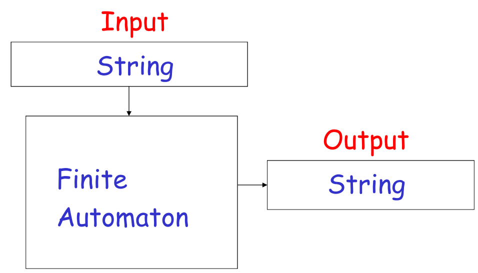
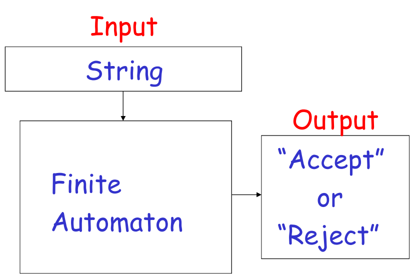
Finite Automata

class 2

Finite Automaton

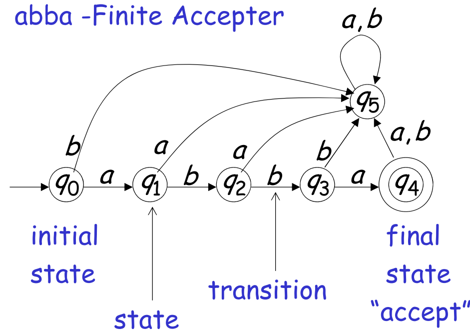


Finite Accepter



Transition Graph

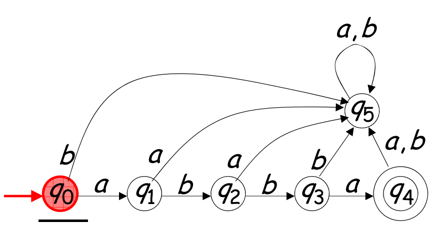
abba -Finite Accepter



Initial Configuration

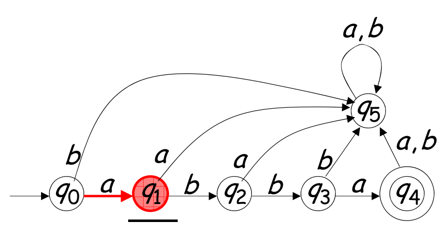
Input String

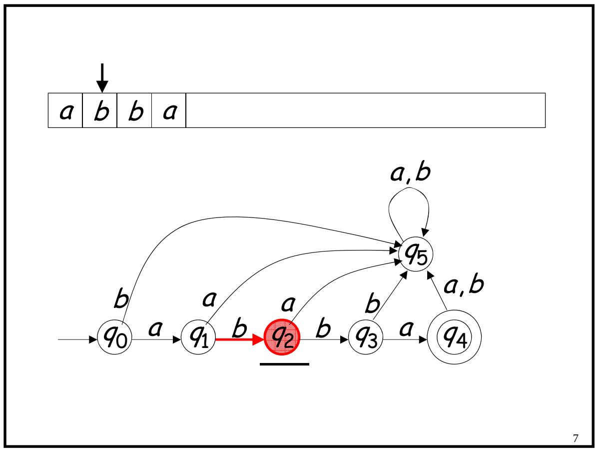
a	b	b	a	

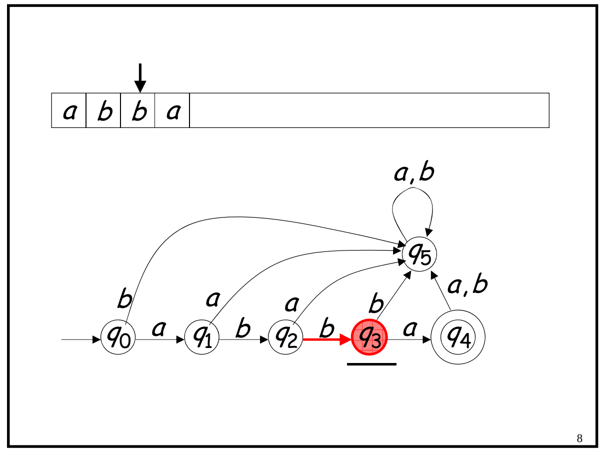


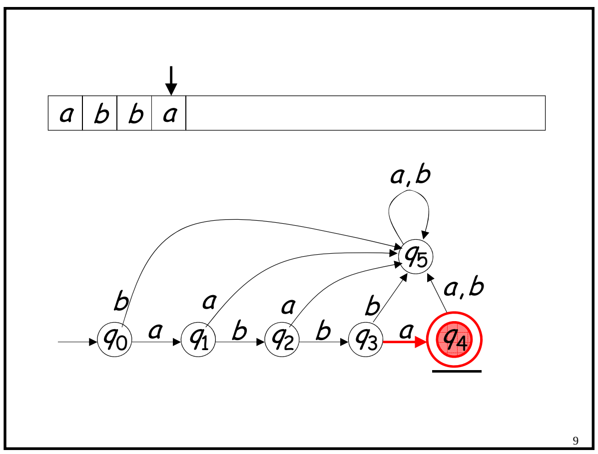
Reading the Input

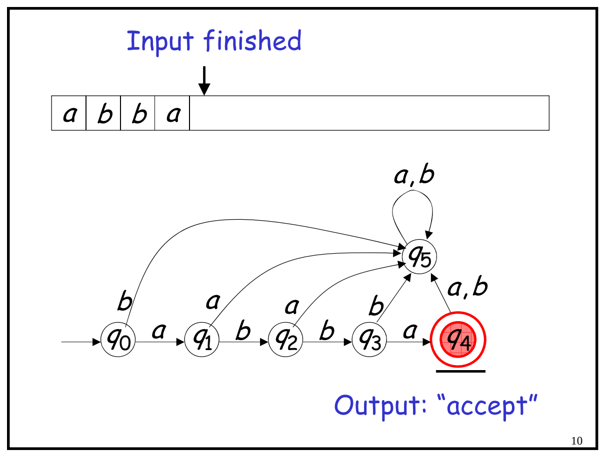
a	b	b	a	

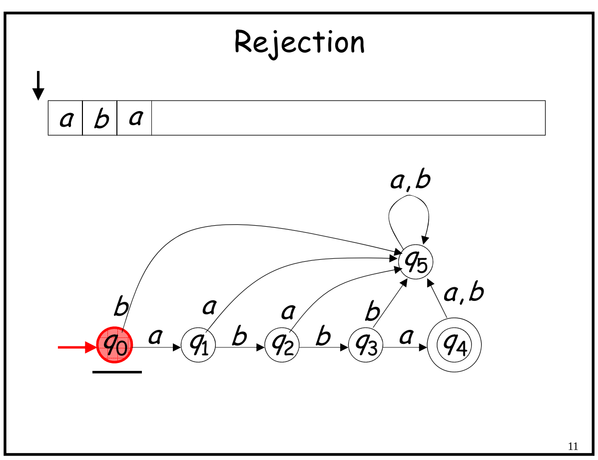


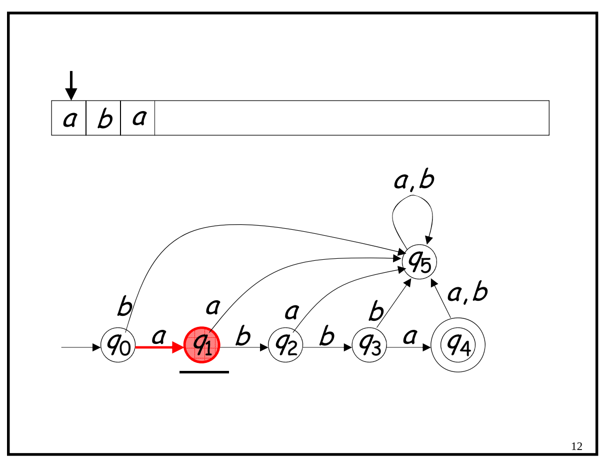

a b b a
a,b
q5
b a a b a,b
q0 a q1 b q2 b q3 a q4
7

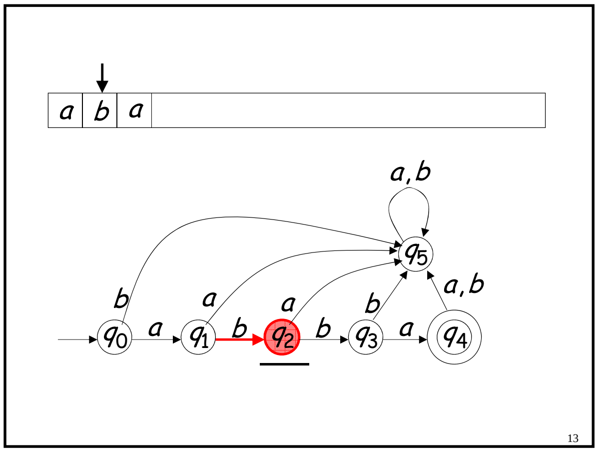

a b b a
a,b
q5
b a a b a,b
q0 a q1 b q2 b q3 a q4
8

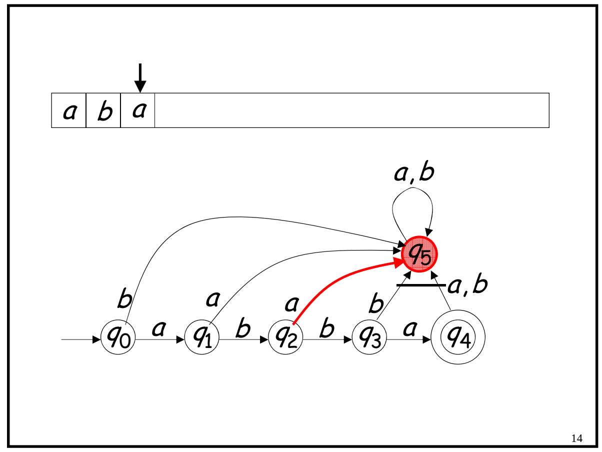

a b b a
a,b
q5
b a a b a,b
q0 a q1 b q2 b q3 a q4
9

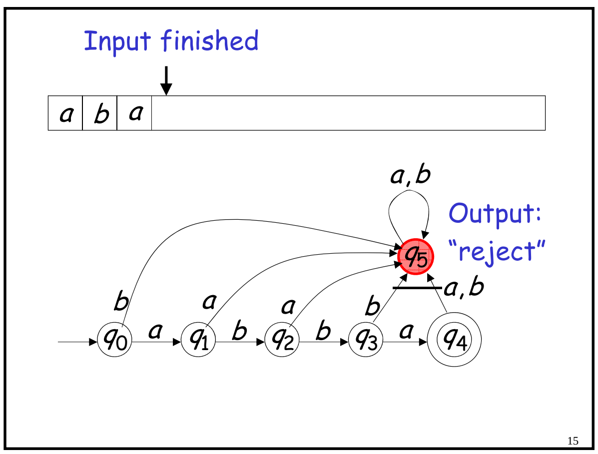

Input finished
a b b a
a,b
q5
b a a b a,b
q0 a q1 b q2 b q3 a q4
Output: "accept"
10

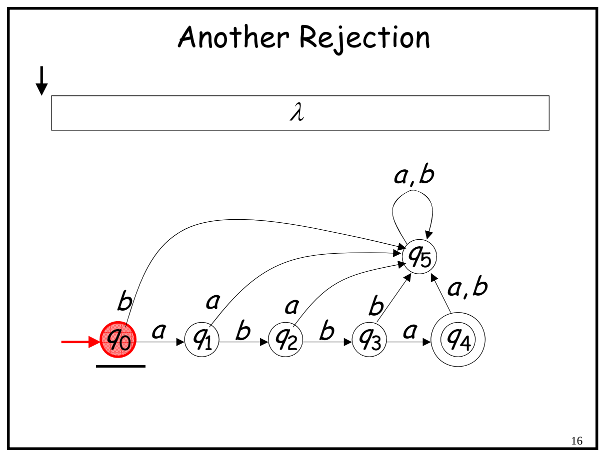

Rejection
a b a
a,b
q5
b a a b a,b
q0 a q1 b q2 b q3 a q4
11

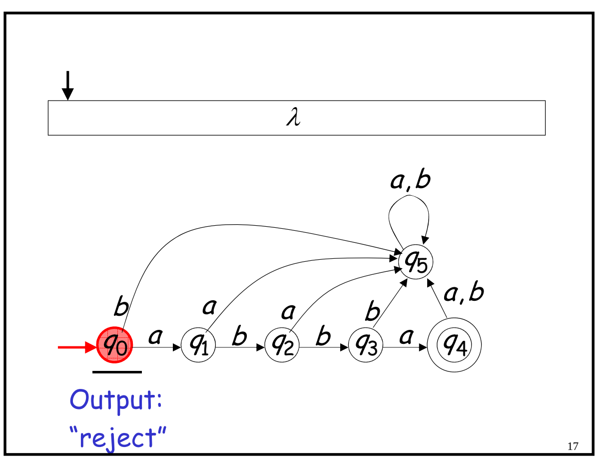

a b a
a,b
q5
b a a b a,b
q0 a q1 b q2 b q3 a q4
12

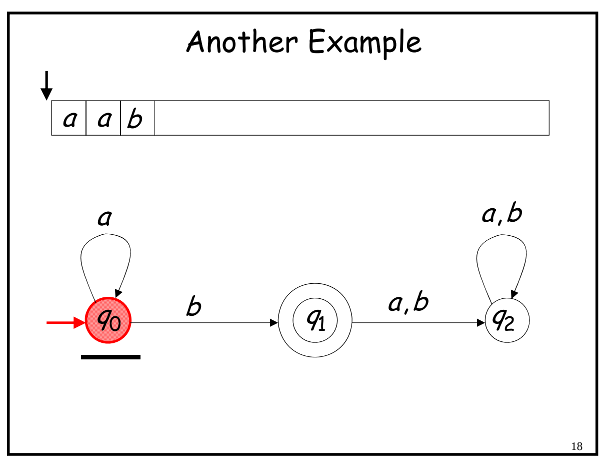

a b a
a,b
b a b a b a
a,b
13

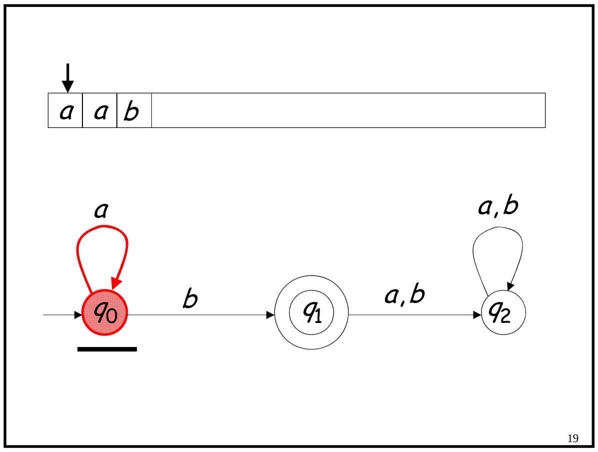

a b a
a,b
b a b a b a
a,b
14

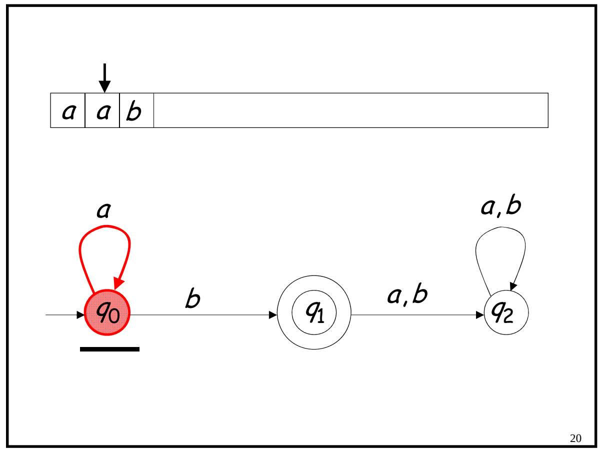

Input finished
a b a
a,b
Output:
"reject"
a,b
15

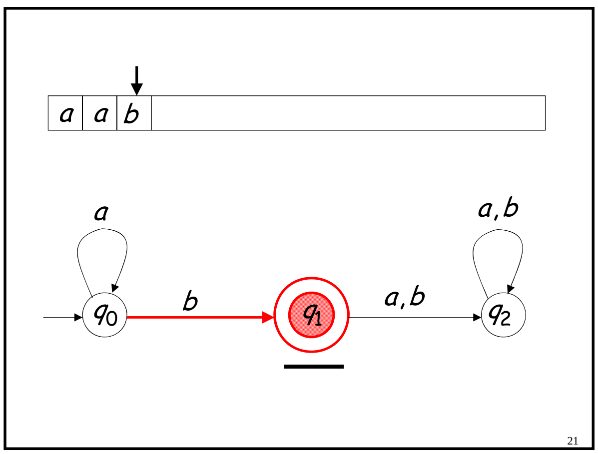

Another Rejection
λ
a,b
a,b
16

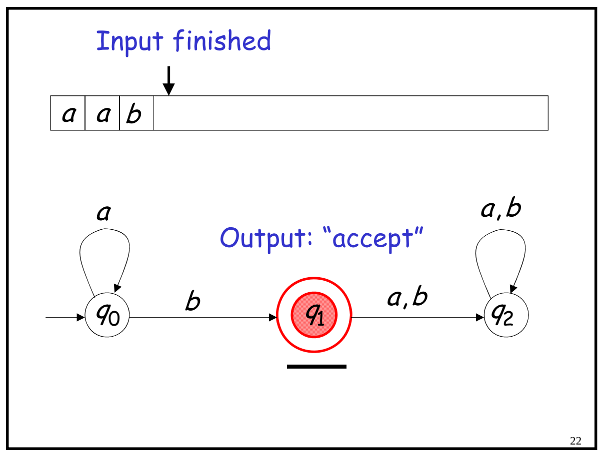

λ
a,b
a,b
Output:
"reject"
17

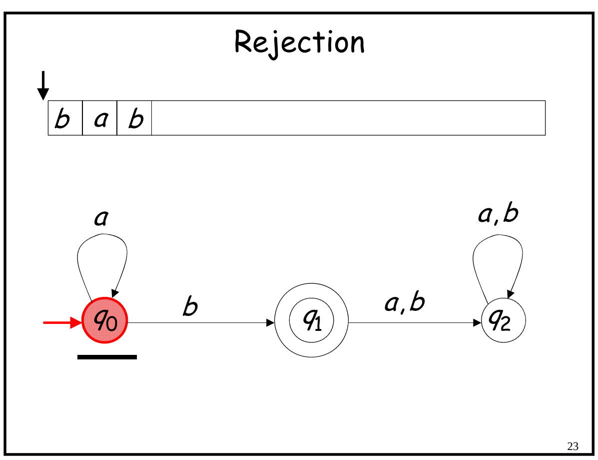

Another Example
a a b
a
b
a,b
a,b
18

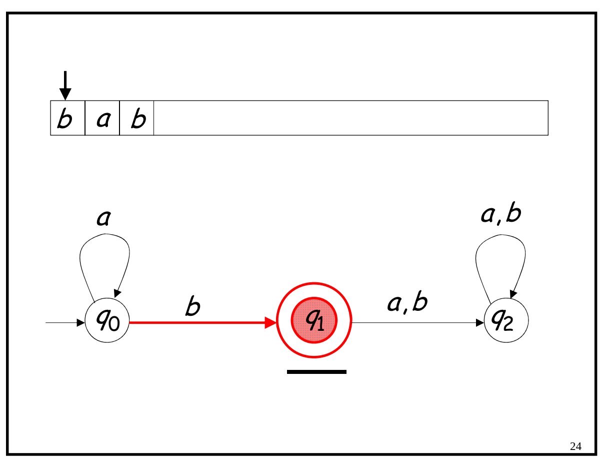

a a b
a
a,b
b
a,b
q_0 q_1 q_2
19

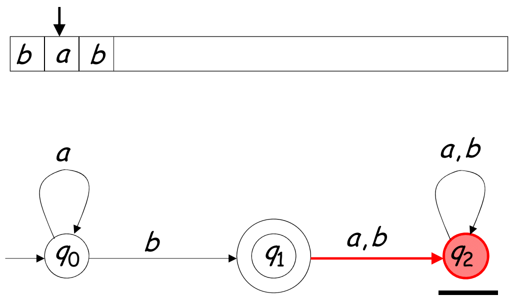

a a b
a
a,b
b
a,b
q_0 q_1 q_2
20


a a b
a
a,b
b
a,b
q_0 q_1 q_2
21

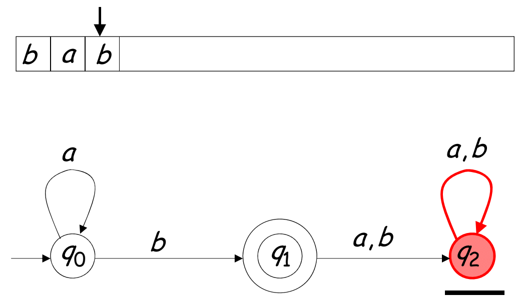

Input finished
a a b
a
Output: "accept"
a,b
b
a,b
q_0 q_1 q_2
22


Rejection
b a b
a
a,b
b
a,b
q_0 q_1 q_2
23


b a b
a
a,b
b
a,b
q_0 q_1 q_2
24

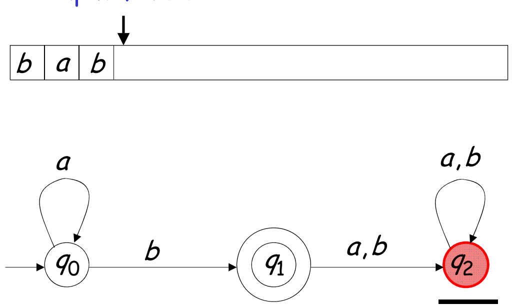


25



26

Input finished



Output: "reject"

27

Formalities

Deterministic Finite Accepter (DFA)

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

Q : set of states

Σ : input alphabet

δ : transition function

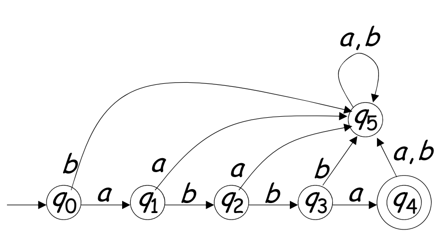
q_0 : initial state

F : set of final states

28

Input Alphabet Σ

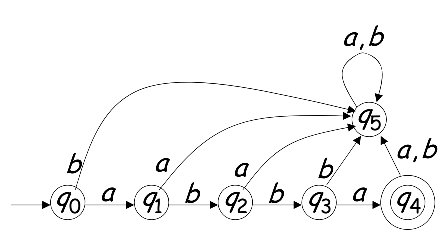
$\Sigma = \{a, b\}$



29

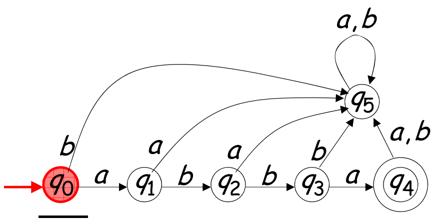
Set of States Q

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



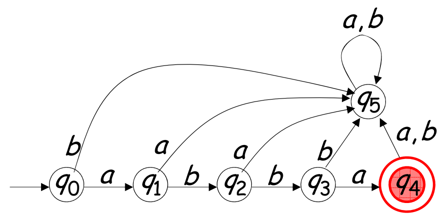
30

Initial State q_0



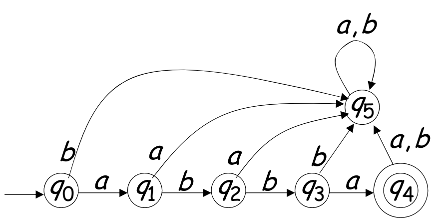
Set of Final States F

$F = \{q_4\}$

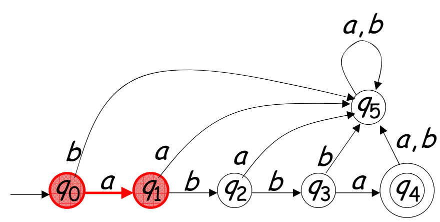


Transition Function δ

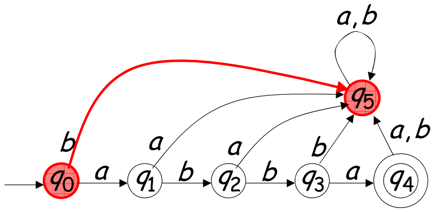
$\delta : Q \times \Sigma \rightarrow Q$



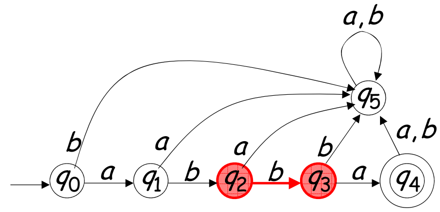
$\delta(q_0, a) = q_1$



$\delta(q_0, b) = q_5$

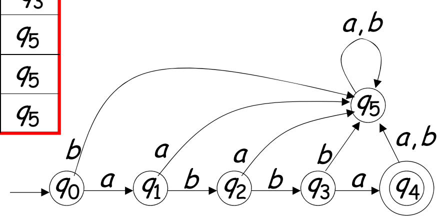


$\delta(q_2, b) = q_3$



Transition Function δ

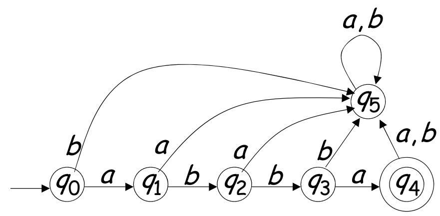
δ	a	b
q_0	q_1	q_5
q_1	q_5	q_2
q_2	q_5	q_3
q_3	q_4	q_5
q_4	q_5	q_5
q_5	q_5	q_5



37

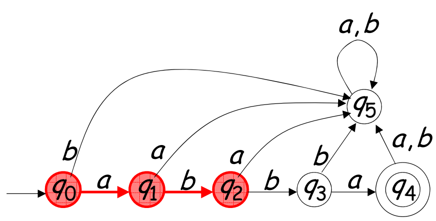
Extended Transition Function $\delta *$

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



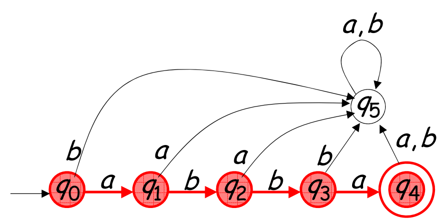
38

$$\delta * (q_0, ab) = q_2$$



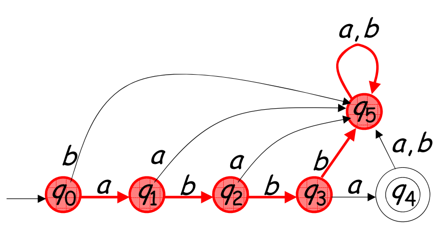
39

$$\delta * (q_0, abba) = q_4$$



40

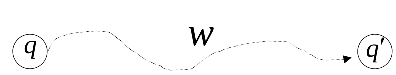
$$\delta * (q_0, abbbaa) = q_5$$



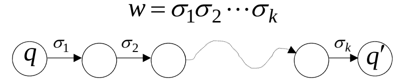
41

Observation: There is a walk from q to q' with label w

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



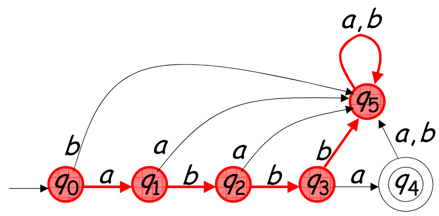
$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



42

Example: There is a walk from q_0 to q_5 with label $abbbaa$

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



43

Recursive Definition

$$\delta^*(q, \lambda) = q$$

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

$\left.\begin{array}{l}\delta^*(q, w\sigma) = q' \\ \delta(q_1, \sigma) = q'\end{array}\right\} \Rightarrow \delta^*(q, w\sigma) = \delta(q_1, \sigma)$

$\left.\begin{array}{l}\delta^*(q, w\sigma) = \delta(q_1, \sigma) \\ \delta^*(q, w) = q_1\end{array}\right\} \Rightarrow \delta^*(q, w\sigma) = \delta(\delta^*(q, w), \sigma)$

44

$\delta^*(q_0, ab) =$

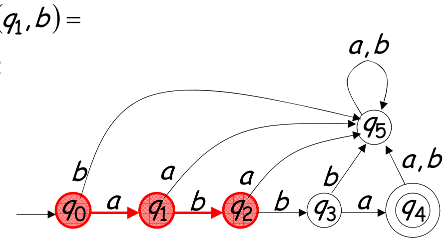
$\delta(\delta^*(q_0, a), b) =$

$\delta(\delta(\delta^*(q_0, \lambda), a), b) =$

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

$\delta(q_1, b) =$

q_2



45

Languages Accepted by DFAs

Take DFA M

Definition:

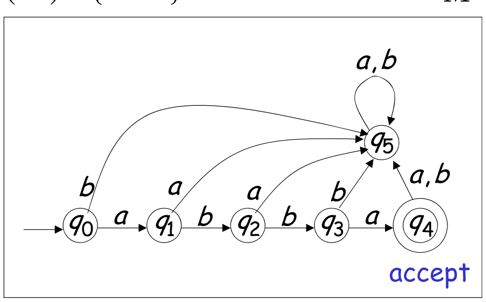
The language $L(M)$ contains all input strings accepted by M

$L(M)$ = { strings that drive M to a final state}

46

Example

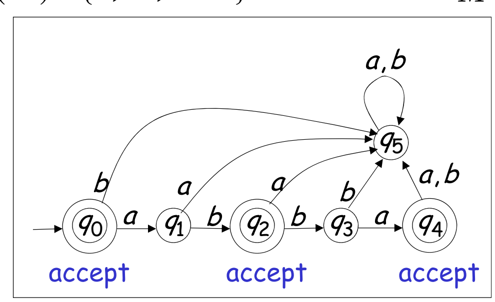
$L(M) = \{abba\}$ $\qquad M$



47

Another Example

$L(M) = \{\lambda, ab, abba\}$ $\qquad M$



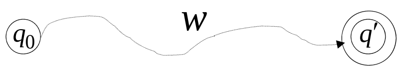
48

Formally

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

Language accepted by M :

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



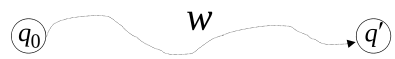
$q' \in F$

49

Observation

Language rejected by M :

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

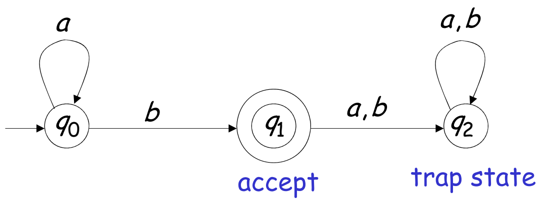


$q' \notin F$

50

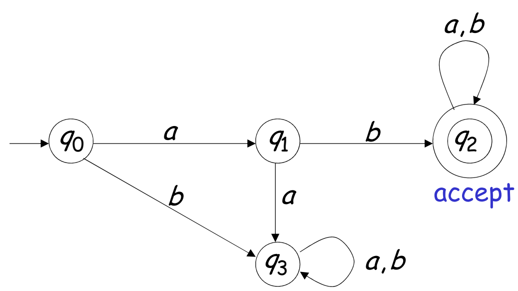
More Examples

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



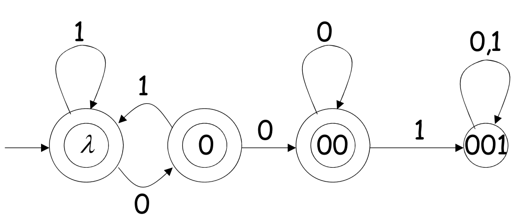
51

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



52

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



53

Regular Languages

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

All regular languages form a language family

54

Examples of regular languages:

$\{abba\}$ $\{\lambda, ab, abba\}$ $\{a^n b : n \geq 0\}$

{ all strings with prefix ab }

{ all strings with prefix ab }

{ all strings without substring 001 }

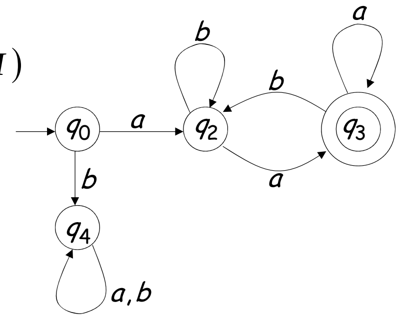
There exist automata that accept these Languages (see previous slides).

55

Another Example

The language $L = \{awa : w \in \{a,b\}^*\}$ is regular:

$L = L(M)$



56

There exist languages which are not Regular:

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

There is no DFA that accepts such a language

(we will prove this later in the class)

57