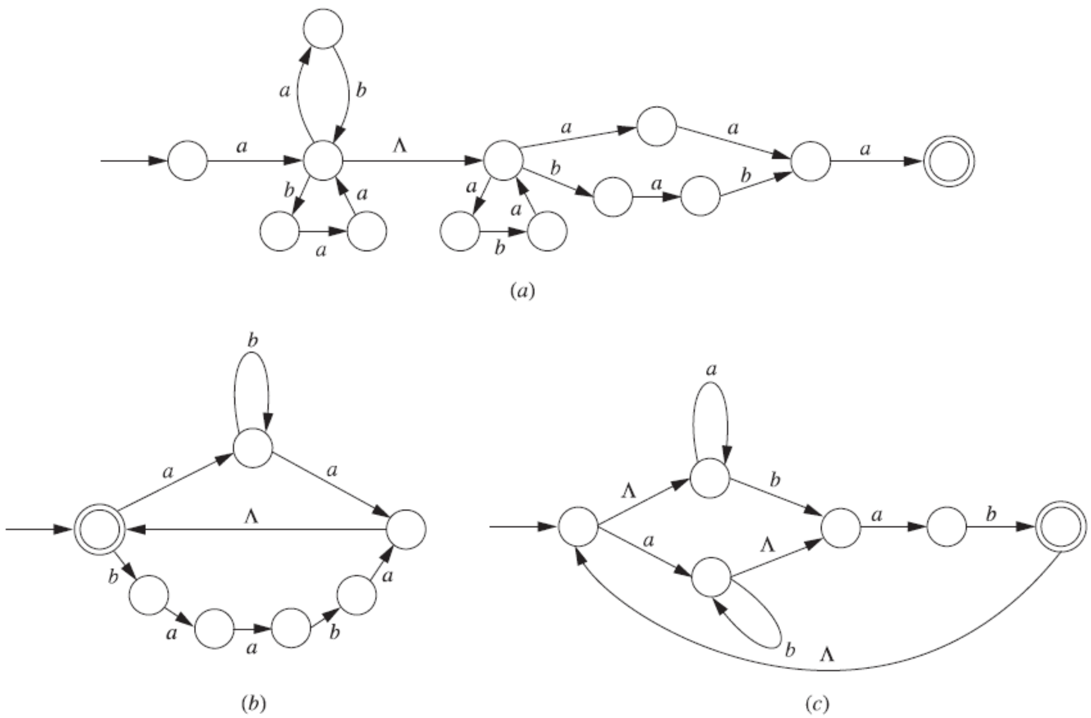
CMPS 130
Spring 2016
Homework Assignment 6
Problems are from Martin 4th edition.

Chapter 3 (p.117): 20abc, 24, 28abc, 31, 37cd, 38ace

1. Problem 3.20abc
   For each of the NFAs shown in Figure 3.35, find a regular expression corresponding to the language it accepts.



   Solution:
   a. $a(ab + baa)^*(aba)^*(aa + bab)a$
   b. $(ab^*a + baaba)^*$
   c. $\big((a^*b + ab^*)ab\big)\big((a^*b + ab^*)ab\big)^*$ This is sometimes written as $\big((a^*b + ab^*)ab\big)^+$

2. Problem 3.24
   Let $M = (Q, \Sigma, q_0, A, \delta)$ be an NFA with no λ-transitions. Show that for every $q \in Q$ and $\sigma \in \Sigma$, $\delta^*(q, \sigma) = \delta(q, \sigma)$.

   Proof:
   Since M has no λ-transitions, the λ-closure operation is trivial, i.e. $\lambda(S) = S$ for any $S \subseteq Q$. Let $q \in Q$ and $\sigma \in \Sigma$. Then

$$\begin{aligned}\delta^*(q,\sigma) &= \delta^*(q,\lambda\sigma) && \text{(meaning of } \lambda) \\ &= \lambda\left(\bigcup_{p\in\delta^*(q,\lambda)} \delta(p,\sigma)\right) && \text{(recursive definition of } \delta^*)\end{aligned}$$

$$= \bigcup_{p \in \delta^*(q,\lambda)} \delta(p,\sigma) \qquad \text{(no } \lambda\text{-transitions in } M\text{)}$$

$$= \bigcup_{p \in \lambda(\{q\})} \delta(p,\sigma) \qquad \text{(recursive definition of } \delta^*\text{)}$$

$$= \bigcup_{p \in \{q\}} \delta(p,\sigma) \qquad \text{(no } \lambda\text{-transitions in } M\text{)}$$

$$= \delta(q,\sigma) \qquad \text{(meaning of above notation)}$$

3. Problem 3.28abc
   Let $M = (Q, \Sigma, q_0, A, \delta)$ be an NFA. This exercise involves properties of the λ-closure of a set S. Since $\lambda(S)$ is defined recursively, structural induction can be used to show that $\lambda(S)$ is a subset of some other set.
   a. Show that if S and T are subsets of Q for which $S \subseteq T$, then $\lambda(S) \subseteq \lambda(T)$.
   b. Show that for any $S \subseteq Q$, $\lambda\big(\lambda(S)\big) = \lambda(S)$.
   c. Show that if $S, T \subseteq Q$, then $\lambda(S \cup T) = \lambda(S) \cup \lambda(T)$.

   Solution:
   Recall the recursive definition of $\lambda(S)$ for $S \subseteq Q$: (1) $S \subseteq \lambda(S)$ and (2) $q \in \lambda(S) \Rightarrow \delta(q,\lambda) \subseteq \lambda(S)$.

   a. **Proof:** Let $q \in \lambda(S)$. We show that $q \in \lambda(T)$ by structural induction on q.
      I. If $q \in S$, then $q \in T \subseteq \lambda(T)$ by the recursive definition of $\lambda(T)$, part (1).
      II. Let $q \in \lambda(S)$ and assume $q \in \lambda(T)$. We must show that $\delta(q,\lambda) \subseteq \lambda(T)$. But this follows from the recursive definition of $\lambda(T)$, part (2).

   b. **Proof:** Since $S \subseteq \lambda(S)$, we have $\lambda(S) \subseteq \lambda\big(\lambda(S)\big)$ by part (a). It remains to show $\lambda\big(\lambda(S)\big) \subseteq \lambda(S)$. Let $q \in \lambda\big(\lambda(S)\big)$. We show that $q \in \lambda(S)$ by structural induction on q.
      I. If $q \in \lambda(S)$, then certainly $q \in \lambda(S)$.
      II. Let $q \in \lambda\big(\lambda(S)\big)$ and assume $q \in \lambda(S)$. We must show that $\delta(q,\lambda) \subseteq \lambda(S)$. But this follows from the recursive definition of $\lambda(S)$, part (2).

   c. **Proof:** Since $S \subseteq S \cup T$ and $T \subseteq S \cup T$, part (a) gives $\lambda(S) \subseteq \lambda(S \cup T)$ and $\lambda(T) \subseteq \lambda(S \cup T)$, and hence $\lambda(S) \cup \lambda(T) \subseteq \lambda(S \cup T)$. It remains to show that $\lambda(S \cup T) \subseteq \lambda(S) \cup \lambda(T)$. Let $q \in \lambda(S \cup T)$. We show that $q \in \lambda(S) \cup \lambda(T)$ by structural induction on q.
      I. If $q \in S \cup T$ then either $q \in S$ or $q \in T$. But $S \subseteq \lambda(S)$ and $T \subseteq \lambda(T)$ by the recursive definitions of $\lambda(S)$ and $\lambda(T)$, so that either $q \in \lambda(S)$ or $q \in \lambda(T)$. Thus $q \in \lambda(S) \cup \lambda(T)$.
      II. Let $q \in \lambda(S \cup T)$ and assume $q \in \lambda(S) \cup \lambda(T)$. We must show that $\delta(q,\lambda) \subseteq \lambda(S) \cup \lambda(T)$. We have either $q \in \lambda(S)$ or $q \in \lambda(T)$. If $q \in \lambda(S)$, then $\delta(q,\lambda) \subseteq \lambda(S)$ by the recursive definition of $\lambda(S)$, part (2). If $q \in \lambda(T)$, then $\delta(q,\lambda) \subseteq \lambda(T)$ by the recursive definition of $\lambda(T)$, part (2). Thus either $\delta(q,\lambda) \subseteq \lambda(S)$ or $\delta(q,\lambda) \subseteq \lambda(T)$. It follows that $\delta(q,\lambda) \subseteq \lambda(S) \cup \lambda(T)$, as required.

4. Problem 3.31

Let $M = (Q, \Sigma, q_0, A, \delta)$ be a DFA, and let $M_1 = (Q, \Sigma, q_0, A, \delta_1)$ be the NFA with no λ-transitions for which $\delta_1(q, \sigma) = \{\delta(q, \sigma)\}$ for every $q \in Q$ and $\sigma \in \Sigma$. Show that for every $q \in Q$ and $x \in \Sigma^*$, $\delta_1^*(q, x) = \{\delta^*(q, x)\}$. Recall that the two functions δ^* and δ_1^* are defined differently.

Proof:

We proceed by structural induction on *x*.

I. Let $x = \lambda$. Then

$$\begin{aligned} \delta_1^*(q, \lambda) &= \lambda(\{q\}) && \text{(base definition of } \delta_1^*) \\ &= \{q\} && \text{(no } \lambda\text{-transitions in } M_1) \\ &= \{\delta^*(q, \lambda)\} && \text{(base definition of } \delta^*) \end{aligned}$$

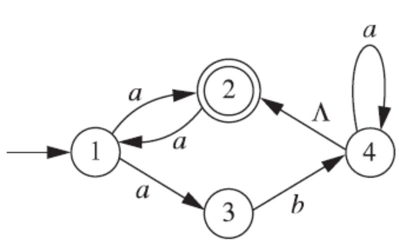
II. Let $x \in \Sigma^*$ and assume $\delta_1^*(q, x) = \{\delta^*(q, x)\}$. We must show $\delta_1^*(q, x\sigma) = \{\delta^*(q, x\sigma)\}$ for any $\sigma \in \Sigma$.

$$\begin{aligned} \delta_1^*(q, x\sigma) &= \lambda\left(\bigcup_{p \in \delta_1^*(q,x)} \delta_1(p, \sigma)\right) && \text{(recursive definition of } \delta_1^*) \\ &= \bigcup_{p \in \delta_1^*(q,x)} \delta_1(p, \sigma) && \text{(no } \lambda\text{-transitions in } M_1) \\ &= \bigcup_{p \in \{\delta^*(q,x)\}} \delta_1(p, \sigma) && \text{(induction hypothesis)} \\ &= \delta_1(\delta^*(q, x), \sigma) && \text{(meaning of the notation)} \\ &= \{\ \delta(\delta^*(q, x), \sigma)\ \} && \text{(definition of } \delta_1) \\ &= \{\ \delta^*(q, x\sigma)\} && \text{(recursive definition of } \delta^*) \end{aligned}$$

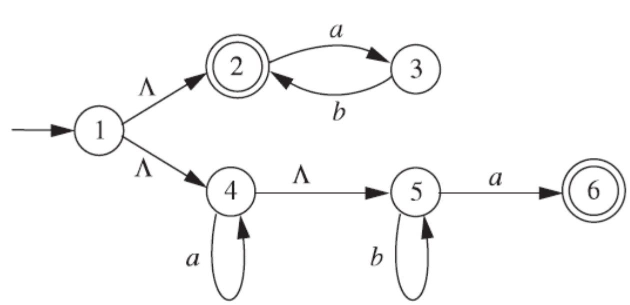
The result now follows for all *x*.

5. Problem 3.37cd

In each part of Figure 3.36 is pictured an NFA. Use the algorithm described in the proof of Theorem 3.17 to draw an NFA with no λ-transitions accepting the same language.



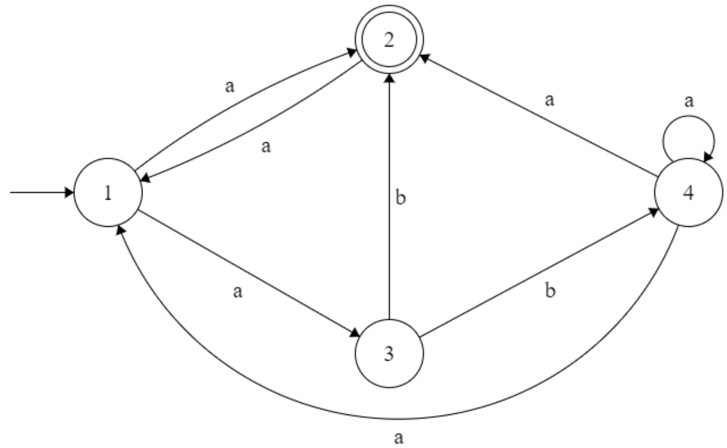
(*c*)



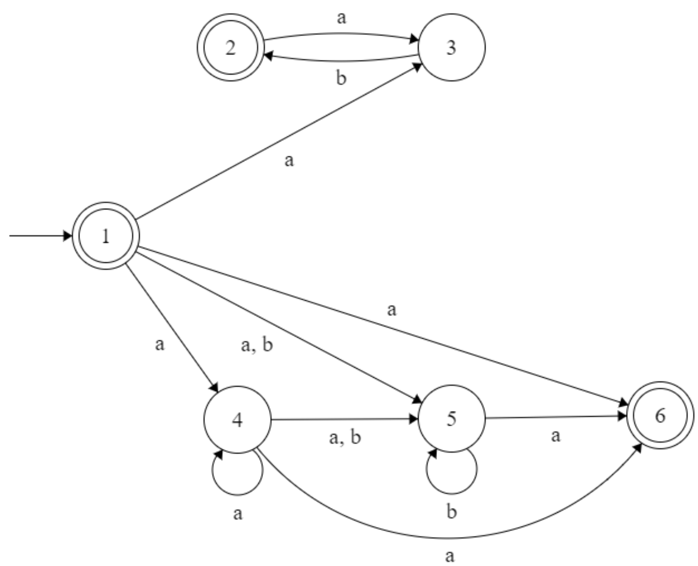
(*d*)

Solution:

a. Since λ is not accepted by the original NFA, there is no need to add to the set of accepting states, i.e. $A_1 = A = \{2\}$.

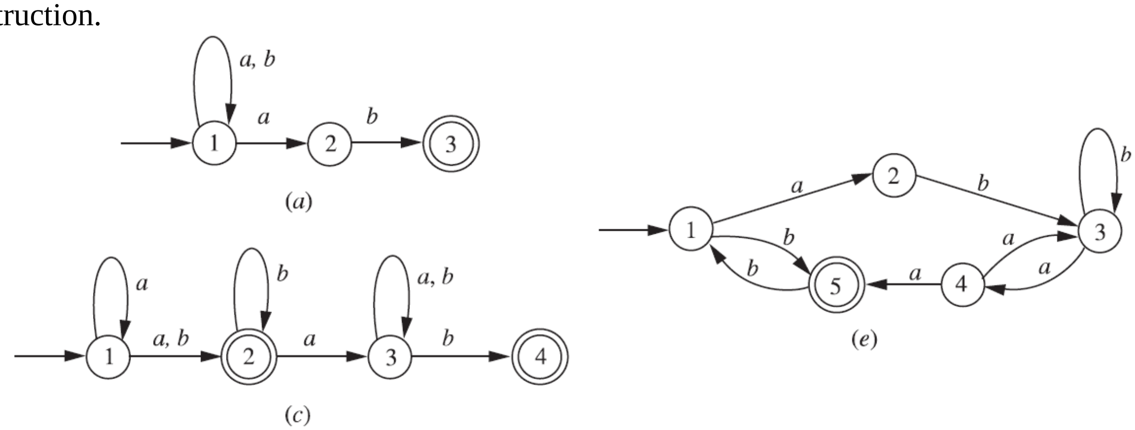


b. Since λ is accepted by the original NFA, we have $A_1 = A \cup \{1\} = \{1, 2, 6\}$.



6. Problem 3.38ace

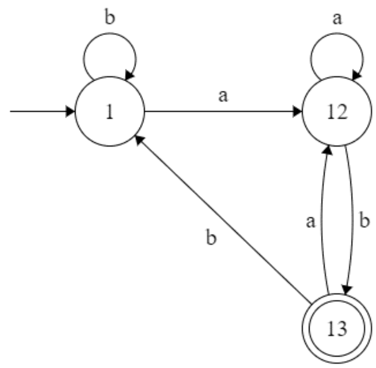
Each part of Figure 3.37 pictures an NFA. Using the subset construction, draw a DFA accepting the same language. Label the final picture so as to make it clear how it was obtained from the subset construction.



Solution:

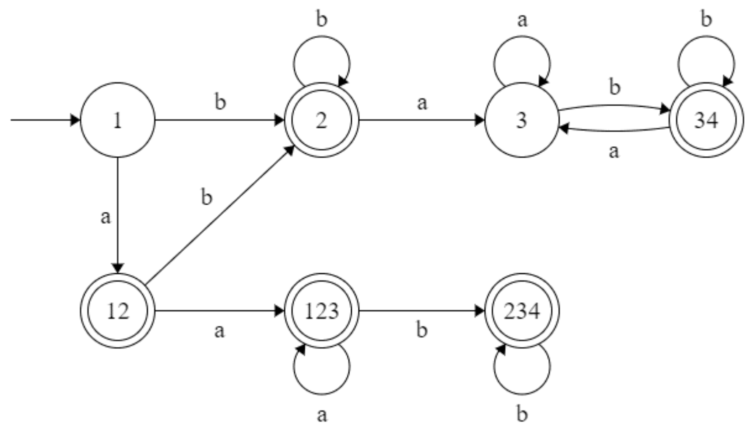
a.

	a	***b***
1	12	1
12	12	13
13	12	1



c.

	a	***b***
1	12	2
12	123	2
2	3	2
3	3	34
123	123	234
34	3	34
234	3	234



e.

	a	***b***
1	2	5
2	∅	3
3	4	3
4	35	∅
5	∅	1
∅	∅	∅
35	4	13
13	24	35
24	35	3