. Thus, if $L \in REG$ then $L^c \in REG$.

c. If $L \in REG$, then there $\exists$ a DFA that idenfities L. Let $M = (\Sigma, Q, q_0, F, \delta)$ be the DFA that identifies L. We can then construct the following NFA M' by making the initial state of M an accepting state, making all of the accepting states of M non-accepting states, and reversing all transitions given by M. Furthermore, we introduce a new initial state, say q_0', that has λ transitions from q_0' to the accepting states of M (which have now become non-accepting states in M').

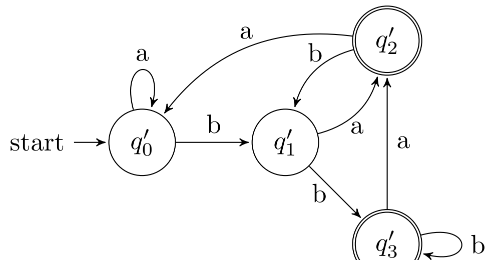
Formally, $M' = (\Sigma, Q', q_0', F', \delta')$ where $Q' = Q \cup \{q_0'\}$, $F' = \{q_0\}$ and $\delta' = \{p | p \in \delta'(q,a) \iff q \in \delta(p,a)$,where$p, q \in Qanda \in \Sigma$, and there are λ transition from q_0 to the accepting states of $M\}$. Since the NFA M' identifies L^R, if $L \in REG$ then $L^R \in REG$.

(3) a.



Where $q_0 = w$ (any word), $q_1 =$ a word that ends with b and $q_2 =$ a word where the 2nd from last letter is b.

b.



Where $q_0' = \{q_0\}$, $q_1' = \{q_0, q_1\}$, $q_2' = \{q_0, q_2\}$ and $q_3' = \{q_0, q_1, q_2\}$. States that are not denoted by prime are defined as above in (a).

c. To create a DFA that identifies L_k we can create a DFA, say M_k, with 2^k states where each state corresponds to one of the k-length words (w) that can be formed using Σ^*. Thus, our set of states, Q_k, is expressed as $Q_k = \Sigma^k = \{a, b\}^k$. This is because to identify L_k, M_k must remember the last k letters since the length of w is not known beforehand. There are 2^k possible words of length k and thus we need 2^k states that remember the last k letterss of each word. For M_k we use the transition function on a state when a letter $l' \in \Sigma$ is read as a left shift operand where all letters in that state are shifted to the left by one and l' is concatenated to the end. Formally, if $l_1 l_2 ... l_k \in \Sigma^k$ is a word of length k and where l_i denotes the position of a letter in that word, then $\delta_k(l_1 l_2 ... l_k, l') = l_2 l_3 ... l_k l'$. The initial state would then be $q_0 = a^k$ because any word with less than length k cannot have a b that is k'th from the last letter of that word Thus, we start with the word of all a's of length k. Finally, the accepting states would be the set of all states that have a b in the first position (i.e. $F_k = \{l_1 l_2 ... l_k \mid l_1 = b\}$)). Overall, $M_k = (\Sigma, Q_k, q_0, F_k, \delta_k)$ as described above.

d. Suppose for the sake of contradiction that there exists a DFA M that identifies the language L with at most 2^{k-1} states. Since there are 2^k possible words of length k by the pigeonhole principle there are two distinct k-length words w_1 and w_2 such that M ends at the same accepting state when given the inputs w_1 and w_2. Pick any i such that w_1 and w_2 are different from each other in the i'th position. Next, construct $w_1' = wa^{k-i}$ and $w_2' = wa^{k-i}$ such that w_1' and $'_2$ contains a b in the k'th position. We see that w_1' and w_2' end in the same accepting state when used as inputs for M. However, M, by our construction, is supposed to end at the same accepting state for exactly only two words which gives us a contradiction. Thus, M must have at least 2^k states to identify L.

(4) Let $N = (\Sigma, Q_N, q_N, F_N, \delta_N)$ be an NFA, and let $M = (\Sigma, Q_M, q_M, F_M, \delta_M)$ be its determinization as defined in class. We prove that for every $w \in \Sigma^*$, $\delta_M^*(q_M, w) = \delta_N^*(q_0, w)$ using induction on the length of w, say n.

Base case. $n = 0 \implies |w| = 0 \implies w = \lambda$.

$$\begin{aligned}
\delta_M^*(q_M, w) &= \delta_M^*(\{q_0\}, \lambda) && w = \lambda,\ q_M = \{q_0\} \\
&= \{q_0\} && \text{by definition} \\
&= \delta_N^*(q_0, \lambda) \\
&= \delta_N^*(q_0, w)
\end{aligned}$$

Since the LHS = RHS the base case holds.
Inductive step. For some $n \geq 0$, assume that $\delta_M^*(q_M, w) = \delta_N^*(q_0, w)$ where $|w| = n$. Now for $k = n$, consider a w'' such that $|w''| = k + 1$ and $w'' = w'a$, where $w' \in \Sigma^*$ such that $|w'| = k$ and $a \in \Sigma$. Now we evaluate $\delta_M^*(q_M, w'')$.

$$\forall p \in Q_M \quad \delta_M^*(p, wa) = \delta_M(\delta_M^*(p, w), a) \qquad \text{where } w \in \Sigma^* \text{ and } a \in \Sigma \tag{1}$$

$$\forall p \in Q_N \quad \delta_N^*(p, wa) = \bigcup_{q \in \delta_N^*(p,w)} \delta_N(q, a) \qquad \text{where } w \in \Sigma^* \text{ and } a \in \Sigma \tag{2}$$

$$S \subseteq Q_N \qquad \delta_M(S, a) = \bigcup_{q \in S} \delta_N(q, a) \qquad \text{where } a \in \Sigma \tag{3}$$

$$\begin{aligned}
\delta_M^*(q_M, w'') &= \delta_M^*(\{q_0\}, w'a) && w'' = w'a,\ q_M = \{q_0\} \\
&= \delta_M(\delta_M^*(\{q_0\}, w'), a) && \text{by property 1} \\
&= \delta_M(\delta_N^*(q_0, w'), a) && \text{by the induction hypothesis} \\
&= \bigcup_{q \in \delta_N^*(q_0, w')} \delta_N(q, a) && \text{by property 2} \\
&= \delta_N^*(q_0, w'a) && \text{by property 3} \\
&= \delta_N^*(q_0, w'') && w'a = w''
\end{aligned}$$

Conclusion. By the principle of induction, we see that for every $w \in \Sigma^*$, $\delta_M^*(q_M, w) = \delta_N^*(q_0, w)$ for $|w| = n$ where $n \geq 0$. $\square$

(5) Assume for the sake of contradiction that A is regular. Then A has a pumping constant p. Now consider the word $w = a^p b^{p+1}$ (i.e. there are p a's and $p+1$ b's). Then we can write w as $w = xyz$ such that the properties of the pumping lemma hold. By the property that states that $|xy| \leq p$ and $|y| \geq p$, we can conclude that y consists of only a's (so does xy). By the lemma $xy^n z \in A\, \forall n \geq 0$, there exists a $w' = xy^2z$ such that $w' \in A$. However, since $|xy| \leq p$ and $|y| \geq p$ there are now at least $p+1$ a's but we also still have $p+1$ b's. Therefore, for w' $\#a$'s $\geq \#b$'s and $w' \notin A$. This is a contradiction to our initial claim that A is regular and we conclude that A is not regular.