A REVISED Solution to Assignment 1

1. (a). We have to show two things: (i) if $L=L^+$, then $L^2 \subseteq L$ and (ii) if $L^2 \subseteq L$, then $L=L^+$.

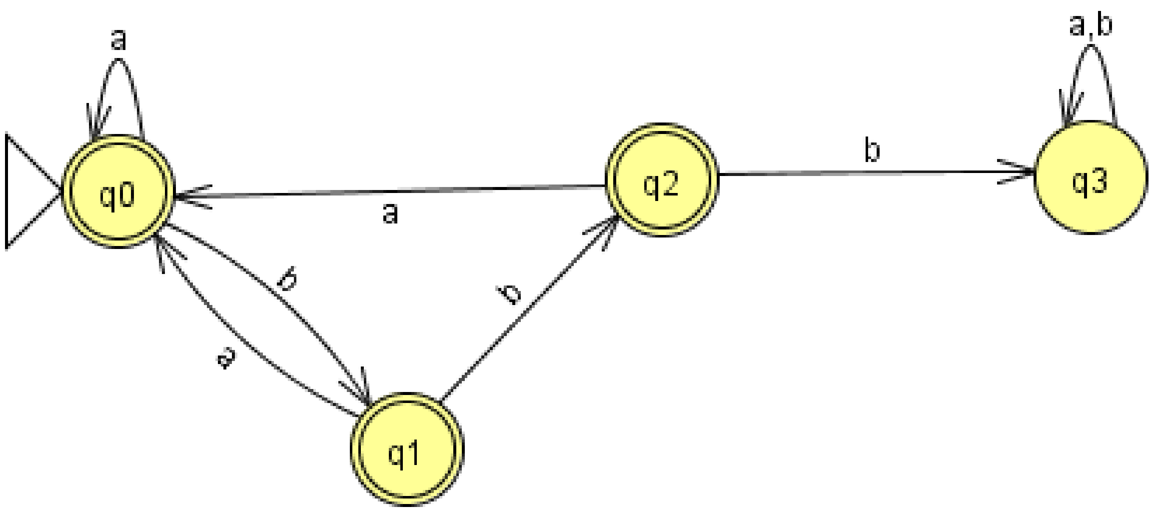
(i) Suppose $L=L^+$. Since $L = L^+ = L \cup L^2 \cup \ldots$, it is obvious that $L^2 \subseteq L^+$ and hence $L^2 \subseteq L$.

(ii) Suppose $L^2 \subseteq L$. We need to show (1) $L \subseteq L^+$ and (2) $L^+ \subseteq L$. It is obvious that (1) holds. To show (2), we will simply use the definition of $\subseteq$ and show that for any string w, whenever w is in L^+, w is in L as well. Suppose w is in L^+. Then there is an integer n such that w is in L^n. This means, there are n strings $x_1,\ldots,x_n$ in L such that $w=x_1\ldots x_n$. We will show the result using the proof by induction method on n, the number of strings that form w. For the basis case (n=1), we have that $w=x_1 \in L^+$, which means $x_1 \in L$, and hence $w \in L$. Induction: suppose the statement holds for some integer n, that is, if s $w=x_1\ldots x_n$ is in L^+, for some integer n, then $w \in L$ and also $x_i \in L$, for $1 \le i \le n$. Let y be any string in L^{n+1}. Then, y can be viewed as having two parts, t and x_{n+1}, that is, $y = t.x_{n+1}$, where $t=x_1\ldots x_n$. It then follows from the basis and the inductive hypothesis that both t and x_{n+1} are present in both L^n as well as in L. This in turn implies that y (which consists of two strings in L) is in L^2, from which we can conclude that $y \in L$, since $L^2 \subseteq L$.

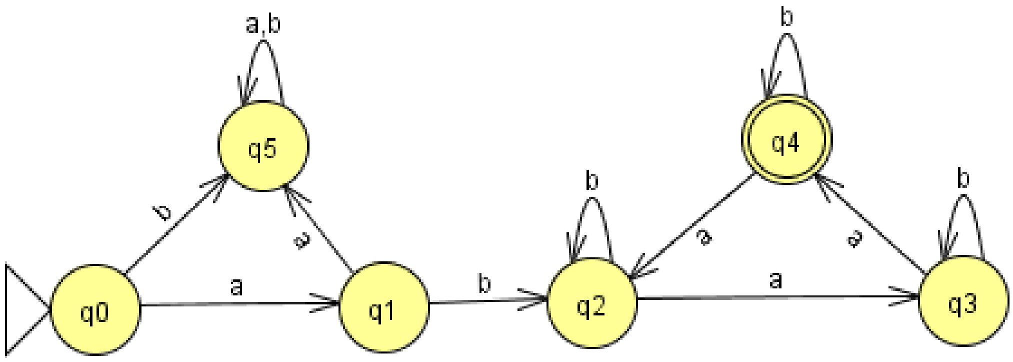
(b). Suppose $L_1 \subseteq L_2$ but $\min\{|x|: x \in L_1\} < \min\{|y|: y \in L_2\}$. This means, there is a string $w \in L_1$ such that $|w| < \min\{|y|: y \in L_2\}$. However, this implies that $w \notin L_2$, which is a contradiction to our assumption that $L_1 \subseteq L_2$.

2. (a). Possible cases: (1) $L_1 = \emptyset$; (2) $L_2 = \{\lambda\}$; (3) $L_2 = \Sigma^*$ and $\lambda \in L_1$; (4) $L_2 = \Sigma^+$ and $\lambda \in L_1$.
(5). $L_1=L_2 = L_k = \{w \in \Sigma^*: |w| = n*k, \text{ for all } n \ge 0\}$, for any fixed integer $k \ge 0$.
(b). Possible cases: (1) $L = \emptyset$; (2) $L = \Sigma^*$; (3) $L = \Sigma^+$.

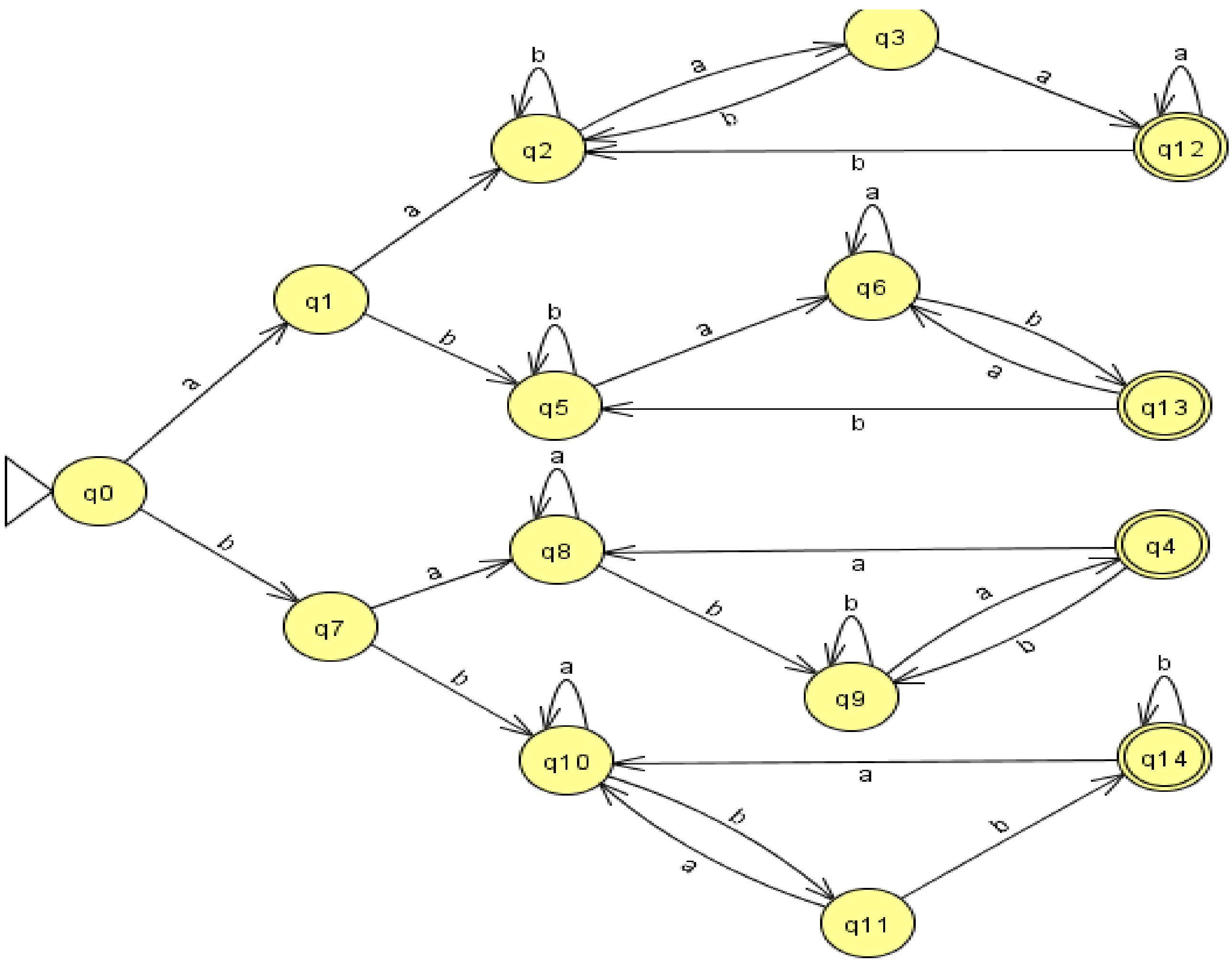
3. (a).



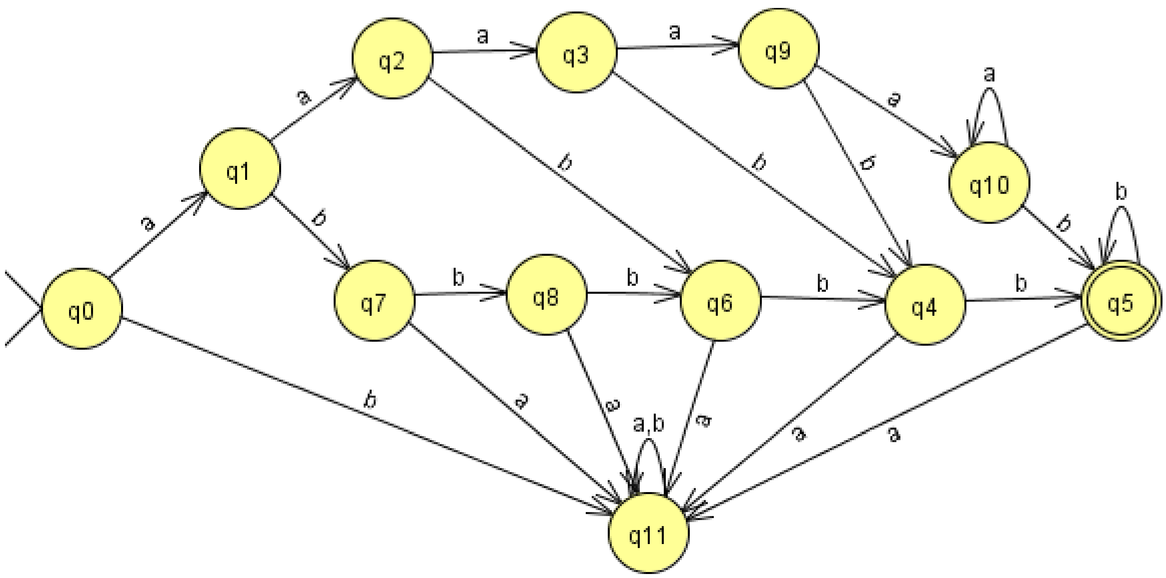
3 . (b).


a,b
b
q5
q4
b
b
b
a
a
a
q0
a
q1
b
q2
a
q3

3. (c).


q3
b
a
a
a
b
q2
q12
b
a
a
q6
q1
b
b
a
b
a
q5
q13
b
a
q0
q8
q4
b
a
a
q7
b
b
a
b
b
q9
b
a
b
q10
q14
a
b
b
a
q11

3. (d).



4. If r1, r2, and r3 are regular expressions for L1, and L2, and L3, then (r1+r2)r3 would be a desired regular expression. Below we give are regular expressions r1 to r3 obtained using the state elimination/reduction technique discussed in the lectures.

   $r1 = (a + ba + bba)^*(\lambda + b + bb)$

   $r2 = abb^*ab^*a(\lambda+b+ ab^*ab^*a)^*$

   $r3 = aa(a+b)^*aa + ab(a+b)^*ab + ba(a+b)^*ba + bb(a+b)^*bb$

5. To answer this question, it is important to note that every string in L satisfies two conditions, one related to the order of symbols (pattern) and the other related to their counts. So for every string x in L, all the a's appear before the b's AND the number of a's is 1 less than the number of b's. From this, we can say that $w \in \overline{L}$, if w has "ba" as a substring or $n_q(w) \neq n_b(w)+1$. The latter can also be expressed as $n_a(w) \geq n_b(w)$ or $n_b(w) > n_a(w)+1$).

6. For this, we can use the DFAs M_1 and M_3 we gave for L_1 and L_3 in question 3, and modify them as follows. Change the final states in M_3 to non-final and vice versa. This yields a DFA N_3 for $\overline{L3}$ shown in Fig. 7-1. We then construct an NFA S for $L_1 \cup \overline{L3}$ by creating a new initial state and connecting it to the initial states of M_1 and N_3 using λ-transitions (Fig. 7-2). As the last step, we use the subset construction technique to convert S into an equivalent DFA, shown in Fig. 7-3.

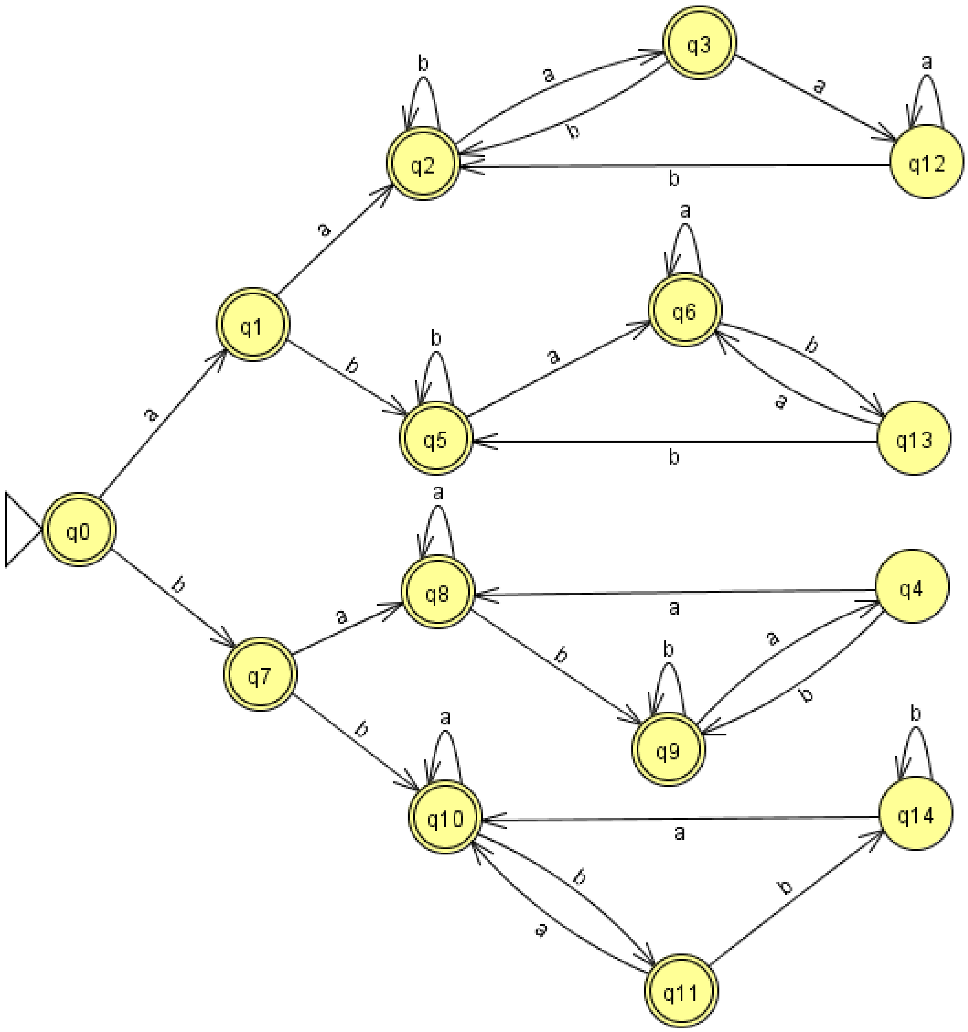


Fig. 7-1- A DFA N_3 for the complement of L_3

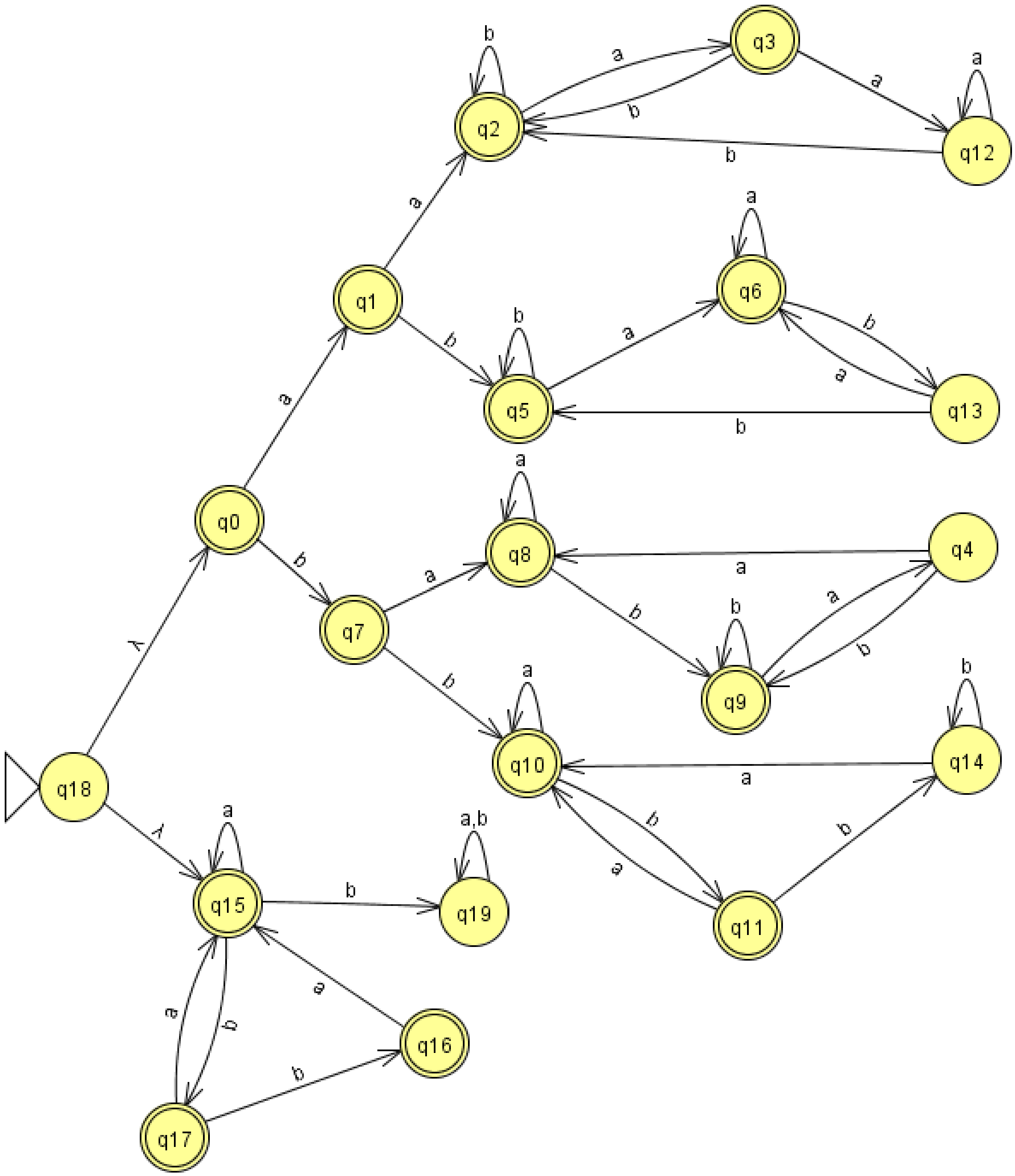


Fig. 7-2- An NFA for ($L_1 \cup \overline{L3}$)