1 2.1

1. Which of the strings 0001, 01101, 00001101 are accepted by the dfa in Figure 2.1?

   Solution: Strings 0001, 01101, and 00001101 are accepted by the dfa in Figure 2.1.

2. ** Translate the graph in Figure 2.5 into δ- notation

   Solution:

$$
\begin{aligned}
\delta(\lambda, 1) &= \lambda, & \delta(\lambda, 0) &= 0, \\
\delta(0, 1) &= \lambda, & \delta(0, 0) &= 00, \\
\delta(00, 0) &= 00, & \delta(00, 1) &= 001, \\
\delta(001, 0) &= 001, & \delta(001, 1) &= 001.
\end{aligned}
$$

3. For $\Sigma = \{a, b\}$, construct dfa's that accept the sets consisting of

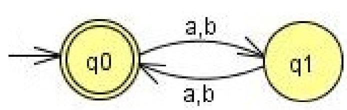
   (a) all strings of even length

   (b) all strings of length greater than 5

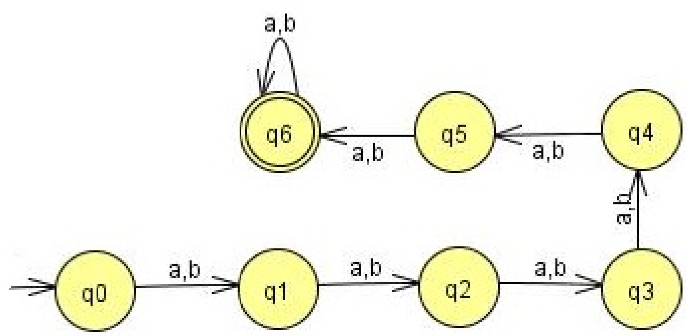
   (c) ** all strings with an even number of a's

   (d) all strings with an even number of a's and an odd number of b's

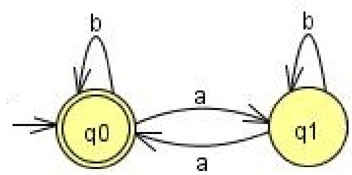
   Solution: (a)



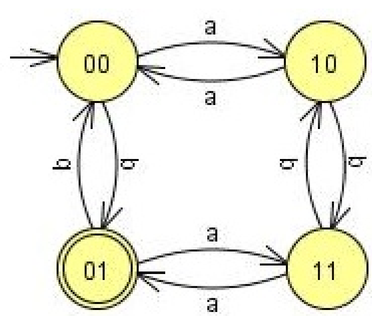
(b) The solution requires seven states as shown below.



(c)



(d) The key for solving this problem is to realize that an even number of a's and an odd number of b's in w can be stated as $n_a(w)$ mod 2 = 0 and $n_b(w)$ mod 2 = 1. Therefore if we label each state in the automaton by a two digits number such that the first digit represents a mod 2 and the second one for b mod 2. the number of a's and b's (in modulo 2) will be traced in an obvious manner. A dfa for the solution of the problem is given below.



4. For $\Sigma = \{a, b\}$, construct dfa's that accept the sets consisting of

(a) ** all strings with exactly one a,

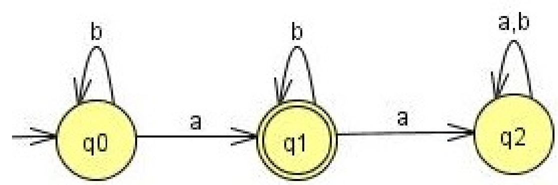
(b) all strings with at least two a's,

(c) all strings with no more than two a's

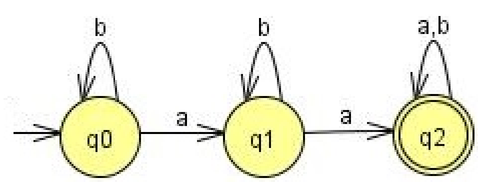
(d) all strings with at least one b and exactly two a's,

(e) all the strings with exactly two a's and more than three b's.

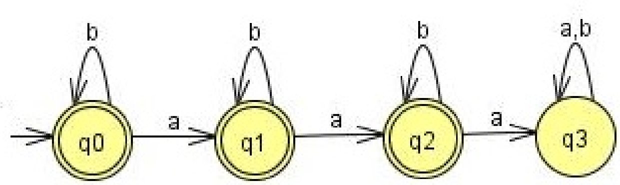
Solution: (a) A solution is given by



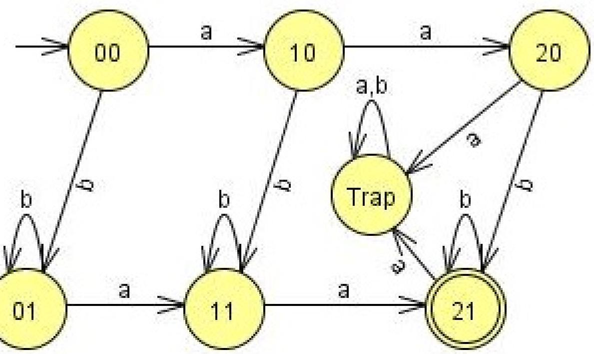
(b) A solution is given by



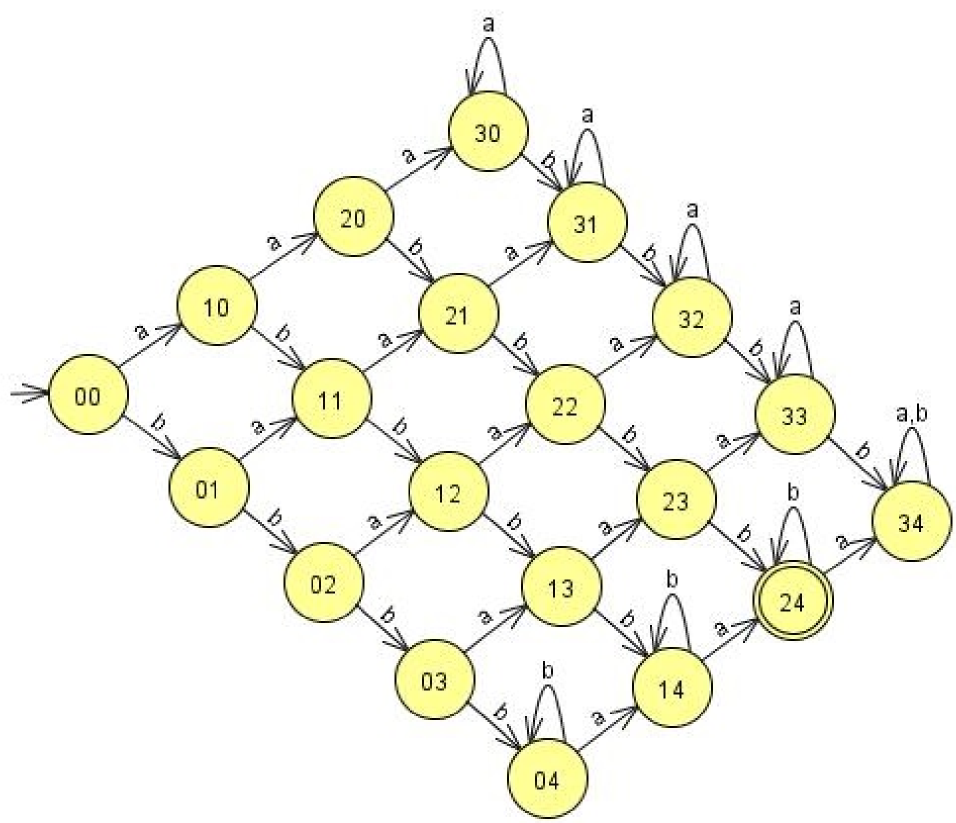
(c) The first three states are all final states until the third input a is encountered in the fourth state.



(d) We label the state by a two digit number such that the first digit represents the number of input a's and the second digit represents the number of b's.



(e) Use the same approach as (d) to label the state by two digit numbers to keep track $n_a(w)$ and $n_b(w)$.



5. Give dfa's for the languages

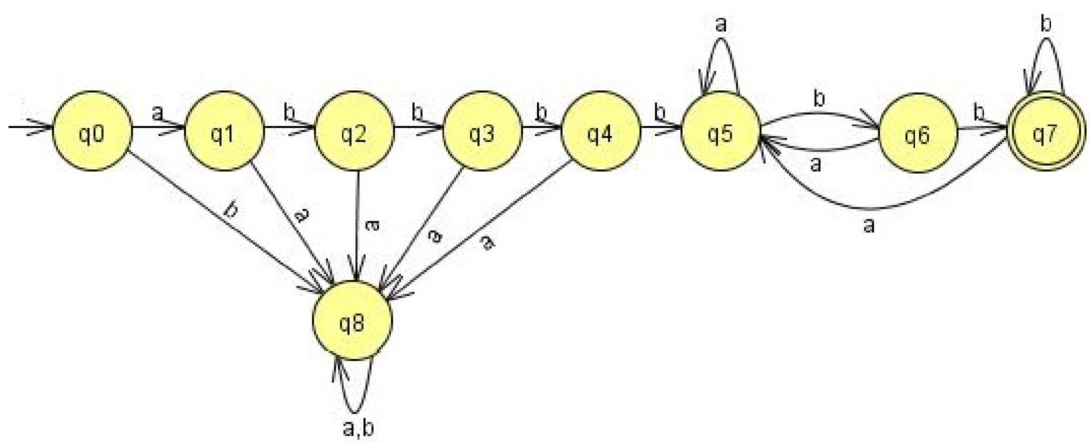
(a) $L = \{ab^4wb^2 : w \in \{a,b\}^*\}$,

(b) $L = \{ab^na^m : n \geq 3, m \geq 2\}$,

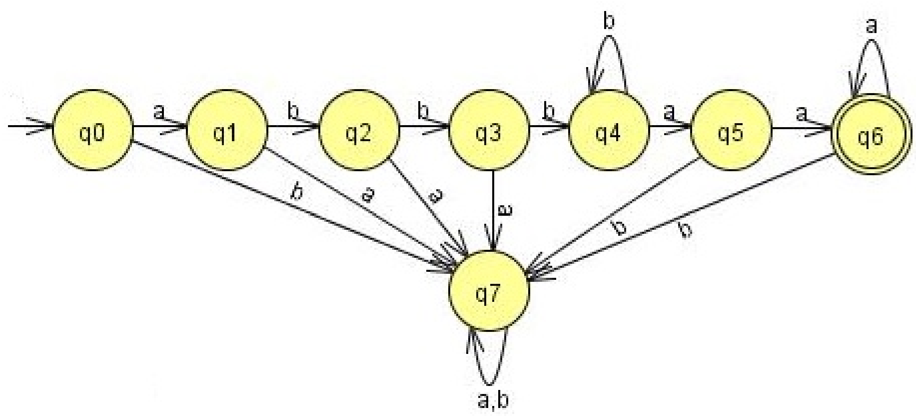
(c) $L = \{w_1abbw_2 : w_1 \in \{a,b\}^*, w_2 \in \{a,b\}^*\}$,

(d) $L = \{ba^n : n \geq 1, n \neq 4\}$.

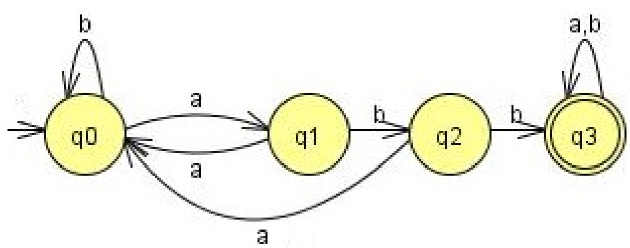
Solution: (a) The key is that any string that can be accept by the solution dfa must consist ab^4 at the begining and two b's at the end. The first requirement is straightforward, and the second requirement is solved between states q_5 to q_7.



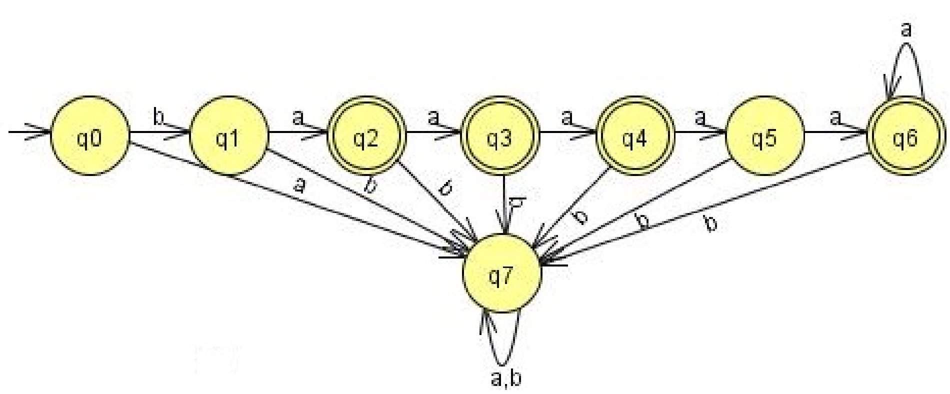
(b)



(c)



(d) The dfa must have final states corresponding to one a, two, three, five, and more a's but not four a's as shown below.



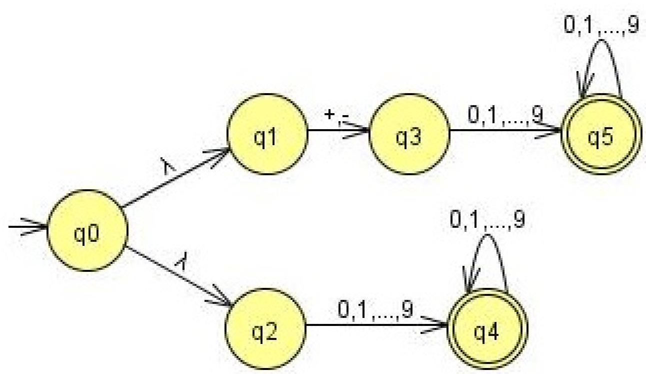
9. Show that if we change Figure 2.6, making q_3 a nonfinal state and making q_0, q_1, q_2 final states, the resulting dfa accepts L.

Solution: If $w \in L$ then the dfa in Figure 2.6 will reject w. That isat the end of string the dfa will be in one of the nonfinal states q_0, q_1, q_2. However, if we change these states to final states, then the resulting dfa accepts w

1 2.2

1. Construct an nfa that accepts all integer numbers in C.

   Solution: Integer numbers in C have the form <*sign*><*magnitude*>. A solution is

2. Prove in detail the claim made in the previous section that if in a transition graph there is a walk labeled w, there must be some walk labeled w of length no more than $\Lambda + (1+\Lambda)\,|w|$.

   Solution: Suppose we have a walk labeled $w = a_1a_2...a_n$ in a graph with Λ λ-transitions.Then any walk labeled w that includes λ- edges can be described by the form

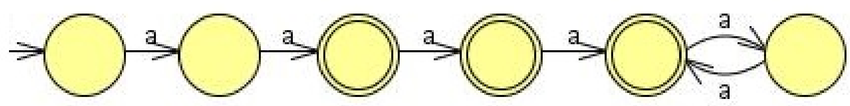
   $$u = \lambda^{k_1}a_1\lambda^{k_2}...\lambda^{k_n}a_n\lambda^{k_{n+1}},$$

   where λ^{k_i} denotes successive traversal k_i λ-edges and a_i denotes the traversal of an edge labeled a_i . Denoting the length of u by $|u|$ we have

   $$\begin{aligned} |u| &= (k_1 + k_2 + ... + k_{n+1}) + (|a_1|... + |a_n|) \\ &= (n+1)\Lambda + n \\ &= \Lambda + (1+\Lambda)|w|. \end{aligned}$$

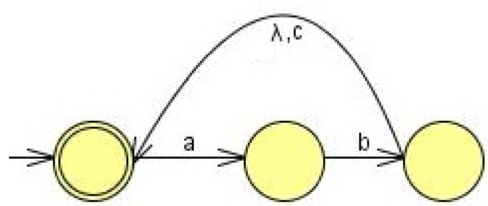
3. ** Find a dfa that accepts the language defined by the nfa in Figure 2.8.

Solution: The nfa in Figure 2.8 accepts the language $\{a^n : n = 3 \text{ or } n \text{ is even}\}$. A dfa for this language is



9. ** Construct an nfa with three states that accepts the language $\{ab, abc\}^*$.

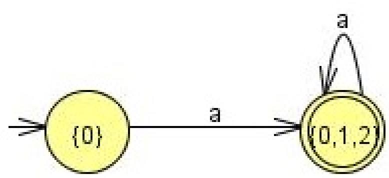
   Solution:



1 2.3

1. ** Use the construction of Theorem 2.2 to convert the nfa in Figure 2.10 to a dfa. Can you see a simpler answer more directly?

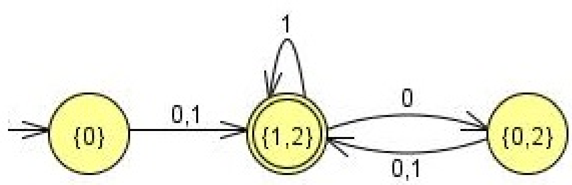
   Solution:



   For a simple answer, note that the accepted language is $\{a^n : n > 1\}$.

2. Convert the nfa in Exercise 13, Section 2.2, into an equivalent dfa.

   Solution: Following the procedure nfa-to-dfa, we have

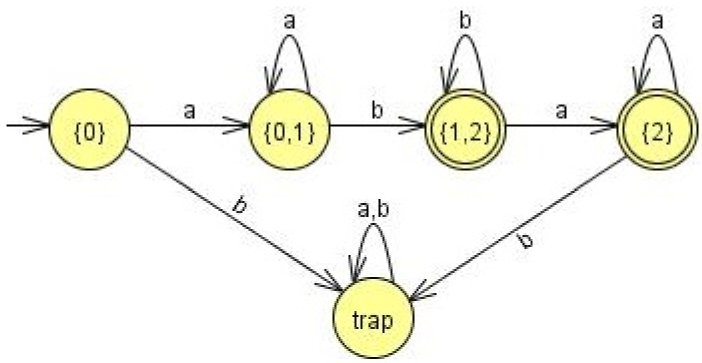


3. ** Convert the nfa defined by

$$\begin{aligned}\delta(q_0, a) &= \{q_0, q_1\}\\ \delta(q_1, b) &= \{q_1, q_2\}\\ \delta(q_2, a) &= \{q_2\}\end{aligned}$$

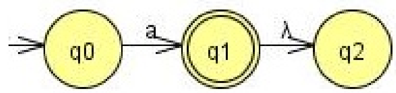
   with initial state q_0 and final state q_2 into an equivalent dfa.

   Solution: This gives the dfa



8. Is it true that for every nfa $M = (Q, \Sigma, \delta, q_0, F)$ the complement of $L(M)$ is equal to the set $\{w \in \Sigma^* : \delta^*(q_0, w) \cap (Q - F) \neq \oslash\}$? If so, prove it; if not, give a counterexample.

   Solution: No. λ-transition make a difference . See the following simple example, that $L = \{a\}$. But $\delta^*(q_0, a) \cap (Q - F) = \{q_2\} \neq \oslash$ which is impossible.



9. Prove that for every nfa with an arbitrary number of final states there is an equivalent nfa with only one final state. Can we make a similar claim for dfa's?

   Solution: Introduce a new final state p_f and for every $q \in F$ add the transitions
$$\delta(q, \lambda) = \{p_f\}.$$
   Then make p_f the only final state. It is a simple matter then to argue that if $\delta^*(q_0, w) \in F$ originally, then $\delta^*(q_0, w) = \{p_f\}$ after the modification, so both the original and the modified nfa's are equivalent.

   Since this construction requires λ-transitions, it cannot be made for dfa's. Generally, it is impossible to have only one final state in a dfa, as can be seen by constructing a dfa that accepts $\{\lambda, a\}$.

1 2.4

1. Consider the dfa with initial state q_0 , final state q_2 and

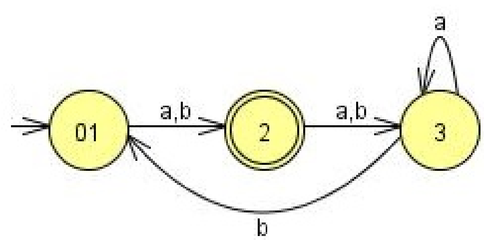
$$\begin{array}{rclr}
\delta(q_0,a) & = & q_2 & \delta(q_0,b)=q_2 \\
\delta(q_1,a) & = & q_2 & \delta(q_1,b)=q_2 \\
\delta(q_2,a) & = & q_3 & \delta(q_2,b)=q_3 \\
\delta(q_3,a) & = & q_3 & \delta(q_3,b)=q_1
\end{array}$$

   Find a minimal equivalent dfa.

   Solution: Using procedure *mark*, we generate the equivalence classes $\{q_0,q_1\},\{q_2\},\{q_3\}$. Then the procedure *reduce* gives

$$\begin{array}{rr}
\hat{\delta}(01,a)=2, & \hat{\delta}(01,b)=2, \\
\hat{\delta}(2,a)=3, & \hat{\delta}(2,b)=3, \\
\hat{\delta}(3,a)=3, & \hat{\delta}(3,b)=01.
\end{array}$$

   Therefore a minimal equivalent dfa is



2. Minimize the number of states in the dfa in Figure 2.16.

   Solution: To simply the notation for the names of the states in Figure 2.16, we let

   $p_0=\{q_0\}$, $p_1=\{q_0,q_1\}$, $p_2=\{q_0,q_1,q_2\}$

   $p_3=\{q_1,q_2\}$, $p_4=\{q_1\}$, $p_5=\{q_2\}$, $p_6=\oslash$.

   After performing the procedures *mark* and *reduce*, the resulting minimal dfa is