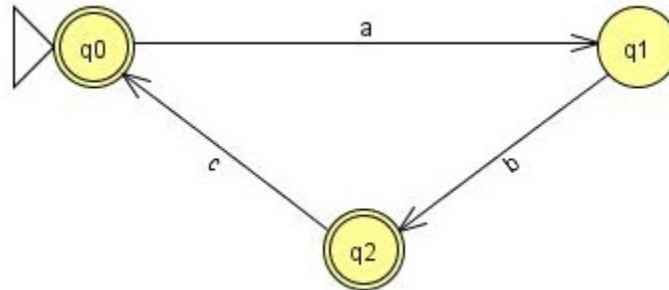


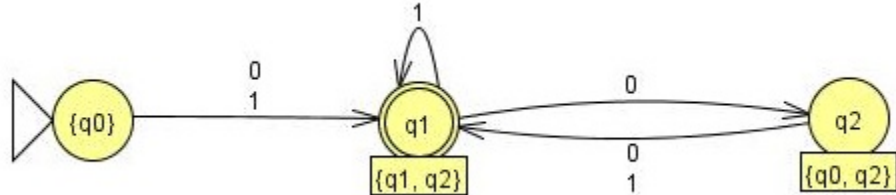
§2.2 8. Construct an NFA with three states that accepts the language  $\{ab, abc\}^*$ .



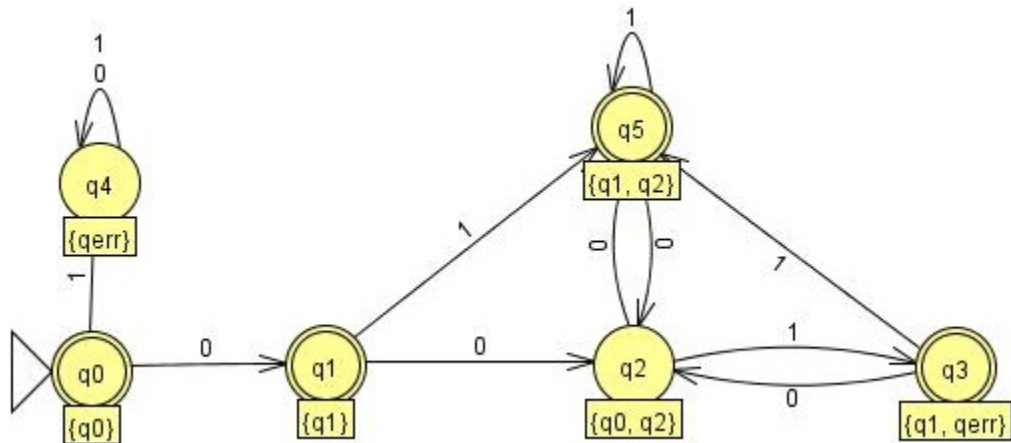
§2.2 13. What is the complement of the language accepted by the NFA in Figure 2.10?

Assuming  $\Sigma = \{a\}$ , we can easily convert this NFA to a DFA. We then can see that it accepts all strings of the form  $aa^*$ . Hence,  $L = \{a^n : n \geq 1\}$ . The complement is  $\bar{L} = \Sigma^* - L$ . So, we have  $\bar{L} = \Sigma^* - \{a^n : n \geq 1\}$ . It is clear to see, then, that  $\bar{L} = \{a^0\}$ , or  $\bar{L} = \{\lambda\}$ .

§2.3 2. Convert the NFA in Exercise 12, Section 2.2, into an equivalent DFA.



§2.3 3. Convert the NFA into an equivalent DFA.

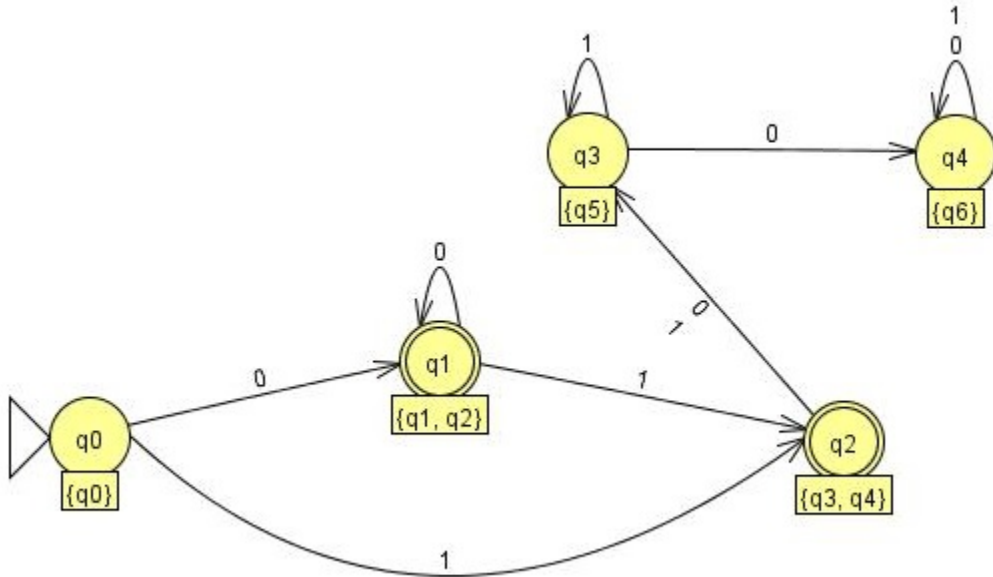


§2.3 12. Show that if  $L$  is regular, so is  $L^R$ .

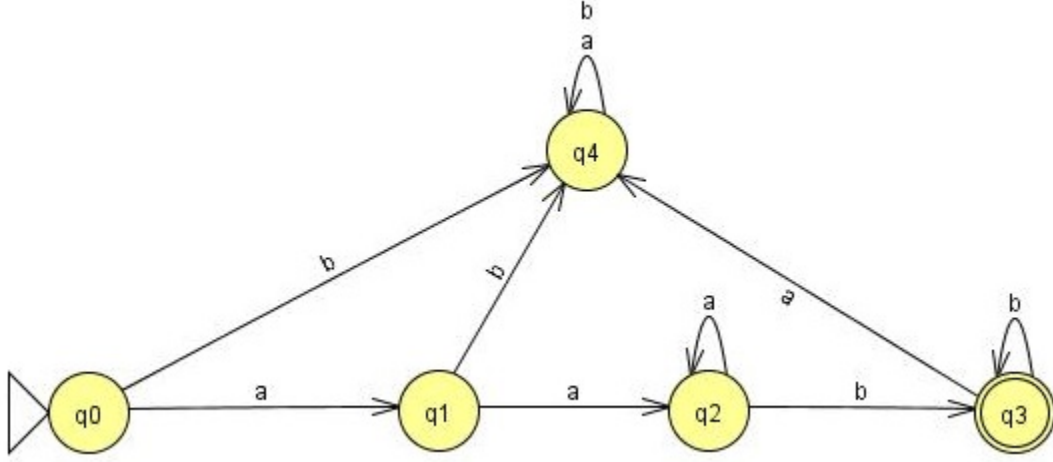
Since  $L$  is regular, we know that a finite accepter  $M$  exists that accepts  $L$ . We know we can construct a transition graph  $G$  with a finite number of vertices (states) and edges (transitions) from  $M$ . For the sake of argument, we may reasonably assume there is only one final state in  $M$  and  $G$ . We may say this machine and graph accept all strings  $w \in L$ . We now need to show that  $L^R$  is regular. By definition, the reverse of  $L$  is  $L^R = \{w^R : w \in L\}$ , where  $w^R$  is the reverse of  $w$ . To show  $L^R$  is regular, we need to find a finite accepter that accepts all strings in  $L^R$ . Since we already have a graph  $G$  that represents the machine  $M$  that accepts  $L$ , we can start here. Because  $G$  is a digraph (and by definition of a finite accepter), states can only transition in certain directions. Specifically,  $G$  is set up to accept only  $w \in L$ . Let us clone this graph and call it  $G_R$ . In the graph  $G_R$ , let us reverse the direction of all non-loop edges. This may create inaccessible states, but these are negligible. Also, let us change the final state in  $G_R$  to the initial state and vice versa. Now we have a graph that accepts strings in the reverse direction of  $G$ . That is,  $G_R$  accepts  $w^R$ . Hence, this graph represents at least a nondeterministic finite accepter that accepts all strings in the language  $L^R$ , which is sufficient to show that  $L^R$  is a regular language.

#QED

§2.4 1. Minimize the number of states in the DFA in Figure 2.16.



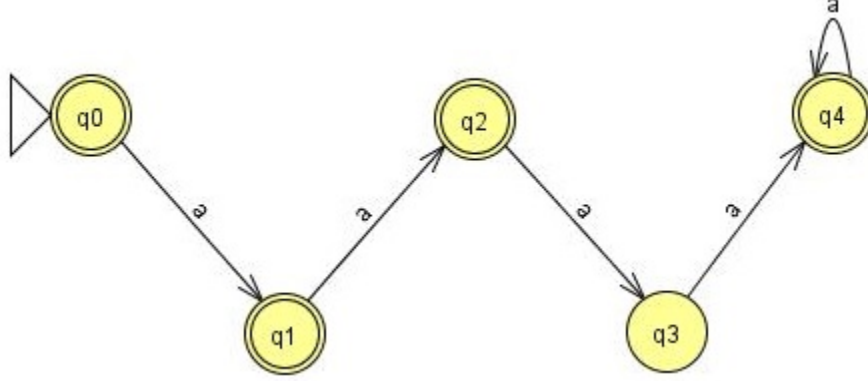
**§2.4 2a.** Find the minimal DFA for  $L = \{a^n b^m : n \geq 2, m \geq 1\}$ . Prove that the result is minimal.



We can use the mark and reduce algorithms to minimize our DFA. We know that the reduce algorithm will produce a DFA  $\widehat{M}$  such that  $L(M) = L(\widehat{M})$  and  $\widehat{M}$  has the smallest number of states possible. We claim that our DFA is  $\widehat{M}$ . Assume that our DFA is not minimal. If we apply the mark and reduce algorithms on  $\widehat{M}$ , then we will produce another DFA  $\widehat{M}_1$  such that  $L(\widehat{M}) = L(\widehat{M}_1)$  and  $\widehat{M}_1$  has fewer states than  $\widehat{M}$ . Following the mark algorithm, we eliminate any inaccessible states, of which there are none. Next, we mark all distinguishable pairs of states  $(p, q)$  such that  $p \in F$  and  $q \notin F$  or vice versa. For our DFA  $\widehat{M}$ , these are the pairs  $(0, 3)$ ,  $(1, 3)$ ,  $(2, 3)$ , and  $(3, 4)$ . Next, we determine whether any other pairs are distinguishable based on the computations  $\delta(p, a) = p_a$  and  $\delta(q, a) = q_a$  for all pairs  $(p, q)$  and all  $a \in \Sigma$ . If  $(p_a, q_a)$  is distinguishable, then so is  $(p, q)$ . Repeating this step, we see all pairs  $(p, q)$  are distinguishable, thus ending the mark algorithm. The first step of the reduce algorithm asks us to create equivalence classes, which gives us five distinct sets, each containing a different state in  $\widehat{M}$ . The next step asks us to create a new state for each equivalence class that contains indistinguishable states. Since we have no such states, there is no need to create these new states. The rest of the steps in the reduce algorithm thus become superfluous. We see, then, that the DFA  $\widehat{M}_1$  produced by mark and reduce is actually  $\widehat{M}$ . This contradicts our assumption that  $\widehat{M}$  is not the minimal DFA. Thus, we have that  $\widehat{M}$  is the minimal DFA.

**#QED**

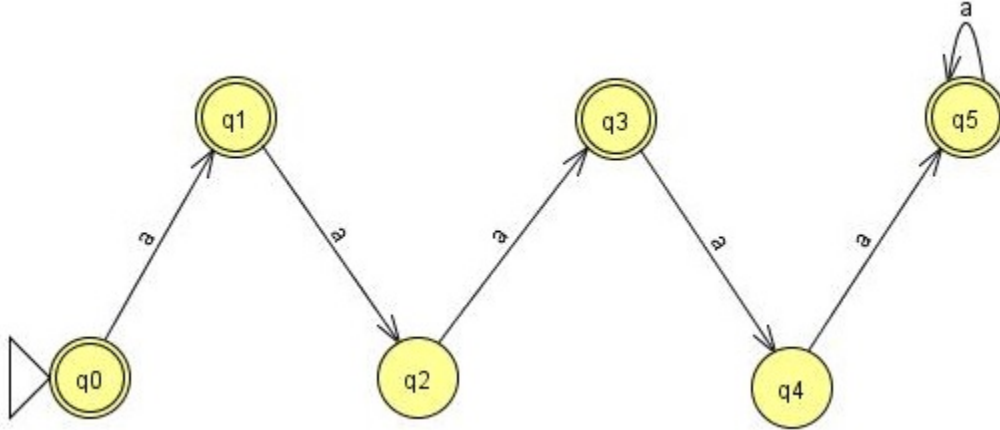
**§2.4 2c.** Find the minimal DFA for  $L = \{a^n : n \geq 0, n \neq 3\}$ . Prove that the result is minimal. Assume  $\Sigma = \{a\}$ .



We can use the mark and reduce algorithms to minimize our DFA. We know that the reduce algorithm will produce a DFA  $\widehat{M}$  such that  $L(M) = L(\widehat{M})$  and  $\widehat{M}$  has the smallest number of states possible. We claim that our DFA is  $\widehat{M}$ . Assume that our DFA is not minimal. If we apply the mark and reduce algorithms on  $\widehat{M}$ , then we will produce another DFA  $\widehat{M}_1$  such that  $L(\widehat{M}) = L(\widehat{M}_1)$  and  $\widehat{M}_1$  has fewer states than  $\widehat{M}$ . Following the mark algorithm, we eliminate any inaccessible states, of which there are none. Next, we mark all distinguishable pairs of states  $(p, q)$  such that  $p \in F$  and  $q \notin F$  or vice versa. For our DFA  $\widehat{M}$ , these are the pairs  $(0, 3)$ ,  $(1, 3)$ ,  $(2, 3)$ , and  $(3, 4)$ . Next, we determine whether any other pairs are distinguishable based on the computations  $\delta(p, a) = p_a$  and  $\delta(q, a) = q_a$  for all pairs  $(p, q)$  and all  $a \in \Sigma$ . If  $(p_a, q_a)$  is distinguishable, then so is  $(p, q)$ . Repeating this step, we see all pairs  $(p, q)$  are distinguishable, thus ending the mark algorithm. The first step of the reduce algorithm asks us to create equivalence classes, which gives us five distinct sets, each containing a different state in  $\widehat{M}$ . The next step asks us to create a new state for each equivalence class that contains indistinguishable states. Since we have no such states, there is no need to create these new states. The rest of the steps in the reduce algorithm thus become superfluous. We see, then, that the DFA  $\widehat{M}_1$  produced by mark and reduce is actually  $\widehat{M}$ . This contradicts our assumption that  $\widehat{M}$  is not the minimal DFA. Thus, we have that  $\widehat{M}$  is the minimal DFA.

**#QED**

**§2.4 2d.** Find the minimal DFA for  $L = \{a^n : n \neq 2 \text{ and } n \neq 4\}$ . Prove that the result is minimal. Assume  $\Sigma = \{a\}$ .



We can use the mark and reduce algorithms to minimize our DFA. We know that the reduce algorithm will produce a DFA  $\widehat{M}$  such that  $L(M) = L(\widehat{M})$  and  $\widehat{M}$  has the smallest number of states possible. We claim that our DFA is  $\widehat{M}$ . Assume that our DFA is not minimal. If we apply the mark and reduce algorithms on  $\widehat{M}$ , then we will produce another DFA  $\widehat{M}_1$  such that  $L(\widehat{M}) = L(\widehat{M}_1)$  and  $\widehat{M}_1$  has fewer states than  $\widehat{M}$ . Following the mark algorithm, we eliminate any inaccessible states, of which there are none. Next, we mark all distinguishable pairs of states  $(p, q)$  such that  $p \in F$  and  $q \notin F$  or vice versa. For our DFA  $\widehat{M}$ , these are the pairs  $(0, 2)$ ,  $(0, 4)$ ,  $(1, 2)$ ,  $(1, 4)$ ,  $(2, 3)$ ,  $(2, 5)$ ,  $(3, 4)$ , and  $(4, 5)$ . Next, we determine whether any other pairs are distinguishable based on the computations  $\delta(p, a) = p_a$  and  $\delta(q, a) = q_a$  for all pairs  $(p, q)$  and all  $a \in \Sigma$ . If  $(p_a, q_a)$  is distinguishable, then so is  $(p, q)$ . Repeating this step, we see all pairs  $(p, q)$  are distinguishable, thus ending the mark algorithm. The first step of the reduce algorithm asks us to create equivalence classes, which gives us six distinct sets, each containing a different state in  $\widehat{M}$ . The next step asks us to create a new state for each equivalence class that contains indistinguishable states. Since we have no such states, there is no need to create these new states. The rest of the steps in the reduce algorithm thus become superfluous. We see, then, that the DFA  $\widehat{M}_1$  produced by mark and reduce is actually  $\widehat{M}$ . This contradicts our assumption that  $\widehat{M}$  is not the minimal DFA. Thus, we have that  $\widehat{M}$  is the minimal DFA.

**#QED**

**§2.4 4.** Minimize the states in the depicted DFA.

