2.3 Equivalence of Deterministic and Nondeterministic Finite Accepters

We now come to a fundamental question. In what sense are dfa's and nfa's different? Obviously, there is a difference in their definition, but this does not imply that there is any essential distinction between them. To explore this question, we introduce the concept of equivalence between automata.

Definition 2.7

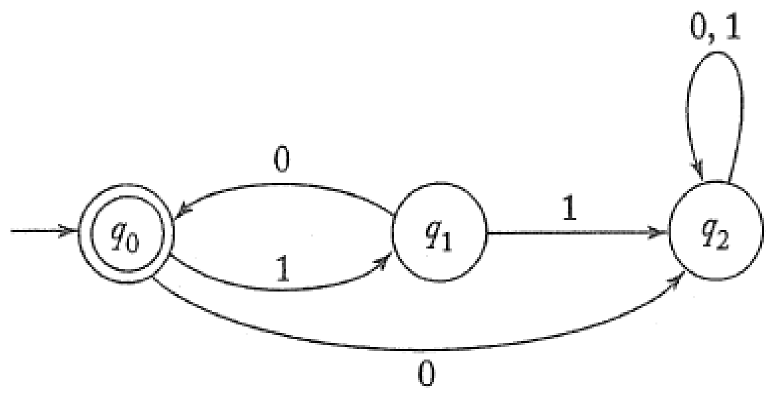
Two finite accepters, M_1 and M_2, are said to be equivalent if

$$L(M_1) = L(M_2),$$

that is, if they both accept the same language.

Example 2.11 The dfa shown in Figure 2.11 is equivalent to the nfa in Figure 2.9 since they both accept the language $\{(10)^n : n \geq 0\}$.

Figure 2.11



Example 2.12 Convert the nfa in Figure 2.12 to an equivalent dfa.

Figure 2.12

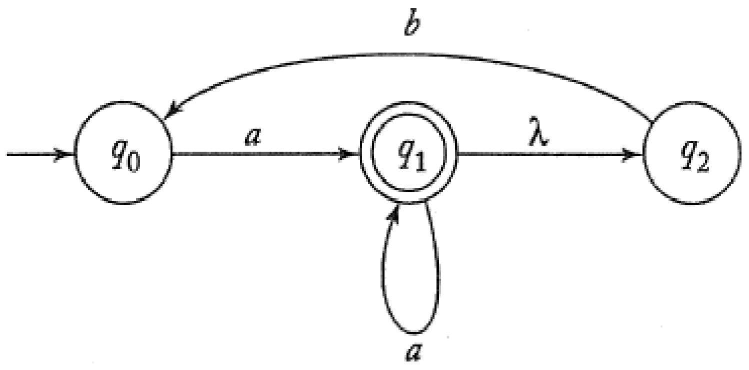
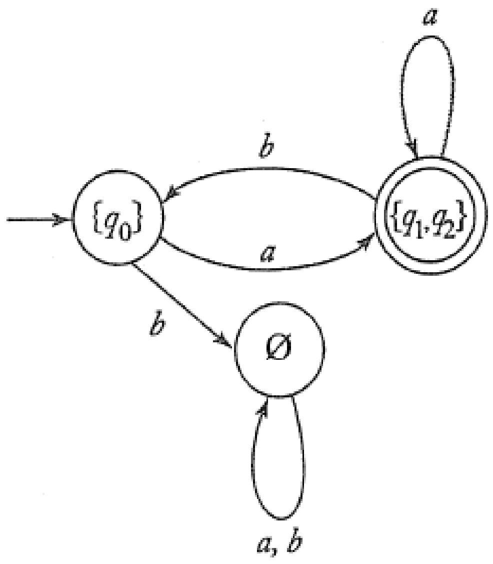


Figure 2.13



Theorem 2.2 Let L be the language accepted by a nondeterministic finite accepter $M_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$. Then there exists a deterministic finite accepter $M_D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ such that

$$L = L(M_D).$$

Proof: Given M_N, we use the procedure *nfa-to-dfa* below to construct the transition graph G_D for M_D. To understand the construction, remember that G_D has to have certain properties. Every vertex must have exactly $|\Sigma|$ outgoing edges, each labeled with a different element of Σ. During the construction, some of the edges may be missing, but the procedure continues until they are all there.

procedure: nfa-to-dfa

1. Create a graph G_D with vertex $\{q_0\}$. Identify this vertex as the initial vertex.

2. Repeat the following steps until no more edges are missing.

   Take any vertex $\{q_i, q_j, ..., q_k\}$ of G_D that has no outgoing edge for some $a \in \Sigma$. Compute $\delta_N^*(q_i, a), \delta_N^*(q_j, a), ..., \delta_N^*(q_k, a)$. If

   $$\delta_N^*(q_i, a) \cup \delta_N^*(q_j, a) \cup ... \cup \delta_N^*(q_k, a) = \{q_l, q_m, ..., q_n\},$$

   create a vertex for G_D labeled $\{q_l, q_m, ..., q_n\}$ if it does not already exist.

   Add to G_D an edge from $\{q_i, q_j, ..., q_k\}$ to $\{q_l, q_m, ..., q_n\}$ and label it with a.

3. Every state of G_D whose label contains any $q_f \in F_N$ is identified as a final vertex.

4. If M_N accepts λ, the vertex $\{q_0\}$ in G_D is also made a final vertex.

Example 2.13 Convert the nfa in Figure 2.14 into an equivalent deterministic machine.

Since $\delta_N(q_0, 0) = \{q_0, q_1\}$, we introduce the state $\{q_0, q_1\}$ in G_D and add an edge labeled 0 between $\{q_0\}$ and $\{q_0, q_1\}$. In the same way, considering $\delta_N(q_0, 1) = \{q_1\}$ gives us the new state $\{q_1\}$ and an edge labeled 1 between it and $\{q_0\}$.

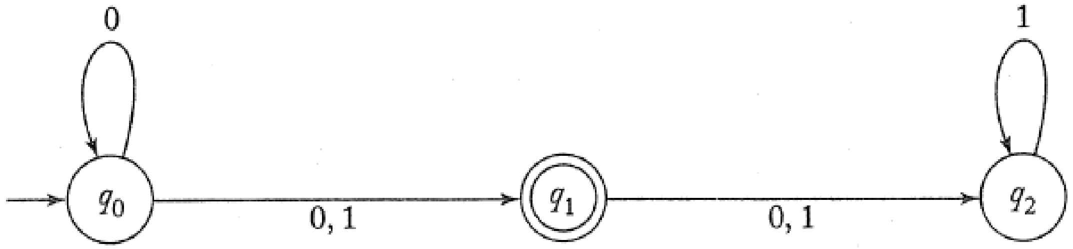
There are now a number of missing edges, so we continue, using the construction of Theorem 2.2. Looking at the state $\{q_0, q_1\}$, we see that there is no outgoing edge labeled 0, so we compute

$$\delta_N^*(q_0, 0) \cup \delta_N^*(q_1, 0) = \{q_0, q_1, q_2\}.$$

This gives us the new state $\{q_0, q_1, q_2\}$ and the transition

$$\delta_D(\{q_0, q_1\}, 0) = \{q_0, q_1, q_2\}.$$

Figure 2.14



Then, using $a = 1$, $i = 0$, $j = 1$, $k = 2$,

$$\delta_N^* (q_0, 1) \cup \delta_N^* (q_1, 1) \cup \delta_N^* (q_2, 1) = \{q_1, q_2\}$$

makes it necessary to introduce yet another state $\{q_1, q_2\}$. At this point, we have the partially constructed automaton shown in Figure 2.15. Since there are still some missing edges, we continue until we obtain the complete solution in Figure 2.16.

Figure 2.15

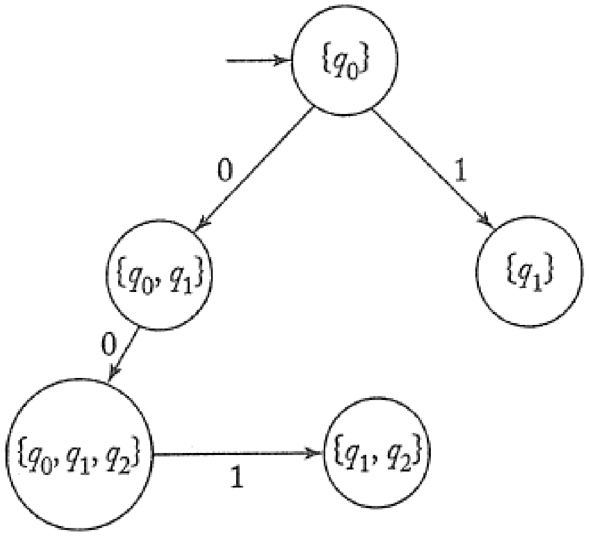
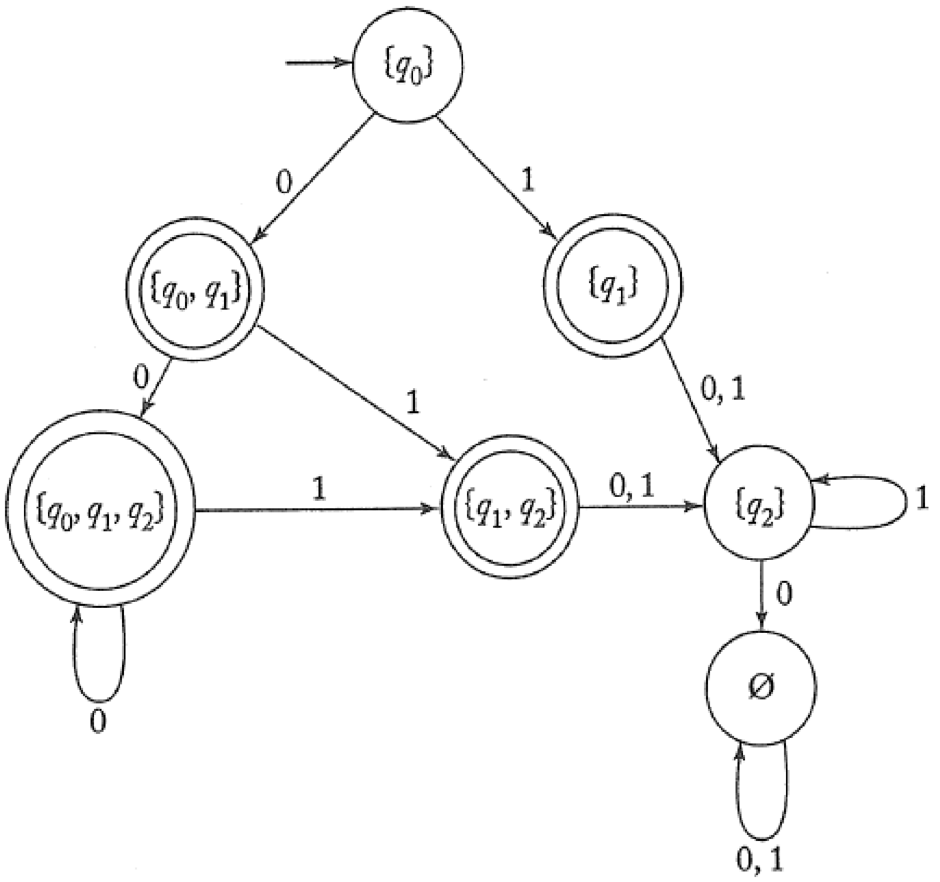


Figure 2.16



$\{q_0\}$
0
1
$\{q_0, q_1\}$
$\{q_1\}$
0
1
0, 1
$\{q_0, q_1, q_2\}$
1
$\{q_1, q_2\}$
0, 1
$\{q_2\}$
1
0
0
$\varnothing$
0, 1