Name: Rollno:

CSE340: Theory of Computation (Homework Assignment 1)

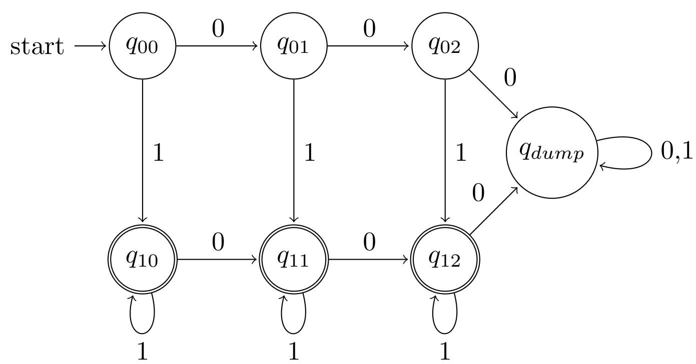
Due Date: 21st August, 2017 (in class)

Total Number of Pages: 4 Total Points 50

Question 1. (18 points) Give DFAs for the following languages.

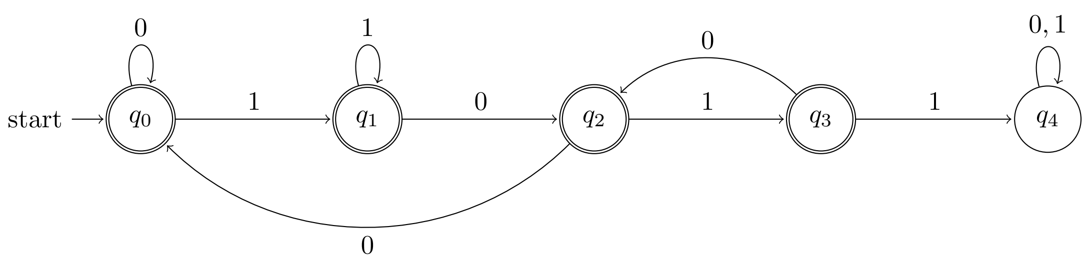
(a) $A = \{x \in \{0,1\}^* \mid \#_0(x) \leq 2 \text{ and } \#_1(x) \geq 1\}$

Solution:



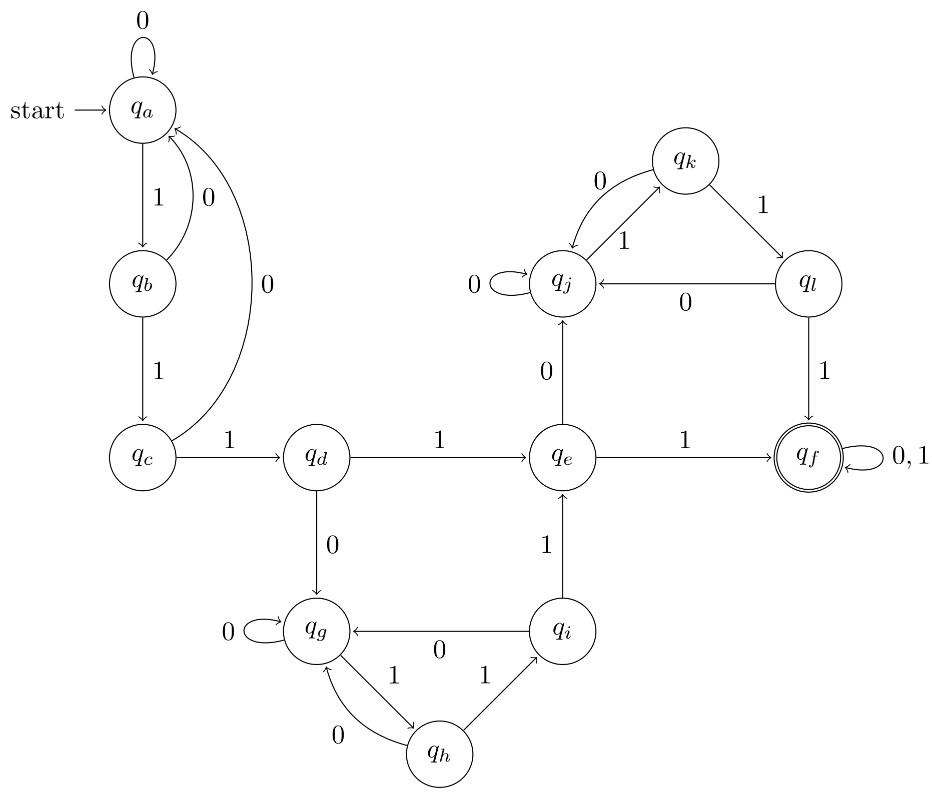
(b) $B = \{x \in \{0,1\}^* \mid x \text{ does not contain the substring } 1011\}$

Solution:



(c) $C = \{x \in \{0,1\}^* \mid x \text{ has at least 3 occurrences of 3 consecutive 1's with overlapping}\}$
(For example the string 11111 is in the language C.)

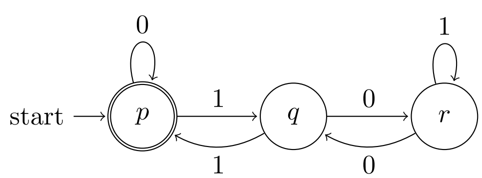
Solution:



Question 2. (12 points) Give DFAs accepting the same language as the following regular expressions using the minimum number of states. Give reason why you cannot have a DFA with lesser number of states.

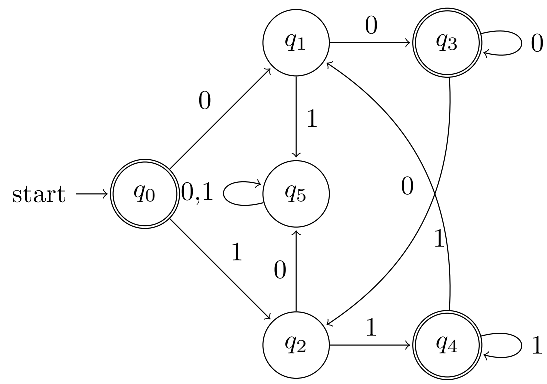
(a) $(0 + 1(01^*0)^*1)^*$

Solution:



(b) $(000^* + 111^*)^*$

Solution:



Question 3. (10 points) For languages L_1 and L_2 over Σ, define

$$\text{Mix}(L_1, L_2) = \{w \in \Sigma^* \mid w = x_1 y_1 x_2 y_2 \ldots x_k y_k, \text{ where } x_1 x_2 \ldots x_k \in L_1 \text{ and } y_1 y_2 \ldots y_k \in L_2, \text{ each } x_i, y_i \in \Sigma^*\}.$$

Show that if L_1 and L_2 are regular then $\text{Mix}(L_1, L_2)$ is also regular.

Solution: Let $D_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ and $D_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ be two DFAs such that $L_1 = L(D_1)$ and $L_2 = L(D_2)$. We define an NFA $N = (Q, \Sigma, \delta q_s, F)$ that accepts $\text{Mix}(L_1, L_2)$ as follows:

- $Q = Q_1 \times Q_2$ is the set of states.
- The transition function δ of N is defined as

$$\delta((x, y), a) \quad = \quad \{(\delta_1(x, a), y), (x, \delta_2(y, a))\}$$

- (q_{01}, q_{02}) is the start state.
- $F = F_1 \times F_2$ is the set of accept states.

Question 4. (10 points) Let Σ and Δ be two alphabets and let $h : \Sigma \to \Delta^*$. Extend h to be a function from Σ^* to Δ^* as follows:

$$\begin{aligned} h(\epsilon) &= \epsilon, \\ h(wa) &= h(w)h(a) \quad \text{where } w \in \Sigma^*,\ a \in \Sigma. \end{aligned}$$

(Such a function h is called a *homomorphism*.)

Now, for $L \subseteq \Sigma^*$,

$$h(L) = \{h(w) \in \Delta^* \mid w \in L\}.$$

Also, for $L \subseteq \Delta^*$,

$$h^{-1}(L) = \{w \in \Sigma^* \mid h(w) \in L\}.$$

(a) Prove that if $L \subseteq \Sigma^*$ is regular, then so is $h(L)$.

Solution: Let R be a regular expression such that $L = L(R)$. Now for every symbol $a \in \Sigma$ and every occurrence of a in R, replace a with the string $h(a)$, to get a new regular expression over Δ (say R'). It is easy to see that $L(R') = h(L)$.

(b) Prove that if $L \subseteq \Delta^*$ is regular, then so is $h^{-1}(L)$.

Solution: Let $D = (Q, \Delta, \delta, q_0, F)$ be a DFA such that $L = L(D)$.
We will construct a DFA $D' = (Q, \Sigma, \delta', q_0, F)$ for $h^{-1}(L)$. Note that the set of states, the start state and the set of accept states of D' is same as that of D and for obvious reasons the alphabet of D' is Σ.
Let $a \in \Sigma$. Define $\delta'(q, a) = p$, where p is the state reached by D on reading the string $h(a)$ from the state q (typically this is denoted by the notation $\delta'(q, a) = \delta(q, h(a))$).