Chapter 4

Nondeterminism and Kleene's Theorem

4.1. (a) $\delta^*(1,b) = \delta(1,b) = \{1,2\}$. $\delta^*(1,bb) = \delta(1,b) \cup \delta(2,b) = \{1,2\} \cup \emptyset = \{1,2\}$.
(b) $\delta^*(ba) = \delta(1,a) \cup \delta(2,a) = \{4,6,7\} \cup \{3\} = \{3,4,6,7\}$. $\delta^*(bab) = \cup_{p \in \{3,4,6,7\}} \delta(p,b) = \{1\} \cup \emptyset \cup \{9\} \cup \{8\} = \{1,8,9\}$.
(c) $\{1,2,9\}$ (d) $\{1,8,9\}$ (e) $\emptyset$

4.2. (a) $\delta^*(1,a) = \cup_{p \in \delta^*(1,\Lambda)} \delta(p,a) = \delta(1,a) = \{1,2\}$. $\delta^*(1,ab) = \delta(1,b) \cup \delta(2,b) = \{1,3\}$.
(b) $\delta^*(1,aba) = \delta(1,a) \cup \delta(3,a) = \{1,2,4\}$. $\delta^*(1,abaa) = \cup_{p \in \{1,2,4\}} \delta(p,a) = \{1,2,3,5\}$.
$\delta^*(1,abaab) = \delta(1,b) \cup \delta(2,b) \cup \delta(3,b) \cup \delta(5,b) = \{1,3,4,5\}$.

4.3. $\delta^*(q,a) = \cup_{p \in \delta^*(q,\Lambda)} \delta(p,a) = \cup_{p \in \{q\}} \delta(p,a) = \delta(q,a)$.

4.4. $\lceil \log_2 n \rceil$ (i.e., the smallest integer $\geq \log_2 n$). Reason: if there is an NFA accepting L with k states, then by the subset construction, there is an FA accepting L with no more than 2^k states. If there were an NFA with fewer than $\lceil \log_2 n \rceil$ states, there would be one with fewer than $\log_2 n$ states, and so there would be an FA with fewer than $2^{\log_2 n} = n$ states.

4.5. For every $q \in Q$, $\delta^*(q,\Lambda) = \{q\}$; for every $q \in Q$, $y \in \Sigma^*$, and $a \in \Sigma$, $\delta^*(q,ay) = \cup_{p \in \delta(q,a)} \delta^*(q,y)$.

4.6. An example is the language $\{\Lambda, 0\} \subseteq \{0\}^*$. Since Λ is in the language, the initial state must be an accepting state. Since 0 is also in the language, the 0-transition from the initial state must go to an accepting state. The only way for this to happen with just one accepting state is for all the strings 0^k to be accepted.

4.7. Yes. If $M = (Q, \Sigma, q_0, A, \delta)$ is any NFA for which $\Lambda \notin L(M)$, we can construct another NFA $M' = (Q \cup \{p\}, \Sigma, q_0, \{p\}, \delta')$ accepting $L(M)$, as follows. The state p is a new state not in Q, and it is the only accepting state of M'. All transitions in M are still present in M'. In addition, for every transition of the form $q \xrightarrow{a} r$, where $r \in A$, there is an additional transition $q \xrightarrow{a} p$ in M'. There are no transitions from p. Then it is easy to see that M' accepts $L(M)$. On the one hand, if $x = a_1 a_2 \ldots a_n \in L(M)$, then there is a sequence of transitions

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} \cdots \xrightarrow{a_{n-1}} q_{n-1} \xrightarrow{a_n} q_n$$

for some $q_n \in A$. The string is accepted by M' because of the sequence of transitions that agrees with this except that the last one is replaced by $q_{n-1} \xrightarrow{a_n} p$. On the other hand, if $x \in L(M')$, then there is a sequence of transitions ending with $q_n \xrightarrow{a_n} p$ by which x is accepted, and the state p doesn't appear earlier in the sequence. Therefore, there is a corresponding sequence of transitions by which M accepts x.

4.8. For both parts, we observe that for any sequence of transitions

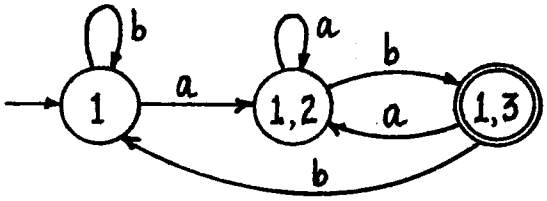
$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} \cdots \xrightarrow{a_{n-1}} q_{n-1} \xrightarrow{a_n} q_n$$

by which a string is accepted by M, each state q_i has the property that it is reachable from q_0 and some element of A is reachable from it. Therefore, neither the modification in (a) nor the one in (b) has any effect on the set of strings accepted by the NFA.

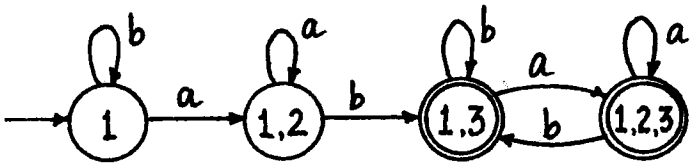
4.9. (a) There can be no more than m^n paths corresponding to x, since at each step there are at most m choices for the next state.

(b) If $x = x_1x_2$, $\delta^*(q_0, x) = \cup_{p\in\delta^*(q_0,x_1)}\delta^*(p, x_2)$. In order to use this formula to compute $\delta^*(q_0, x)$, it is necessary to know only the states in $\delta^*(q_0, x_1)$, not all the possible paths by which those states are reached. The computation tree really contains more information than we need; "pruning" it means ignoring everything except the set of states the NFA might end up at at each level of the tree.

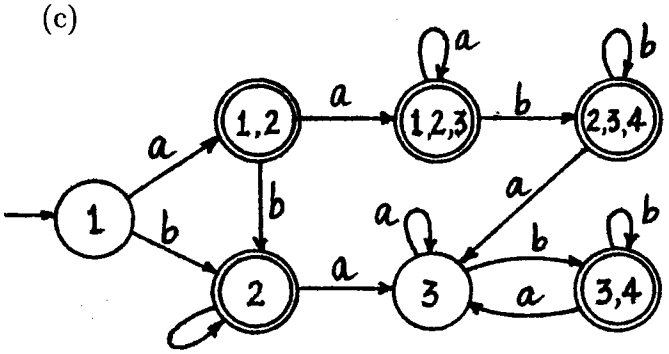
4.10. (a)



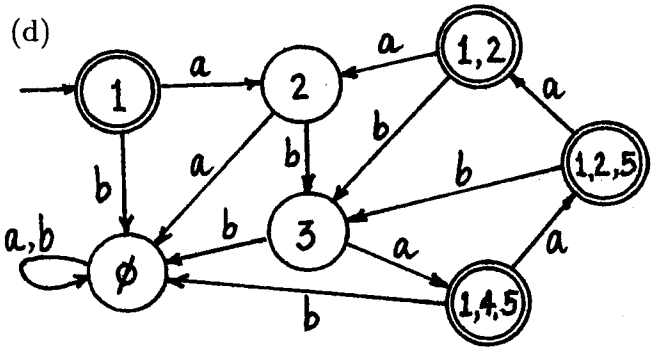
(b)



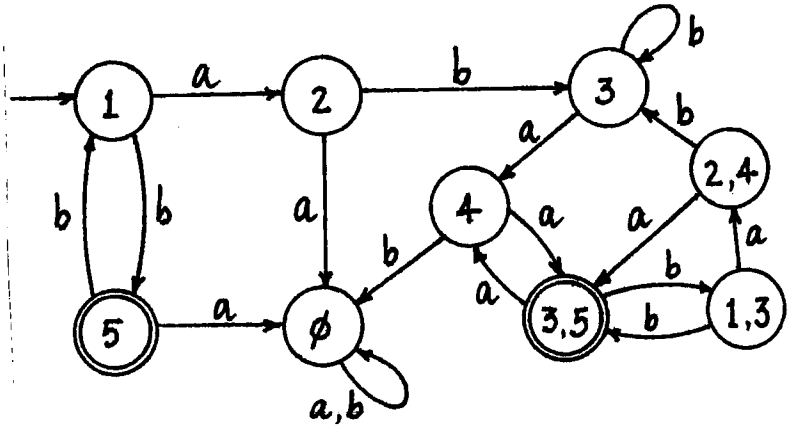
(c)



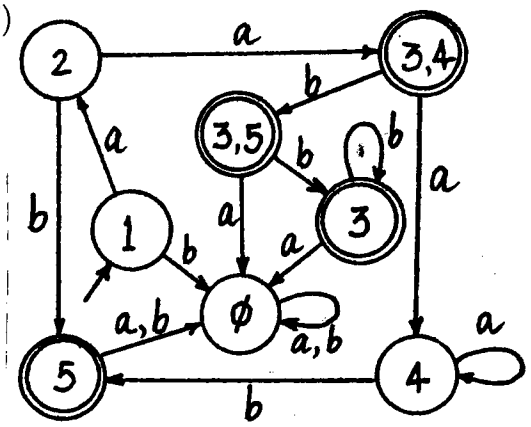
(d)



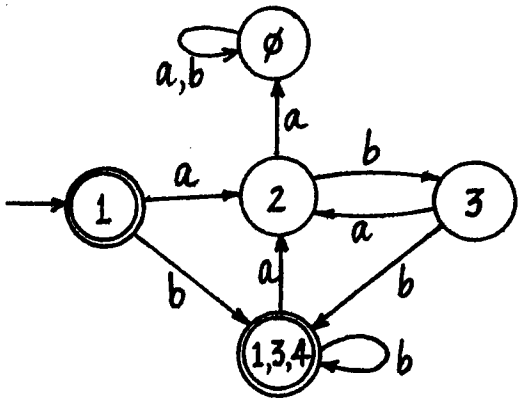
(e)



4.10 (f)



(g)



4.11. No, because there may be a string x so that in M, one sequence of transitions corresponding to x ends at an accepting state and another sequence doesn't. Then M' has this property also, which means that $x \in L(M)$ and $x \in L(M')$. An example can easily be constructed with just two states.

4.12. The conclusion of Theorem 3.4 does not hold in the case of unions. For example, if M_1 has an a-transition from q_1 to an accepting state, and M_2 has no a-transitions from q_2, then there are no a-transitions from (q_1, q_2), and so a is not accepted by M even though it is in $L(M_1) \cup L(M_2)$.

The conclusion holds in the case of intersections. On the one hand, if $x = a_1 a_2 \dots a_n$ is accepted by both M_1 and M_2, then there are sequences of transitions

$$q_1 \xrightarrow{a_1} q_1^{(1)} \xrightarrow{a_2} q_1^{(2)} \xrightarrow{a_3} \dots \xrightarrow{a_{n-1}} q_1^{(n-1)} \xrightarrow{a_n} q_1^{(n)}$$

and

$$q_2 \xrightarrow{a_1} q_2^{(1)} \xrightarrow{a_2} q_2^{(2)} \xrightarrow{a_3} \dots \xrightarrow{a_{n-1}} q_2^{(n-1)} \xrightarrow{a_n} q_2^{(n)}$$

where the $q_1^{(i)}$'s are elements of Q_1, with $q_1^{(n)} \in A_1$, and the $q_2^{(i)}$'s are elements of Q_2, with $q_2^{(n)} \in A_2$. In this case, the sequence of transitions

$$(q_1, q_2) \xrightarrow{a_1} (q_1^{(1)}, q_2^{(1)}) \xrightarrow{a_2} \dots \xrightarrow{a_n} (q_1^{(n)}, q_2^{(n)})$$

is defined in M and causes x to be accepted by M. This argument can essentially be reversed, so that if x is accepted by M, then it is accepted by both M_1 and M_2.

The conclusion does not hold in the case of differences. The example given for unions is also a counterexample here.

4.13. (a) yes (b) no (c) yes

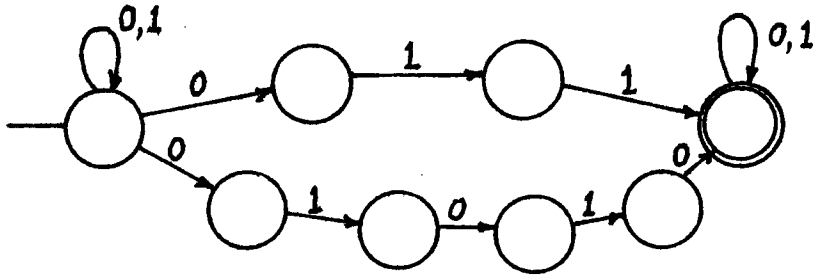
4.14. $aa^*b^*(a+b)(a+ba^*b^*(a+b))^*$

4.15. (a) $a(ab+baa)^*(aba)^*(aa+bab)a$
(b) $(ab^*a+baaba)^*$ (c) $((a^*b+ab^*)ab)^+$

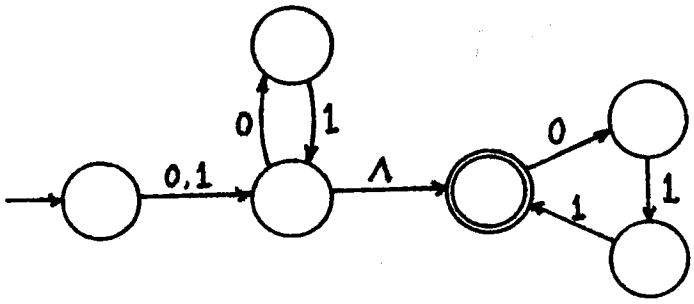
4.16. (a) $\{2,3,5\}$ (b) $\{1,2,5\}$ (c) $\{1,2,3,4,5\}$ (d) $\{3,5\}$ (e) $\{1,2,4,5\}$ (f) $\{1,2,3,4,5\}$

4.17. $\{1,3,4,6\}$

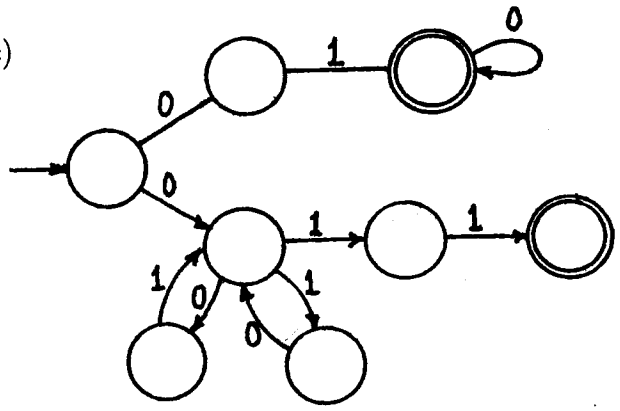
4.18. (a)



(b)



(c)



4.19. The new NFA-Λ, M^r, is constructed as follows. It has the same states as M, in addition to one new one, p_0, which will be the initial state of M^r. The only accepting state of M^r is q_0. All the transitions of M are present in M^r but in reverse; for example, if the transition $q \xrightarrow{a} r$ appears in M, then $r \xrightarrow{a} q$ appears in M^r, and if $q \xrightarrow{\Lambda} r$ appears in M, $r \xrightarrow{\Lambda} q$ appears in M'. Finally, M^r also has Λ-transitions from p_0 to all the states in the set A.

For any sequence of transitions in M from q_0 to $q_f \in A$, there is a sequence in M^r that starts at q_0, goes to q_f by a Λ-transition, and then follows the original path in reverse, ending up at q_0. On the other hand, for any sequence of transitions by which a string is accepted in M^r, the reverse sequence (except for the transition involving p_0) appears in M, so that the reverse of the string is accepted by M.

4.20. (a) Yes. (b) Yes. (c) No. The string ab is accepted by the first but not the second. (d) Yes. (e) No. The null string is accepted by the second but not the first.

4.21. The proof is by structural induction. We first observe that since there are no Λ-transitions in M_1, $\delta_1^*(q,\Lambda) = \{q\}$ and $\delta_1^*(q,xa) = \cup_{p\in\delta_1^*(q,x)}\delta_1(p,a)$, for every $q \in Q$, every

$x \in \Sigma^*$, and every $a \in \Sigma$.

For the basis step, we've already observed that $\delta_1^*(q, \Lambda) = \{q\}$, and $\delta^*(q, \Lambda) = q$ by definition of δ^*. Suppose that for some y, $\delta_1^*(q, y) = \{\delta^*(q, y)\}$ for every q. Then for $a \in \Sigma$,

$$\begin{aligned}
\delta_1^*(q, ya) &= \cup_{p \in \delta_1^*(q,y)} \delta_1(p, a) \\
&= \cup_{p \in \{\delta^*(q,y)\}} \delta_1(p, a) \\
&= \delta_1(\delta^*(q, y), a) \\
&= \{\delta(\delta^*(q, y), a)\} \\
&= \{\delta^*(q, ya)\}
\end{aligned}$$

The second equality uses the induction hypothesis, the fourth the definition of δ_1, and the last the definition of δ^*.

4.22. Yes, because for any set S, the Λ-closure of S is the same, no matter which way $\delta_1(q, \Lambda)$ is defined.

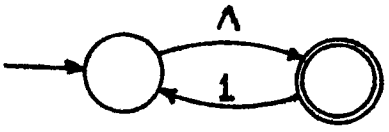
4.23. Let $M = (Q, \Sigma, q_0, A, \delta)$ be the NFA-Λ. Then the FA $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$ can be defined as follows. $Q_1 = 2^Q$; $q_1 = \{q_0\}$; A_1 is the set $\{q \in Q_1 \mid q_0 \in q \text{ or } q \cap A \neq \emptyset\}$, if $\Lambda(\{q_0\}) \cap A \neq \emptyset$, or $\{q \in Q_1 \mid q \cap A \neq \emptyset\}$, if $\Lambda(\{q_0\}) \cap A = \emptyset$.

4.24. If $x \neq \Lambda$, and $\delta_1^*(q_0, x)$ contains a state q for which $\Lambda(\{q\}) \cap A \neq \emptyset$, then $q \in \delta_1^*(q_0, x)$ since $\delta_1^*(q_0, x) = \delta^*(q_0, x)$ and the second set is Λ-closed. As we observed in the proof of Theorem 4.2, however, $\delta_1^*(q_0, \Lambda)$ is $\{q_0\}$ by definition, and this is *not* $\delta^*(q_0, \Lambda)$ if $\{q_0\}$ is not Λ-closed.

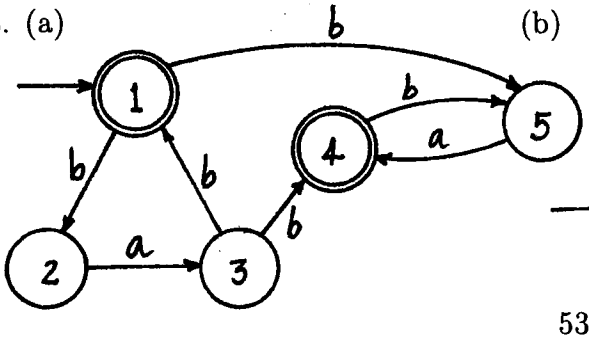
4.25. L^+

4.26. In both cases, a single state can be added. In (a), the new state is the initial state, and there is a Λ-transition from it to the state that was the initial state of the original machine. In (b), the new state is the only accepting state, and there are Λ-transitions from each of the previous accepting states to the new one. Otherwise, the transitions are the same as in the original.

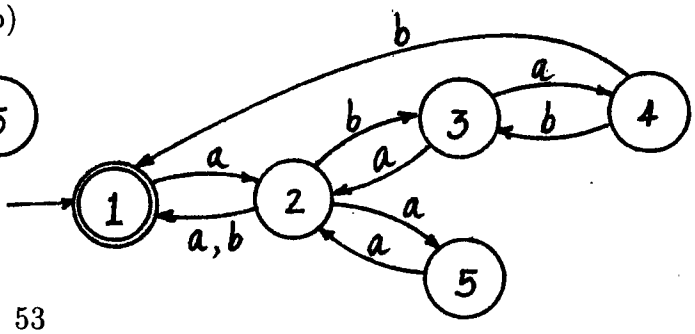
4.27. (This works for both cases.)



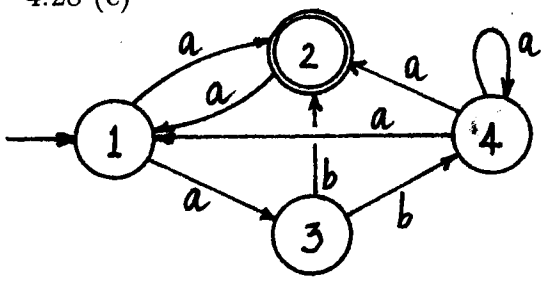
4.28. (a)



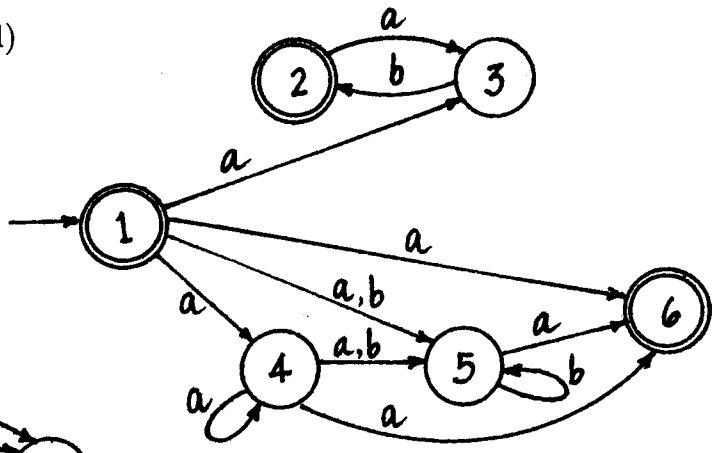
(b)



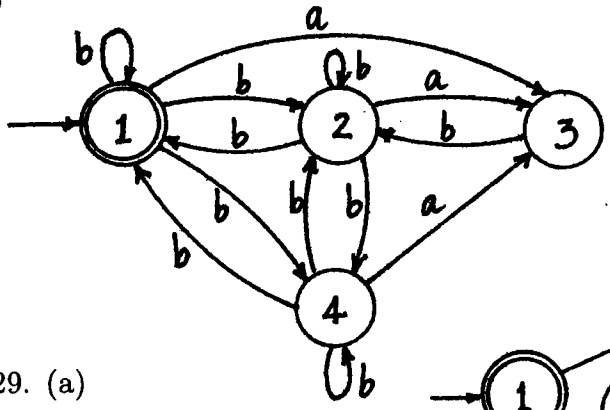
4.28 (c)



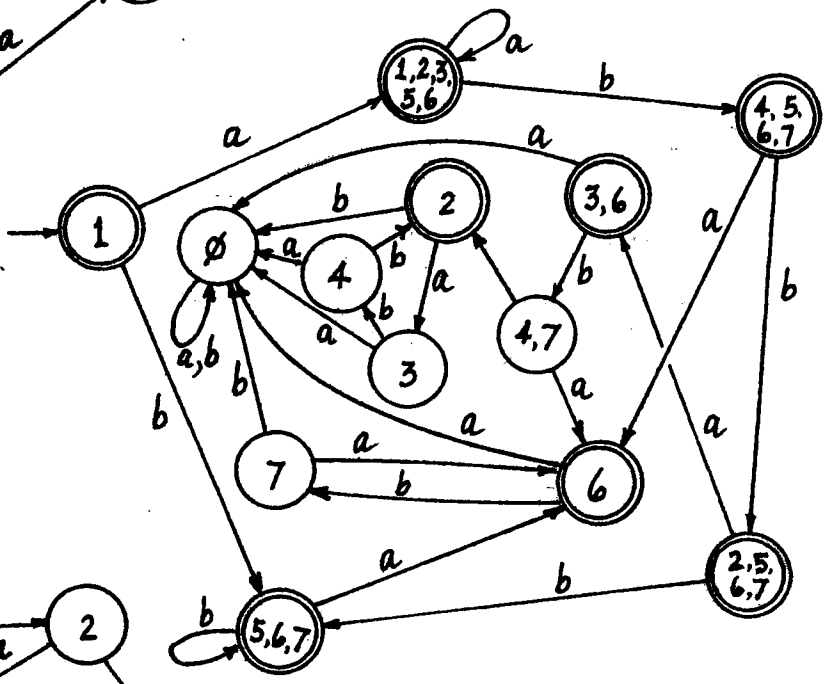
(d)



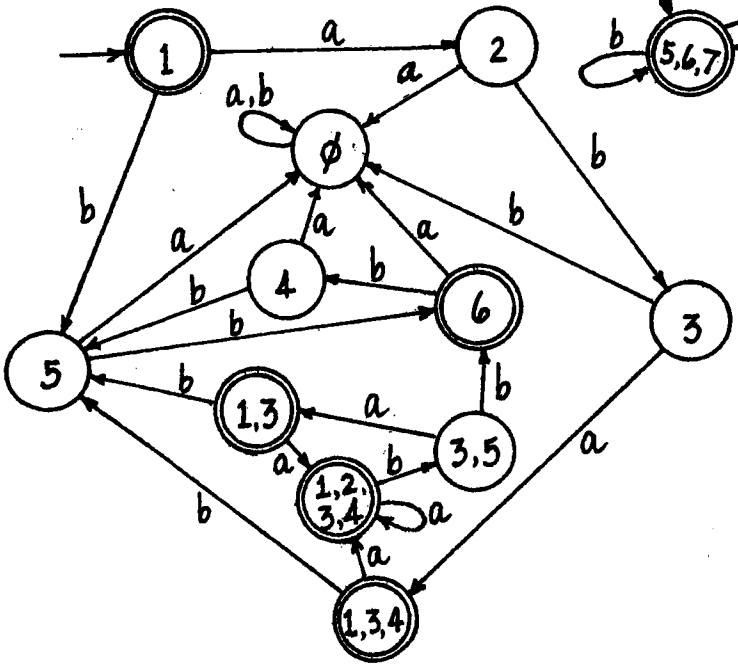
(e)



4.29. (a)

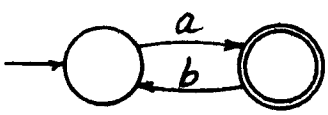


(b)

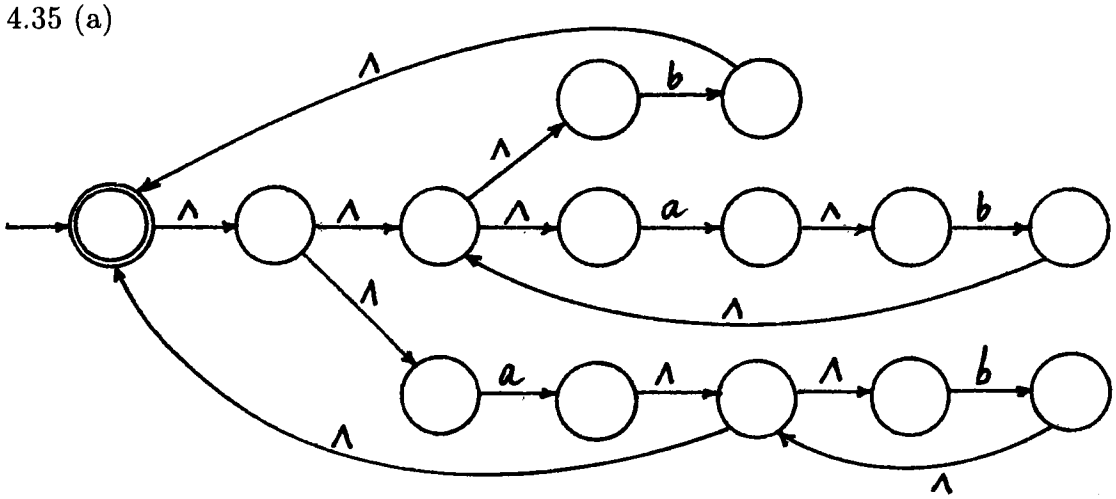


4.32 and 4.33. If M_1 and M_2 are one-state NFAs accepting $\{a\}^*$ and $\{b\}^*$, respectively, then neither construction is correct.

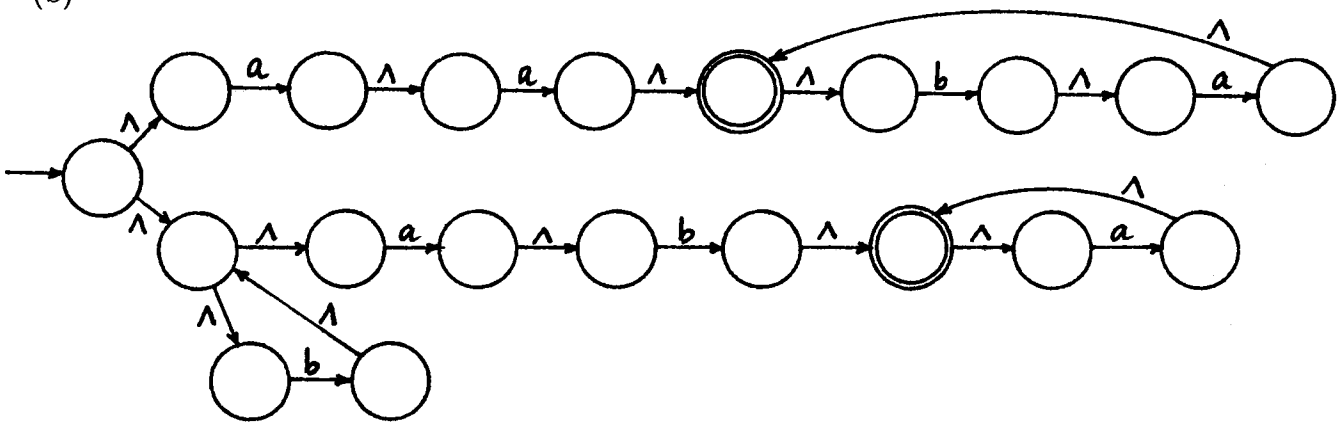
4.34. If M_1 is the machine shown, accepting $L = \{a\}\{ba\}^*$, this construction would allow ab, which isn't in L^*, to be accepted.



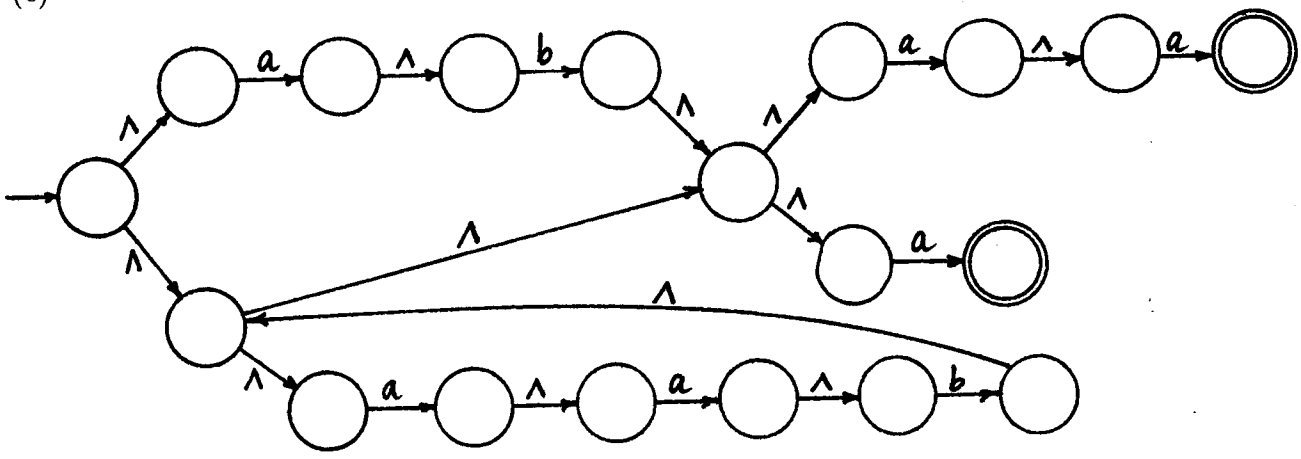
4.35 (a)



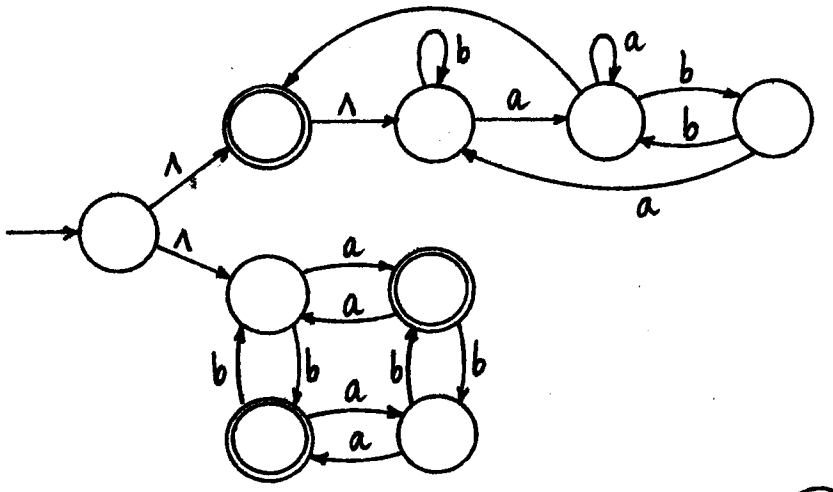
(b)



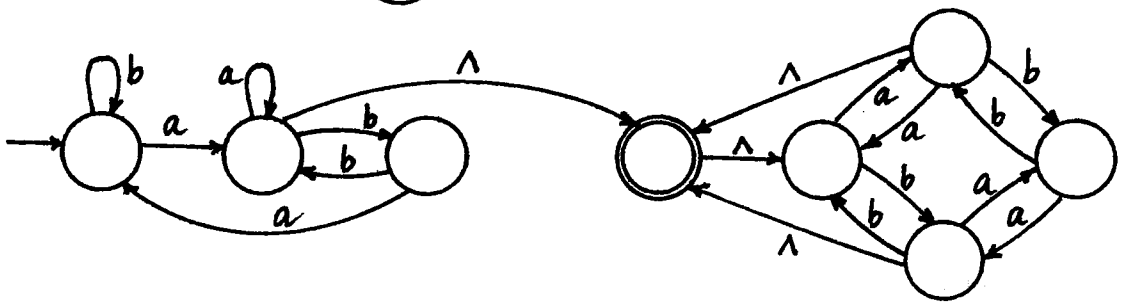
(c)



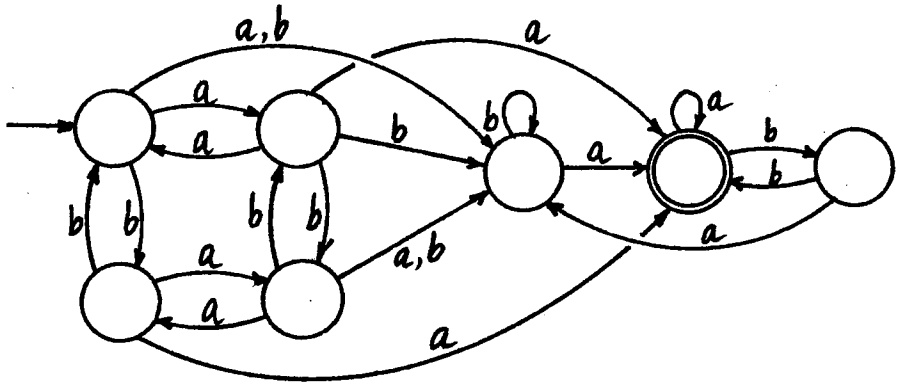
4.36. (e)

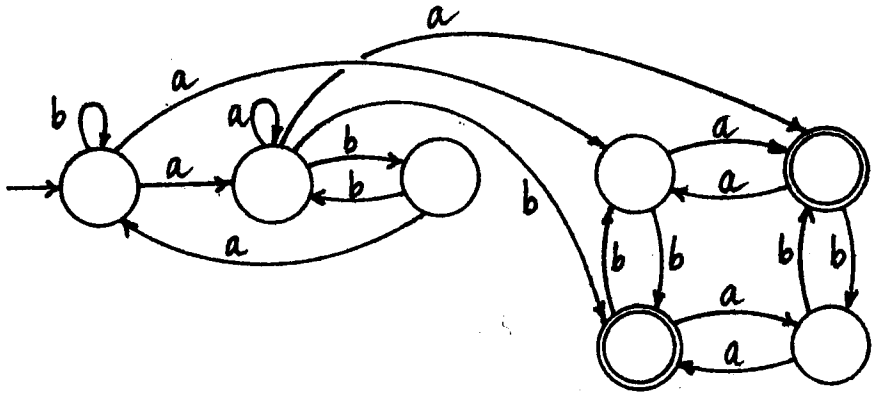

b
a
b
Λ
a
b
Λ
a
Λ
a
a
b
b
a
b
b
a
a

(f)


b
a
a
Λ
b
a
b
Λ
Λ
a
a
b
b
b
a
a
Λ

4.37.


a,b
a
a
a
b
b
b
b
b
b
a
a
a
b
b
a
a
a,b
a


a
a
b
a
a
b
a
b
b
a
a
b
b
b
b
b
a
a

4.38. Here are the tables with the regular expressions $r(p,q,k)$ for $0 \leq k \leq 2$.

p	$r(p,1,0)$	$r(p,2,0)$	$r(p,3,0)$
1	Λ	b	a
2	a	Λ	b
3	b	a	Λ

p	$r(p,1,1)$	$r(p,2,1)$	$r(p,3,1)$
1	Λ	b	a
2	a	$\Lambda + ab$	$b + aa$
3	b	$a + bb$	$\Lambda + ba$

p	$r(p,1,2)$	$r(p,2,2)$	$r(p,3,2)$
1	$(ba)^*$	$b(ab)^*$	$a + b(ab)^*(b + aa)$
2	$a(ba)^*$	$(ab)^*$	$(ab)^*(b + aa)$
3	$b + (a + bb)(ab)^*a$	$(a + bb)(ab)^*$	$\Lambda + ba + (a + bb)(ab)^*(b + aa)$

The language corresponds to the regular expression $r(1,3,3) = r(1,3,2)+r(1,3,2)r(3,3,2)^*r(3,3,2)$, or $r(1,3,2)r(3,3,2)^*$, or

$$(a + b(ab)^*(b + aa))(ba + (a + b)(ab)^*(b + aa))^*$$

(b) Here are simplified tables with the regular expressions $r(p,q,k)$ for $0 \leq k \leq 2$.

p	$r(p,1,0)$	$r(p,2,0)$	$r(p,3,0)$
1	Λ	$a + b$	$\emptyset$
2	$\emptyset$	$\Lambda + a$	b
3	b	a	Λ

p	$r(p,1,1)$	$r(p,2,1)$	$r(p,3,1)$
1	Λ	$a + b$	$\emptyset$
2	$\emptyset$	$\Lambda + a$	b
3	b	$a + ba + bb$	Λ

p	$r(p,1,2)$	$r(p,2,2)$	$r(p,3,2)$
1	Λ	$(a + b)a^*$	$(a + b)a^*b$
2	$\emptyset$	a^*	a^*b
3	b	$(a + ba + bb)a^*$	$\Lambda + (a + ba + bb)a^*b$

The language corresponds to the regular expression $r(1,2,2) + r(1,3,2)r(3,3,2)^*r(3,2,2)$, or

$$(a + b)a^* + (a + b)a^*b((a + ba + bb)a^*b)^*(a + ba + bb)a^*$$

(c) Here is the table $r(p,q,0)$.

p	$r(p,1,0)$	$r(p,2,0)$	$r(p,3,0)$	$r(p,4,0)$
1	Λ	$a + b$	$\emptyset$	$\emptyset$
2	$\emptyset$	Λ	$a + b$	$\emptyset$
3	$\emptyset$	$\emptyset$	Λ	$a + b$
4	a	b	$\emptyset$	Λ

The table $r(p,q,1)$ is exactly the same except that $r(4,2,1) = b + a(a + b)$. The table $r(p,q,2)$ is the same as $r(p,q,1)$ except that $r(1,3,2) = (a+b)^2$ and $r(4,3,2) = a(a+b)^2 + b(a + b)$.

Here is the table $r(p, q, 3)$.

p	$r(p,1,0)$	$r(p,2,0)$	$r(p,3,0)$	$r(p,4,0)$
1	Λ	$a+b$	$(a+b)^2$	$(a+b)^3$
2	$\emptyset$	Λ	$a+b$	$(a+b)^2$
3	$\emptyset$	$\emptyset$	Λ	$a+b$
4	a	$b+a(a+b)$	$a(a+b)^2+b(a+b)$	$\Lambda+a(a+b)^3+b(a+b)^2$

The language corresponds to the expression $r(1,2,4)+r(1,3,4)$, or $r(1,2,3)+r(1,4,3)r(4,4,3)^*r(4,2,3$ $r(1,3,3)+r(1,4,3)r(4,4,3)^*r(4,3,3)$, or (rearranging and simplifying slightly)

$$a+b+(a+b)^2+(a+b)^3(a(a+b)^3+b(a+b)^2)^*)(b+(\Lambda+a)(a+b)^2)$$

(d) Here are the tables with the regular expressions $r(p, q, k)$ for $0 \leq k \leq 2$.

p	$r(p,1,0)$	$r(p,2,0)$	$r(p,3,0)$	$r(p,4,0)$
1	$\Lambda+b$	$\emptyset$	a	$\emptyset$
2	a	Λ	b	$\emptyset$
3	$\emptyset$	a	Λ	b
4	$\emptyset$	b	$\emptyset$	$\Lambda+a$

p	$r(p,1,1)$	$r(p,2,1)$	$r(p,3,1)$	$r(p,4,1)$
1	b^*	$\emptyset$	b^*a	$\emptyset$
2	ab^*	Λ	$b+ab^*a$	$\emptyset$
3	$\emptyset$	a	Λ	b
4	$\emptyset$	b	$\emptyset$	$\Lambda+a$

p	$r(p,1,2)$	$r(p,2,2)$	$r(p,3,2)$	$r(p,4,2)$
1	b^*	$\emptyset$	b^*a	$\emptyset$
2	ab^*	Λ	$b+ab^*a$	$\emptyset$
3	aab^*	a	$\Lambda+ab+aab^*a$	b
4	bab^*	b	$bb+bab^*a$	$\Lambda+a$

Here are the regular expressions $r(p, q, 3)$:
$r(1,1,3) = b^* + b^*a(ab+aab^*a)^*aab^*$
$r(1,2,3) = b^*a(ab+aab^*a)^*a$
$r(1,3,3) = b^*a(ab+aab^*a)^*$
$r(1,4,3) = b^*a(ab+aab^*a)^*b$
$r(2,1,3) = ab^* + (b+ab^*a)(ab+aab^*a)^*aab^*$
$r(2,2,3) = \Lambda + (b+ab^*a)(ab+aab^*a)^*a$
$r(2,3,3) = (b+ab^*a)(ab+aab^*a)^*$
$r(2,4,3) = (b+ab^*a)(ab+aab^*a)^*b$
$r(3,1,3) = (ab+aab^*a)^*aab^*$
$r(3,2,3) = (ab+aab^*a)^*a$
$r(3,3,3) = (ab+aab^*a)^*$

$r(3,4,3) = (ab + aab^*a)^*b$
$r(4,1,3) = bab^* + (bb + bab^*a)(ab + aab^*a)^*aab^*$
$r(4,2,3) = b + (bb + bab^*a)(ab + aab^*a)^*a$
$r(4,3,3) = (bb + bab^*a)(ab + aab^*a)^*$
$r(4,4,3) = a + (bb + bab^*a)(ab + aab^*a)^*b$

Finally, the language corresponds to the regular expression $r(1,1,4)$, which is
$r(1,1,3) + r(1,4,3)r(4,4,3)^*r(4,1,3) = b^* + b^*a(ab + aab^*a)^*aab^* \quad +$
$b^*a(ab + aab^*a)^*b(b^* + b^*a(ab + aab^*a)^*aab^*)^*(bab^* + (bb + bab^*a)(ab + aab^*a)^*aab^*)$.

4.39. If $\delta(q, \Lambda) = \{p\}$, $\delta(p, a) \neq \emptyset$, and $\delta(q, a) = \emptyset$, for example, each of these definitions would say that $\delta^*(q, a) = \emptyset$, and so none of them is correct.

4.40. (a) We must show two things: first, that for every $s \in S$, $s \in \Lambda(T)$, and second, that for any $t \in \Lambda(T)$, $\Lambda(\{t\}) \subseteq \Lambda(T)$. The first is true because $S \subseteq T$ and $T \subseteq \Lambda(T)$ by definition. The second is part of the definition of $\Lambda(T)$.

(b) We know from the definition of Λ-closure that $\Lambda(S) \subseteq \Lambda(\Lambda(S))$. To show the opposite inclusion using structural induction, we must show that for every $s \in \Lambda(S)$, $s \in \Lambda(S)$, and for every $t \in \Lambda(S)$, $\Lambda(\{t\}) \subseteq \Lambda(S)$. The first is trivial, and the second is part of the definition of $\Lambda(S)$.

(c) The statement $\Lambda(S \cup T) \subseteq \Lambda(S) \cup \Lambda(T)$ is easily shown by structural induction. The opposite inclusion follows from the two statements $\Lambda(S) \subseteq \Lambda(S \cup T)$ and $\Lambda(T) \subseteq \Lambda(S \cup T)$, both of which are true by part (a).

(d) From (c), we have the result when $|S| = 2$, and it is easy to extend the result to arbitrary n by induction.

(e) $\Lambda(S \cap T)$ is a subset both of $\Lambda(S)$ and of $\Lambda(T)$, by part (a), and therefore $\Lambda(S \cap T) \subseteq \Lambda(S) \cap \Lambda(T)$.

(f) We always have $S \subseteq \Lambda(S)$ and therefore $\Lambda(S)' \subseteq S' \subseteq \Lambda(S')$. If the first and third of these two sets are equal, then the equality of the first two implies that $S = \Lambda(S)$, and the equality of the second and third implies that $S' = \Lambda(S')$. The converse is also clearly true: if $S = \Lambda(S)$ and $S' = \Lambda(S')$, then $\Lambda(S)' = \Lambda(S')$. Thus, the condition is true precisely when there is neither a Λ-transition from an element of S to an element of S' nor a Λ-transition from an element of S' to an element of S. (A more concise way to say this is to say that S and S' are both Λ-closed—see the next exercise.)

4.41. (a) For any two sets S and T, we have $S \cup T \subseteq \Lambda(S \cup T) \subseteq \Lambda(S) \cup \Lambda(T)$. (The second inclusion is easily checked using structural induction.) Therefore, if S and T are Λ-closed, $S \cup T = \Lambda(S \cup T)$.

(b) For any two sets S and T, we have $S \cap T \subseteq \Lambda(S) \cap \Lambda(T) \subseteq \Lambda(S) \cap \Lambda(T)$. If S and T are Λ-closed, it follows that $S \cap T = \Lambda(S \cap T)$.

(c) $\Lambda(S)$ is a Λ-closed set containing S as a subset. If T is any other such set, then since $S \subseteq T$, $\Lambda(S) \subseteq \Lambda(T)$ (by Exercise 4.40(a)), and therefore $\Lambda(S) \subseteq T$ because T is Λ-closed.

4.42. We show the formula for an NFA-Λ, and the formula for NFAs will follow as a special

case. The proof is by structural induction on y and will be very enjoyable for those who like mathematical notation. The basis step is for $y = \Lambda$. For any $q \in Q$ and any $x \in \Sigma^*$,

$$\bigcup_{p \in \delta^*(q,x)} \delta^*(p, \Lambda) = \bigcup_{p \in \delta^*(q,x)} \Lambda(\{p\}) = \Lambda(\bigcup_{p \in \delta^*(q,x)} \{p\}) = \Lambda(\delta^*(q,x)) = \delta^*(q,x) = \delta^*(q, x\Lambda)$$

In this formula, we are using parts of Exercise 4.40. The first equality uses the definition of δ^* for an NFA-Λ. The second uses part (d) of Exercise 4.40. The third is simply the definition of union. The fourth uses part (b) of Exercise 4.40 (and the fact that according to the definition, $\delta^*(q,x)$ is in fact $\Lambda(S)$ for some set S).

Now assume that for the string y,

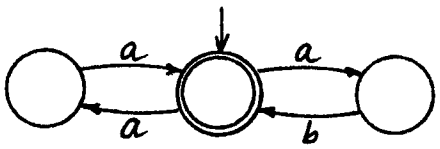
$$\delta^*(q, xy) = \bigcup_{p \in \delta^*(q,x)} \delta^*(p, y)$$

for every state q and every string $x \in \Sigma^*$. Then for any $a \in \Sigma$,

$$\begin{aligned}
\delta^*(q, x(ya)) &= \delta^*(q, (xy)a) \\
&= \Lambda(\bigcup_{p \in \delta^*(q,xy)} \delta(p, a)) \\
&= \Lambda(\{s \mid \exists p(p \in \delta^*(q, xy) \wedge s \in \delta(p, a))\}) \\
&= \Lambda(\{s \mid \exists p(p \in \bigcup_{r \in \delta^*(q,x)} \delta^*(r, y) \wedge s \in \delta(p, a))\}) \\
&= \Lambda(\{s \mid \exists r(r \in \delta^*(q, x) \wedge \exists p(p \in \delta^*(r, y) \wedge s \in \delta(p, a)))\}) \\
&= \Lambda(\{s \mid \exists r(r \in \delta^*(q, x) \wedge s \in \bigcup_{p \in \delta^*(r,y)} \delta(p, a))\}) \\
&= \Lambda(\bigcup_{r \in \delta^*(q,x)} (\bigcup_{p \in \delta^*(r,y)} \delta(p, a))) \\
&= \bigcup_{r \in \delta^*(q,x)} \Lambda(\bigcup_{p \in \delta^*(r,y)} \delta(p, a)) \\
&= \bigcup_{r \in \delta^*(q,x)} \delta^*(r, ya)
\end{aligned}$$

The second equality uses the definition of δ^*. The third is just a restatement of the definition of union. The fourth uses the induction hypothesis. The fifth, sixth, and seventh are also just restatements involving the definitions of various unions. The eighth uses part (c) of Exercise 4.40 (actually, the generalization of part (c) from the union of two sets to an arbitrary union). The ninth is the definition of δ^*.

4.43. The language accepted by each M_1^c is a subset of $L(M)$. However, there may be strings in $L(M)$ not accepted by any M_1^c. If M is the FA below, there are two possible M_1^c's, each with an unreachable state, and neither accepts the string $aaab$.



4.44. (a) The set of suffixes of elements of L. On the one hand, if $x = yz \in L$, $q \in \delta^*(q_0, y)$, and $q_f \in \delta^*(q, z)$, then M_1 contains a Λ-transition from q_0 to q, which implies that z is accepted by M_1. On the other hand, if $z = b_1 b_2 \dots b_m \in L(M)$, where each $b_i \in \Sigma \cup \{\Lambda\}$, and the states $q_0 = p_0, p_1, p_2, \dots, p_m = q_f$ are such that $p_i \in \delta_1(p_{i-1}, b_i)$ for each i (where δ_1 is the transition function of M_1), then the assumptions regarding q_0 and q_f imply that the only one of these transitions that may not be present in M is the first one, which may be one of the Λ-transitions added to M_1. It follows that there is some string y so that $p_1 \in \delta^*(q_0, y)$ and that $q_f \in \delta^*(q_0, yz)$. In other words, z is a suffix of an element of L.

(b) An argument like the one in part (a) shows that M_2 accepts the strings that are prefixes of elements of L.

(c) M_3 accepts the language of substrings of elements of L.

(d) M_4 accepts the set of all strings that can be obtained from elements of L by deleting zero or more substrings: that is, strings of the form $y_1 y_2 \dots y_n$ for which there are strings $x_0, x_1, \dots, x_n$ satisfying $x_0 y_1 x_1 y_2 x_2 \dots x_{n-1} y_n x^n \in L$.

4.45. (a) It is at least clear that the two one-state NFAs accepting 0^* and 1^* cannot be joined so that the composite NFA accepts the union of the two languages. No matter which state was chosen as the initial state, either 01 or 10 would have to be accepted. It is easy to show by looking at cases that no two-state NFA can work.

(b) A simple condition that makes it possible to incorporate M_1 and M_2 into a composite NFA is that there are no transitions to the initial state of M_1, and either $\Lambda \in L(M_1)$ or $\Lambda \notin L(M_2)$. In this case, for each $a \in \Sigma$, a-transitions can be drawn from the initial state of M_1 to every state q in M_2 for which $q \in \delta_2(q_2, a)$ (where q_2 is the initial state of M_2). If M_1 contains transitions to q_1, the initial state, then it may still be possible to draw transitions from q_1 to states of M_2, provided every string x with $q_1 \in \delta_1^*(q_1, x)$ is a prefix of an element of $L(M_2)$.

4.46. (a) $f(\Lambda) = f(\Lambda\Lambda) = f(\Lambda)f(\Lambda)$, and it follows that $f(\Lambda) = \Lambda$.

(b) We observe that for any $L_1, L_2 \subseteq \Sigma_1^*$, we have $f(L_1 \cup L_2) = f(L_1) \cup f(L_2)$, $f(L_1 L_2) = f(L_1)f(L_2)$, and $f(L_1^*) = f(L_1)^*$. (The last two are because f is a homomorphism.) Now we can show very easily, using structural induction, that if L is regular, so is $f(L)$. First, if L is any of the regular languages $\emptyset$, $\{\Lambda\}$, or $\{a\}$ (where $a \in \Sigma_1$), then $f(L)$ is regular, since $f(\emptyset) = \emptyset$, $f(\{\Lambda\}) = \{\Lambda\}$, and $f(\{a\}) = \{f(a)\}$. Second, if $f(L_1)$ and $f(L_2)$ are regular, then so are $f(L_1 \cup L_2)$, $f(L_1 L_2)$, and $f(L_1^*)$.

(In fact, it is easy to check that if r is a regular expression representing L, then a regular expression representing $f(L)$ can be obtained by replacing every alphabet symbol a in r by the string $f(a)$.)

(c) Suppose $M = (Q, \Sigma_2, q_0, A, \delta)$ is an FA accepting $L \subseteq \Sigma_2^*$. We define an FA $M_1 = (Q, \Sigma_1, q_0, F, \delta_1)$ accepting $f^{-1}(L)$ as follows: for $q \in Q$ and $a \in \Sigma_1$, $\delta_1(q, a) = \delta^*(q, f(a))$. Then it follows easily from the fact that f is a homomorphism that for every $q \in Q$ and every $x \in \Sigma_1^*$, $\delta_1^*(q, x) = \delta^*(q, f(x))$. Therefore, M_1 accepts x if and only if M accepts $f(x)$. This means that for any $x \in \Sigma_1^*$, $x \in f^{-1}(L)$ if and only if M_1 accepts x.

4.47. The reduction step in which p is eliminated is to redefine $e(q,r)$ for every pair (q,r) where neither q nor r is p. (This includes the pairs (q,q).) This is done using the formula

$$e(q,r) = e(q,r) + e(q,p)e(p,p)^*e(p,r)$$

We illustrate this process for the FA shown in part (b) of Figure 4-22. The first picture shows the NFA-Λ with the states q_0 and q_f; the subsequent pictures describe the machines obtained by eliminating the states 1, 2, and 3, in that order. The regular expression $e(q_0, q_f)$ in the last picture describes the language. (Note that it looks the same as the regular expression obtained in the solution to Exercise 4.38(b), although the two algorithms do not always produce regular expressions that look the same.)