CSC 339 – Theory of Computation
Spring 2022-2023

3. Deterministic Finite Automata

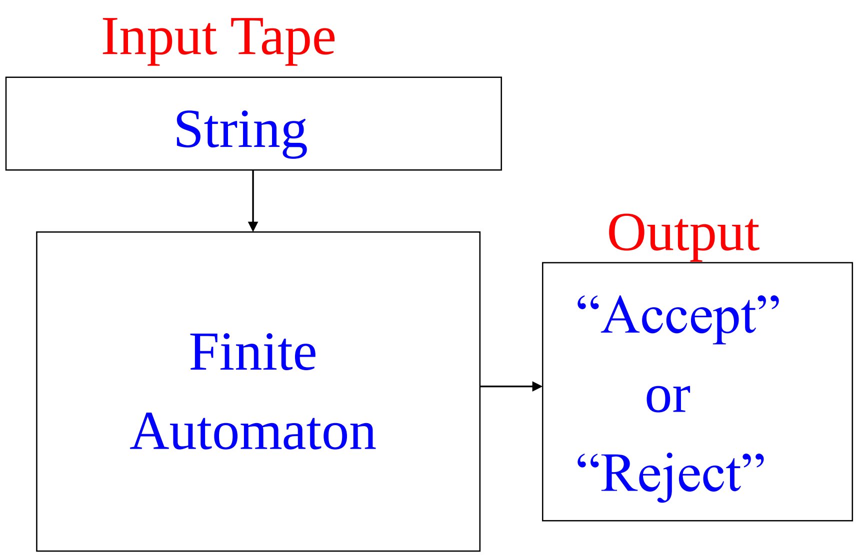
Outline

- Introduction
- Deterministic Finite Automata (DFA)
- Examples
- Languages accepted
- Formal definition
- Regular languages

Introduction

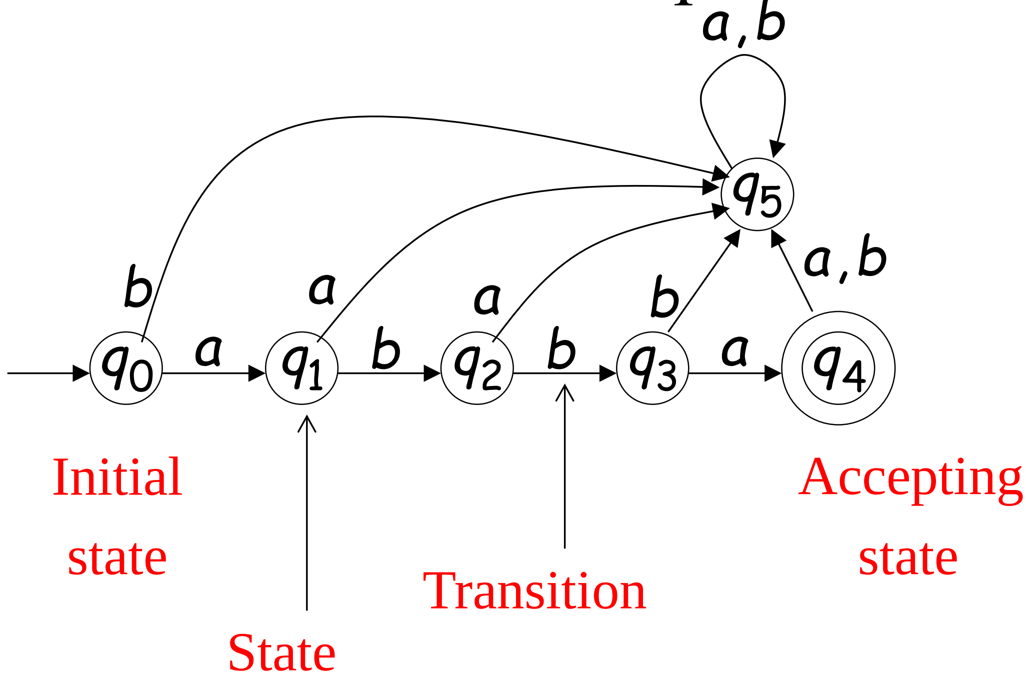
- An ***automaton*** (plural: ***automata***) is a mathematical model of a computing device.
- Why build models?
  - Mathematical simplicity: easier to manipulate abstract models of computers than actual computers.
  - Large classes of real computers are just special cases of more general models.
- Goal:
  - Figure out in which cases we can build automata for particular languages.
- A ***finite automaton*** is a simple type of mathematical machine for determining whether a string is contained within some language.

Deterministic Finite Automaton (DFA)



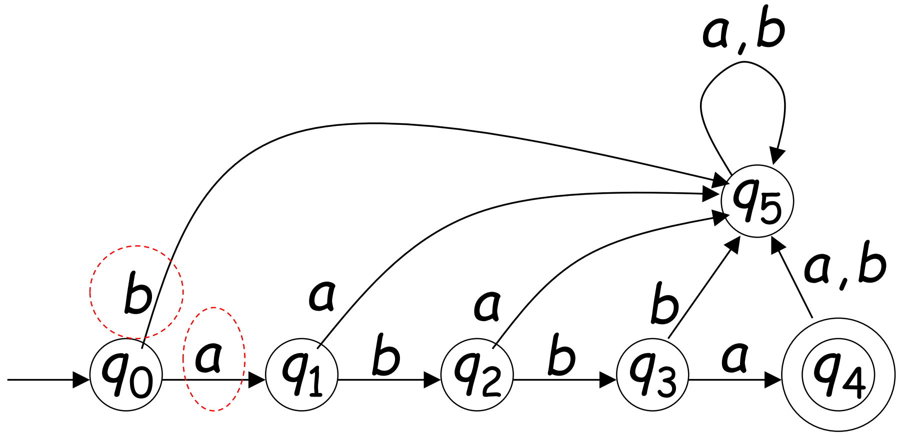
4

Transition Graph



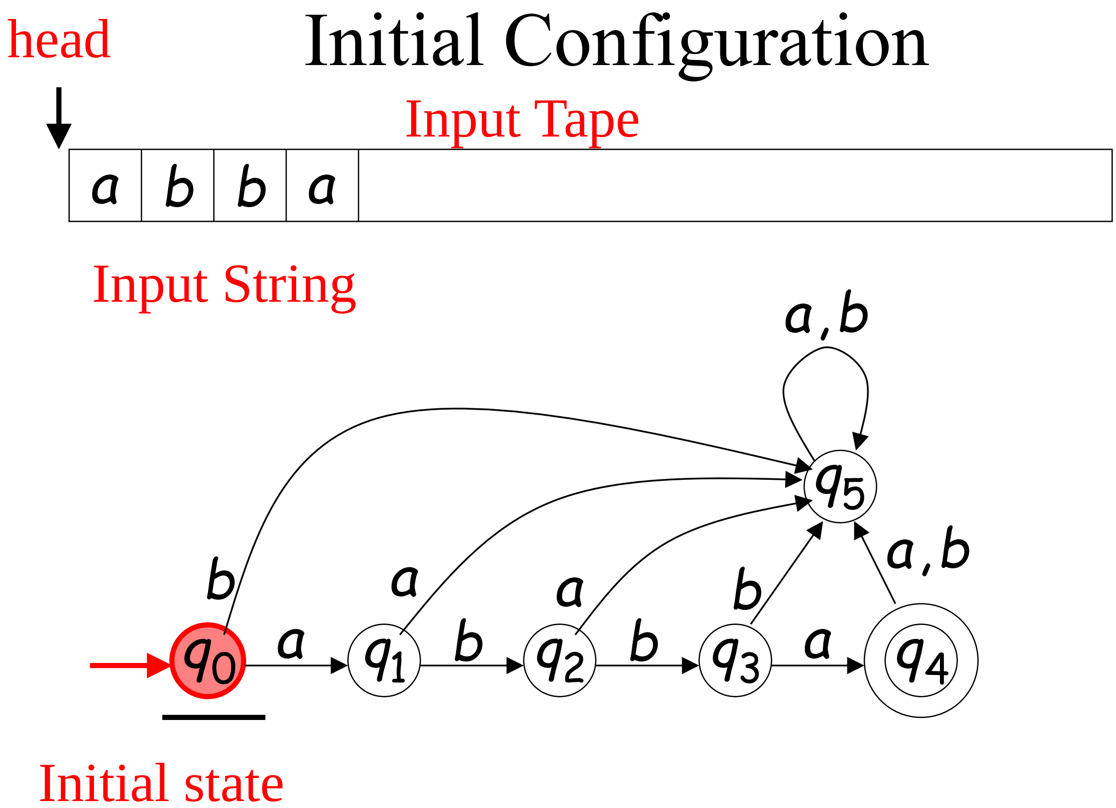
5

Alphabet $\Sigma = \{a, b\}$



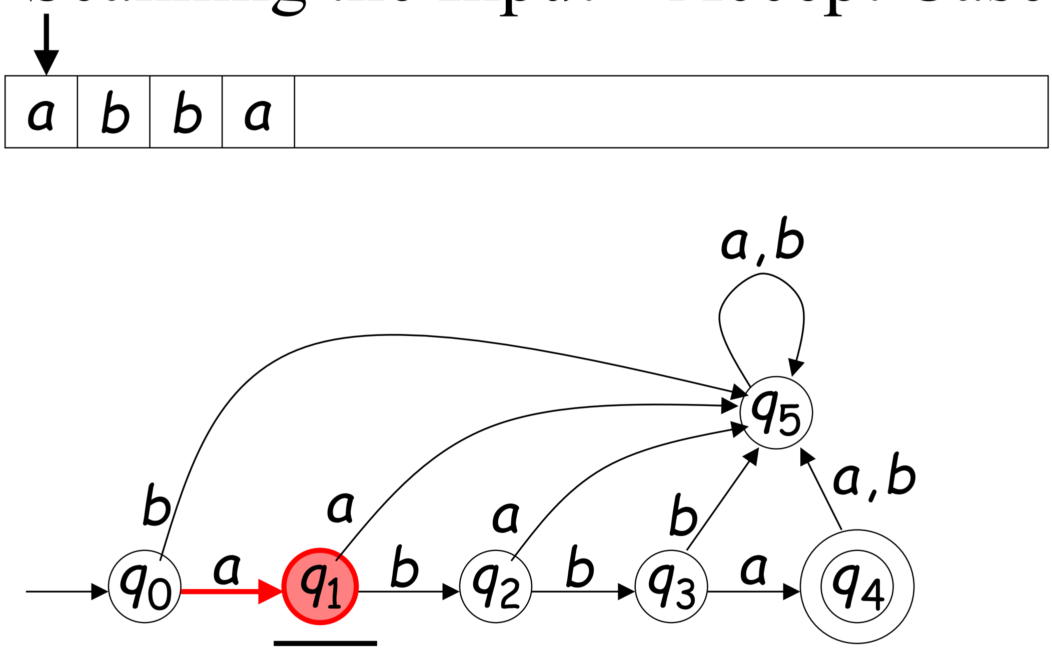
For every state, there is a transition for every symbol in the alphabet.

Initial Configuration

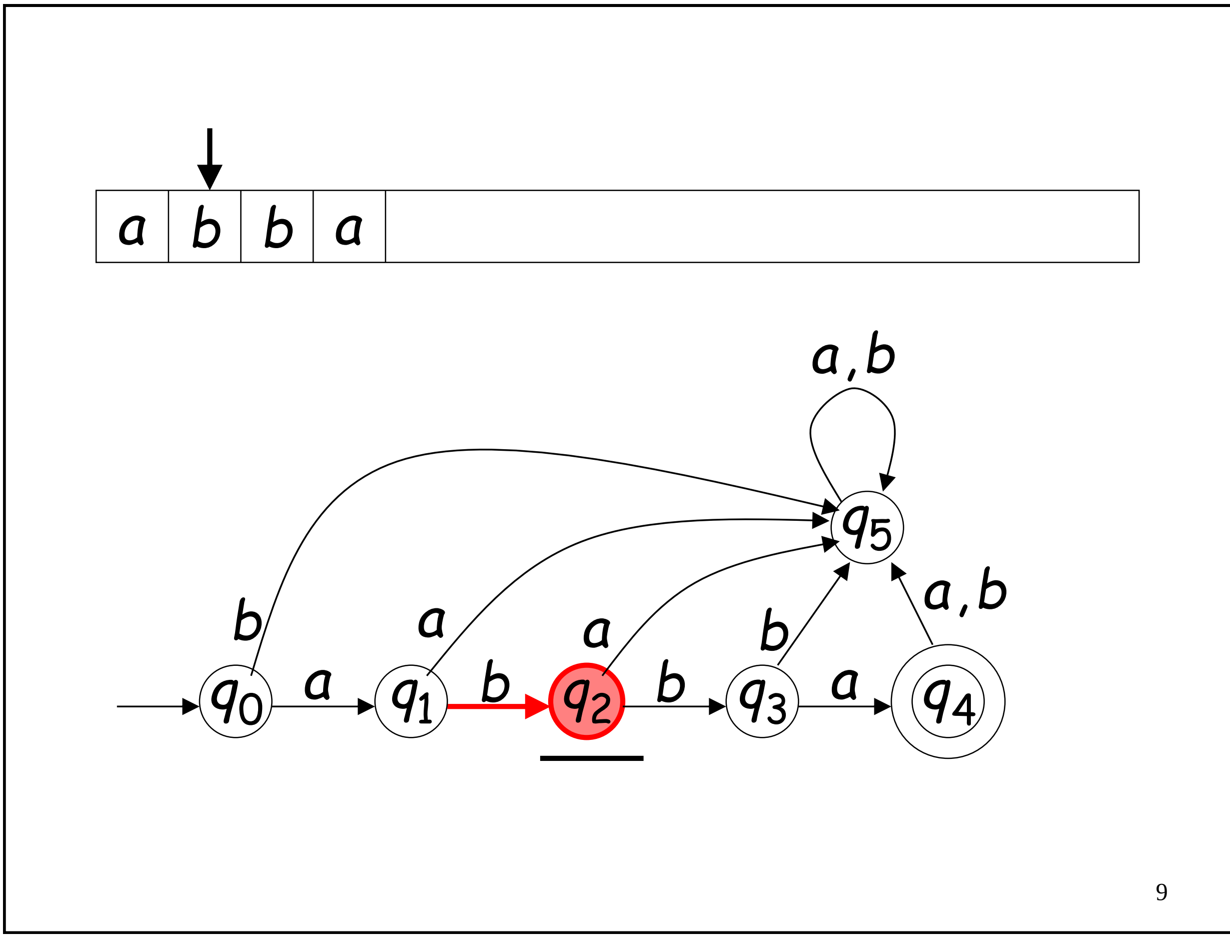


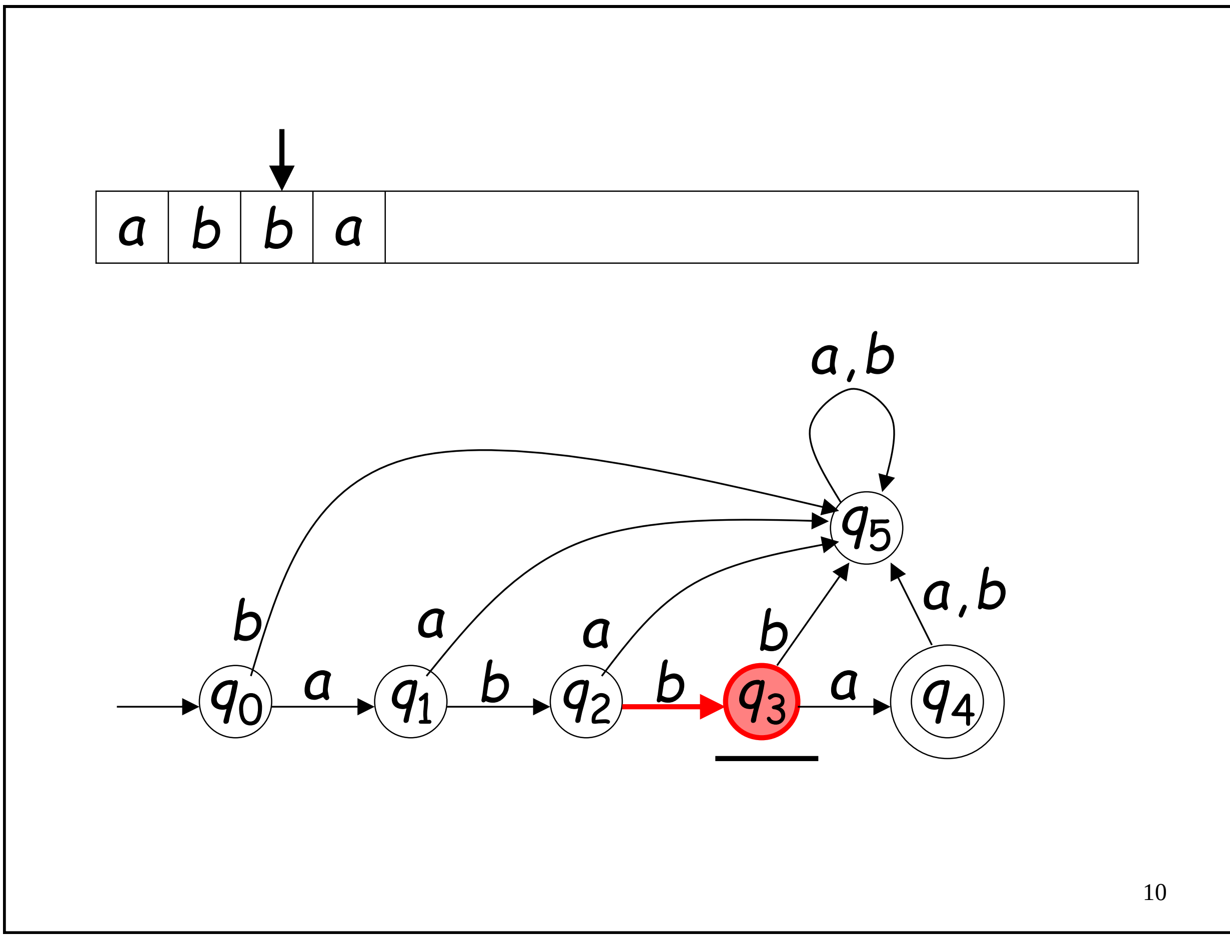
7

Scanning the Input – Accept Case

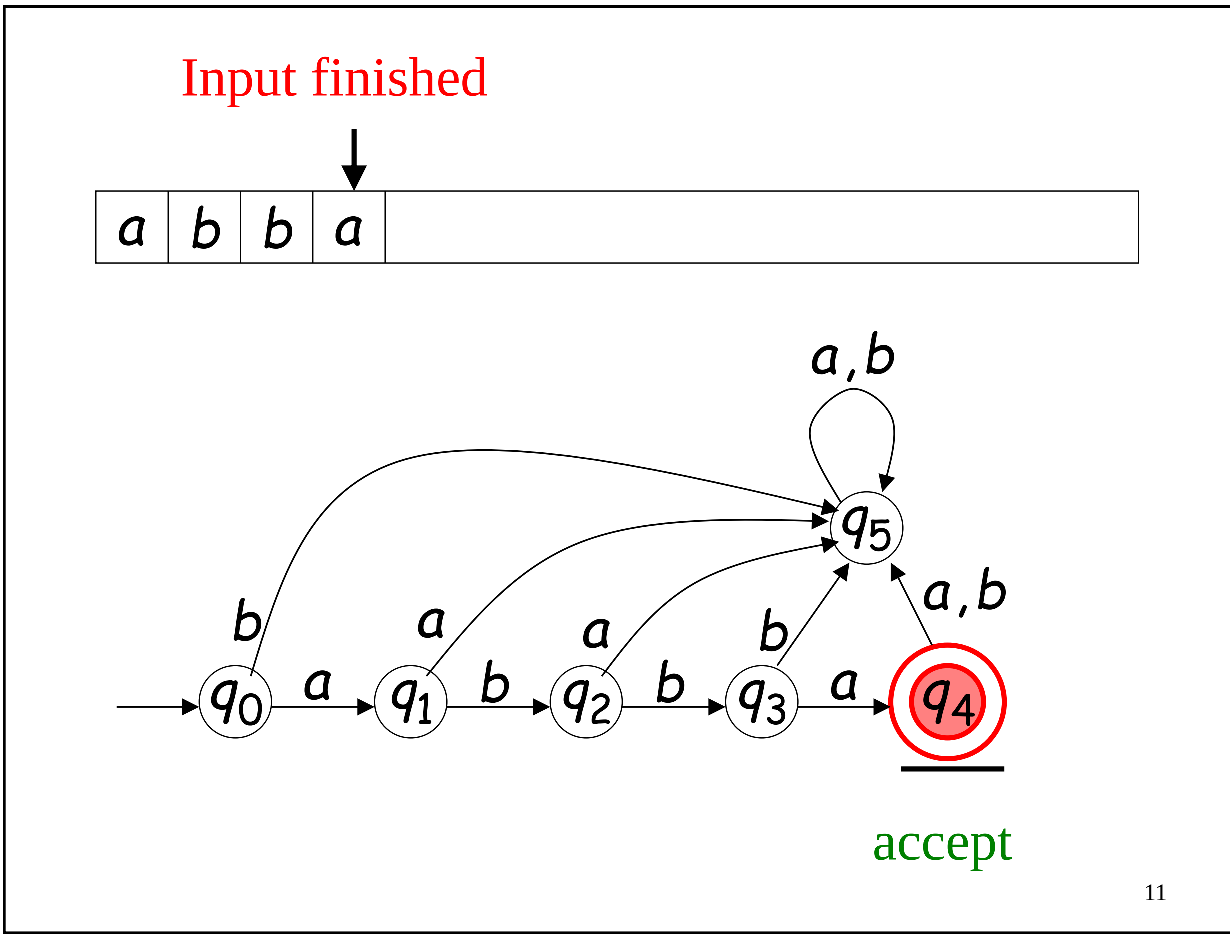


8


a
b
b
a
a,b
q_5
b
a
a
b
a,b
q_0
a
q_1
b
q_2
b
q_3
a
q_4
9



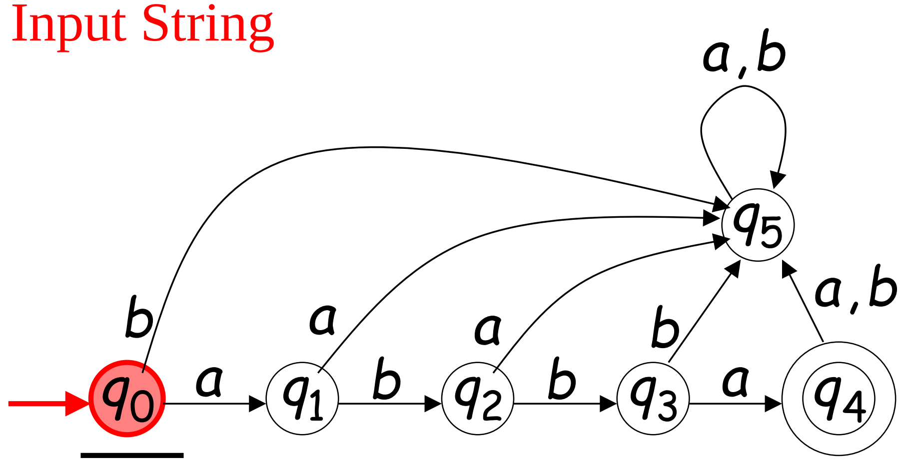
a
b
b
a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4
10


Input finished
a
b
b
a
a,b
q_5
a,b
b
a
a
b
q_0
a
q_1
b
q_2
b
q_3
a
q_4
accept
11

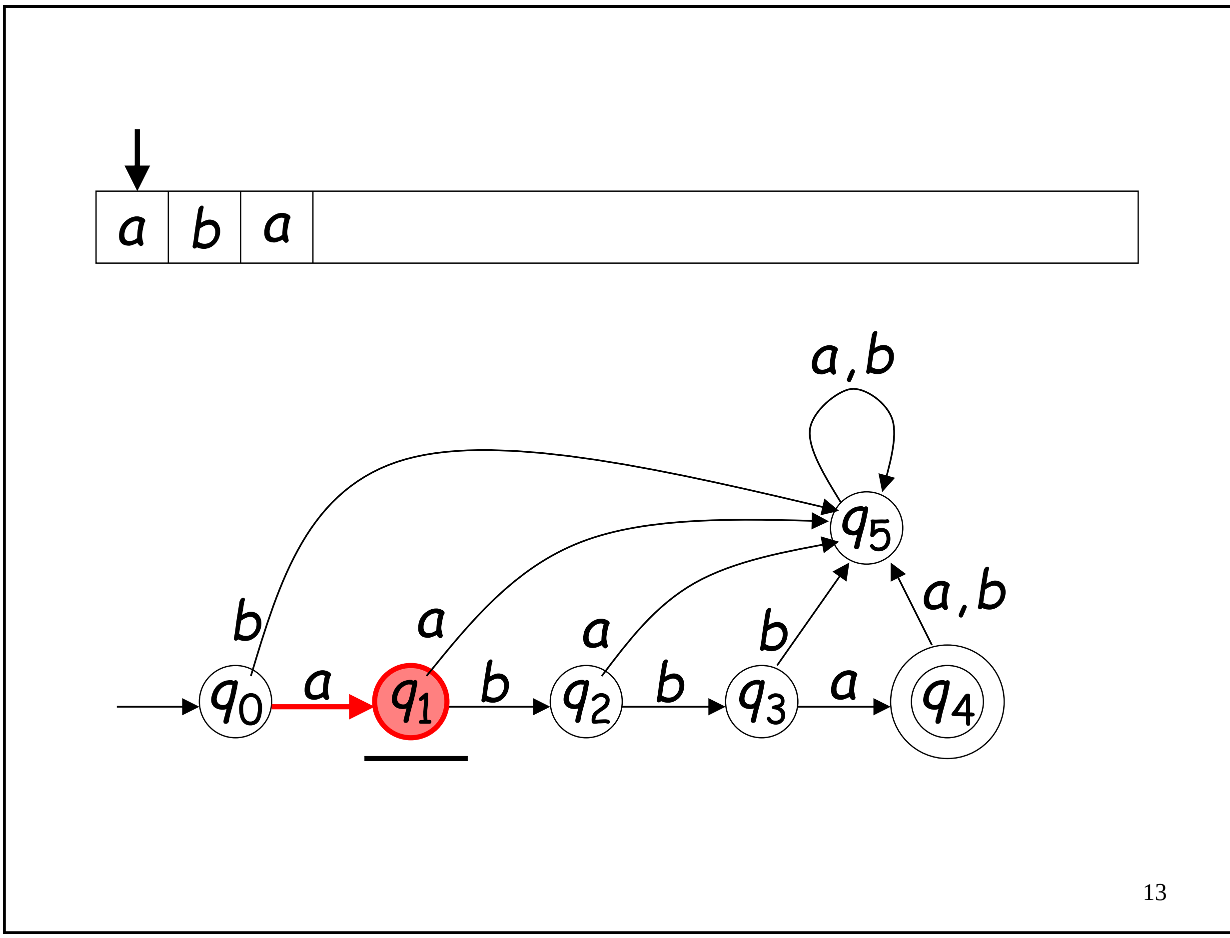
Scanning the Input – Reject Case

a	b	a	

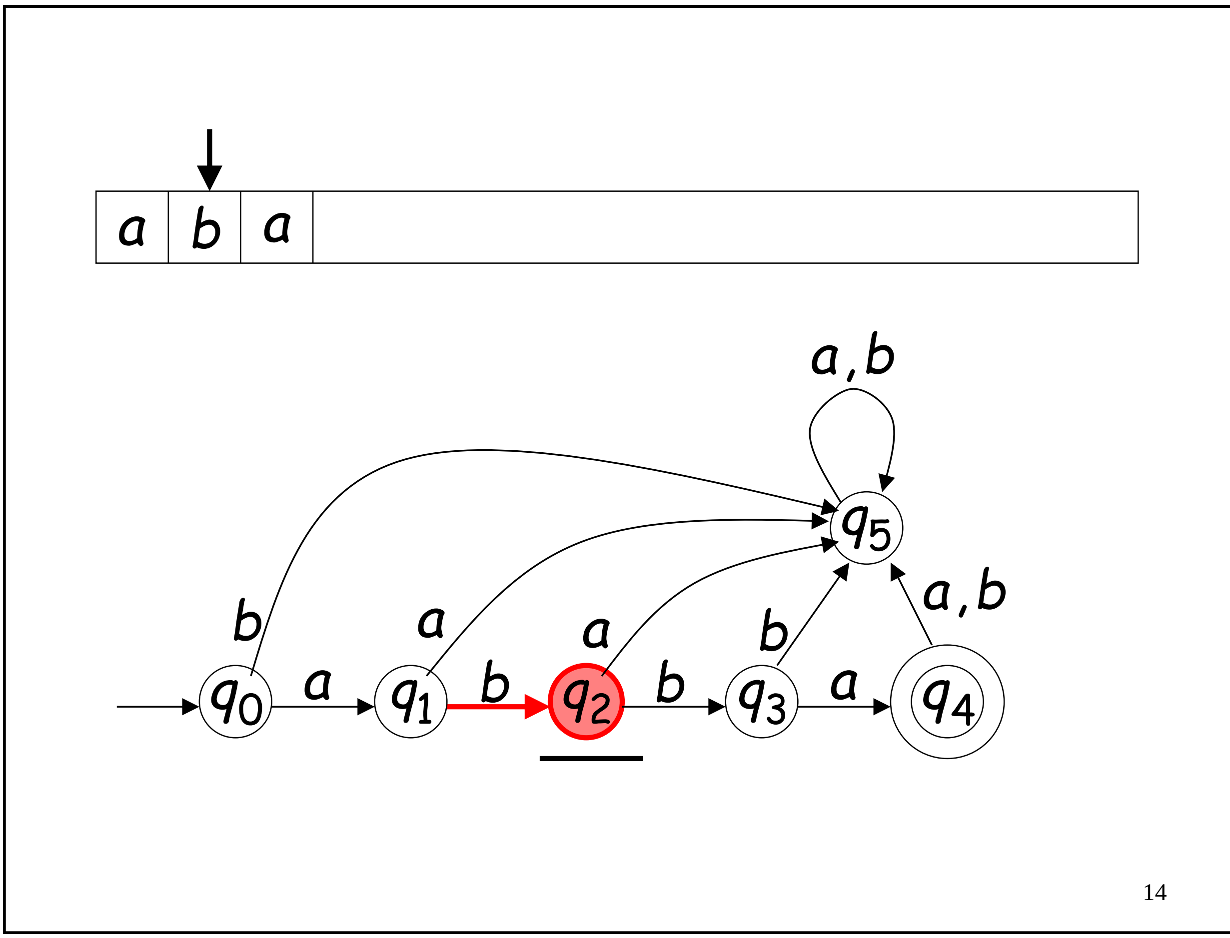
Input String



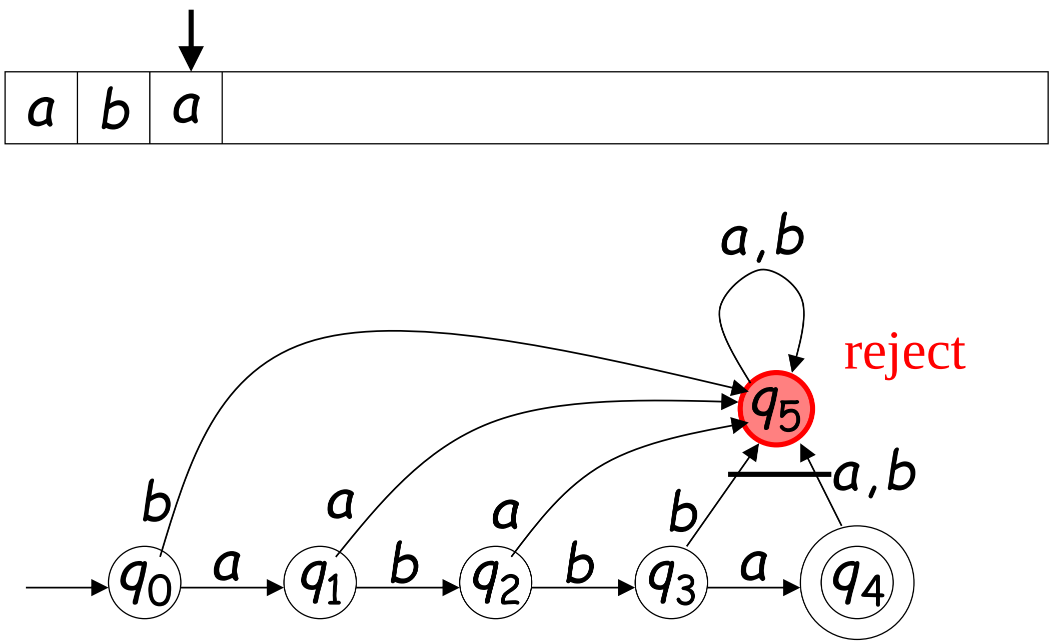
12



a
b
a
a,b
q5
a,b
b
a
a
b
q0
a
q1
b
q2
b
q3
a
q4
13


a
b
a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4
14

Input finished



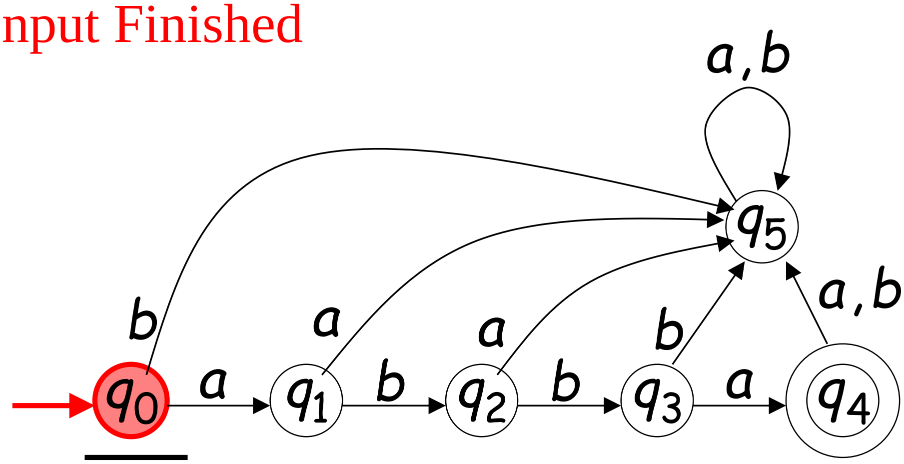
15

Scanning the Input – Another Reject Case

Tape is empty

(ε)

Input Finished

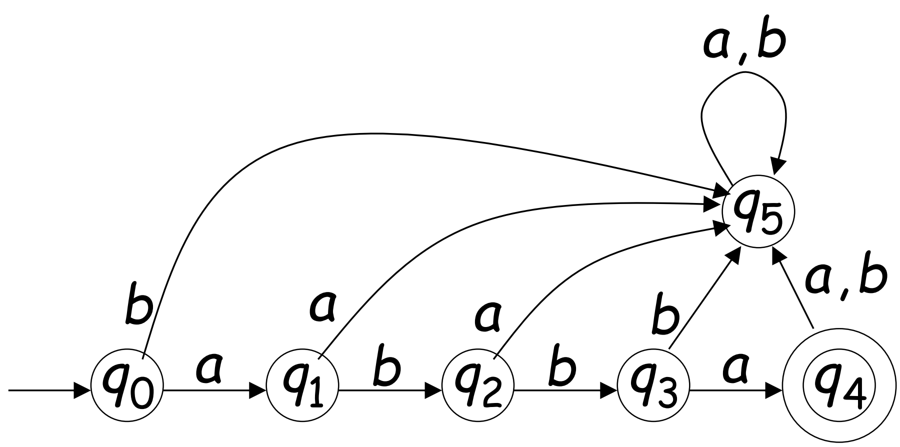


reject

16

Language Accepted

Language accepted: $L = \{abba\}$



17

Language Accepted

To accept a string:

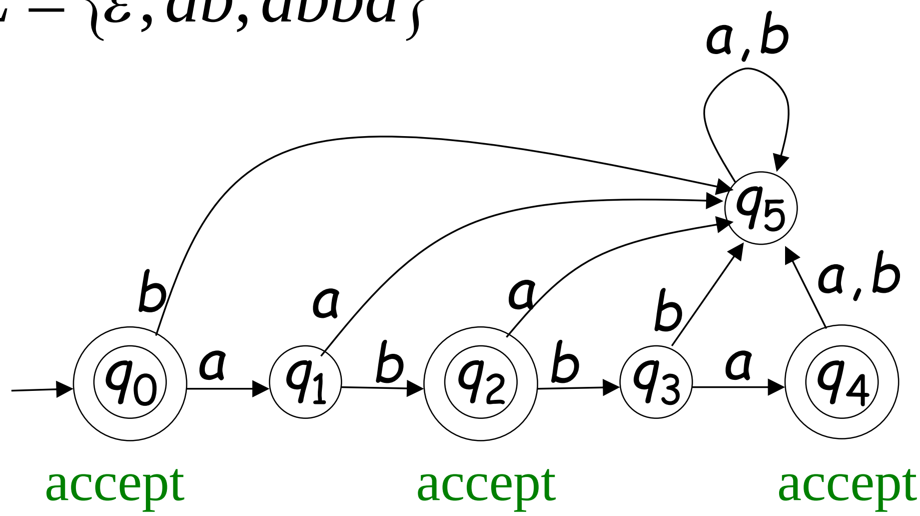
All the input string is scanned and the last state is accepting.

To reject a string:

All the input string is scanned and the last state is non-accepting.

Language Accepted

$$L = \{\varepsilon, ab, abba\}$$

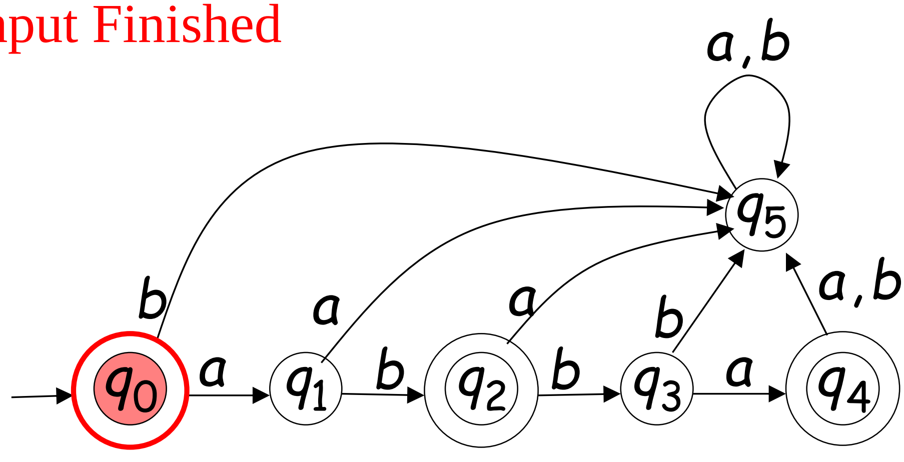


19

Empty Tape

(ε)

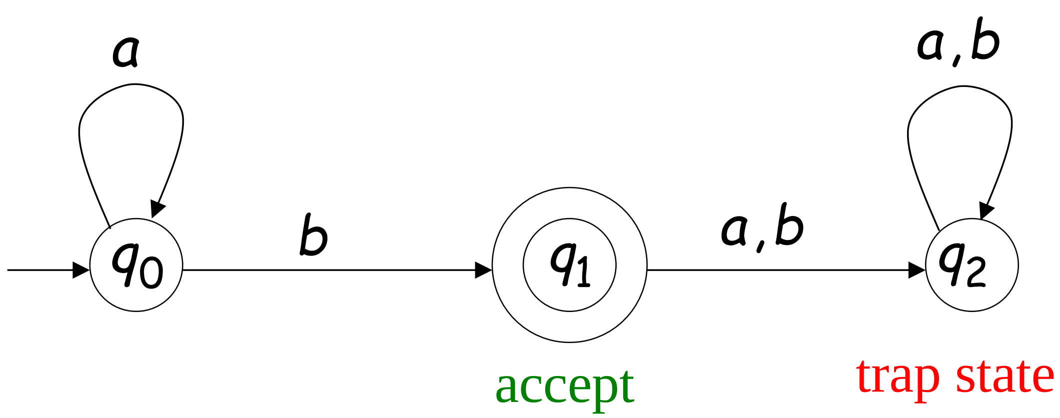
Input Finished



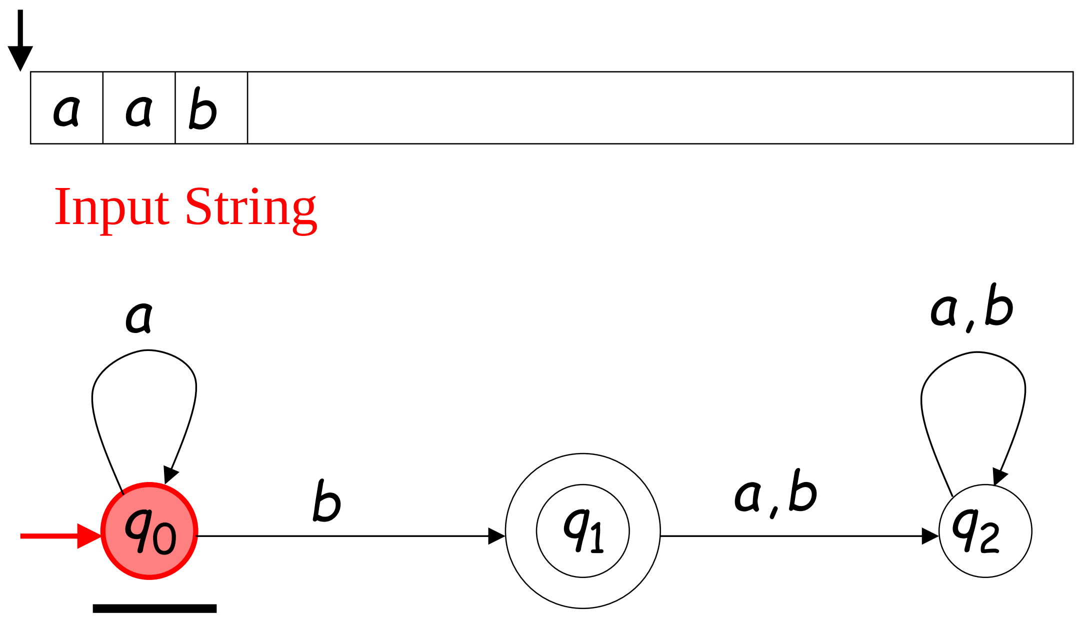
accept

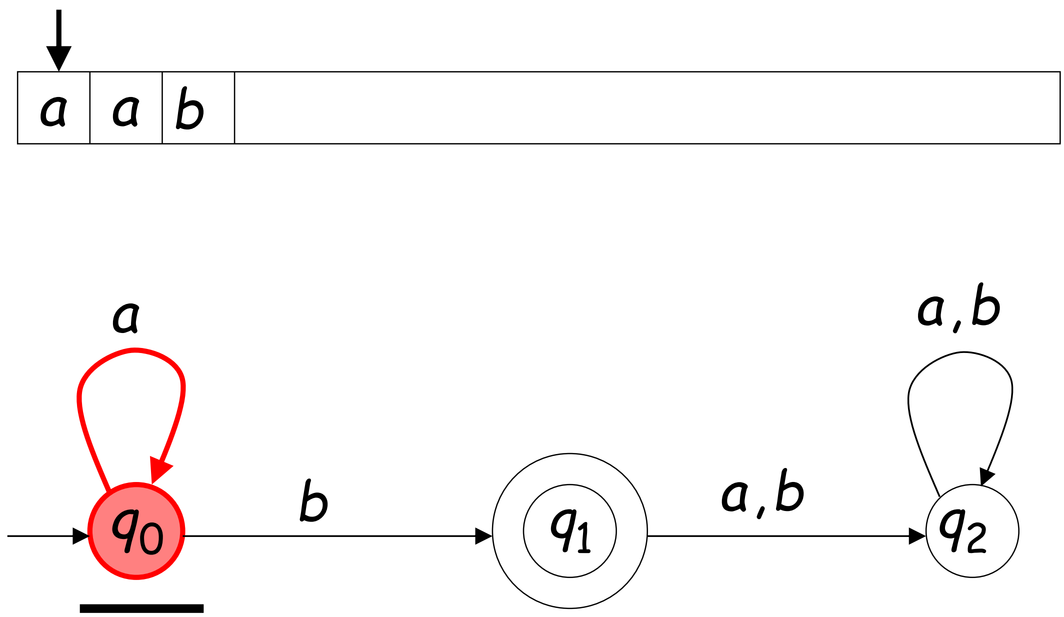
20

Other Examples

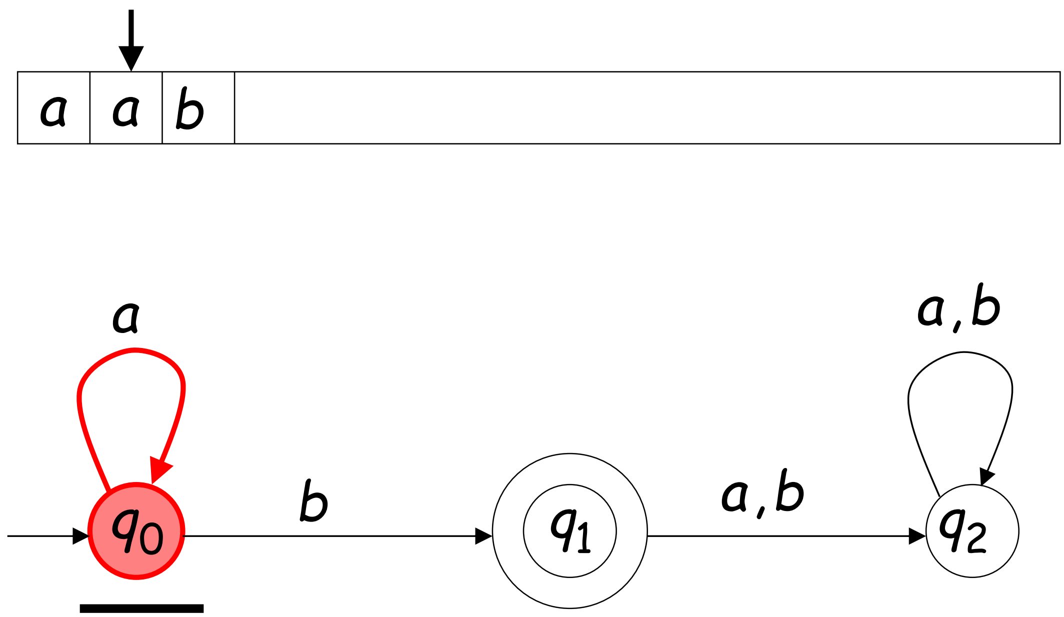


21

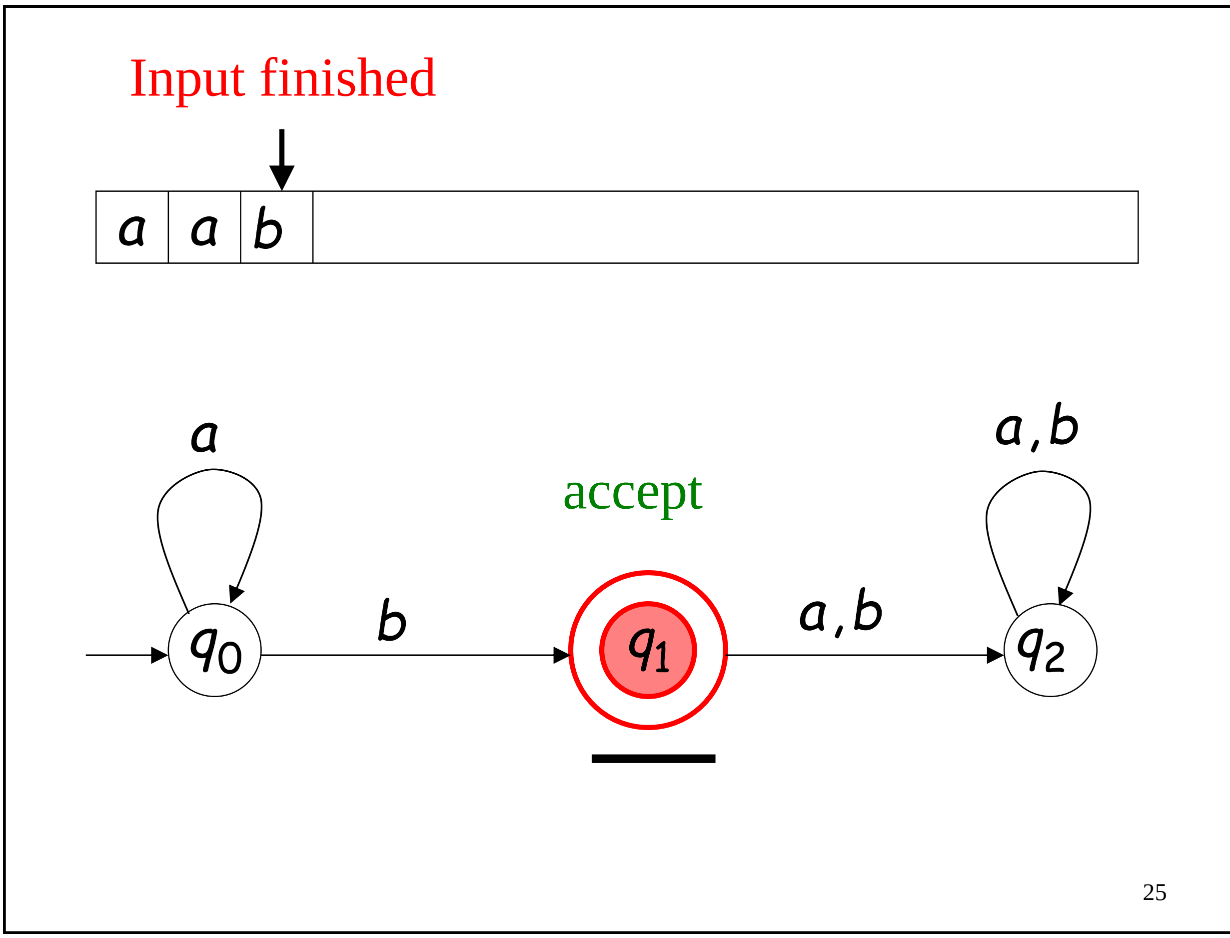

a
a
b
Input String
a
a,b
b
a,b
q_0
q_1
q_2

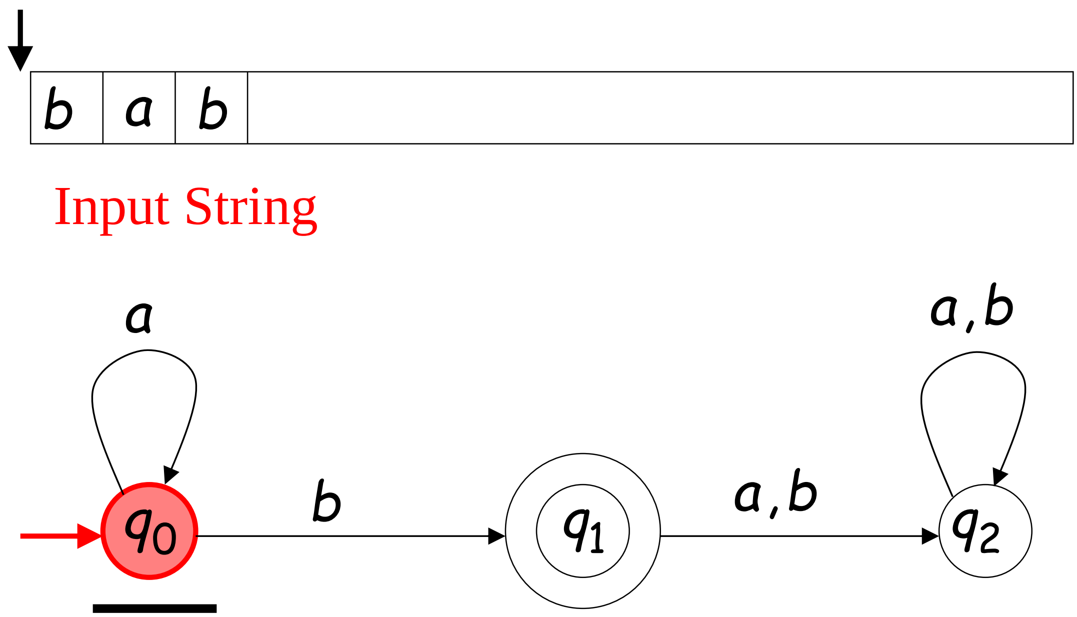


23

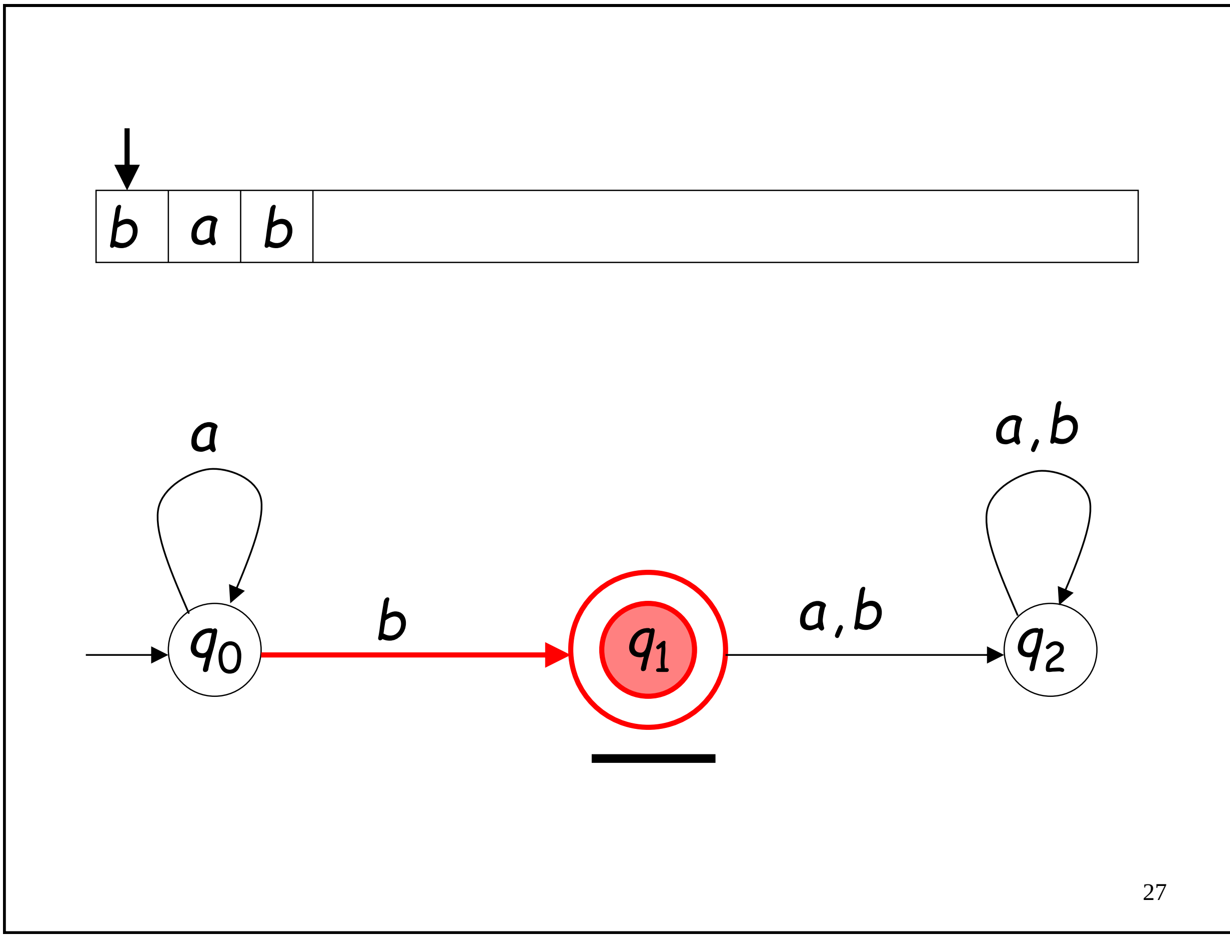


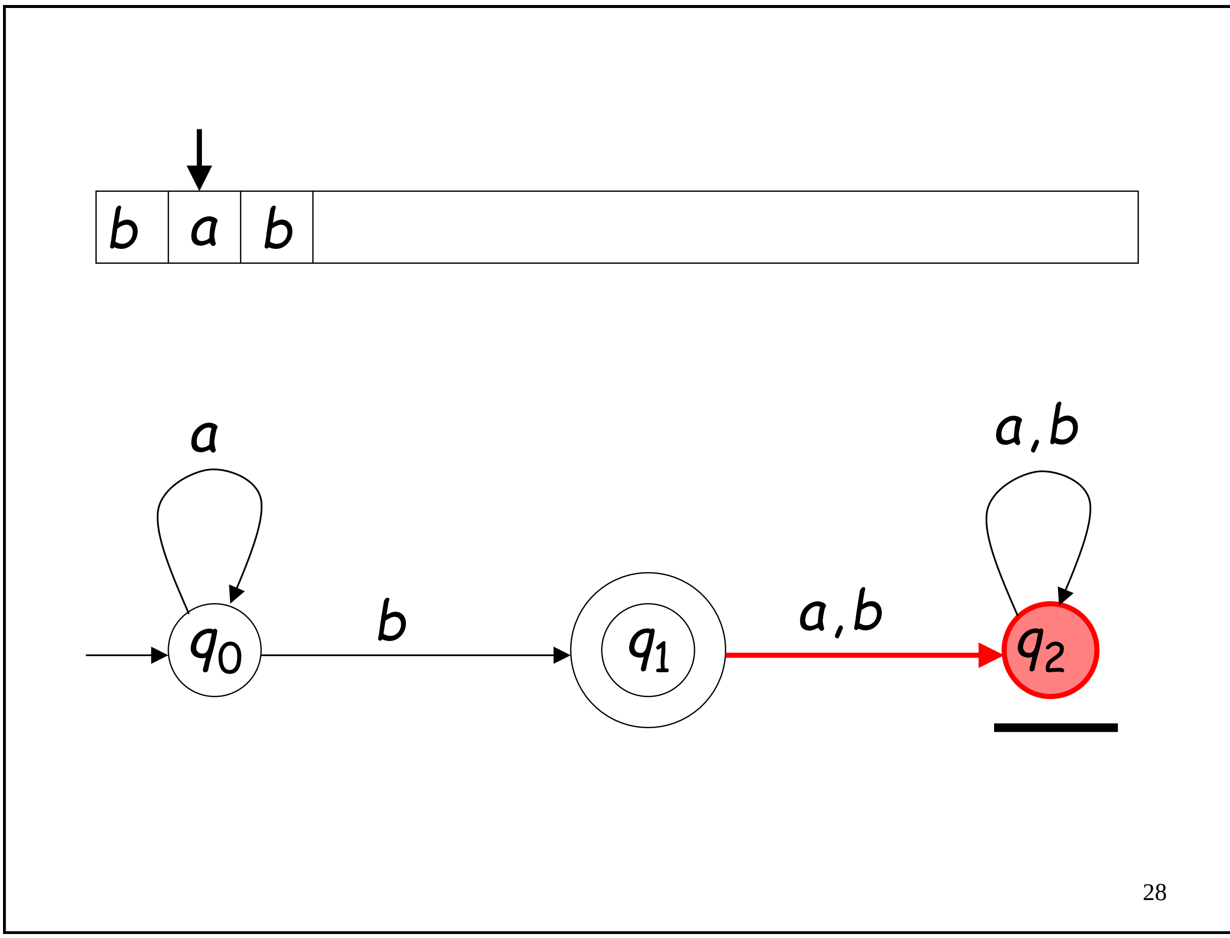
24

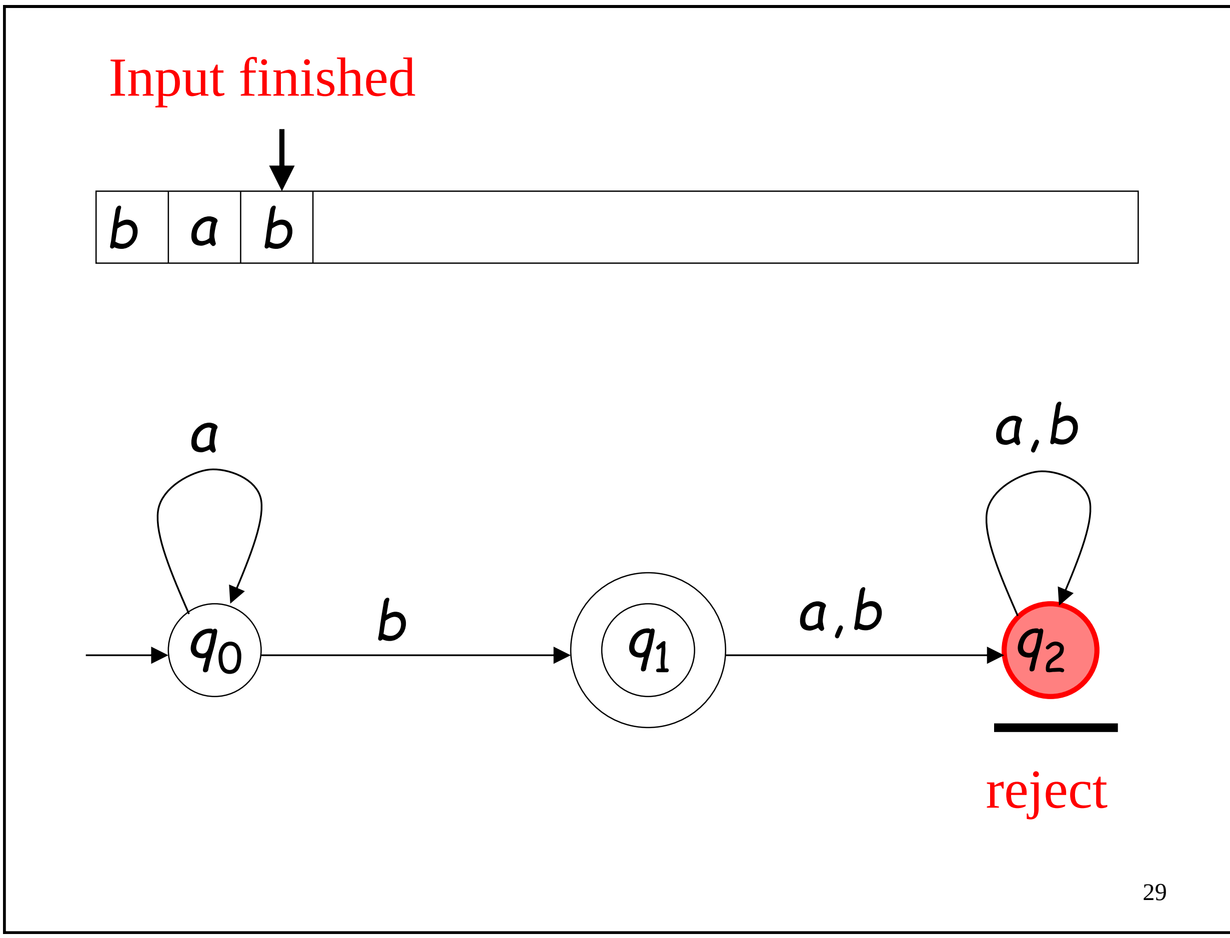

Input finished
a
a
b
a
accept
a, b
b
q_0
q_1
a, b
q_2
25



26

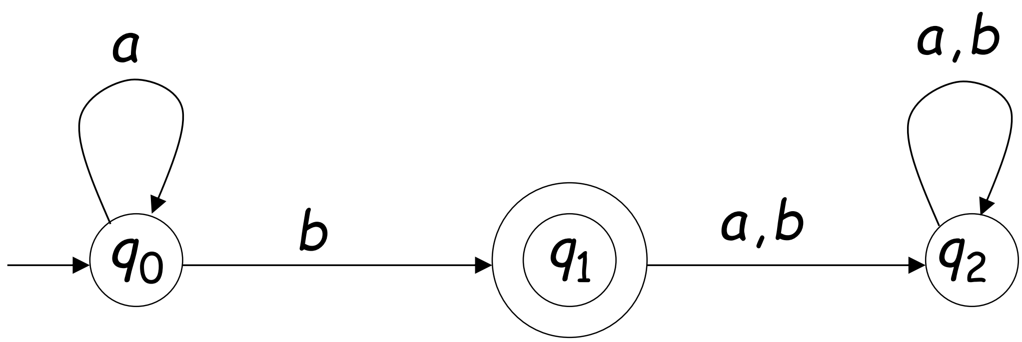

b
a
b
a
a,b
b
a,b
q_0
q_1
q_2
27


b
a
b
a
b
a,b
a,b
q_0
q_1
q_2
28



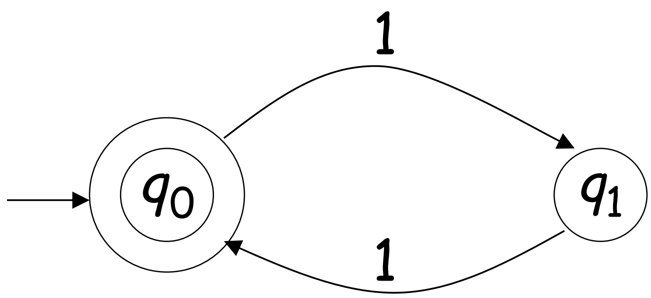
Input finished
b
a
b
a
a,b
b
a,b
q_0
q_1
q_2
reject
29

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



Other Examples

Alphabet: $\Sigma = \{1\}$



Language accepted:

$$EVEN = \{x : x \in \Sigma^* \text{ and } x \text{ is even}\}$$
$$= \{\varepsilon, 11, 1111, 111111, \ldots\}$$

31

Formal Definition

- Deterministic Finite Automaton (DFA)

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

Q : set of states

Σ : input alphabet

δ : transition function

q_0 : initial state

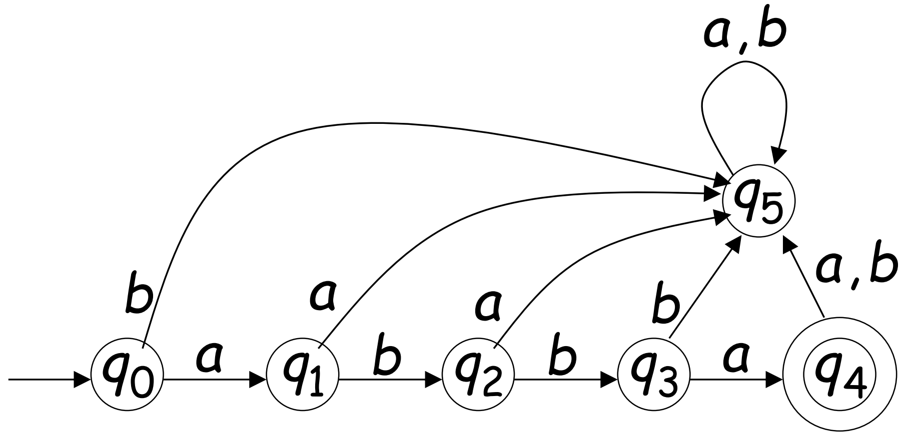
F : set of accepting states

32

Set of States Q

Example:

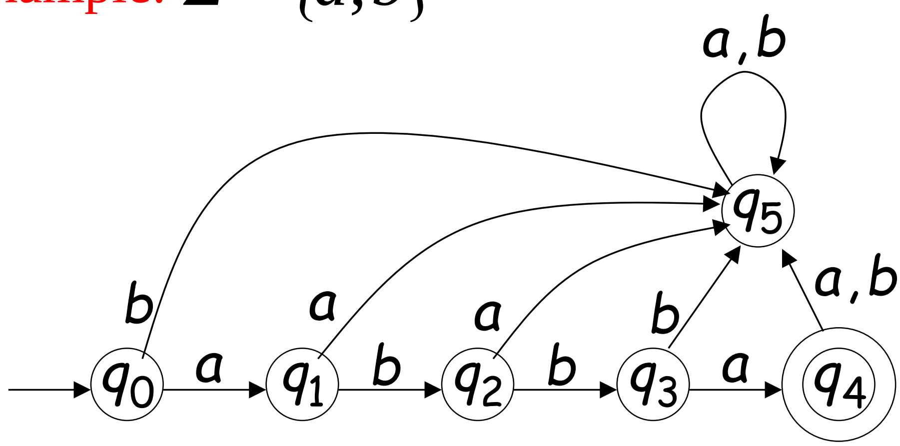
$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$$



Input Alphabet Σ

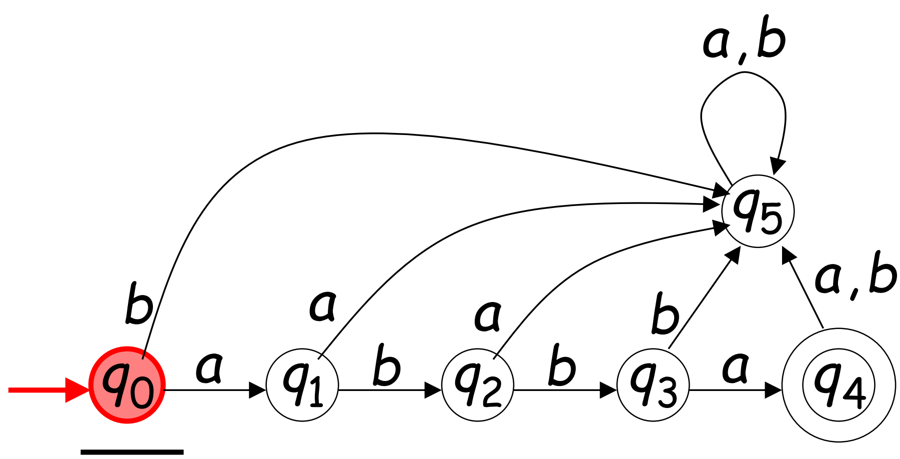
$\varepsilon \notin \Sigma$: the input alphabet does not contain ε

Example: $\Sigma = \{a, b\}$



Initial State q_0

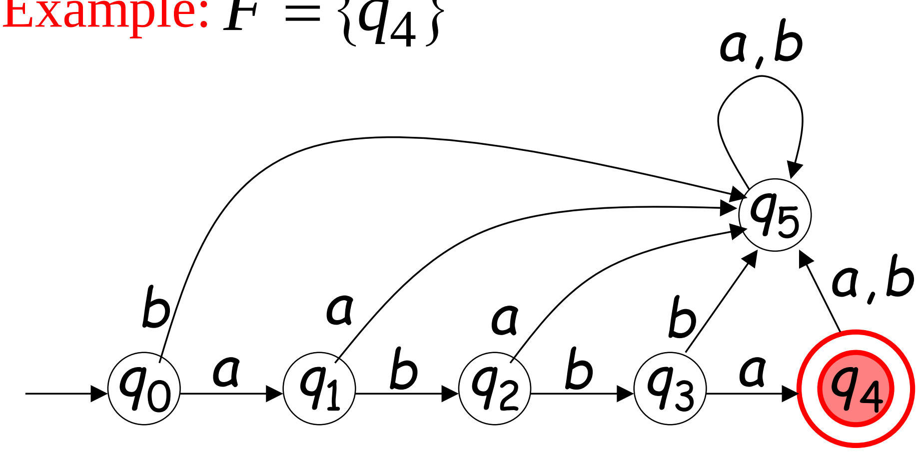
Example:



35

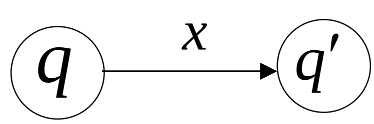
Set of Accepting States $F \subseteq Q$

Example: $F = \{q_4\}$



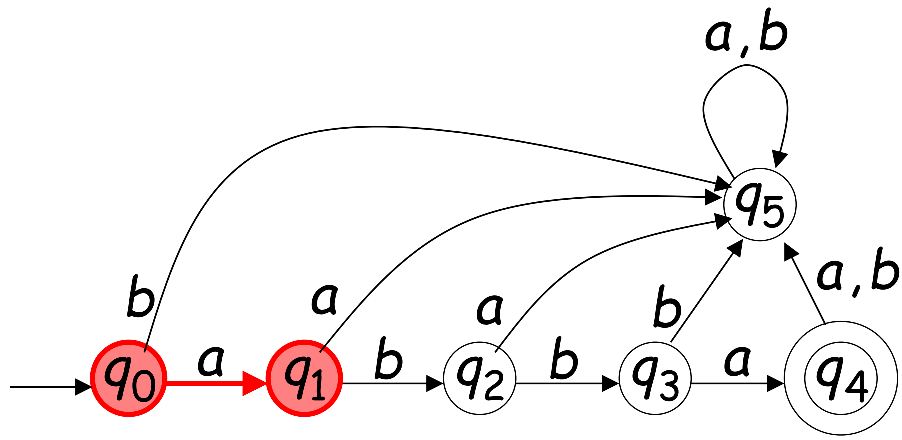
Transition Function $\delta : Q \times \Sigma \to Q$

$$\delta(q, x) = q'$$

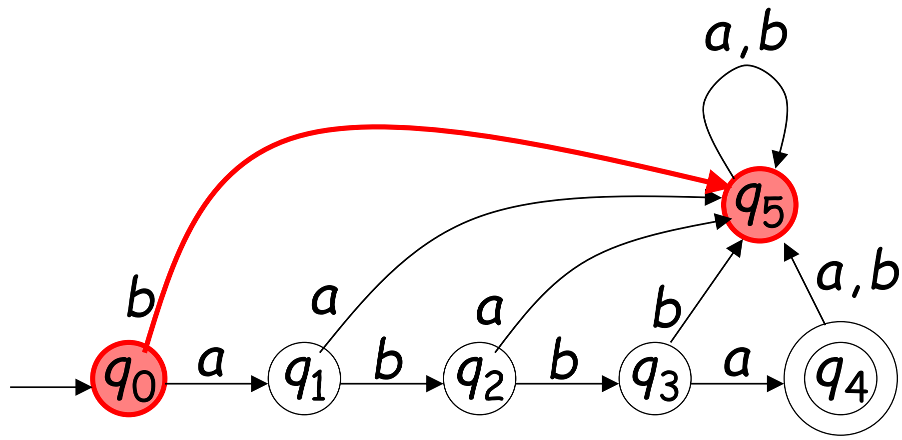


Describes the result of a transition from state q with symbol x

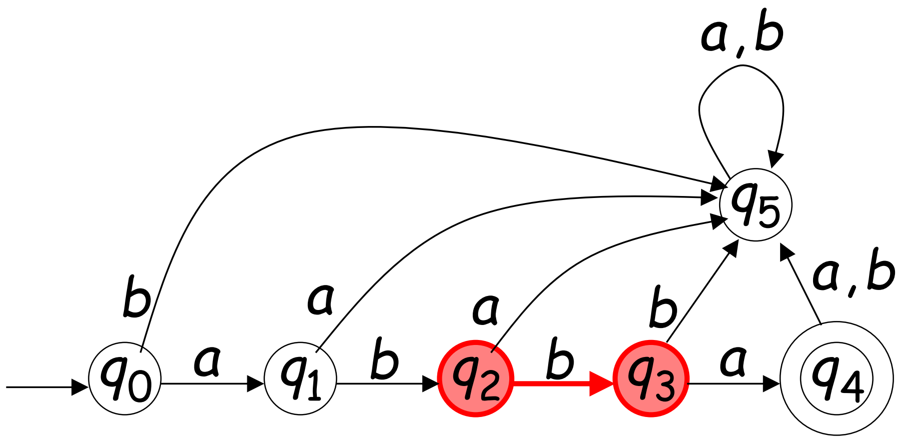
Example: $\delta(q_0, a) = q_1$



$$\delta(q_0, b) = q_5$$



$$\delta(q_2, b) = q_3$$



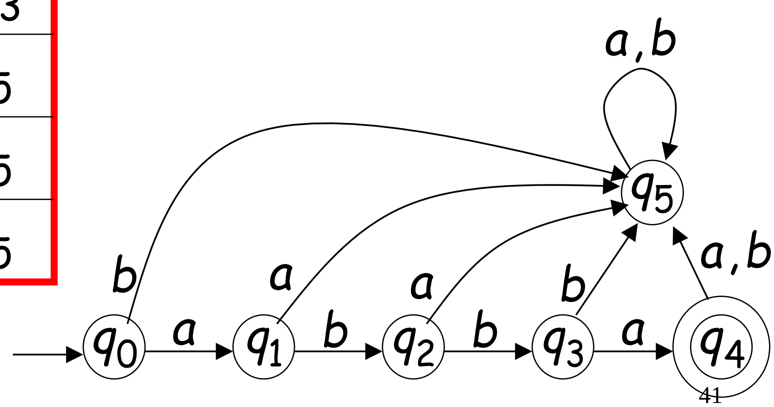
40

Transition Table for δ

symbols

states

δ	a	b
q_0	q_1	q_5
q_1	q_5	q_2
q_2	q_5	q_3
q_3	q_4	q_5
q_4	q_5	q_5
q_5	q_5	q_5



41

Extended Transition Function

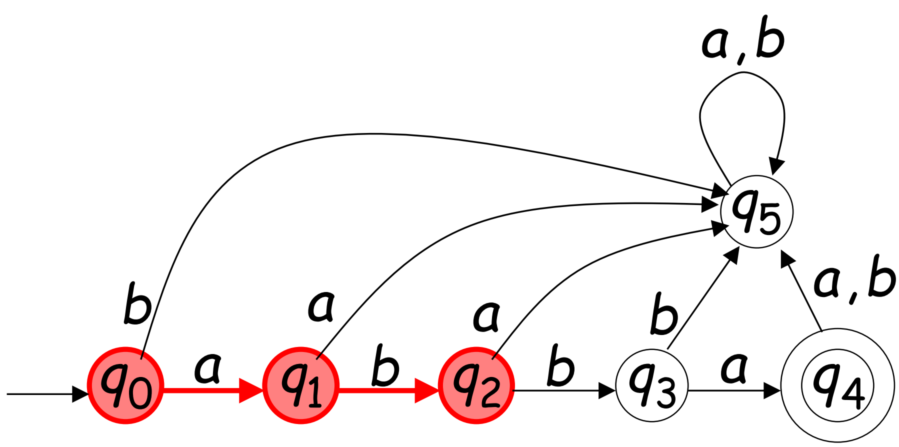
$$\delta^* : Q \times \Sigma^* \to Q$$

$$\delta^*(q, w) = q'$$

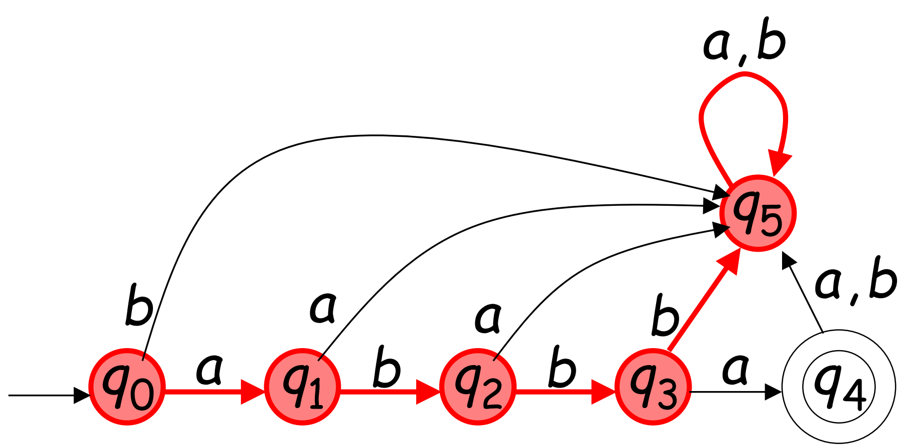
Describes the resulting state after scanning string w from state q

42

Example: $\delta^*(q_0, ab) = q_2$

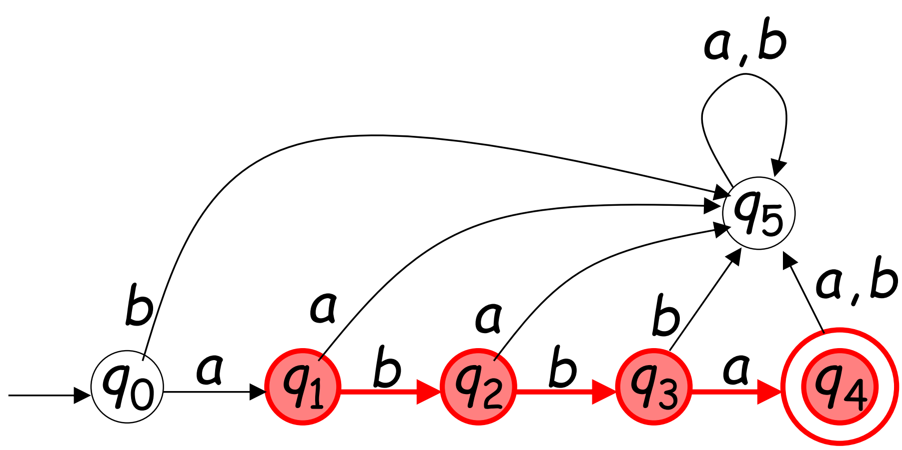


$$\delta^*(q_0, abbbaa) = q_5$$



44

$$\delta^*(q_1, bba) = q_4$$



45

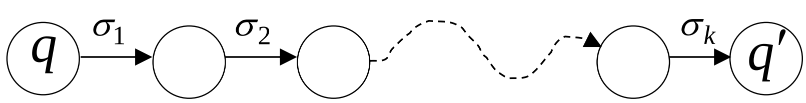
Special Case

For any state q: $\delta^{*}(q, \varepsilon) = q$

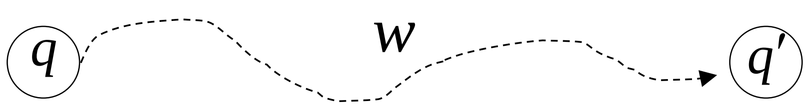
In general: $\delta^*(q, w) = q'$

implies that there is a walk of transitions

$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



states may be repeated



47

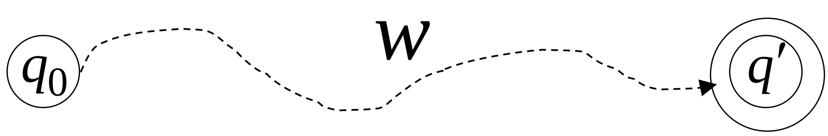
Language Accepted by DFA

Language of DFA M :

It is denoted as $L(M)$ and contains all the strings accepted by M

We say that a language L' is accepted (or recognized) by DFA M if $L(M) = L'$

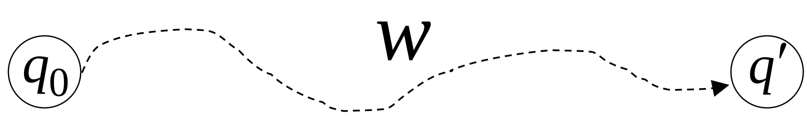
- For a DFA $M = (Q, \Sigma, \delta, q_0, F)$
- Language accepted by M :
- $L(M) = \{ w \in \Sigma^* : \delta^*(q_0, w) \in F \}$



$q' \in F$

Language rejected by M :

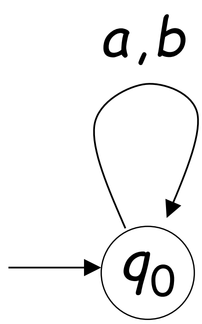
$$\overline{L(M)} = \left\{ w \in \Sigma^* : \delta^*(q_0, w) \notin F \right\}$$

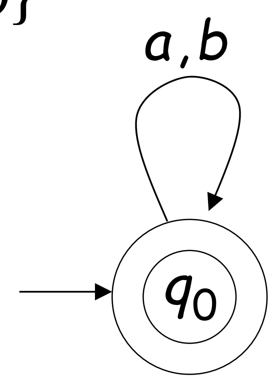


$q' \notin F$

Other DFA Examples

$$\Sigma = \{a, b\}$$





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

Empty language

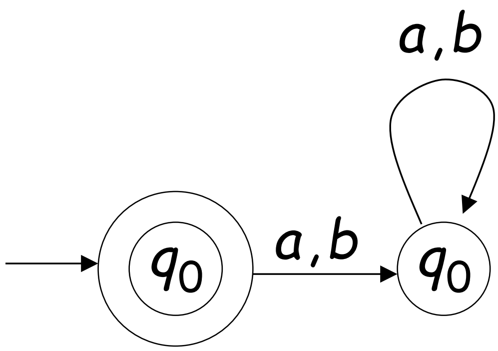
$$L(M) = \Sigma^*$$

All strings

51

Other DFA Examples

$$\Sigma = \{a, b\}$$



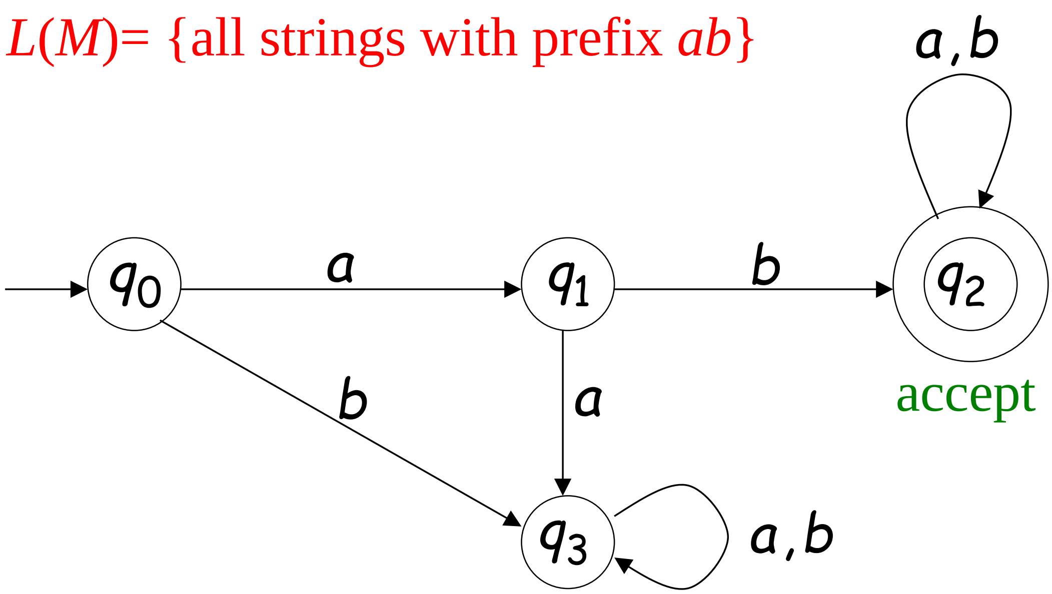
$$L(M) = \{\varepsilon\}$$

Language of the empty string

Other DFA Examples

$\Sigma = \{a, b\}$

$L(M)$= {all strings with prefix *ab*}

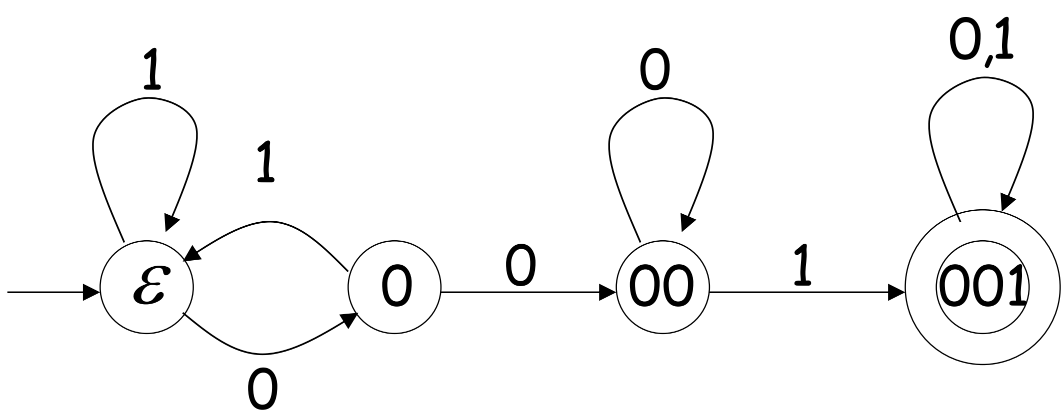


53

Other DFA Examples

$\Sigma = \{0,1\}$

L(*M*)= {all binary strings containing substring 001}

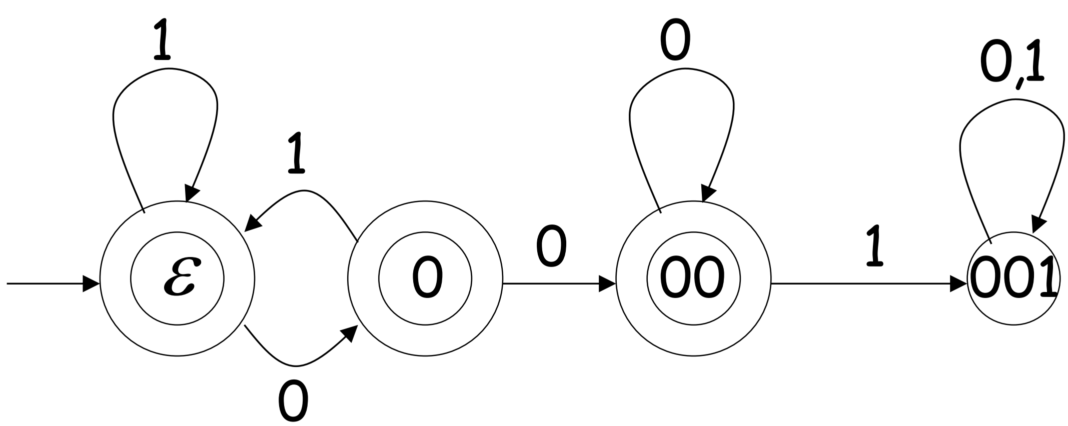


54

Other DFA Examples

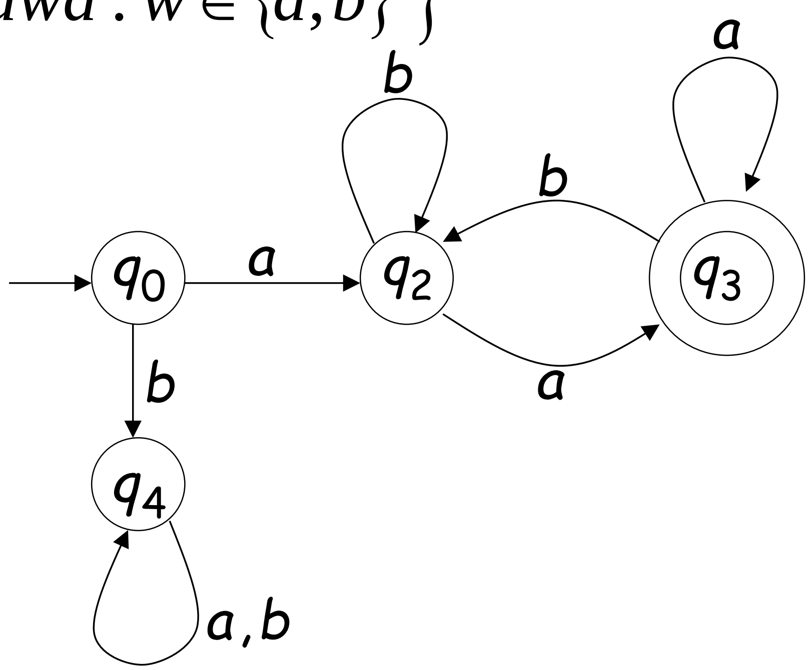
$\Sigma = \{0,1\}$

L(*M*) = {all binary strings without substring 001}



Other DFA Examples

$$L(M) = \left\{ awa : w \in \{a,b\}^* \right\}$$



56

Regular Languages

Definition:

- A language L is regular if there is a DFA M that accepts it ($L(M)=L$)
- The languages accepted by all DFAs form the family of regular languages

Examples of Regular Languages

$\{abba\}$ $\{\varepsilon, ab, abba\}$

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

{all strings in $\{a,b\}^*$ with prefix ab}

{all binary strings without substring 001}

$\{x : x \in \{1\}^*$ and x is even$\}$

$\{\ \}$ $\{\varepsilon\}$ $\{a,b\}^*$

There exist automata that accept these languages

Examples of Non-Regular Languages

There exist languages that are not Regular:

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

$$ADDITION = \{x + y = z \ : x = 1^n, y = 1^m, z = 1^k, n + m = k\}$$

There is no DFA that accepts these languages

(proof: later class)

59

Readings

- Textbook:
  - Part 1, Section 1.1 (Finite Automata)