Formal languages and Automata
形式语言与自动机

Chapter2 FINITE AUTOMATA

Beijing University of Posts and Telecommunications
北京邮电大学
刘咏彬 liuyb@bupt.edu.cn

2023.9

NONDETERMINISTIC FINITE ACCEPTERS

- Nondeterminism means a choice of moves for an automaton.
- Rather than prescribing a unique move in each situation, we allow a set of possible moves.
- Formally, we achieve this by defining the transition function so that its range is a set of possible states.

DEFINITION 2.4

- A nondeterministic finite accepter or nfa is defined by the quintuple

$$M = (Q, \Sigma, \delta, q_0, F)$$

  where Q, Σ, q_0, F are defined as for deterministic finite accepters, but

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \rightarrow 2^Q$$

- For instance, the current state is q_1, the symbol a is read, and

$$\delta\ (q_1, a) = \{q_0, q_2\}$$

nfa EXAMPLE 2.7

EXAMPLE 2.7

Consider the transition graph in Figure 2.8. It describes a nondeterministic accepter since there are two transitions labeled a out of q_0.

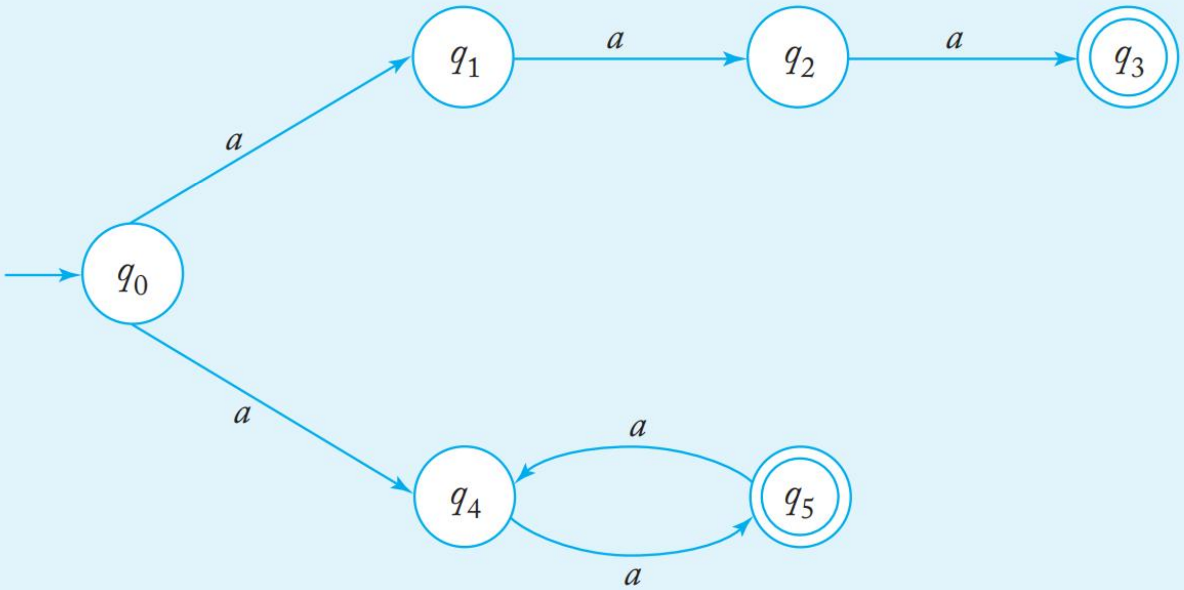


FIGURE 2.8

nfa EXAMPLE 2.7

EXAMPLE 2.7

Consider the transition graph in Figure 2.8. It describes a nondeterministic accepter since there are two transitions labeled a out of q_0.

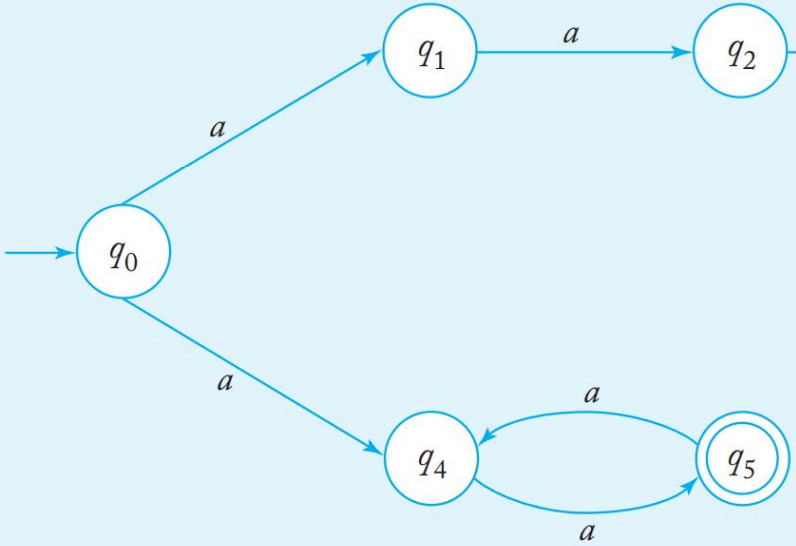


FIGURE 2.8

transition function 转移函数

$$M = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{a\}, \delta, q_0, \{q_3, q_5\})$$

$$\delta(q_0, a) = \{q_1, q_2\}$$
$$\delta(q_1, a) = \{q_2\}$$
$$\delta(q_2, a) = [q_3\}$$
$$\delta(q_3, a) = \Phi$$
$$\delta(q_4, a) = \{q_5\}$$
$$\delta(q_5, a) = \{q_4\}$$

nfa EXAMPLE 2.7

EXAMPLE 2.7

Consider the transition graph in Figure 2.8. It describes a nondeterministic accepter since there are two transitions labeled a out of q_0.

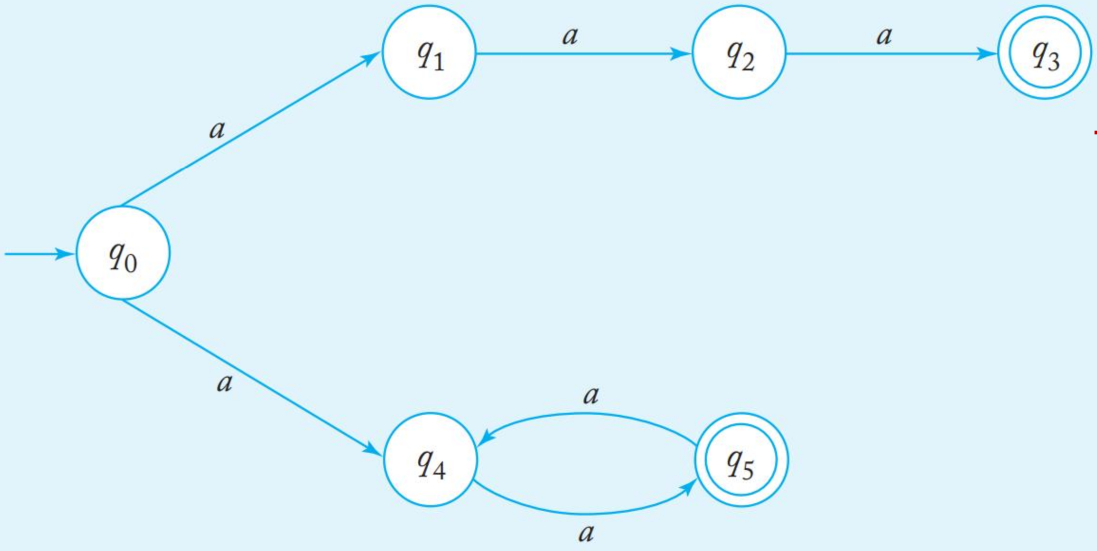


FIGURE 2.8

transition table 状态转移表

States	Input Symbol
	a
$\rightarrow q_0$	$\{q_1, q_4\}$
q_1	$\{q_2\}$
q_2	$\{q_3\}$
$*q_3$	Φ
q_4	$\{q_5\}$
$*q_5$	$\{q_4\}$

nfa EXAMPLE 2.7,

EXAMPLE 2.7

Consider the transition graph in Figure 2.8. It describes a nondeterministic accepter since there are two transitions labeled a out of q_0.

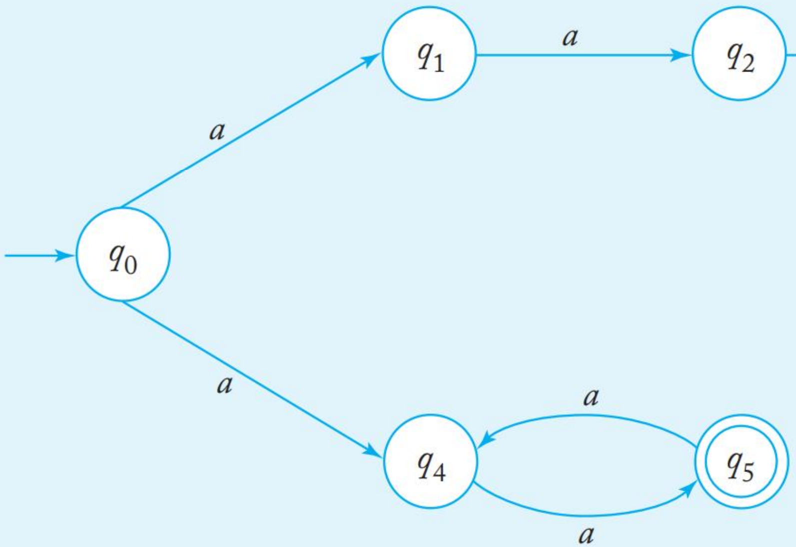
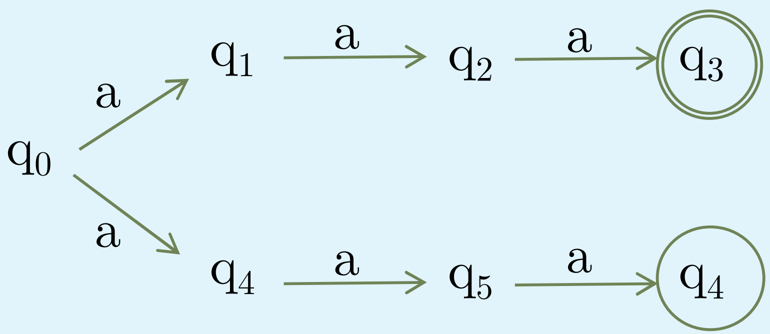


FIGURE 2.8

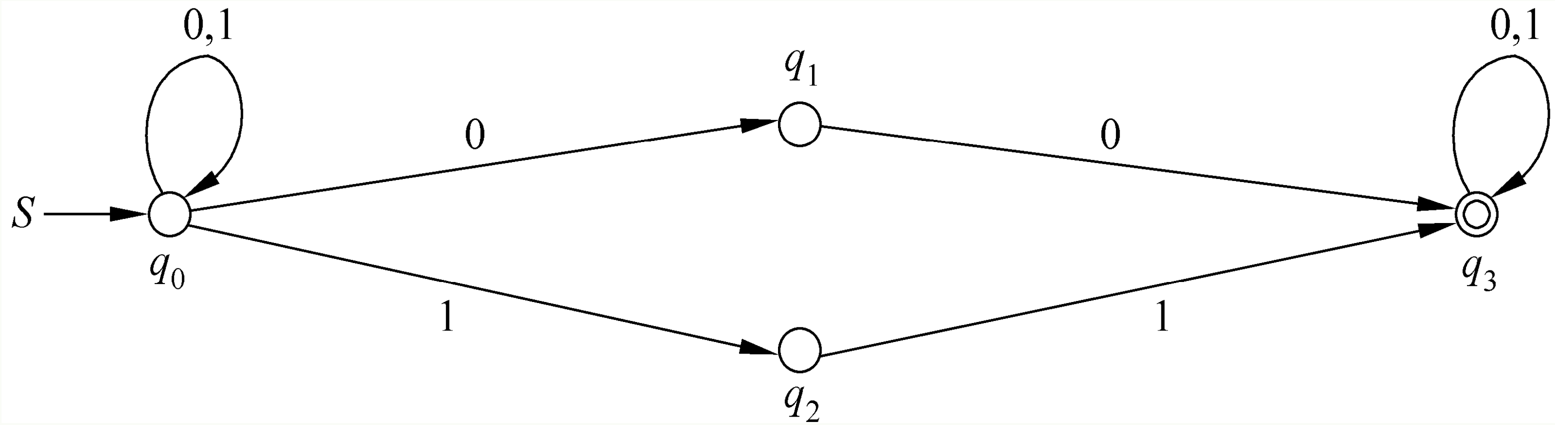
The process of recognizing the string 'aaa':



$$\delta^*(q_0, aaa) = \{q_3, q_4\}$$

补充例子

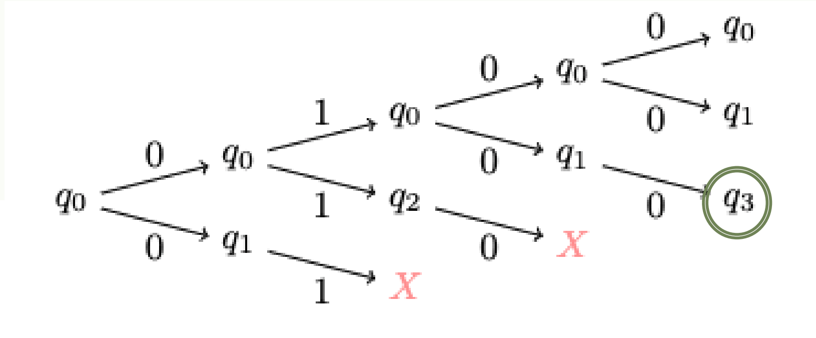
- 接受{x|x∈{0，1}*，且x含有子串00或11}的nfa对应的转移图和转移表。

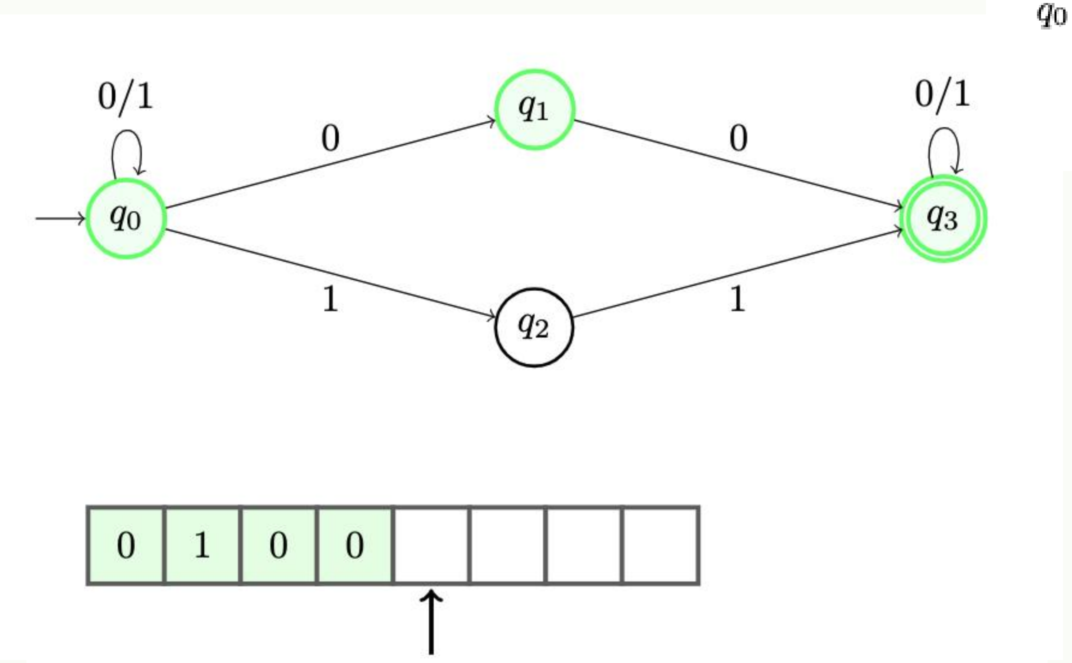


状态说明	状态	输入字符	
		0	1
启动状态	q_0	$\{q_0, q_1\}$	$\{q_0, q_2\}$
	q_1	$\{q_3\}$	Φ
	q_2	Φ	$\{q_3\}$
终止状态	q_3	$\{q_3\}$	$\{q_3\}$

补充例子

- How does an NFA analyze an input string?
- Suppose the input string is "0100."





$$\delta^*(q_0, 0100) = \{q_0, q_1, q_3\}$$

EXAMPLE 2.8

- A nondeterministic automaton is shown in Figure 2.9.
- It is nondeterministic not only because several edges with the same label originate from one vertex, but also because it has a λ-transition（空转移）.

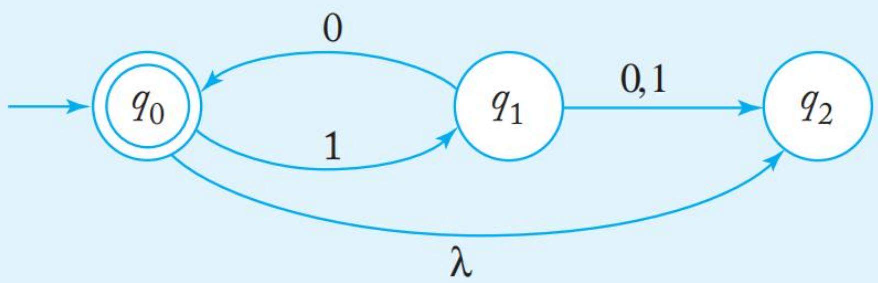


FIGURE 2.9

Nfa's transition function

- DEFINITION 2.5
  - Extended transition function(扩展的转移函数)

$$\delta^* : Q \times \Sigma^* \rightarrow 2^Q$$

  - $\delta^*(q_i, w)$ contains q_j if and only if there is a walk in the transition graph from q_i to q_j labeled w.

$$\delta^*(q_i, w) = \begin{cases} \{q_i\} & w = \lambda \\ \bigcup_{p \in \delta^*(q_i, v)} \delta(p, a) & w = \end{cases}$$

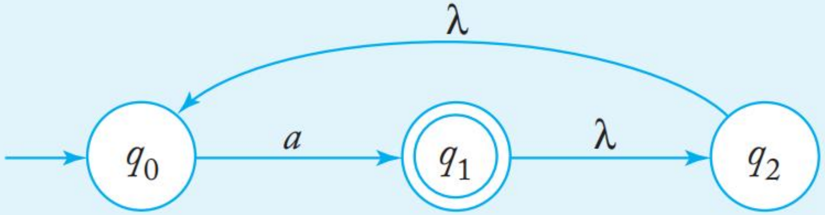
EXAMPLE 2.9

EXAMPLE 2.9

Figure 2.10 represents an nfa. It has several λ-transitions and some undefined transitions such as $\delta(q_2, a)$.

Suppose we want to find $\delta^*(q_1, a)$ and $\delta^*(q_2, \lambda)$. There is a walk labeled a involving two λ-transitions from q_1 to itself. By using some of the λ-edges twice, we see that there are also walks involving λ-transitions to q_0 and q_2. Thus,

$$\delta^*(q_1, a) = \{q_0, q_1, q_2\}.$$



EXAMPLE 2.9

Since there is a λ-edge between q_2 and q_0, we have immediately that $\delta^*(q_2, \lambda)$ contains q_0. Also, since any state can be reached from itself by making no move, and consequently using no input symbol, $\delta^*(q_2, \lambda)$ also contains q_2. Therefore,

$$\delta^*(q_2, \lambda) = \{q_0, q_2\}.$$

Using as many λ-transitions as needed, you can also check that

$$\delta^*(q_2, aa) = \{q_0, q_1, q_2\}.$$

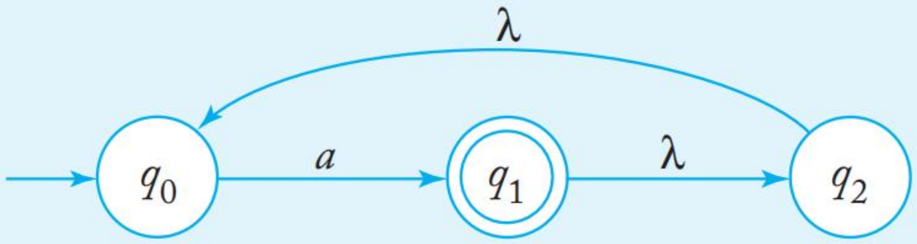


FIGURE 2.10

$$\delta^*(q_0, \lambda) = ?$$
$$\delta^*(q_1, \lambda) = ?$$

$$\delta^*(q_0, \lambda) = \{q_0\}$$
$$\delta^*(q_1, \lambda) = \{q_0, q_1, q_2\}$$

Language accepted by nfa

- DEFINITION 2.6

  The language L accepted by an nfa $M = (Q, \Sigma, \delta, q_0, F)$ is defined as

  $$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \cap F \neq \emptyset\} .$$

  In words, the language consists of all strings w for which there is a walk labeled w from the initial vertex of the transition graph to some final vertex.

nonacceptance in nfa

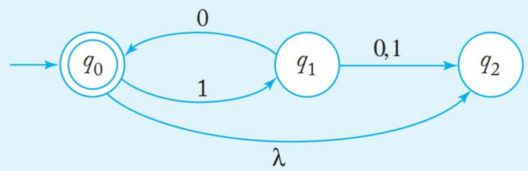


FIGURE 2.9

EXAMPLE 2.10

What is the language accepted by the automaton in Figure 2.9? It is easy to see from the graph that the only way the nfa can stop in a final state is if the input is either a repetition of the string 10 or the empty string. Therefore, the automaton accepts the language $L = \{(10)^n : n \geq 0\}$.

What happens when this automaton is presented with the string $w = 110$? After reading the prefix 11, the automaton finds itself in state q_2, with the transition $\delta(q_2, 0)$ undefined. We call such a situation a **dead configuration**, and we can visualize it as the automaton simply stopping without further action. But we must always keep in mind that such visualizations are imprecise and carry with them some danger of misinterpretation. What we can say precisely is that

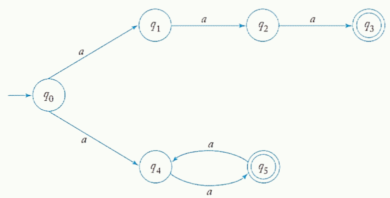
$$\delta^*(q_0, 110) = \varnothing.$$

Thus, no final state can be reached by processing $w = 110$, and hence the string is not accepted.

Why Nondeterminism?

- Digital computers are deterministic
  - Can simulate nondeterminism.

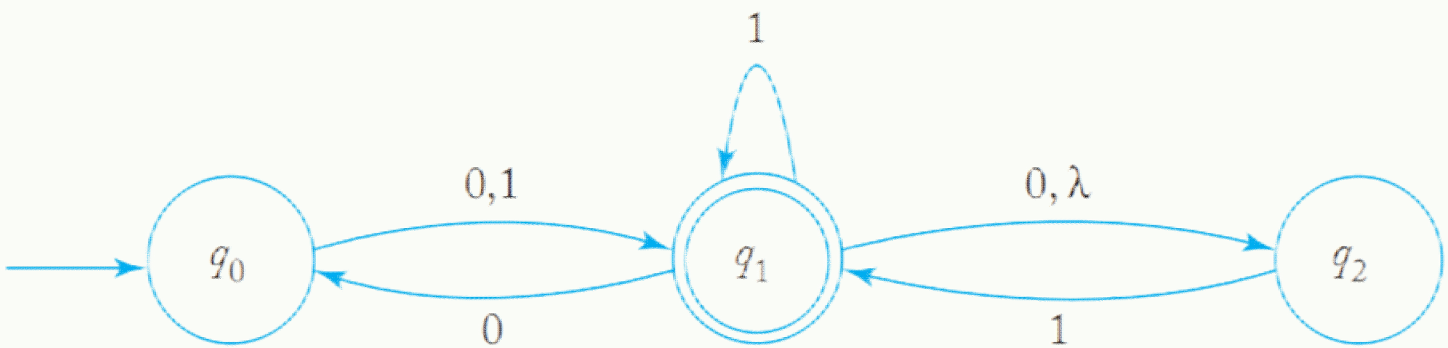
- Nondeterministic algorithm
  - Can do exhaustive search without backtracking
  - Nondeterministic machines can serve as models of search-and-backtrack algorithms.



  - Helpful in solving problems easily.
  - An effective mechanism for describing some complicated languages concisely,e.g., $\mathrm{S} \to \mathrm{aSb}|\lambda$

EXERCISE

- Which of the strings 00, 01001, 10010, 000, 0000 are accepted by the following nfa?



EXERCISE

设计 NFA 识别 $L = \{w \in \{0, 1\}^* \mid w$ 倒数第 3 个字符是 $1\}$。

EQUIVALENCE OF DFA AND NFA

- DEFINITION 2.7

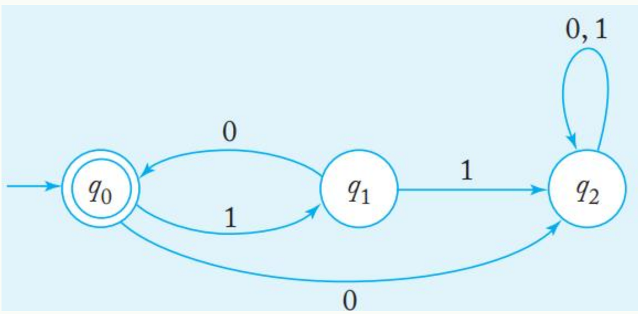
  Two finite accepters, M_1 and M_2, are said to be equivalent （等价的） if

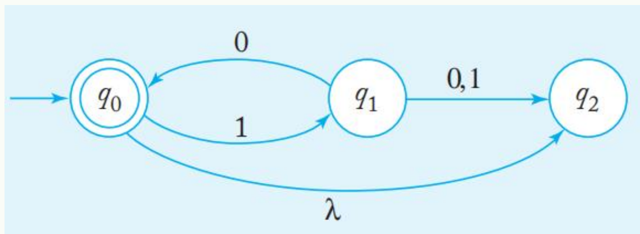
  $$L(M_1) = L(M_2),$$

  that is, if they both accept the same language.

- EXAMPLE 2.11

  - There are generally many accepters for a given language,e.g., the nfa and dfa below both accept the language $\{(10)^n : n \geq 0\}$.





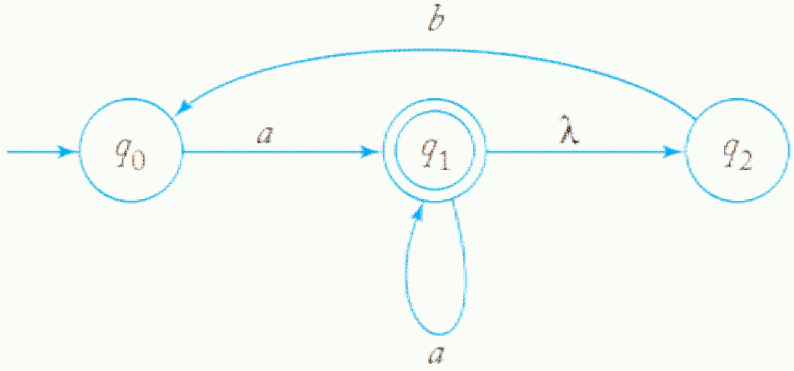
EQUIVALENCE OF DFA AND NFA

- Constructive Proof
- After an nfa has read a string w, we may not know exactly what state it will be in, but we can say that it must be in one state of a set of possible states, say $\{q_i, q_j, \ldots, q_k\}$.
- Label the states of the dfa with a set of states in such a way that, after reading w, the equivalent dfa will be in a single state labeled $\{q_i, q_j, \ldots, q_k\}$.
- Since for a set of |Q| states there are exactly $2^{|Q|}$ subsets, the corresponding dfa will have a finite number of states.

EXAMPLE 2.12

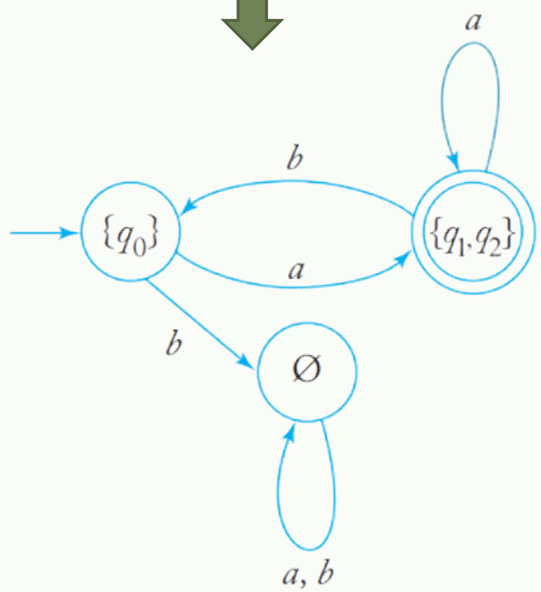
- Convert the nfa to an equivalent dfa.

$$\delta(\{q_0\}, a) = \{q_1, q_2\}.$$
$$\delta(\{q_0\}, b) = \emptyset.$$



- A state labeled □ represents an impossible move for the nfa and, there_x0002_fore, means nonacceptance of the string. Consequently, this state in the dfa must be a nonfinal trap state.

$$\delta(\{q_1, q_2\}, a) = \{q_1, q_2\}.$$
$$\delta(\{q_1, q_2\}, b) = \{q_0\}.$$



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 $L = L(M_D)$
- (see next page)

THEOREM 2.2

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.

FIGURE 2.14

EXAMPLE 2.13

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

FIGURE 2.14

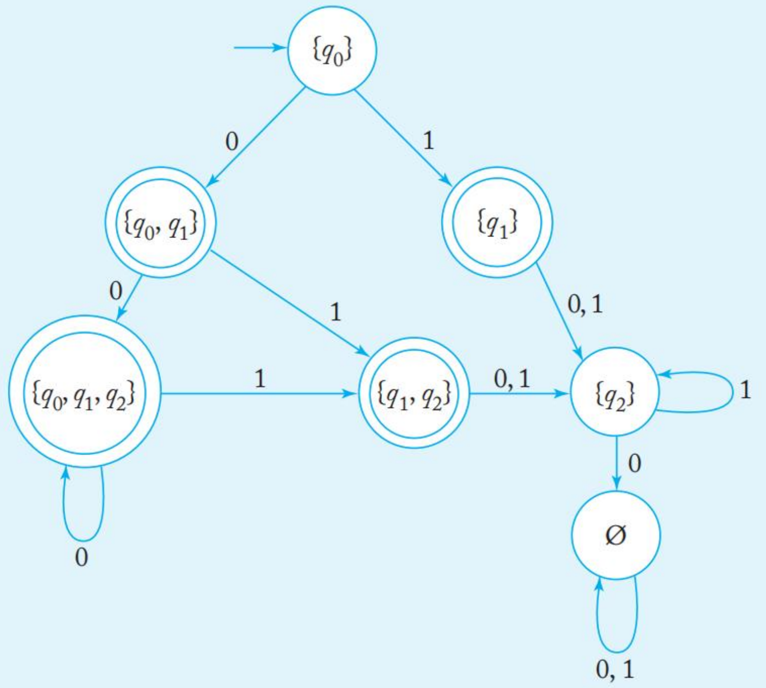


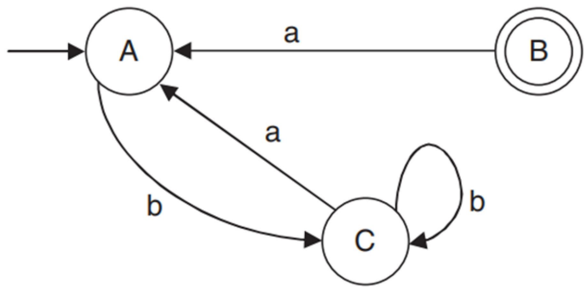
FIGURE 2.16

EXERCISE

1 An NFA with a set of states Q is converted to an equivalent DFA with a set of states Q′. Find which is true.

a) $Q' = Q$ b) $Q' \subseteq Q$
c) $Q \subseteq Q'$ d) None of these

2 The language accepted by the given FA is



a) (ab)* b) bb*a
c) b(ba)*a d) Null

3 In the previous FA, B is called

a) Dead state
b) Inaccessible state
c) Both a and b
d) None of these

END