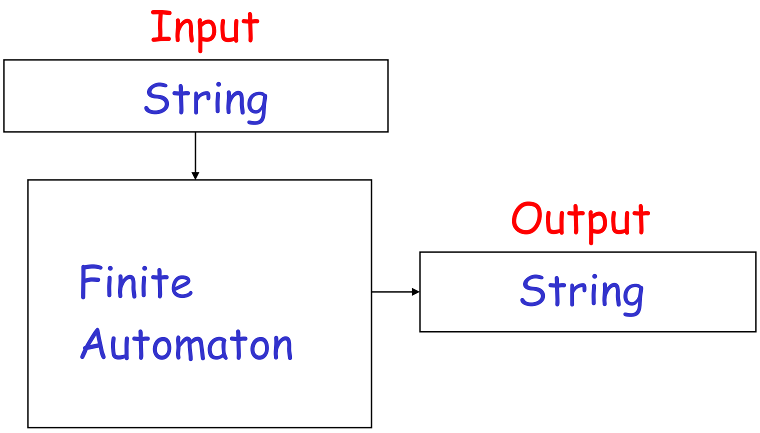
Finite Automata

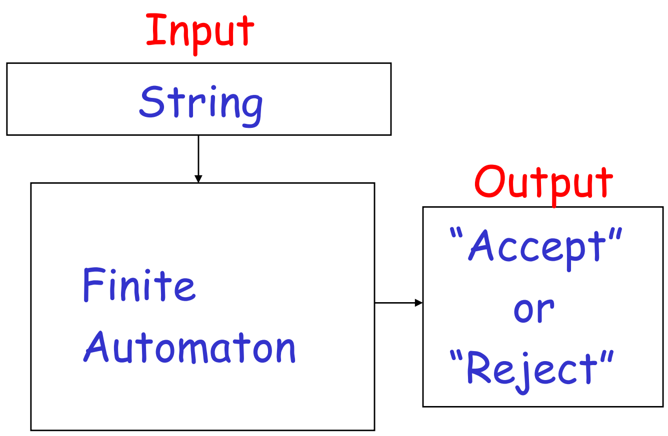
1

Finite Automaton



2

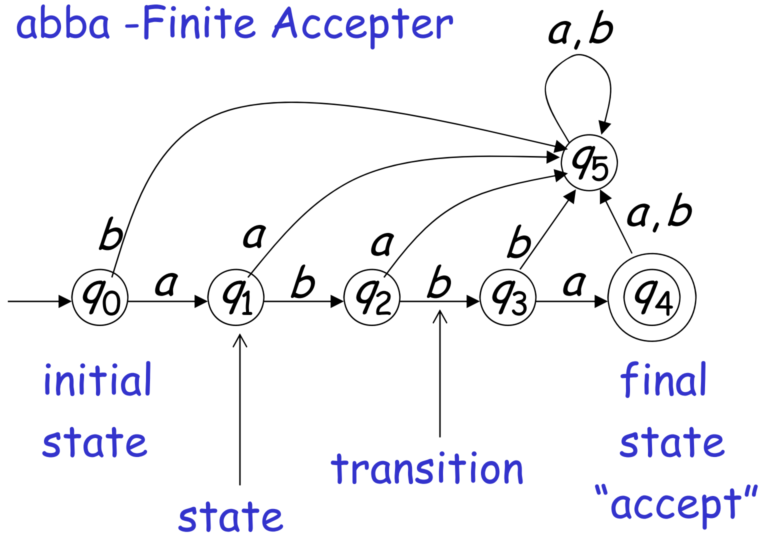
Finite Accepter



3

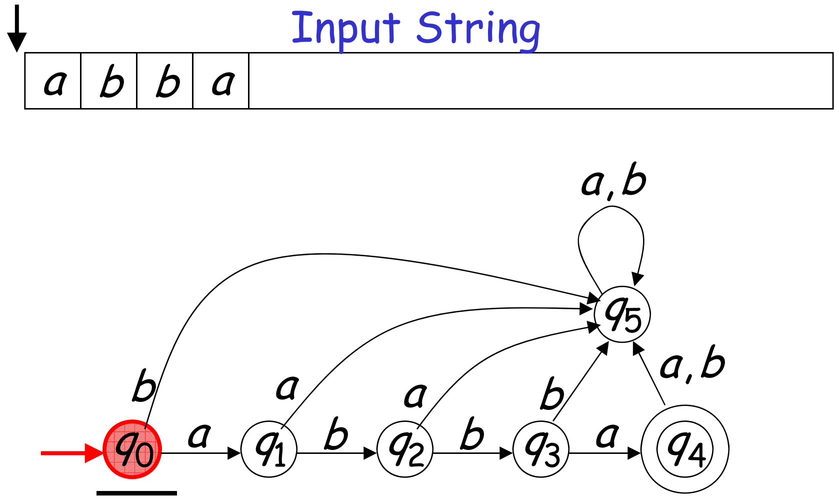
Transition Graph

abba -Finite Accepter



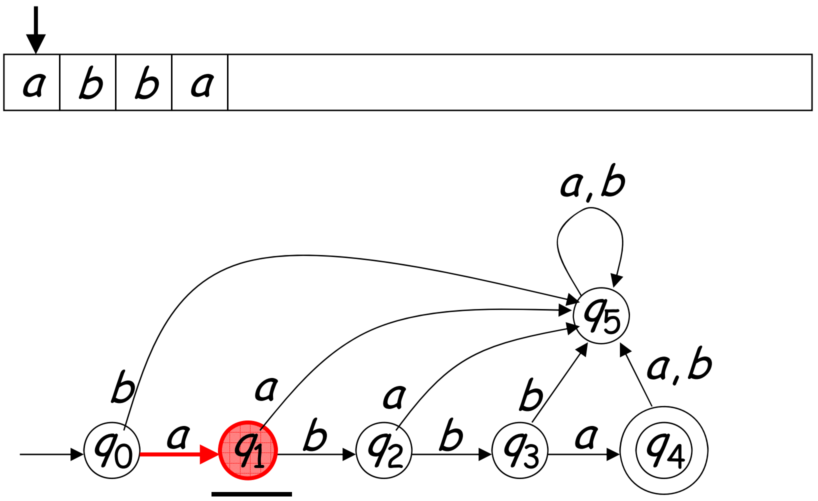
4

Initial Configuration

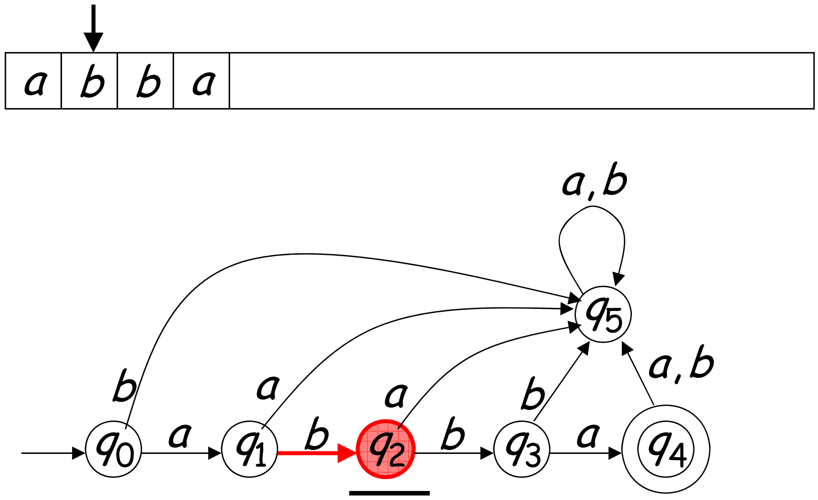


5

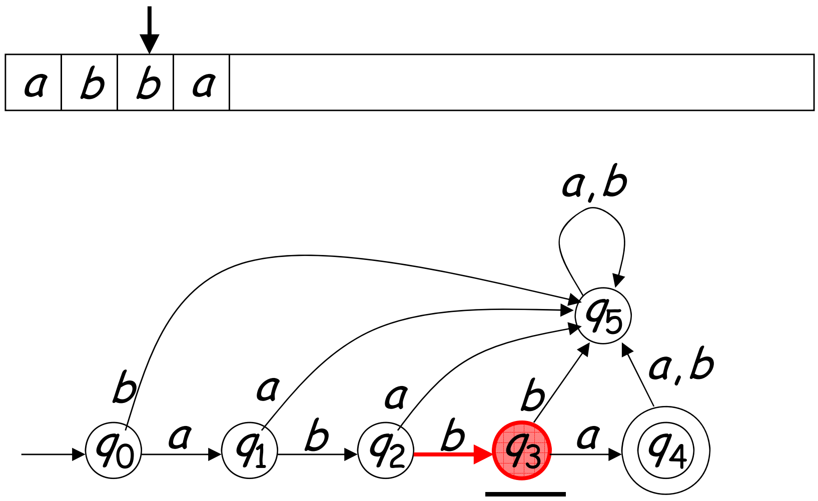
Reading the Input



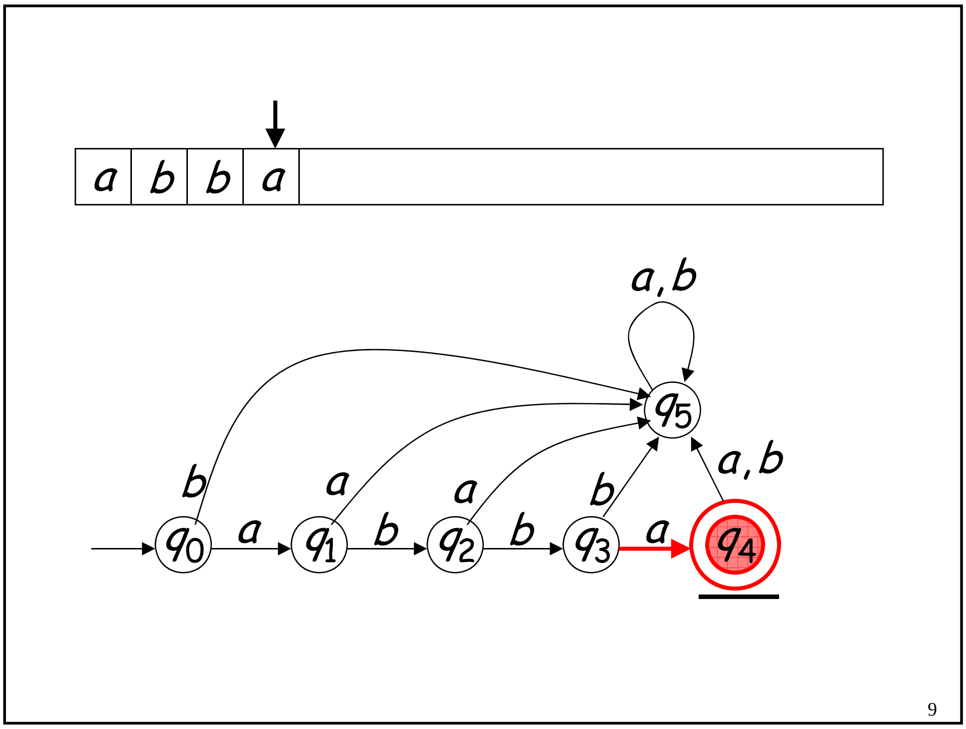
6

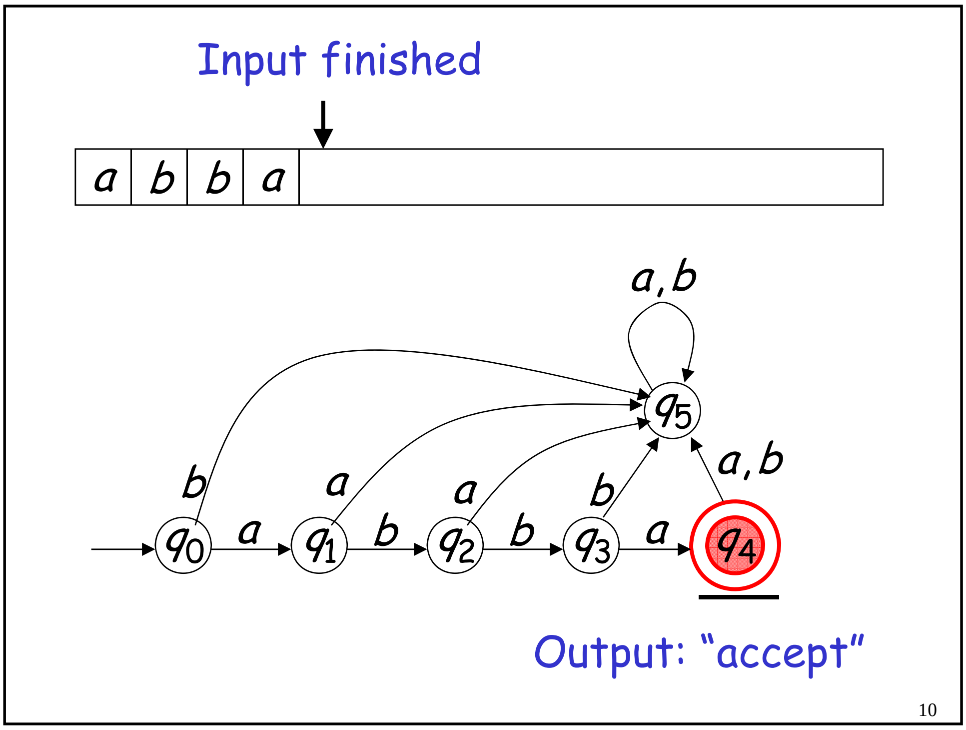

a
b
b
a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4

7

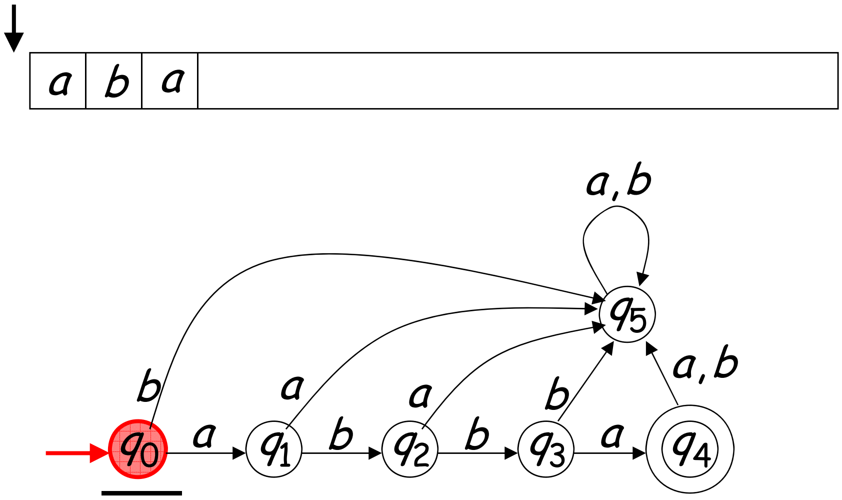

a
b
b
a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4

8

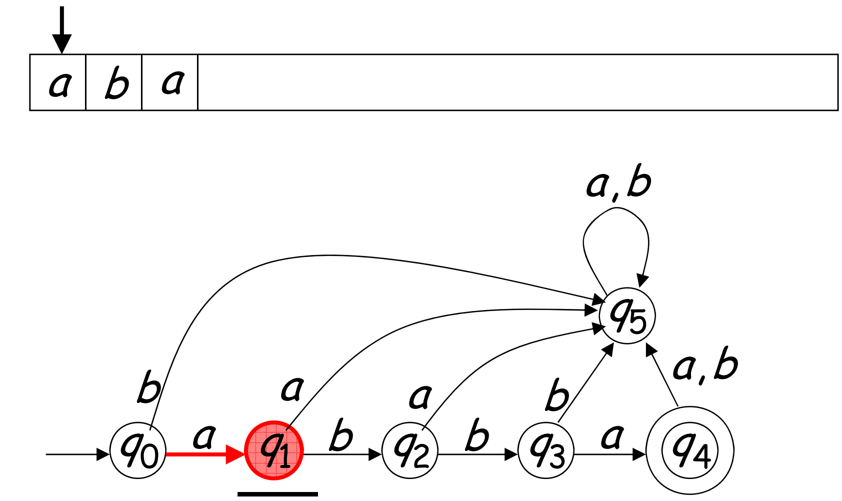

a b b a
a,b
b
a
a
b
a,b
q_0 a q_1 b q_2 b q_3 a q_4
q_5
9


Input finished
a b b a
a,b
b
a
a
b
a,b
q_0 a q_1 b q_2 b q_3 a q_4
q_5
Output: "accept"
10

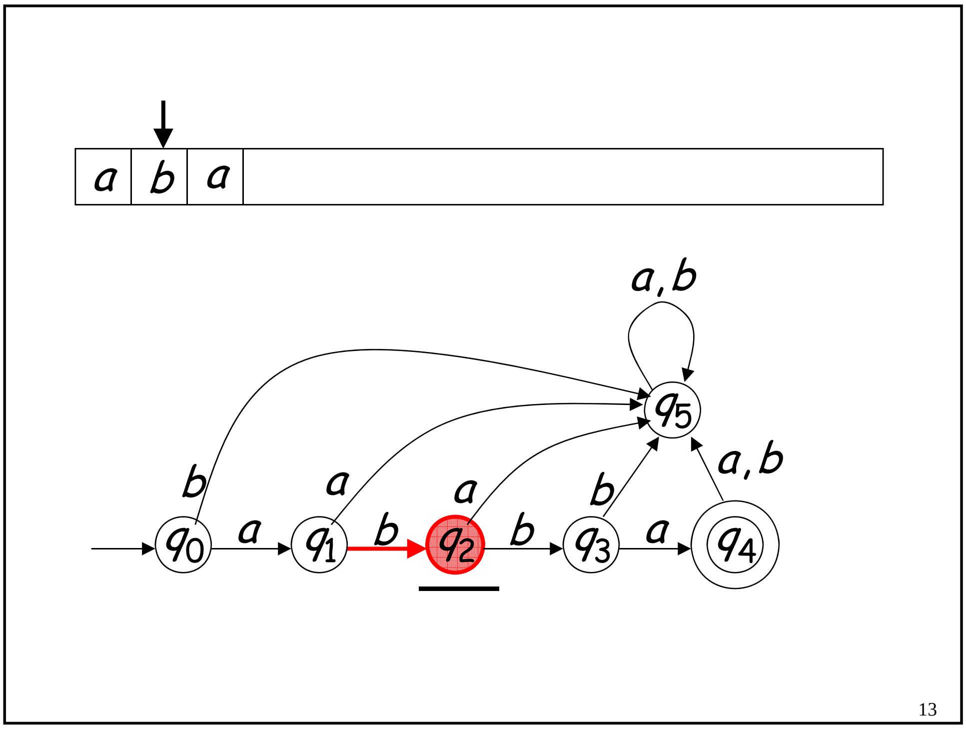
Rejection



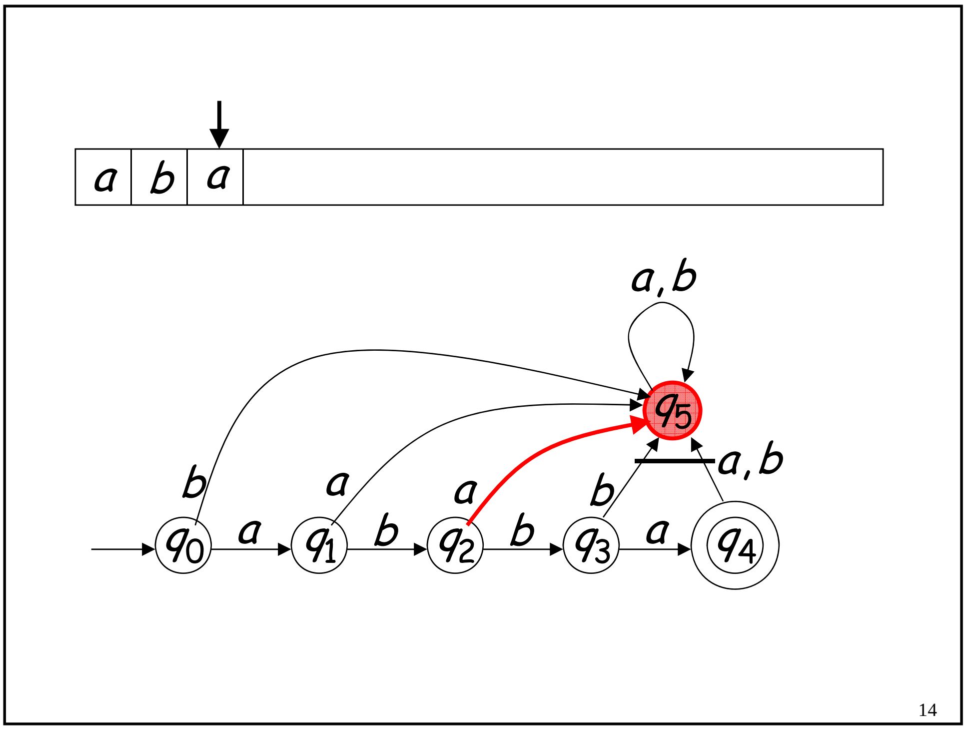
a b a
a,b
q5
b
a
a
b
a,b
q0 a q1 b q2 b q3 a q4



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

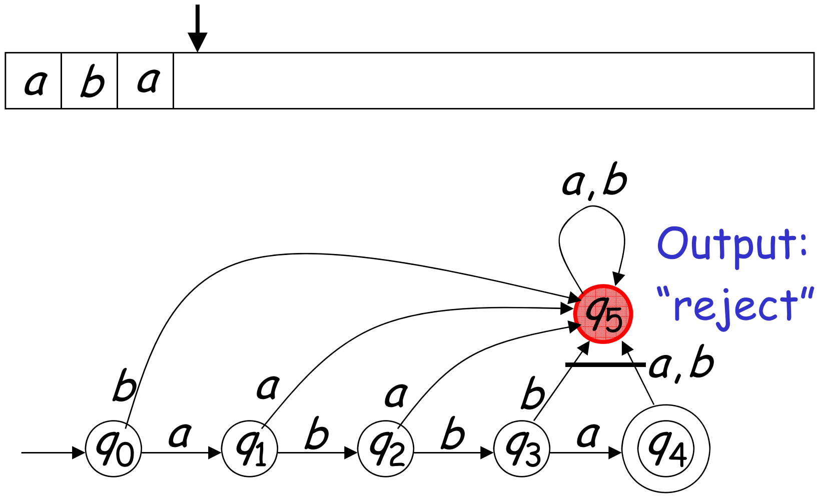


a b a
a,b
q5
b
a
a
a,b
b
q0 a q1 b q2 b q3 a q4
13



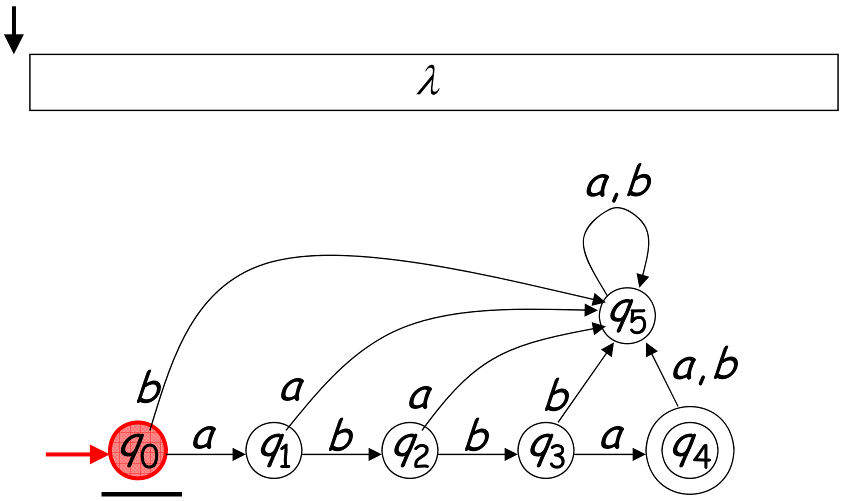
a b a
a,b
q5
a,b
b
a
a
b
q0 a q1 b q2 b q3 a q4
14

Input finished

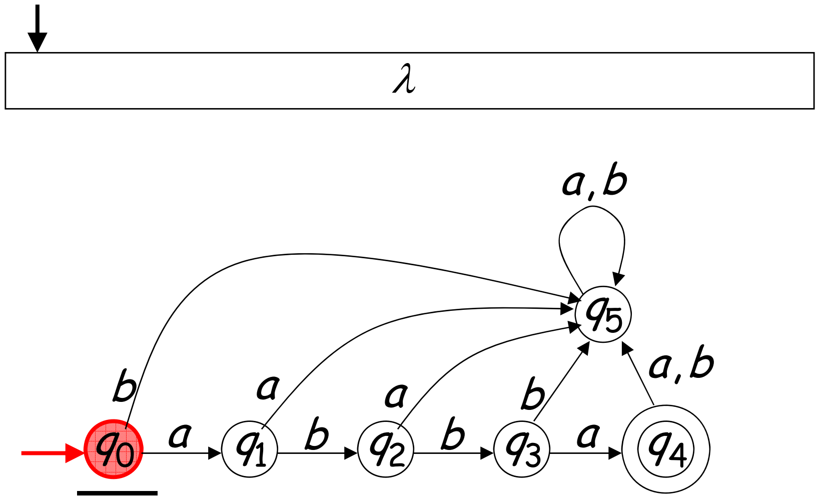


15

Another Rejection

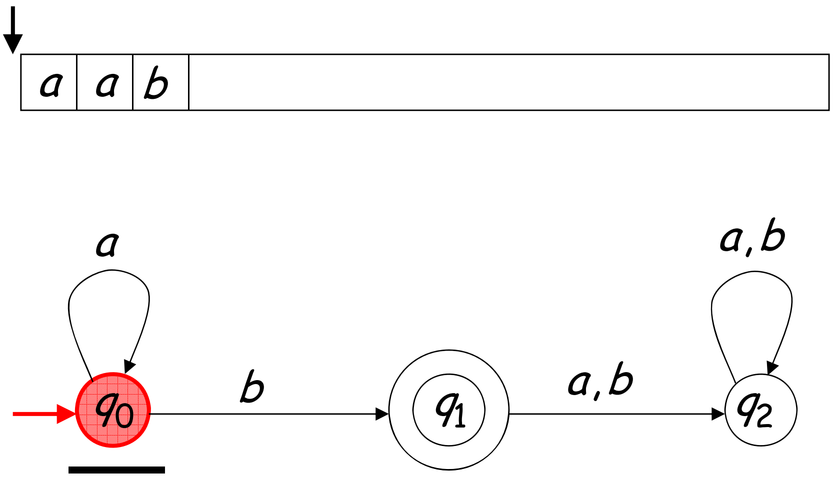


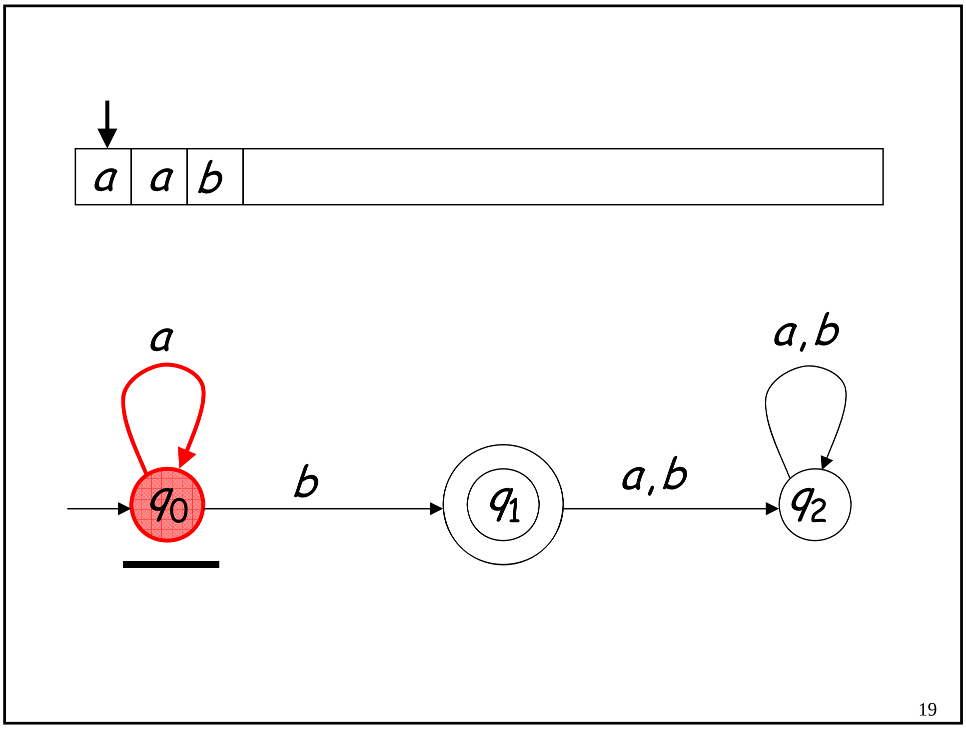
16



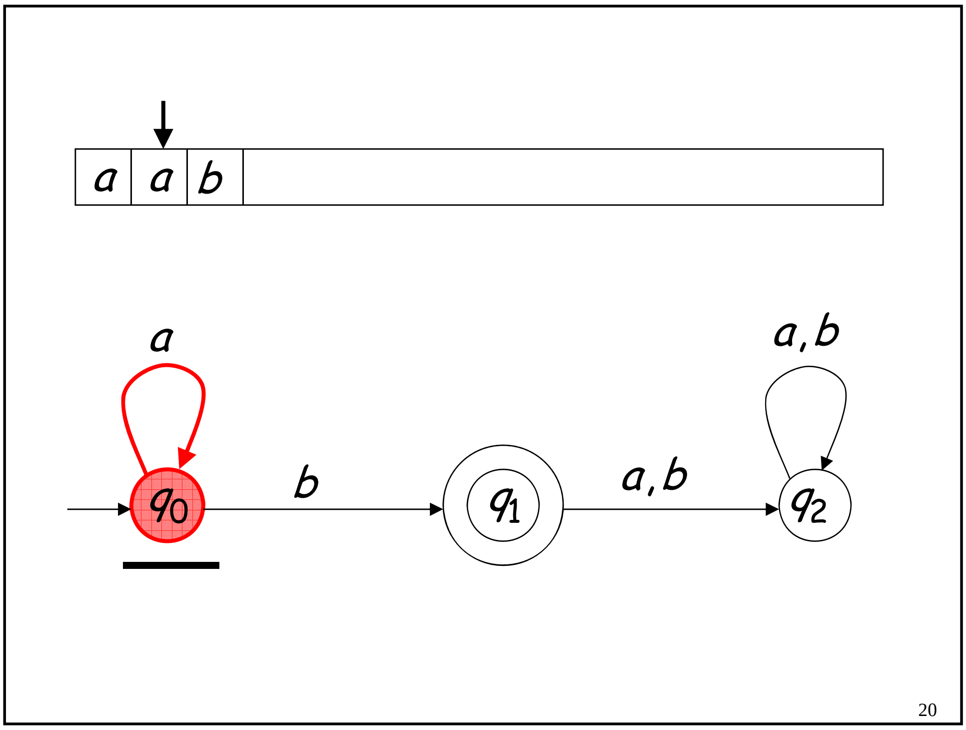
Output:
"reject"

Another Example

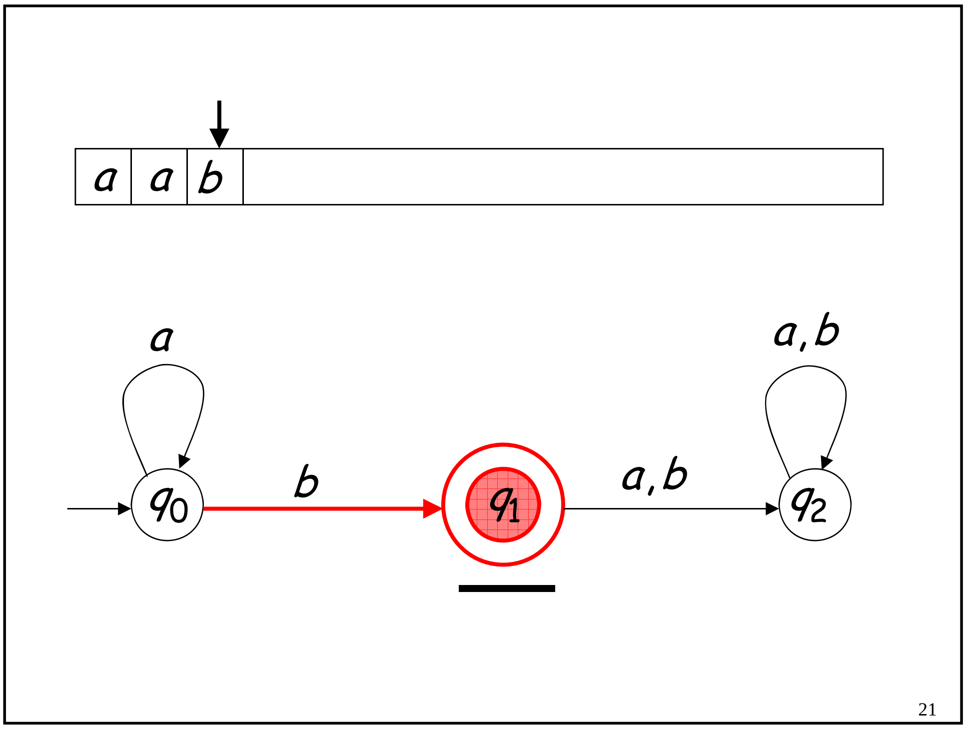


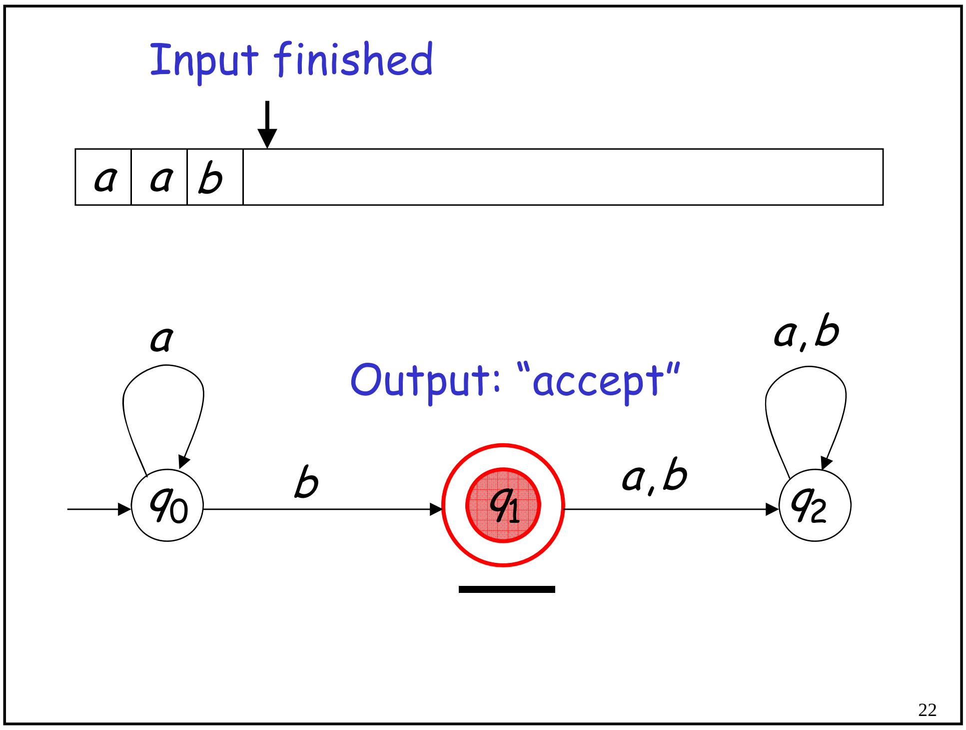


19

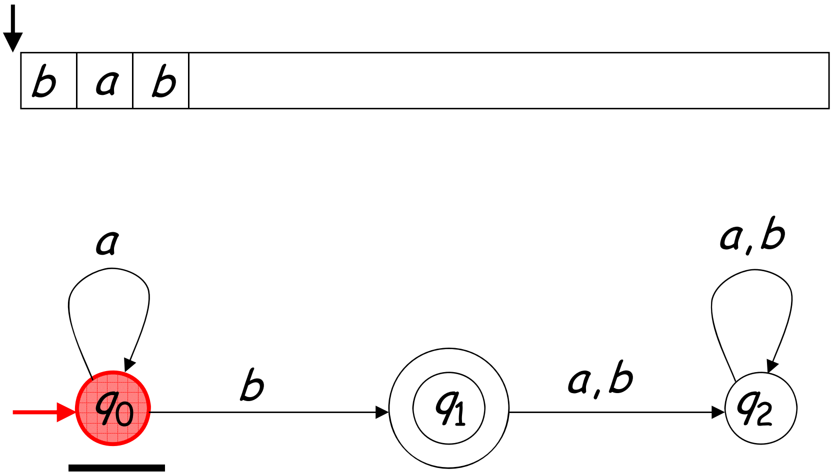


20

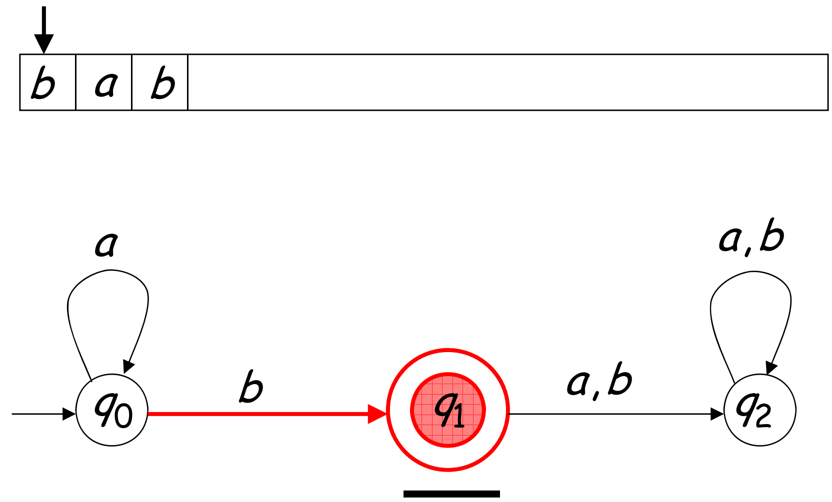

a a b
a
a,b
b
a,b
q0
q1
q2
21


Input finished
a a b
a
a,b
Output: “accept”
b
a,b
q0
q1
q2
22

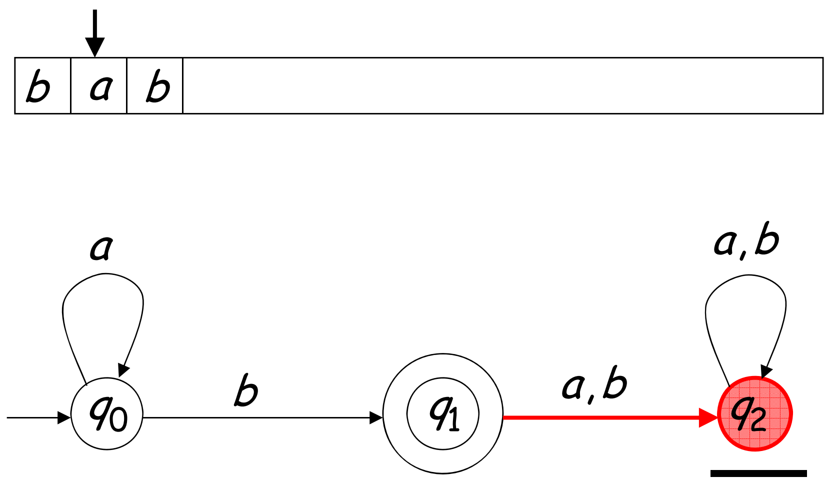
Rejection



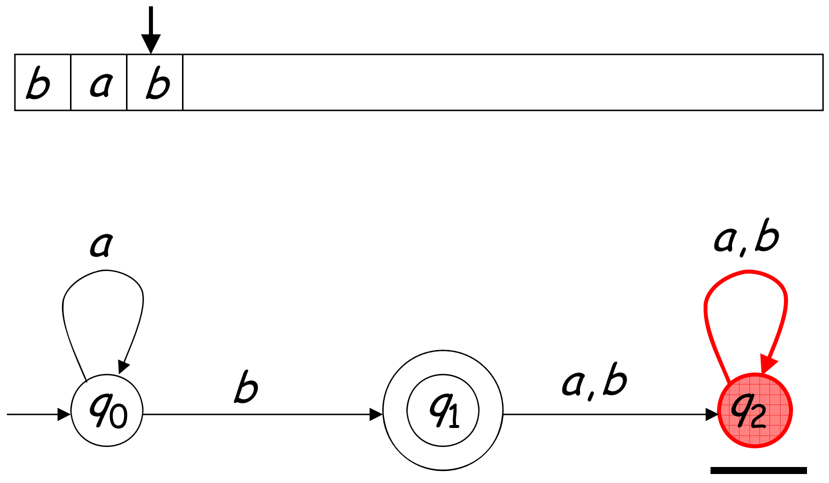
23



24

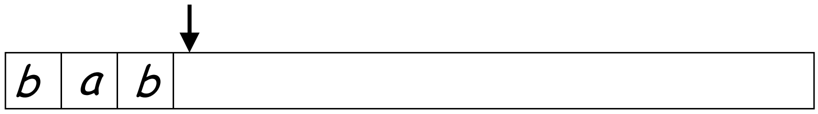


25

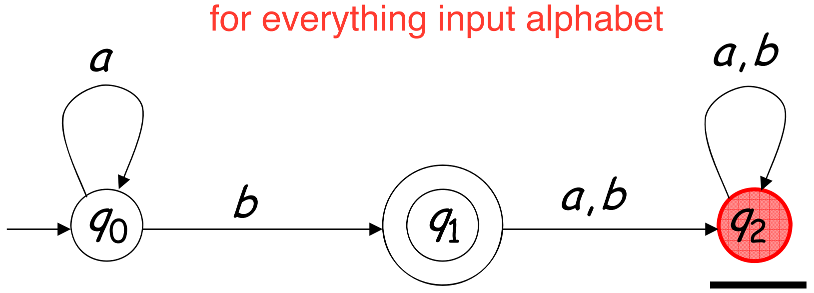


26

Input finished



DFA, for each state, we need to define transition for everything input alphabet



Output: "reject"

27

Formalities

Deterministic Finite Accepter (DFA)

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

Q : set of states

Σ : input alphabet set

δ : transition function

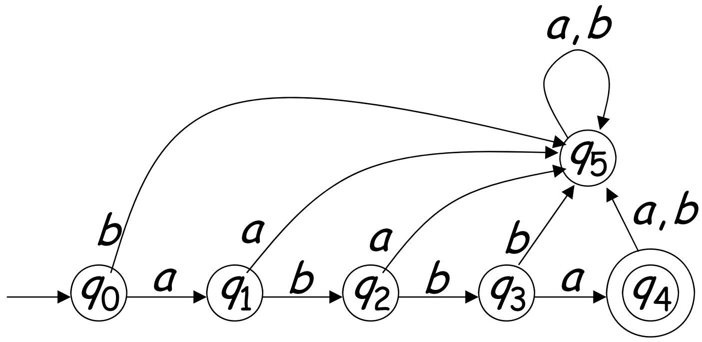
q_0 : initial state

F : set of final states

28

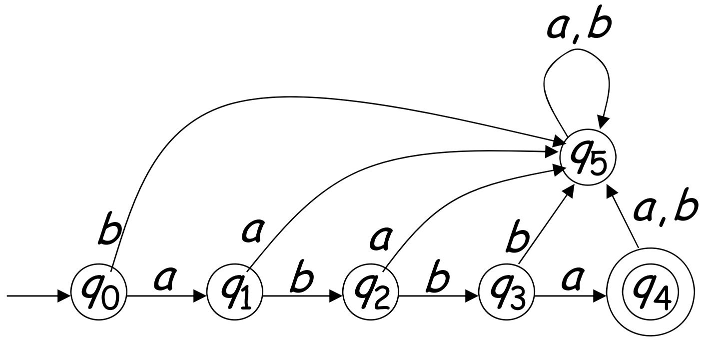
Input Alphabet Σ

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

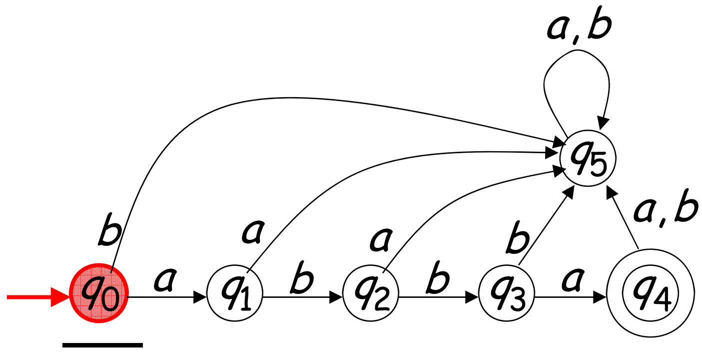


Set of States Q

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



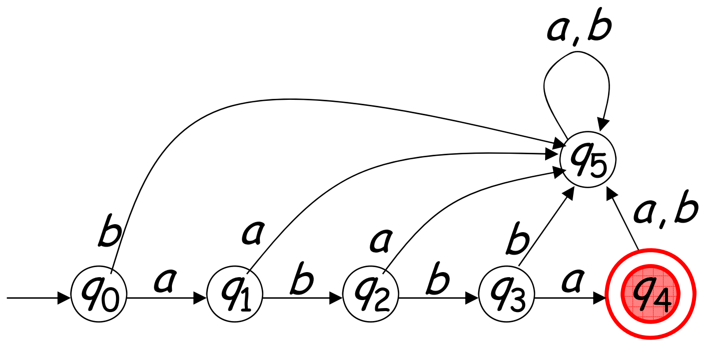
Initial State q_0



31

Set of Final States F

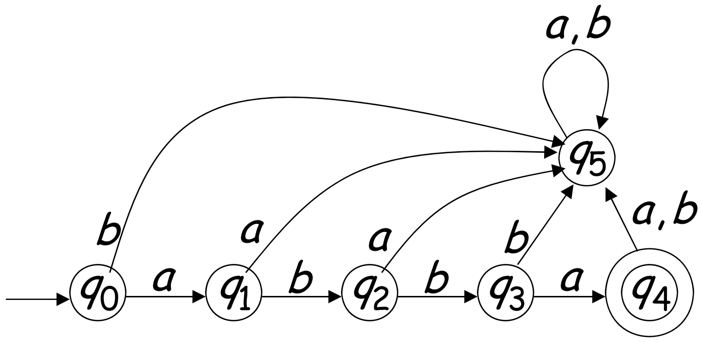
$$F = \{q_4\}$$



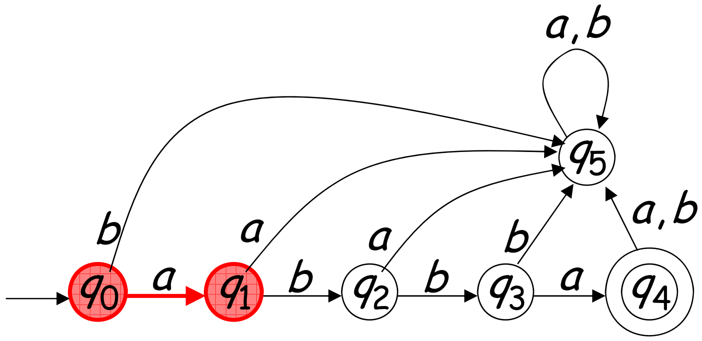
32

Transition Function δ

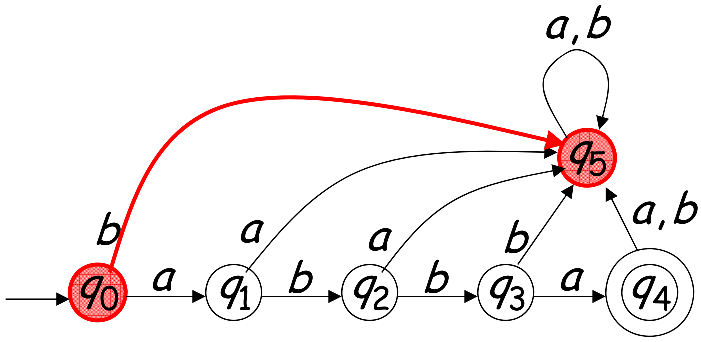
$$\delta : Q \times \Sigma \rightarrow Q$$



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

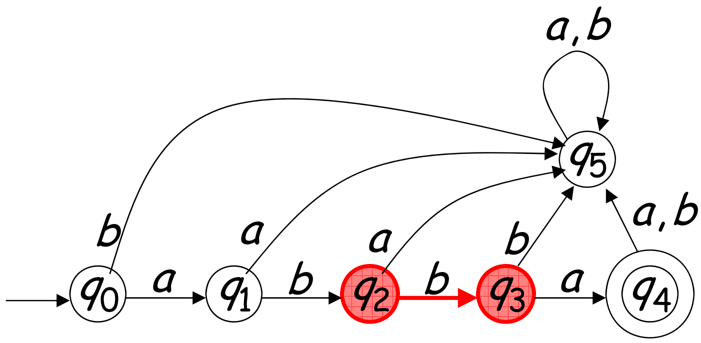


$$\delta(q_0, b) = q_5$$



35

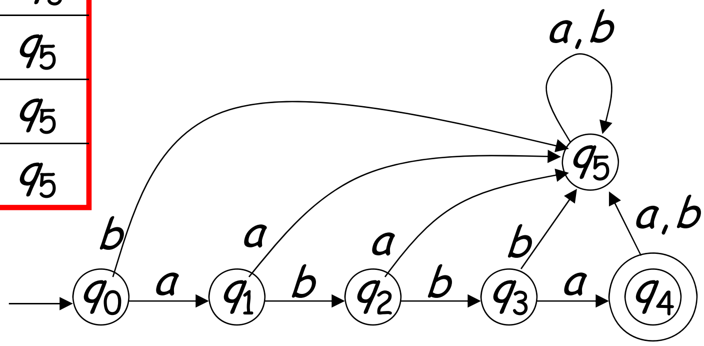
$$\delta(q_2, b) = q_3$$



36

Transition Function δ

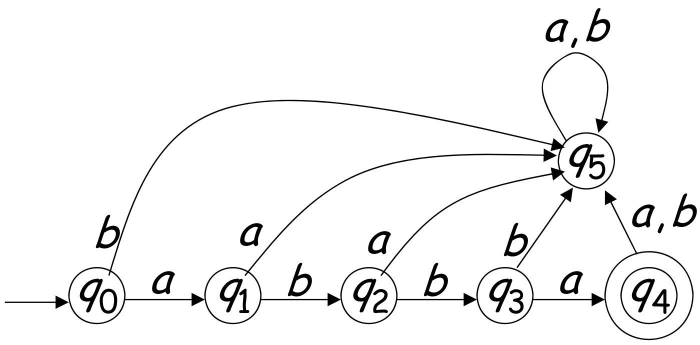
δ	a	b
q_0	q_1	q_5
q_1	q_5	q_2
q_2	q_5	q_3
q_3	q_4	q_5
q_4	q_5	q_5
q_5	q_5	q_5



37

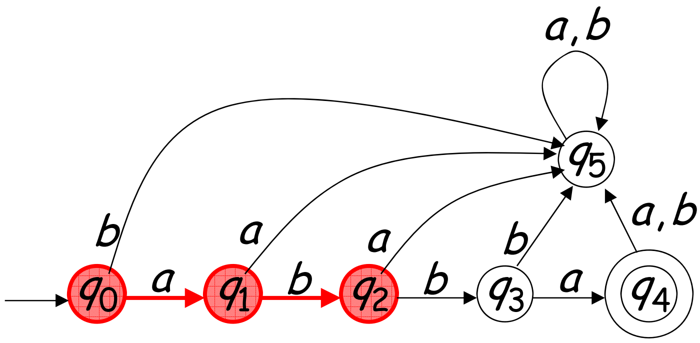
Extended Transition Function δ^*

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

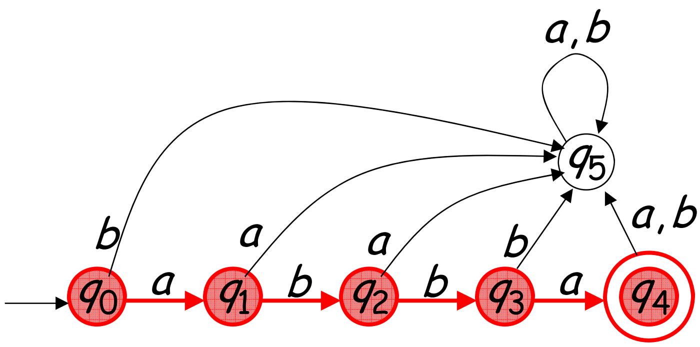


38

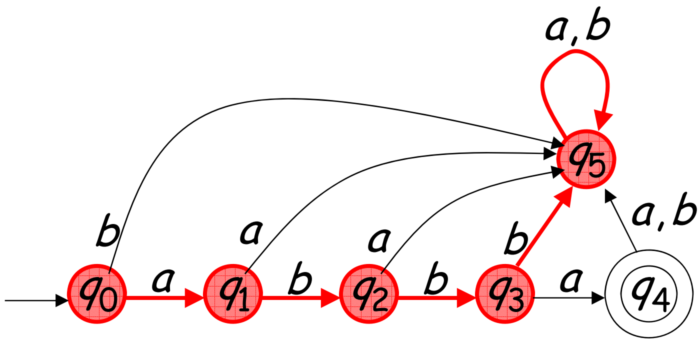
$$\delta^*(q_0, ab) = q_2$$



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



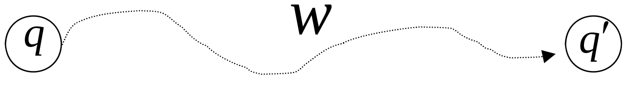
$$\delta^*(q_0, abbbaa) = q_5$$



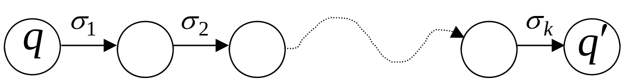
41

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

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



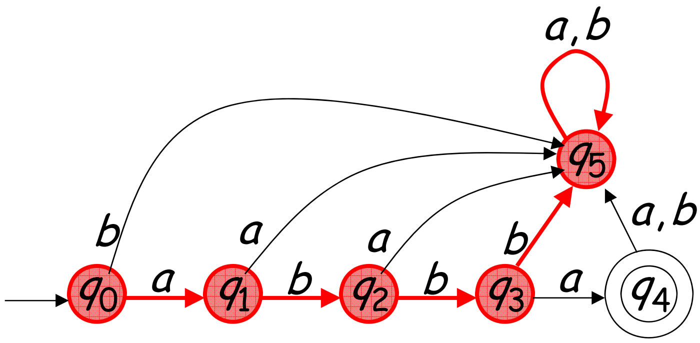
$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



42

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

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



Recursive Definition

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

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

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

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

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

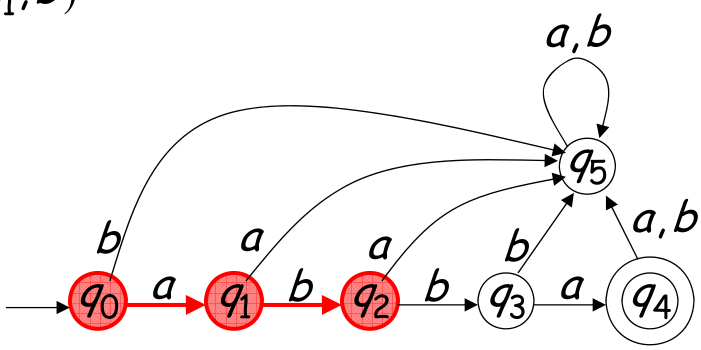
$\delta(\delta^*(q_0, a), b) =$

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

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

$\delta(q_1, b) =$

q_2



Languages Accepted by DFAs

Take DFA M

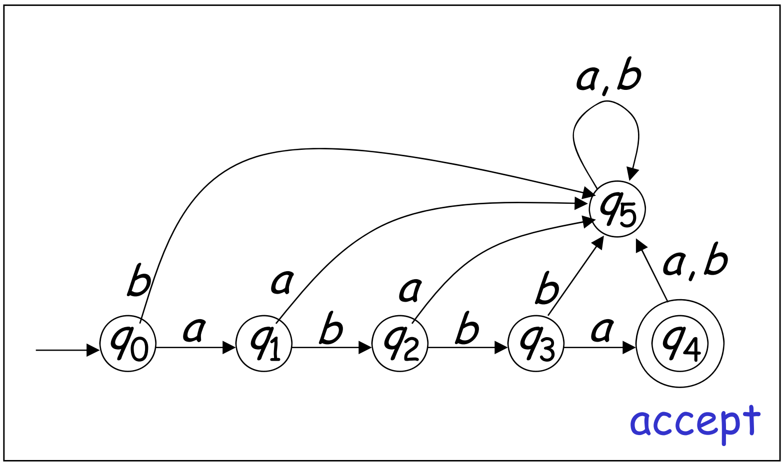
Definition:

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

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

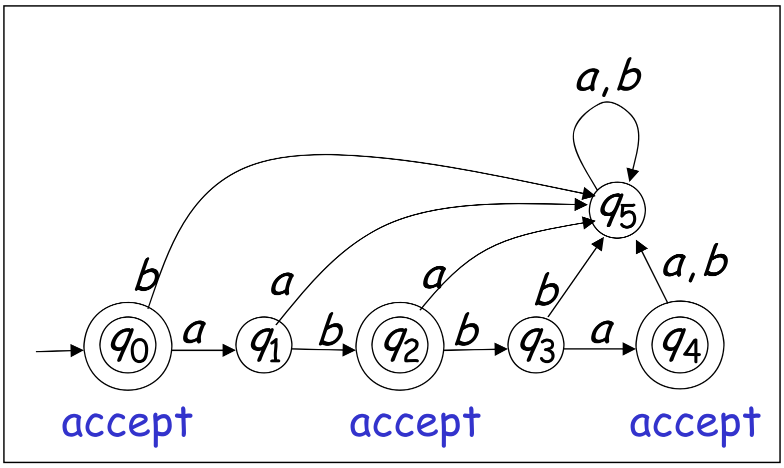
Example

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



Another Example

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

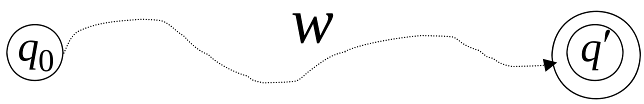


Formally

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

Language accepted by M :

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

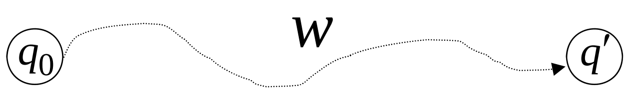


$q' \in F$

Observation

Language rejected by M :

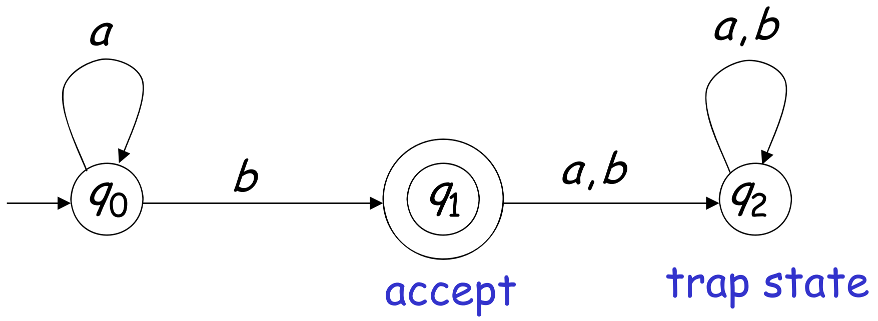
$$\overline{L(M)} = \{ w \in \Sigma^* : \delta^*(q_0, w) \notin F \}$$



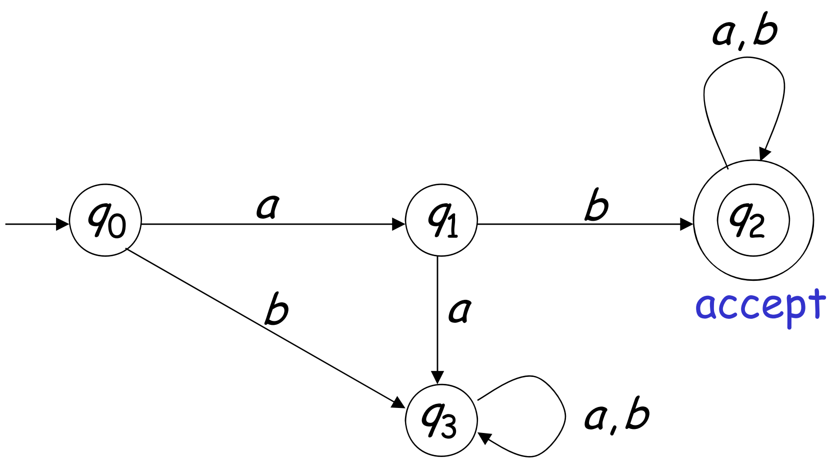
$q' \notin F$

More Examples

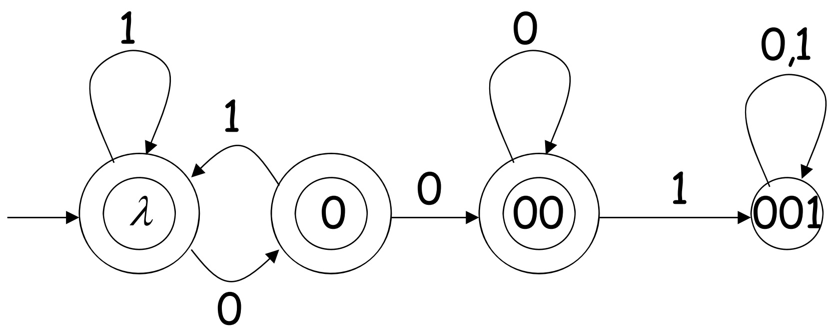
$$L(M)=\{a^n b : n \geq 0\}$$



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



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



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

Examples of regular languages:

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

{ all strings with prefix ab }

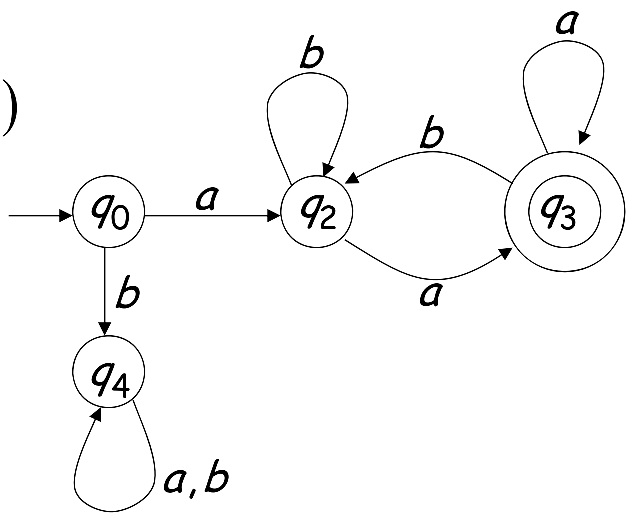
{ all strings without substring 001 }

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

Another Example

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

$L = L(M)$



There exist languages which are <u>not</u> 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