CSE322
Minimization of finite Automaton & REGULAR LANGUAGES

Lecture #5

Definitions

Definition 3.10 Two states q_1 and q_2 are equivalent (denoted by $q_1 \equiv q_2$) if both $\delta(q_1, x)$ and $\delta(q_2, x)$ are final states, or both of them are nonfinal states for all $x \in \Sigma^*$.

As it is difficult to construct $\delta(q_1, x)$ and $\delta(q_2, x)$ for all $x \in \Sigma^*$ (there are an infinite number of strings in Σ^*), we give one more definition.

Definition 3.11 Two states q_1 and q_2 are k-equivalent ($k \geq 0$) if both $\delta(q_1, x)$ and $\delta(q_2, x)$ are final states or both nonfinal states for all strings x of length k or less. In particular, any two final states are 0-equivalent and any two nonfinal states are also 0-equivalent.

Properties

We mention some of the properties of these relations.

Property 1 The relations we have defined, i.e. equivalence and k-equivalence, are equivalence relations, i.e. they are reflexive, symmetric and transitive.

Property 2 By Theorem 2.1, these induce partitions of Q. These partitions can be denoted by π and π_k, respectively. The elements of π_k are k-equivalence classes.

Property 3 If q_1 and q_2 are k-equivalent for all $k \geq 0$, then they are equivalent.

Property 4 If q_1 and q_2 are $(k + 1)$-equivalent, then they are k-equivalent.

Property 5 $\pi_n = \pi_{n+1}$ for some n. (π_n denotes the set of equivalence classes under n-equivalence.)

Construction of Minimum Automaton

Step 1 (Construction of π_0). By definition of 0-equivalence, $\pi_0 = \{Q_1^0, Q_2^0\}$ where Q_1^0 is the set of all final states and $Q_2^0 = Q - Q_1^0$.

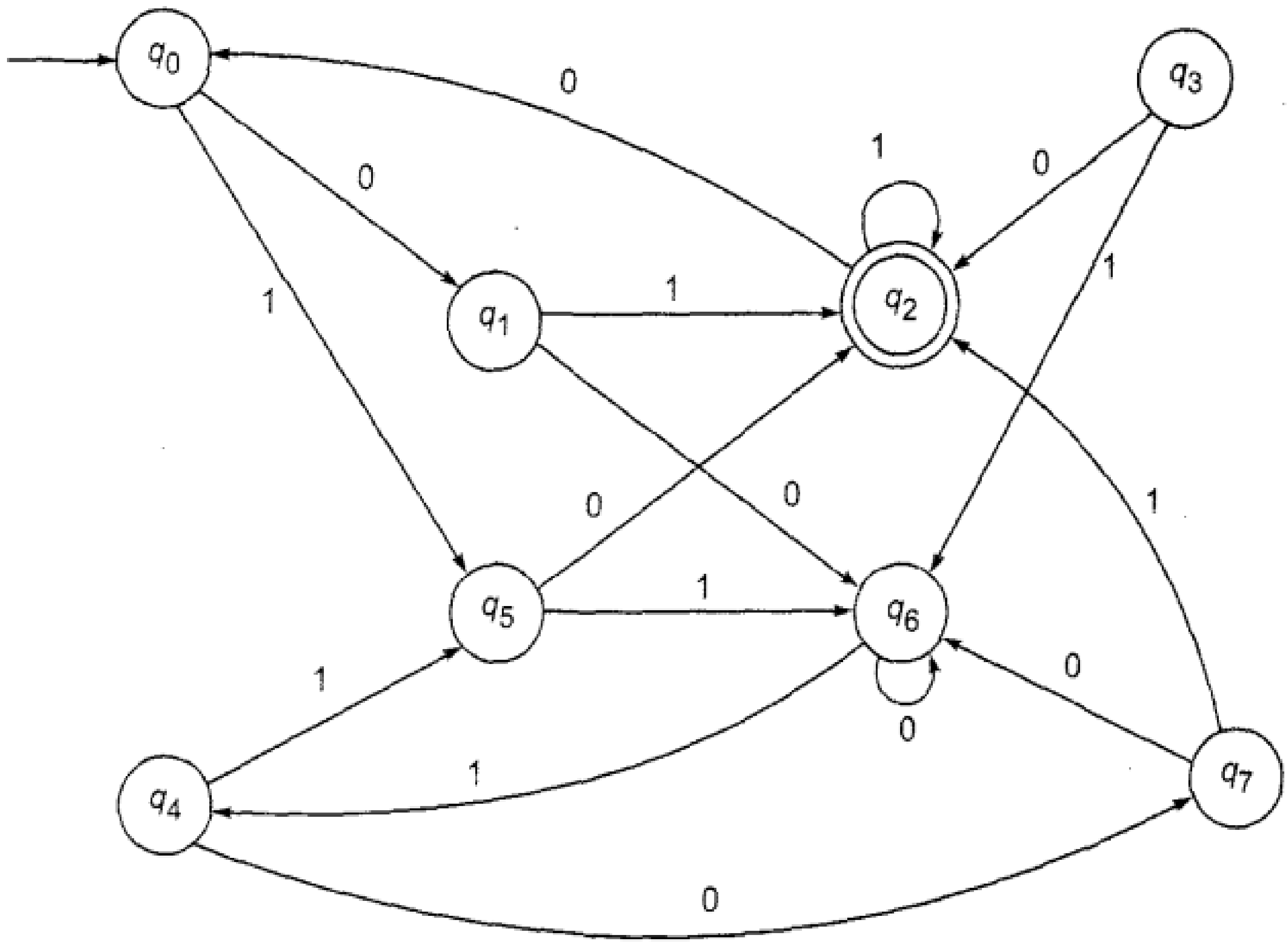
Step 2 (Construction of π_{k+1} from π_k). Let Q_i^k be any subset in π_k. If q_1 and q_2 are in Q_i^k, they are $(k + 1)$-equivalent provided $\delta(q_1, a)$ and $\delta(q_2, a)$ are k-equivalent. Find out whether $\delta(q_1, a)$ and $\delta(q_2, a)$ are in the same equivalence class in π_k for every $a \in \Sigma$. If so, q_1 and q_2 are $(k + 1)$-equivalent. In this way, Q_i^k is further divided into $(k + 1)$-equivalence classes. Repeat this for every Q_i^k in π_k to get all the elements of π_{k+1}.

Step 3 Construct π_n for $n = 1, 2, \ldots$ until $\pi_n = \pi_{n+1}$.

Step 4 (Construction of minimum automaton). For the required minimum state automaton, the states are the equivalence classes obtained in step 3, i.e. the elements of π_n. The state table is obtained by replacing a state q by the corresponding equivalence class $[q]$.

Problem

- Construct a minimum state automaton equivalent to finite automaton



State/Σ	0	1
$\rightarrow q_0$	q_1	q_5
q_1	q_6	q_2
q_2	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

By applying step 1, we get

$$Q_1^0 = F = \{q_2\}, \qquad Q_2^0 = Q - Q_1^0$$

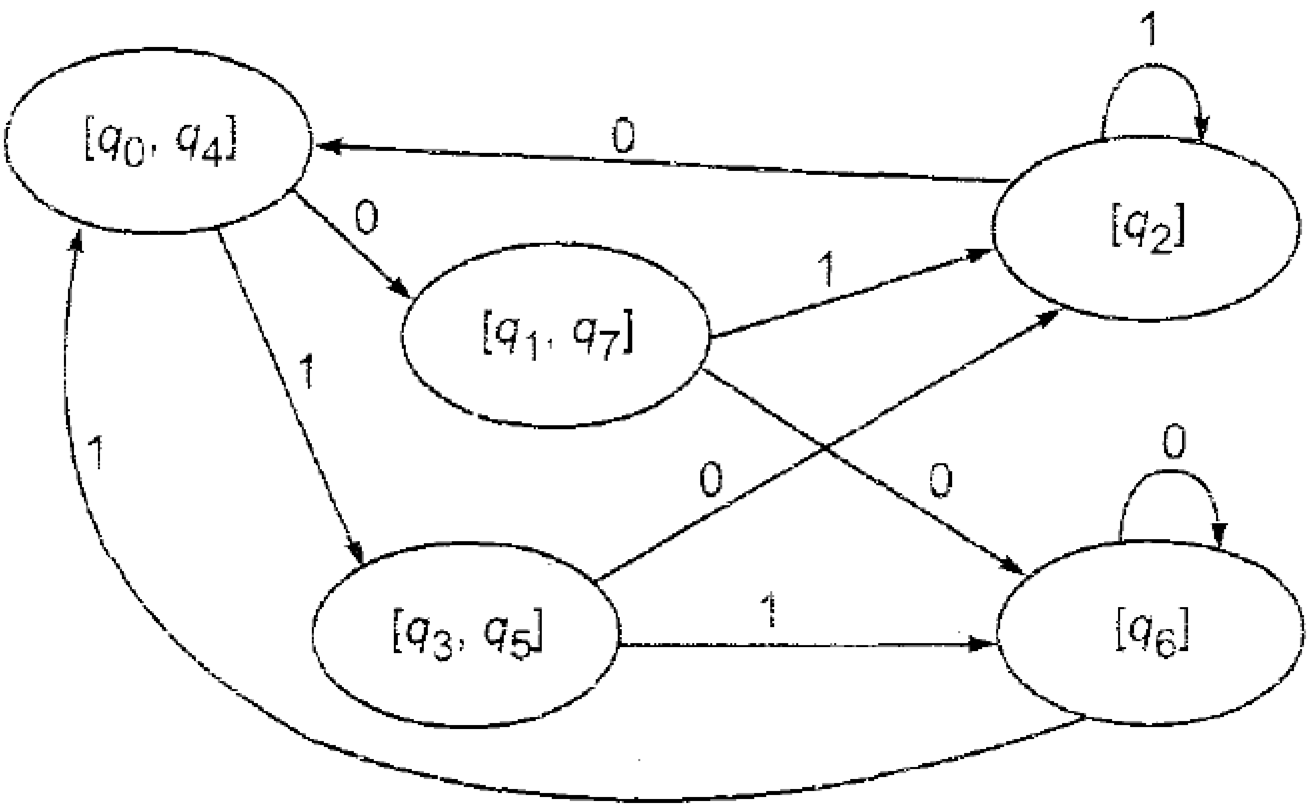
So,

$$\pi_0 = \{\{q_2\}, \{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}\}$$

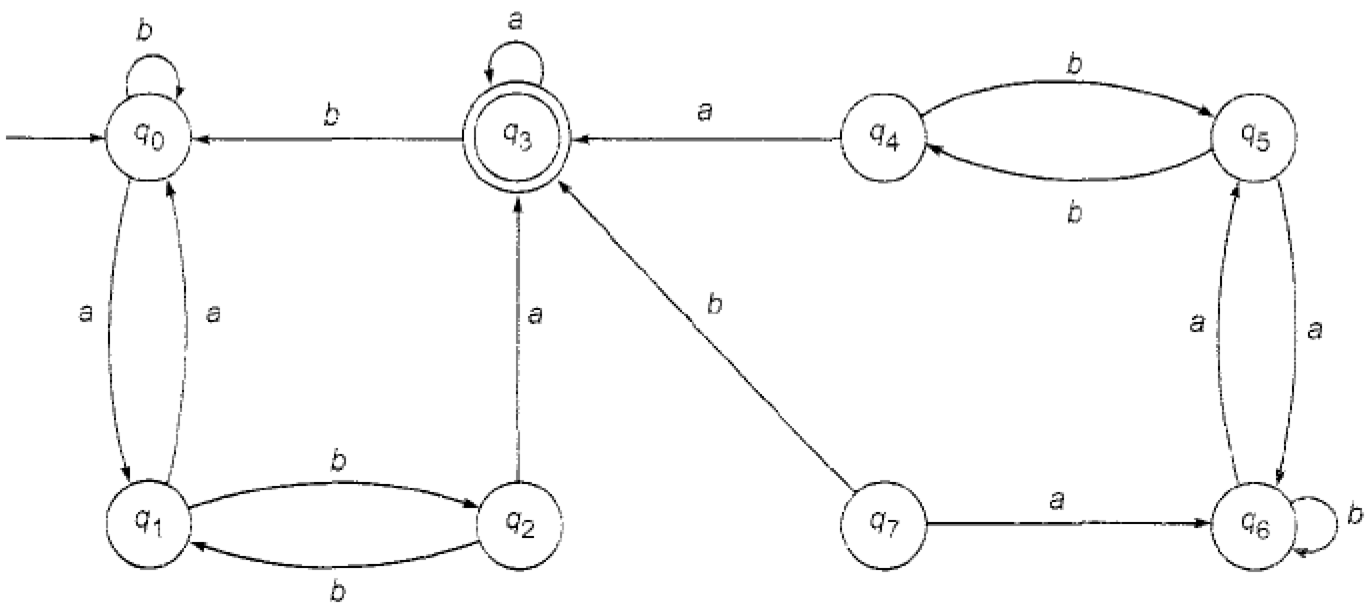
$$\pi_1 = \{\{q_2\}, \{q_0, q_4, q_6\}, \{q_1, q_7\}, \{q_3, q_5\}\}$$

$$\pi_3 = \{\{q_2\}, \{q_0, q_4\}, \{q_6\}, \{q_1, q_7\}, \{q_3, q_5\}\}$$

Solution



Construct the minimum state automaton equivalent to the transition diagram given by Fig. 3.14.



State/Σ	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_3	q_0
q_4	q_3	q_5
q_5	q_6	q_4
q_6	q_5	q_6
q_7	q_6	q_3

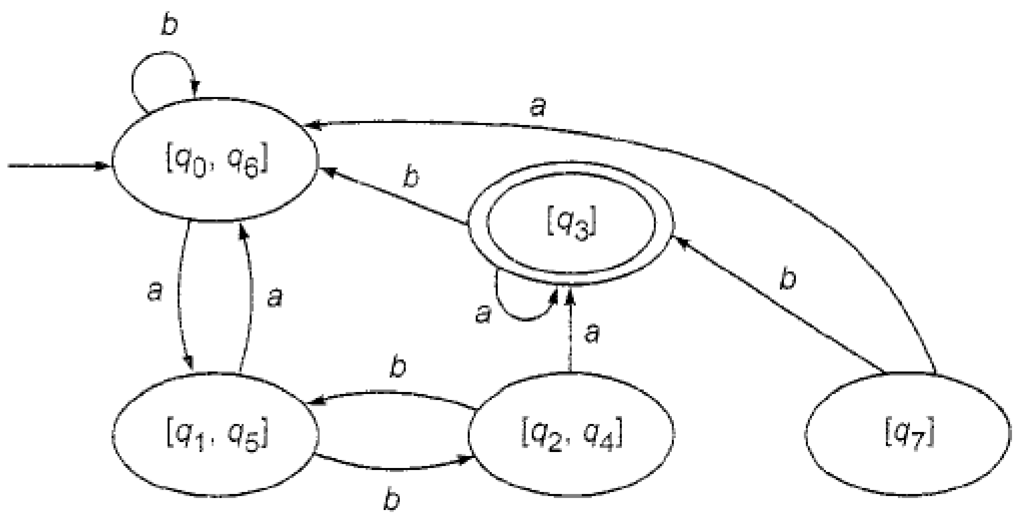
$$\pi_0 = \{\{q_3\}, \{q_0, q_1, q_2, q_4, q_5, q_6, q_7\}\}$$

$$\pi_1 = \{\{q_3\}, \{q_0, q_1, q_5, q_6\}, \{q_2, q_4\}, \{q_7\}\}$$

$$\pi_2 = \{\{q_3\}, \{q_0, q_6\}, \{q_1, q_5\}, \{q_2, q_4\}, \{q_7\}\}$$

$$\pi_3 = \{\{q_3\}, \{q_0, q_6\}, \{q_1, q_5\}, \{q_2, q_4\}, \{q_7\}\}$$

State/Σ	*a*	*b*
$[q_0, q_6]$	$[q_1, q_5]$	$[q_0, q_6]$
$[q_1, q_5]$	$[q_0, q_6]$	$[q_2, q_4]$
$[q_2, q_4]$	$[q_3]$	$[q_1, q_5]$
$[q_3]$	$[q_3]$	$[q_0, q_6]$
$[q_7]$	$[q_0, q_6]$	$[q_3]$



b
a
$[q_0, q_6]$
b
$[q_3]$
a
a
a
b
a
b
$[q_1, q_5]$
$[q_2, q_4]$
$[q_7]$
b

Minimize the given automata

EXAMPLE 3.21

Construct a minimum state automaton equivalent to a DFA whose transition table is defined by Table 3.30.

TABLE 3.30 DFA of Example 3.21

State	a	b
$\to q_0$	q_1	q_2
q_1	q_4	q_3
q_2	q_4	q_3
q_3	q_5	q_6
q_4	q_7	q_6
q_5	q_3	q_6
q_6	q_6	q_6
q_7	q_4	q_6

Regular Languages

Definition:

A language L is regular if there is FA M such that $L = L(M)$

Observation:

All languages accepted by FAs form the family of regular languages

Examples of regular languages:

$$\{abba\} \qquad \{\lambda, ab, abba\}$$

$$\{awa : w \in \{a,b\}^*\} \quad \{a^n b : n \geq 0\}$$

{ all strings with prefix } ab

{ all strings without substring } 001

There exist automata that accept these Languages (see previous slides).

There exist languages which are not Regular:

Example:

$$L = \{a^n b^n : n \geq 0\}$$

There is no FA that accepts such a language

(we will prove this later in the class)