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

REDUCTION OF THE NUMBER OF STATES IN FINITE AUTOMATA

- Any dfa defines a unique language, but the converse is not true.
- For a given language, there are many dfa’ s that accept it.
- The two dfa’ s depicted in Figure 2.17(a) and 2.17(b) are equivalent

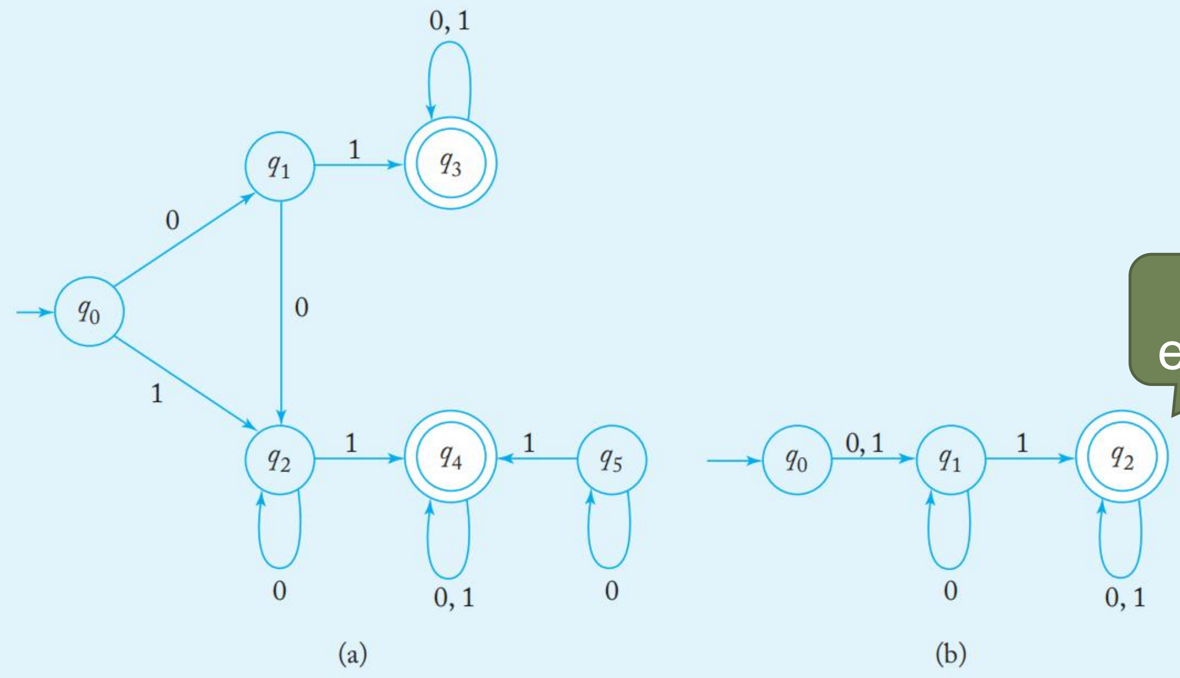


FIGURE 2.17

storage efficiency

Indistinguishable and Distinguishable(1)

- DEFINITION 2.8

  Two states p and q of a dfa are called indistinguishable (不可区分的) if

  $$\delta^*(p, w) \in F \text{ implies } \delta^*(q, w) \in F,$$

  and

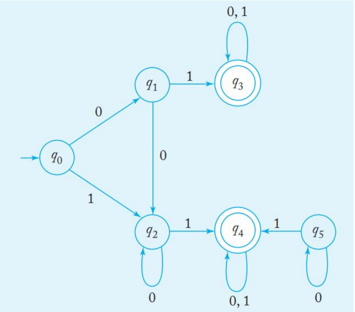
  $$\delta^*(p, w) \notin F \text{ implies } \delta^*(q, w) \notin F,$$

  for **all** $w \in \Sigma^*$.

  If, on the other hand, there **exists** some string $w \in \Sigma^*$ such that

  $$\delta^*(p, w) \in F \text{ and } \delta^*(q, w) \notin F,$$

  or vice versa, then the states p and q are said to be distinguishable （可区分的） by astring w.



Indistinguishable and Distinguishable(2)

- Two states are either indistinguishable or distinguishable.
- Indistinguishability has the properties of an **equivalence relation (等价关系)**:

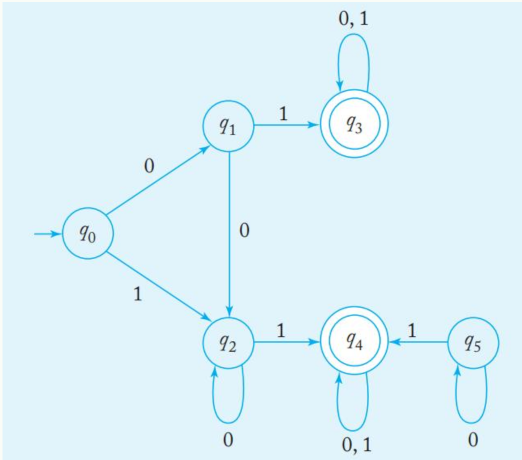
  If p and q are indistinguishable and if q and r are also indistinguishable, then so are p and r, and all three states are indistinguishable.

Procedure: mark (1)

- Finding pairs of distinguishable states
- procedure: mark

1. Remove all inaccessible (不可达的) states.
2. Consider all pairs of states (p, q). If $p \in F$ and $q \notin F$ or vice versa, mark the pair (p, q) as distinguishable.
3. Repeat the following step until no previously unmarked pairs are marked:

   For all pairs (p, q) and all $a \in \Sigma$, compute $\delta(p, a) = p_a$ and $\delta(q, a) = q_a$. If the pair (p_a, q_a) is marked as distinguishable, mark (p, q) as distinguishable.



1. Remove q5
2. fill in the distinguishable table

q_1				
q_2				
q_3	X	X	X	
q_4	X	X	X	
	q_0	q_1	q_2	q_3

Procedure: mark (2)

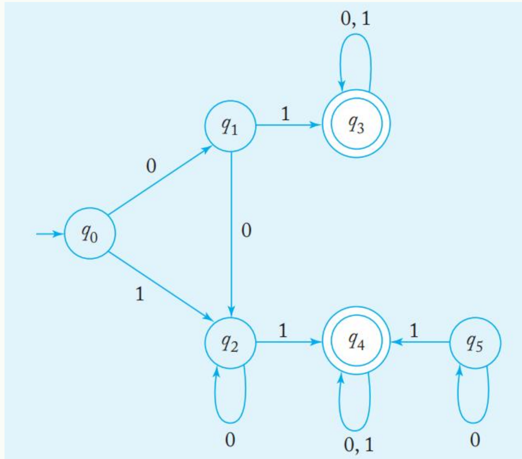
- Finding pairs of distinguishable states
- Procedure: mark

  1. Remove all inaccessible (不可达的) states.

  2. Consider all pairs of states (p, q). If $p \in F$ and $q \notin F$ or vice versa, mark the pair (p, q) as distinguishable.

  3. Repeat the following step until no previously unmarked pairs are marked:

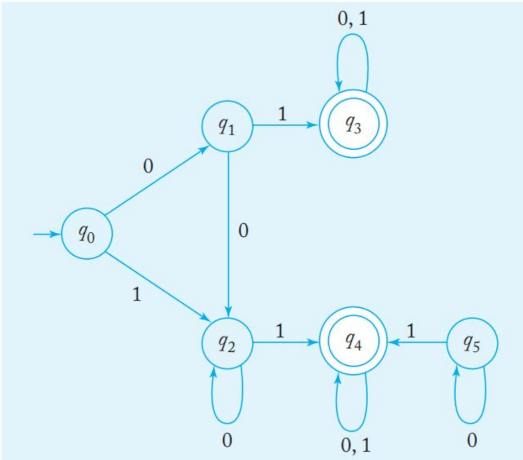
     For all pairs (p, q) and all $a \in \Sigma$, compute $\delta(p, a) = p_a$ and $\delta(q, a) = q_a$. If the pair (p_a, q_a) is marked as distinguishable, mark (p, q) as distinguishable.



1. Remove q5
2. fill in the distinguishable table. mark all the final state and non-final state pairs

q_1				
q_2				
q_3	X	X	X	
q_4	X	X	X	
	q_0	q_1	q_2	q_3

EXAMPLE OF PROCEDURE MARK



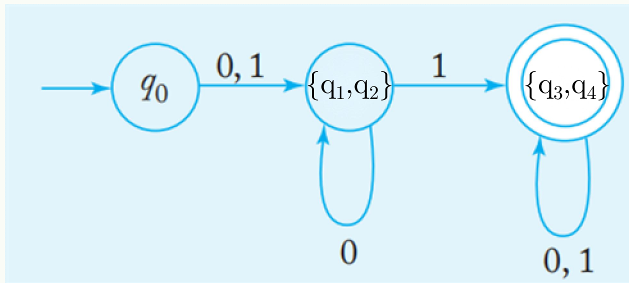
1. Remove q5
2. fill in the distinguishable table. mark all the final state and non-final state pairs

3. $\delta(q_0, 1)=q_2, \delta(q_1, 1)=q_3$, (q_2,q_3) are distinguishable, so mark (q_0,q_1) as distinguishable $\delta(q_0, 1)=q_2, \delta(q_2, 1)=q_4$, (q_2,q_4) are distinguishable, so mark (q_0,q_2) as distinguishable

4. Starting from q_3 or q_4, reading any binary string leads to a transition to an final state, hence (q_3, q_4) are indistinguishable.

5. $\delta(q_1, 0)=q_2, \delta(q_2, 0)=q_2$, $\delta(q_1, 1)=q_3, \delta(q_2, 1)=q_4$, (q_3, q_4) is indistinguishable, hence (q_1, q_2) are indistinguishable

q_1	X			
q_2	X			
q_3	X	X	X	
q_4	X	X	X	
	q_0	q_1	q_2	q_3



Proof of Procedure Mark (1)

- THEOREM 2.3

  The **procedure mark**, applied to any dfa $M = (Q, \Sigma, \delta, q_0, F)$, terminates and determines **all** pairs of distinguishable states.

- Proof:

  Note first that states q_i and q_j are distinguishable with a string of length n if and only if there are transitions

$$\delta(q_i, a) = q_k \tag{2.5}$$

and

$$\delta(q_j, a) = q_l \tag{2.6}$$

for some $a \in \Sigma$, with q_k and q_l distinguishable by a string of length $n-1$.

Proof of Procedure Mark (2)

We First to show that at the completion of the nth pass through the loop in step 3, all states distinguishable by strings of length n or less have been marked.

Proof by induction:

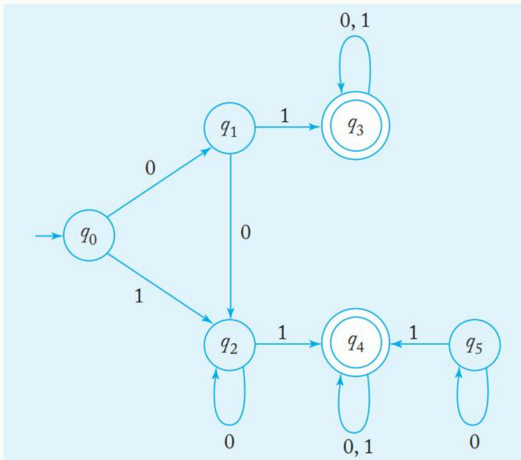
- In step 2, we mark all pairs indistinguishable by λ, so we have a basis with $n = 0$ for an induction.
- Assume that the claim is true for all $i = 0, 1, \ldots, n - 1$. all states distinguishable by strings of length up to $n - 1$ have been marked.
- Because of (2.5) and (2.6) above, at the end of this pass, all states distinguishable by strings of length up to n will bemarked.
- By induction then, we can claim that, for any n, at the completion of the nth pass, all pairs distinguishable by strings of length n or less have been marked.

Proof of Procedure Mark (3)

- To show that this procedure marks all distinguishable states, assume that the loop terminates after n passes. This means that during the nth pass **no new states were marked**.
- From (2.5) and (2.6), it then follows that there cannot be any states distinguishable by a string of length n, but not distinguishable by any shorter string.
- But if there are no states distinguishable only by strings of length n, there cannot be any states distinguishable only by strings of length n+ 1, and so on.
- As a consequence, when the loop terminates, all distinguishable pairs have been marked.

Procedure mark

- The procedure mark can be implemented by partitioning the states into equivalence classes.
- Whenever two states are found to be distinguishable, they are immediately put into separate equivalence classes.
- mark $\{q_0, q_1, q_2, q_3, q_4\}$
  - 0th pass distinguishable: $\{q_0, q_1, q_2\}, \{q_3, q_4\}$
  - 1th pass distinguishable: $\{q_0\}, \{q_1, q_2\}, \{q_3, q_4\}$
  - 2th pass distinguishable: stop



EXAMPLE 2.15

Consider the automaton in Figure 2.18.

In the second step of procedure *mark* we partition the state set into final and nonfinal states to get two equivalence classes $\{q_0, q_1, q_3\}$ and $\{q_2, q_4\}$. In the next step, when we compute

$$\delta(q_0, 0) = q_1$$

and

$$\delta(q_1, 0) = q_2,$$

we recognize that q_0 and q_1 are distinguishable, so we put them into different sets. So $\{q_0, q_1, q_3\}$ is split into $\{q_0\}$ and $\{q_1, q_3\}$. Also, since $\delta(q_2, 0) = q_3$ and $\delta(q_4, 0) = q_4$, the class $\{q_2, q_4\}$ is split into $\{q_2\}$ and $\{q_4\}$. The rest of the computations show that no further splitting is needed.

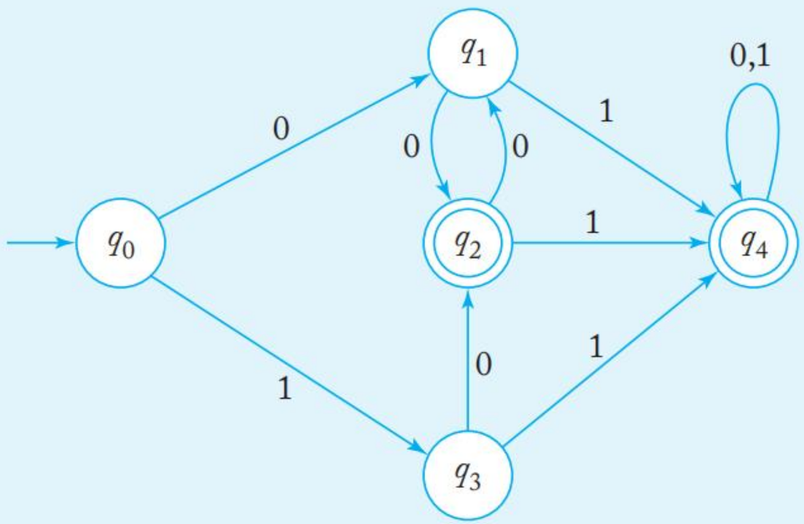


FIGURE 2.18

Construction of the Minimal Dfa

- Procedure: reduce

Given a dfa $M = (Q, \Sigma, \delta, q_0, F)$, we construct a reduced dfa $\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\delta}, q_0, \widehat{F})$ as follows.

1. Use procedure mark to generate the equivalence classes, say$\{q_i, q_j, \cdots, q_k\}$,as described.
2. For each set $\{q_i, q_j, \cdots, q_k\}$ of such indistinguishable states, create a state labeled $ij\cdots k$ for $\widehat{M}$.
3. For each transition rule of M of the form

$$\delta(q_r, a) = q_p$$

find the sets to which q_r and q_p belong. If $q_r \in \{q_i, q_j, \cdots, q_k\}$ and $q_p \in \{q_l, q_m, \cdots, q_n\}$, add to $\widehat{\delta}$ a rule

$$\widehat{\delta}(ij\cdots k) = lm\cdots n.$$

5. The initial state $\widehat{q}_0$ is that state of $\widehat{M}$ whose label includes the 0.
6. $\widehat{F}$ is the set of all the states whose label contains i such that $q_i \in F$

EXAMPLE 2.16

Continuing with Example 2.15, we create the states in Figure 2.19. Since, for example,

$$\delta(q_1, 0) = q_2,$$

there is an edge labeled 0 from state 13 to state 2. The rest of the transitions are easily found, giving the minimal dfa in Figure 2.19.

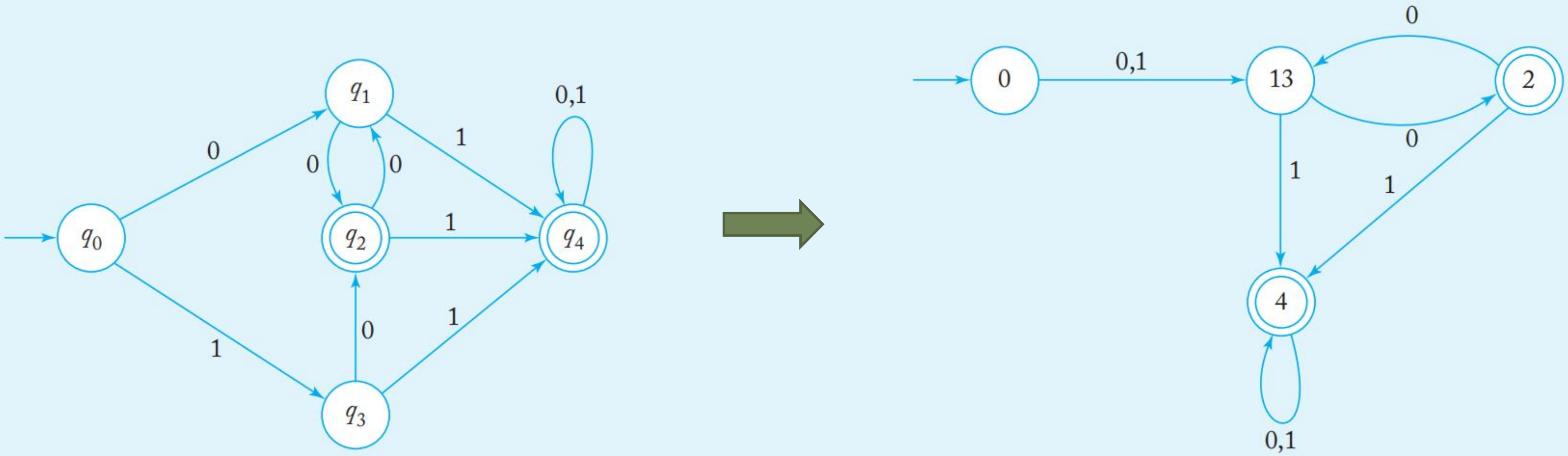


FIGURE 2.19

Minimal Dfa

- THEOREM 2.4
  - Given any dfa M, application of the procedure *reduce* yields another dfa $\widehat{M}$ such that

$$L(M) = L(\widehat{M})$$

  - Furthermore, $\widehat{M}$ is minimal in the sense that there is no other dfa with a smaller number of states that also accepts $L(M)$.

Proof $\widehat{M}$ is equivalent to the original dfa

- This is relatively easy and we can use inductive arguments similar to those used in establishing the equivalence of dfa’s and nfa’s.
- $\delta^*(q_i, w) = q_j$ if and only if the label of is of the $\widehat{\delta}^*(q_i, w) = q_j$ form $\cdots j \cdots$

Proof $\widehat{M}$ is minimal (1)

- Proof by contradiction
  - Assume that there is an equivalent dfa M_1, with transition function δ_1 and initial state q_0, equivalent to $\widehat{M}$, but with fewer states.
  - Since there are no inaccessible states in $\widehat{M}$, there must be distinct strings $w_1, w_2, \cdots, w_m$ such that
$$\hat{\delta}^*(p_0, w_i) = p_i,\ i = 1,2, \cdots, m$$
  - But since M_1 has fewer states than $\widehat{M}$ strings, say w_k and w_l, such that
$$\delta_1{}^*(q_0, w_k) = \delta_1{}^*(q_0, w_l)$$

Proof $\widehat{M}$ is minimal (2)

- Since p_k and p_l are distinguishable, there must be some string x such that $\widehat{\delta}^*(p_0, w_k x) = \widehat{\delta}^*(p_k, x)$ is a final state, and $\widehat{\delta}^*(p_0, w_l x) = \widehat{\delta}^*(p_l, x)$ is a nonfinal state (or vice versa).
  - In other words, $w_k x$ is accepted by $\widehat{M}$ and $w_l x$ is not.
  - But note that
  - $\delta_1{}^*(q_0, w_k x) = \delta_1{}^*(\delta_1{}^*(q_0, w_k), x)$
    $= \delta_1{}^*(\delta_1{}^*(q_0, w_l), w_k)$
- Thus, M_1 either accepts both $w_k x$ and $w_l x$ or rejects both, contradicting the assumption that $\widehat{M}$ and M_1 are equivalent. This contradiction provesthat M_1 cannot exist.

补充

- For any regular language regular language L, in an isomorphic sense（在同构意义上）, the minimal dfa of L is unique.

- Dfa isomorphism means that for any regular language RL, the minimal DFA accepting RL has a unique structure where states are one-to-one corresponded and state transitions are also corresponded accordingly between the DFAs.

END