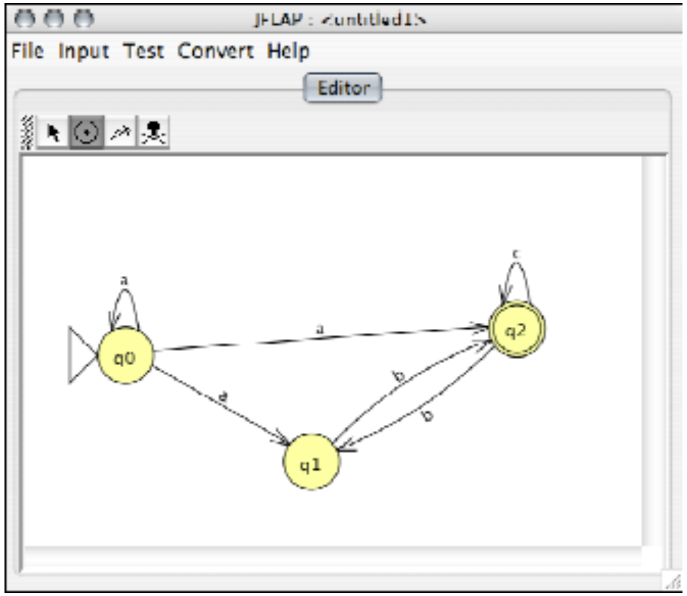
2016

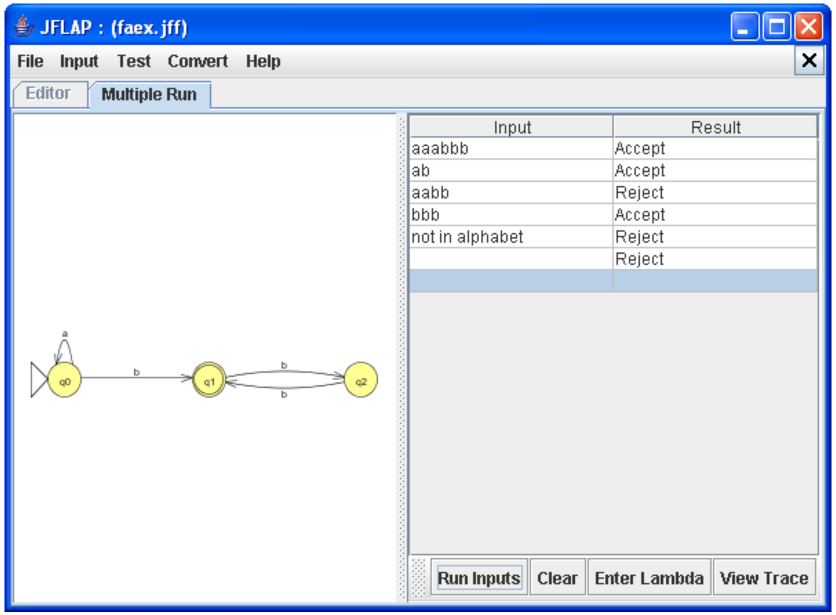
Theory of Computation

Kun-Ta Chuang
Department of Computer Science and Information Engineering
National Cheng Kung University

Announcement

- JFLAP (Java Formal Languages and Automata Package)
  - http://www.jflap.org/



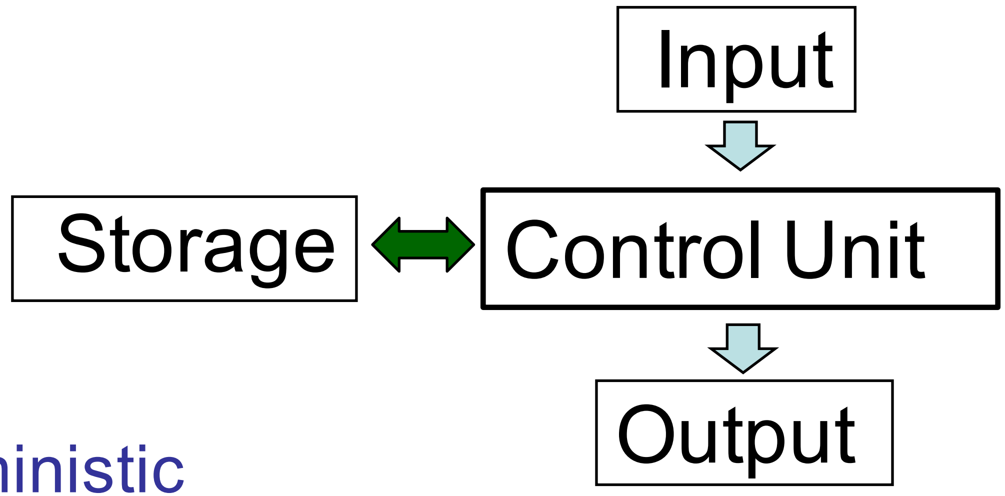


Outline

1. Deterministic Finite Accepters (DFA)
2. Nondeterministic Finite Accepters (NFA)
3. Equivalence of DFA and NFA
4. Reduction of the Number of States in FA*

Automata

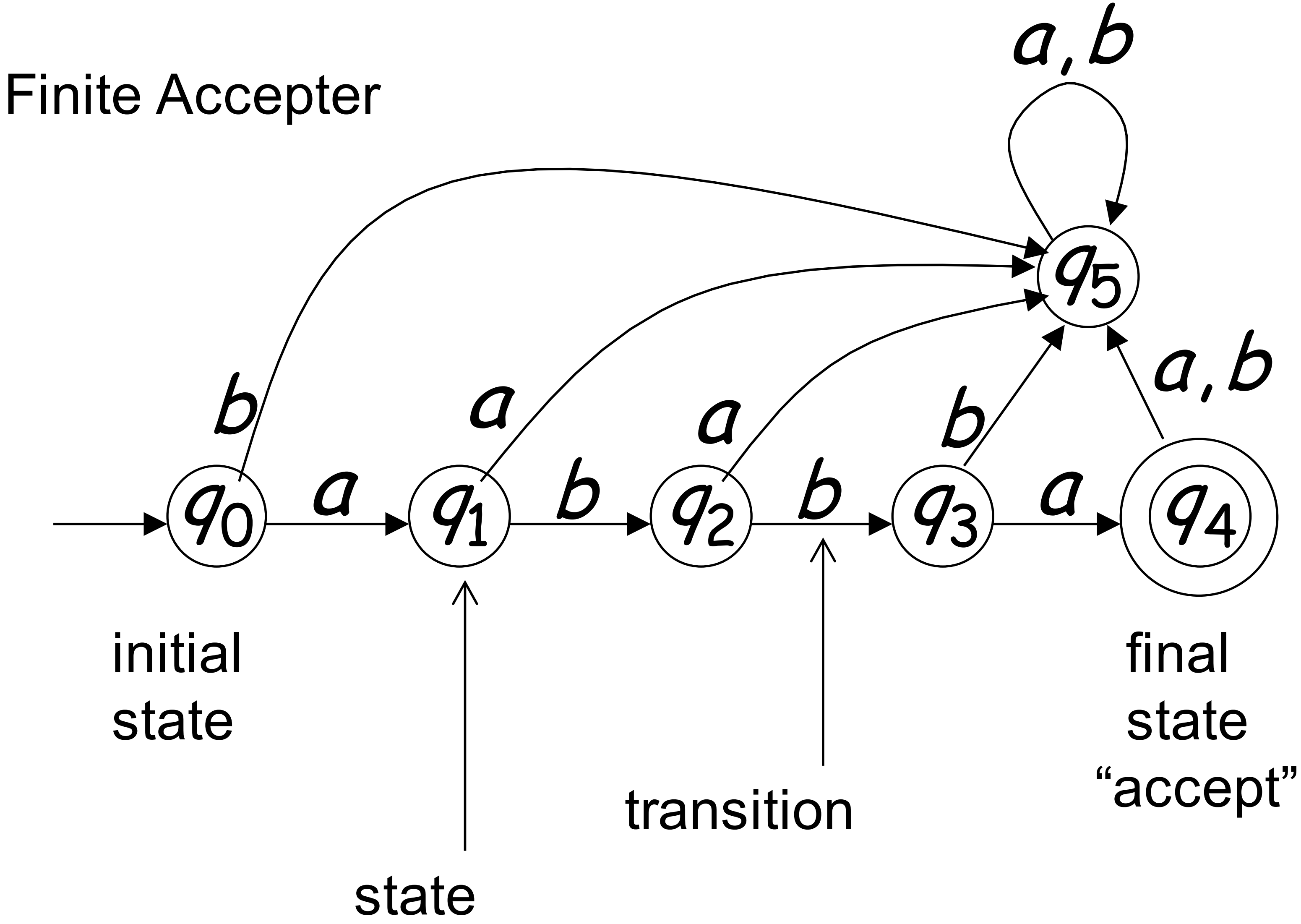
Automaton:
An abstract model of a digital computer



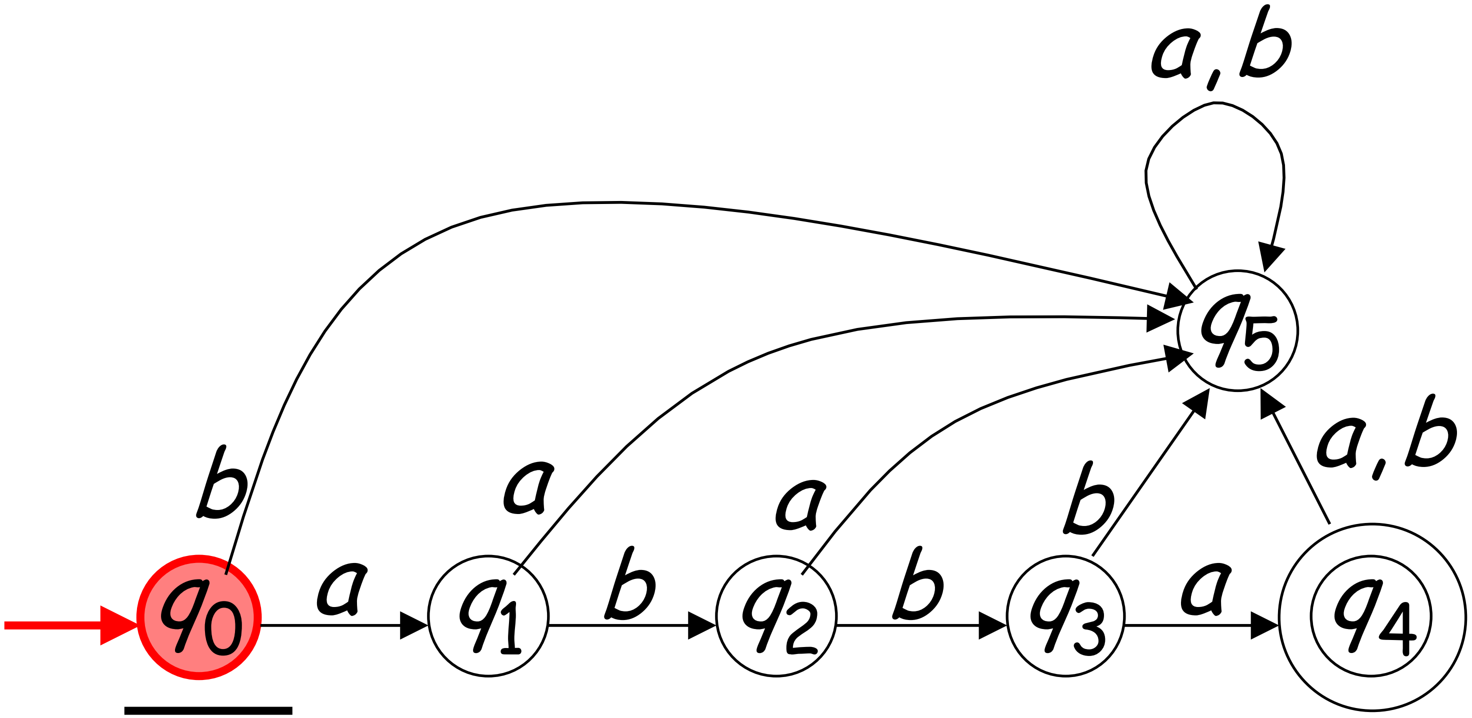
•Deterministic V.S. Nondeterministic

•An automaton whose output is YES or NO
Accepter
•An automaton whose output are strings of symbols
Transducer

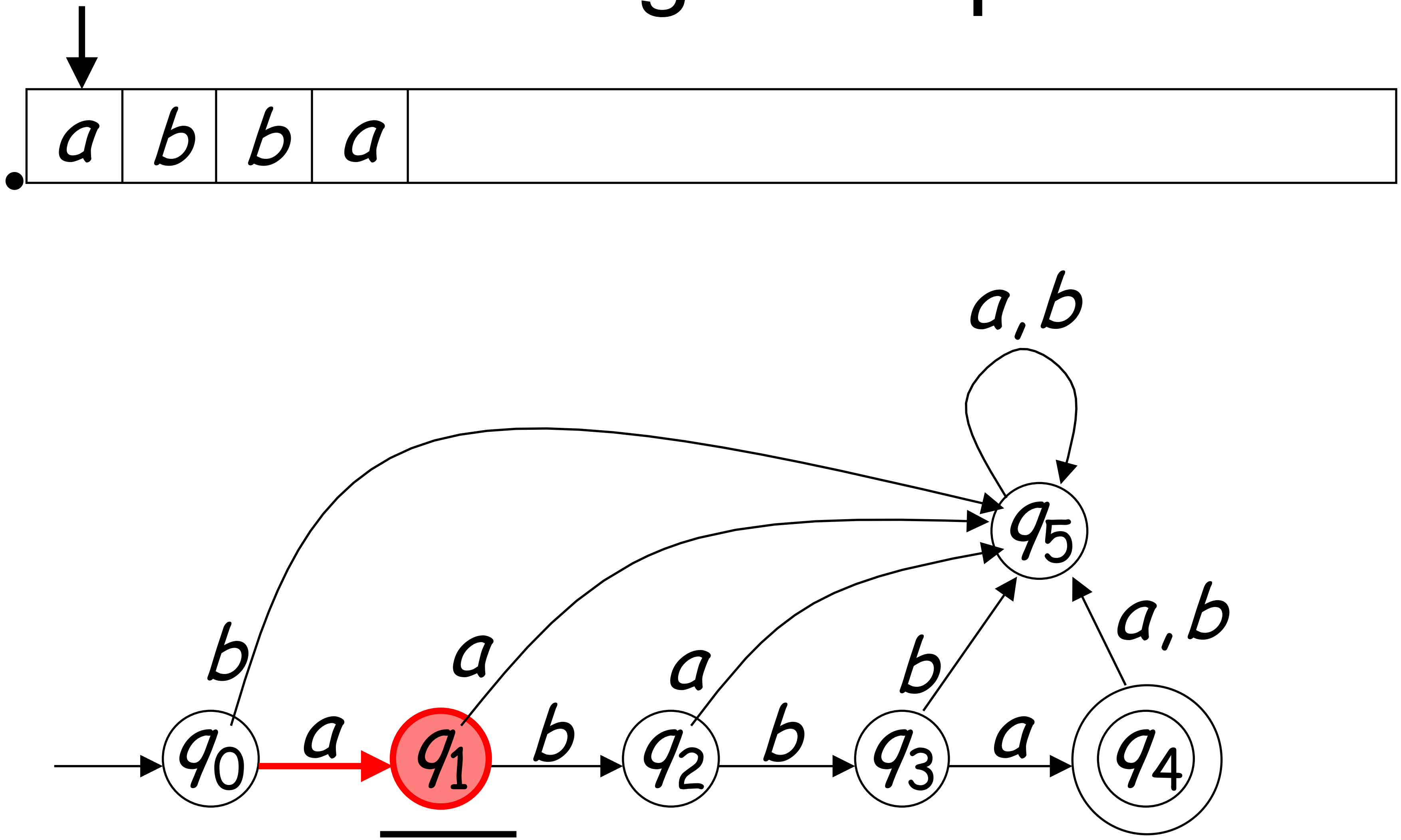
Transition Graph

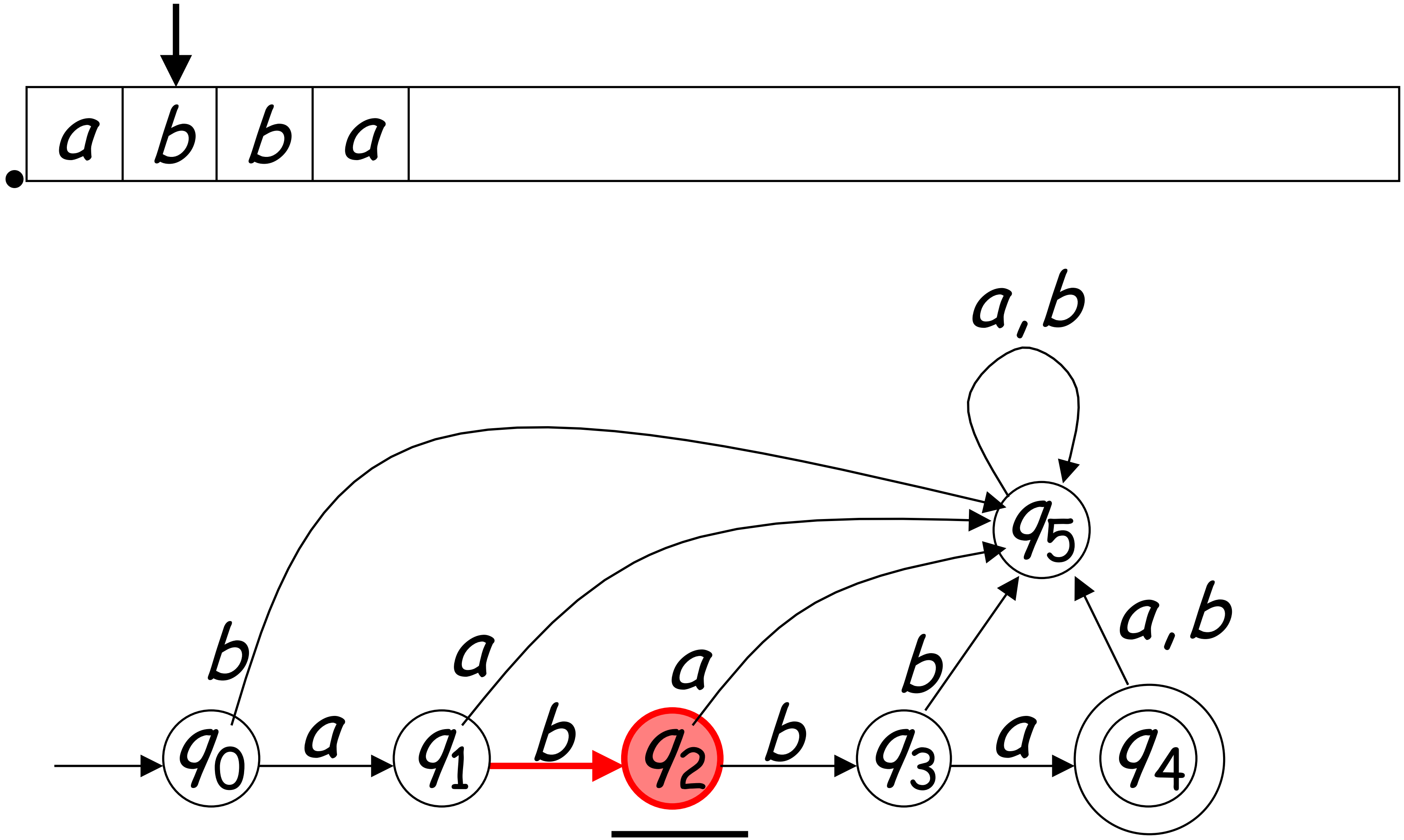


Initial Configuration

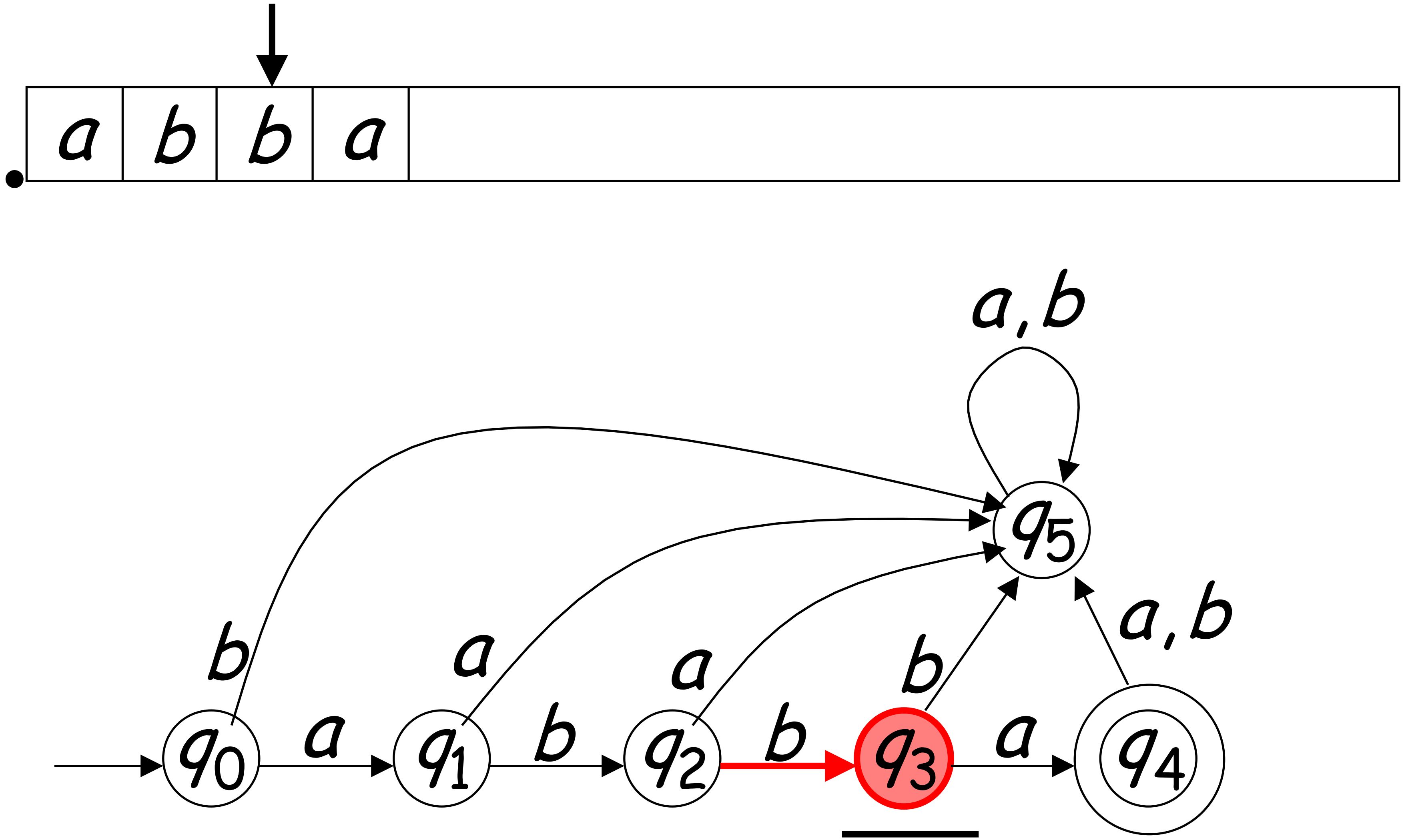


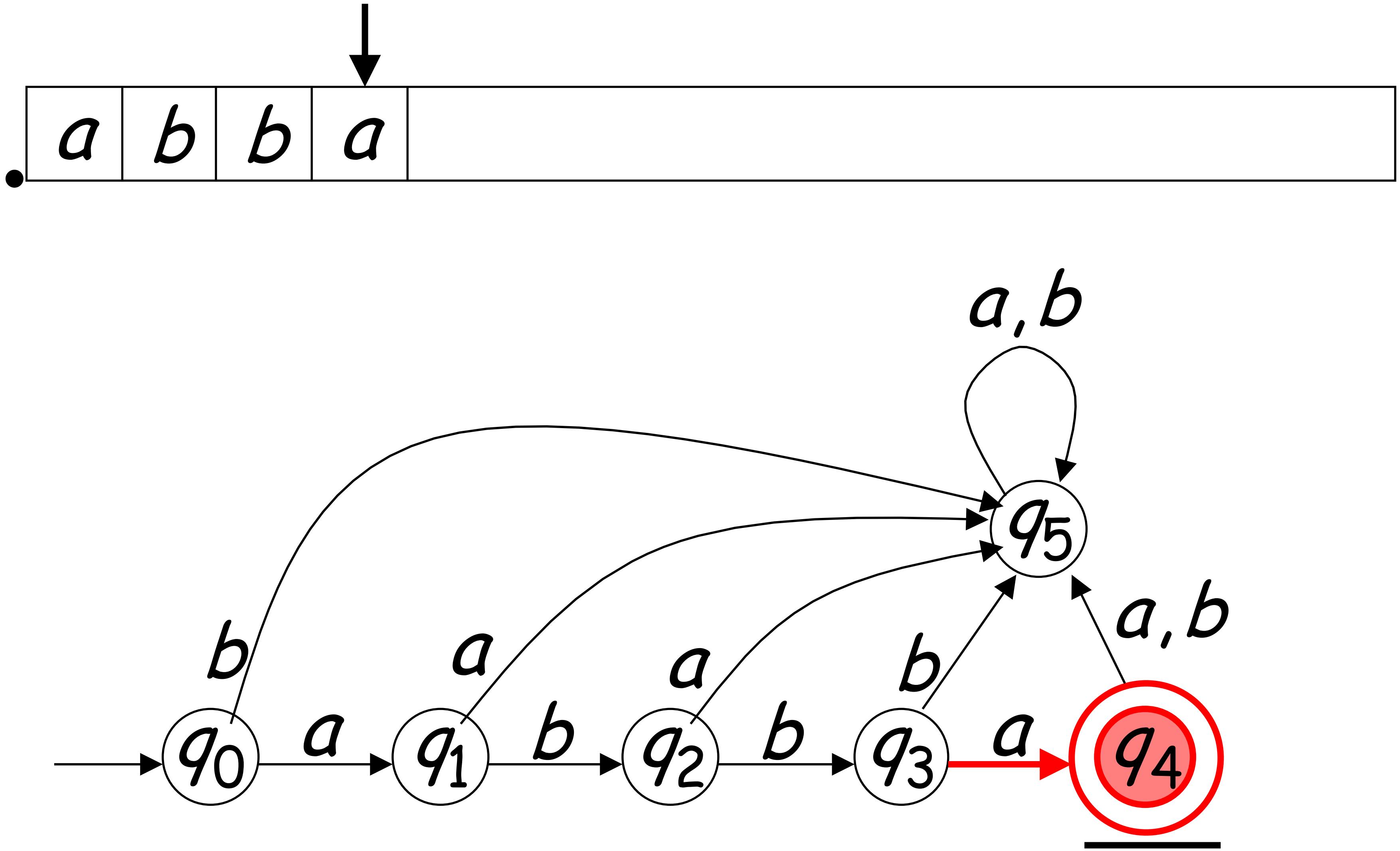
Reading the Input





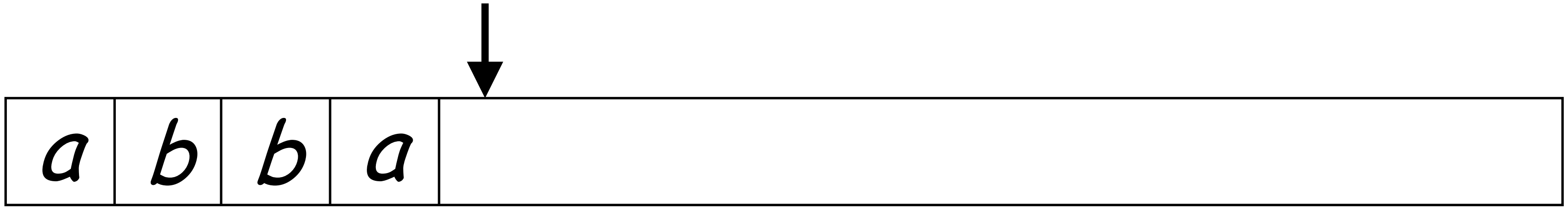
a b b a
a,b
b
a
a
b
a,b
a
b
b
a
q0
q1
q2
q3
q4
q5

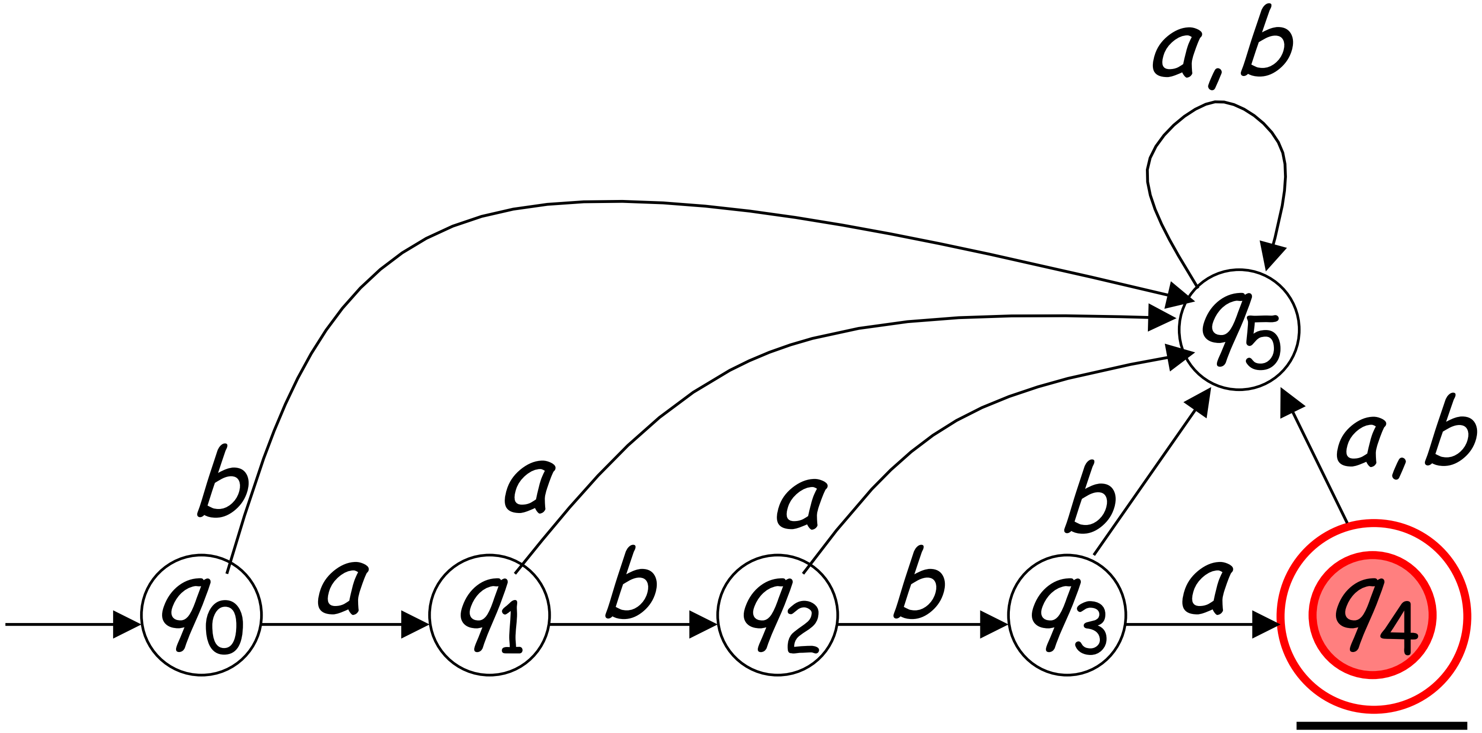

a b b a
a,b
q_5
a,b
b
a
a
b
q_0
a
q_1
b
q_2
b
q_3
a
q_4



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

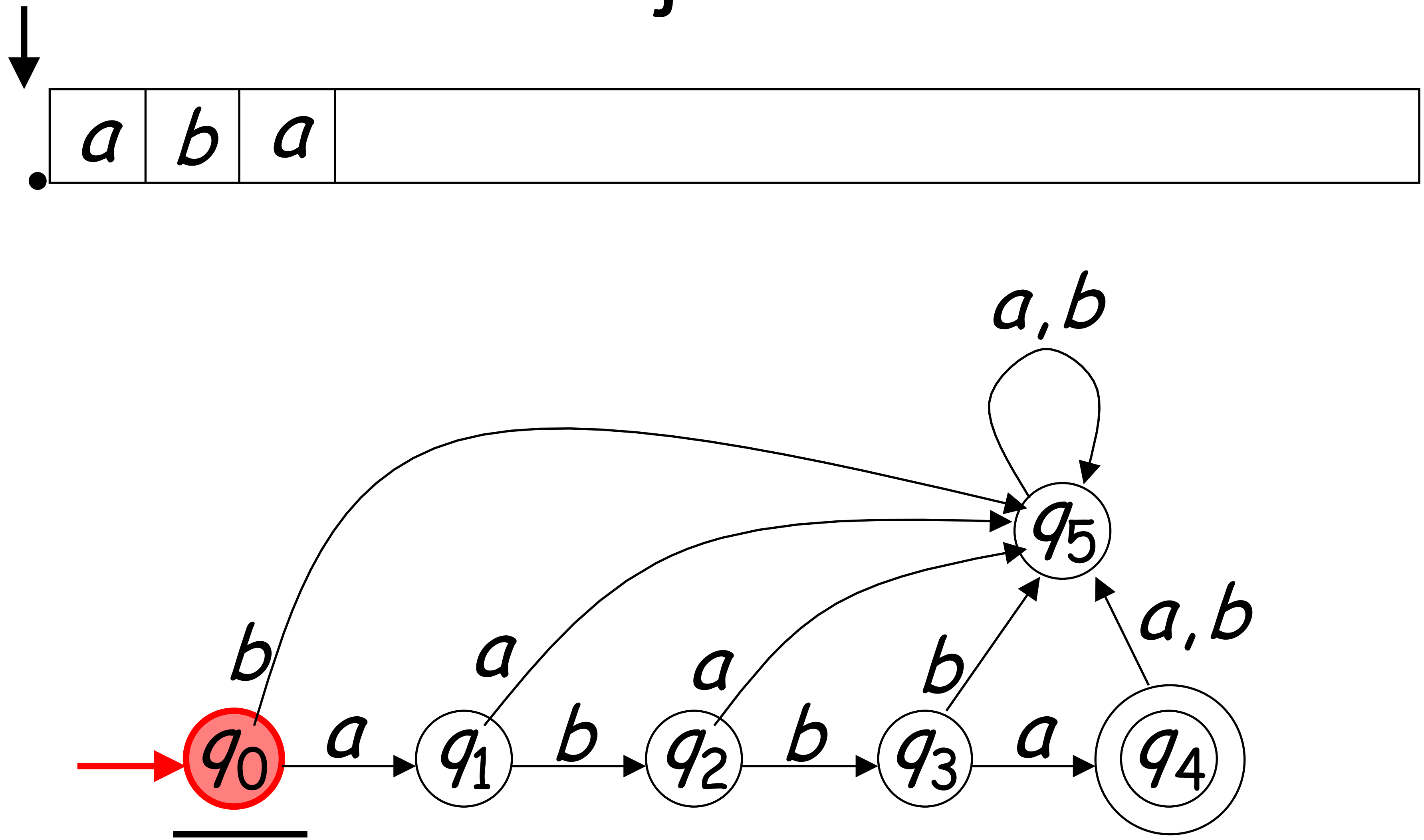
Input finished

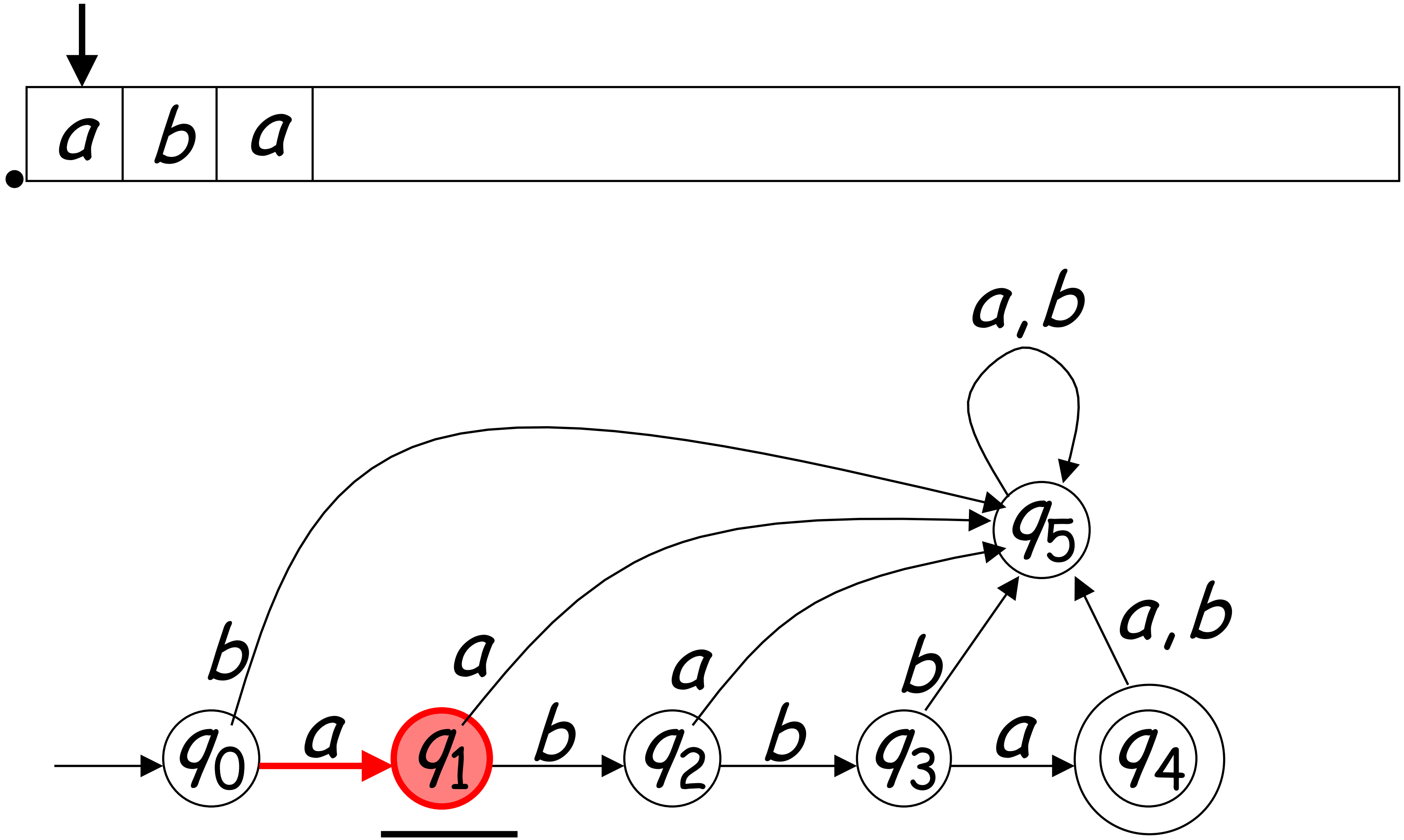


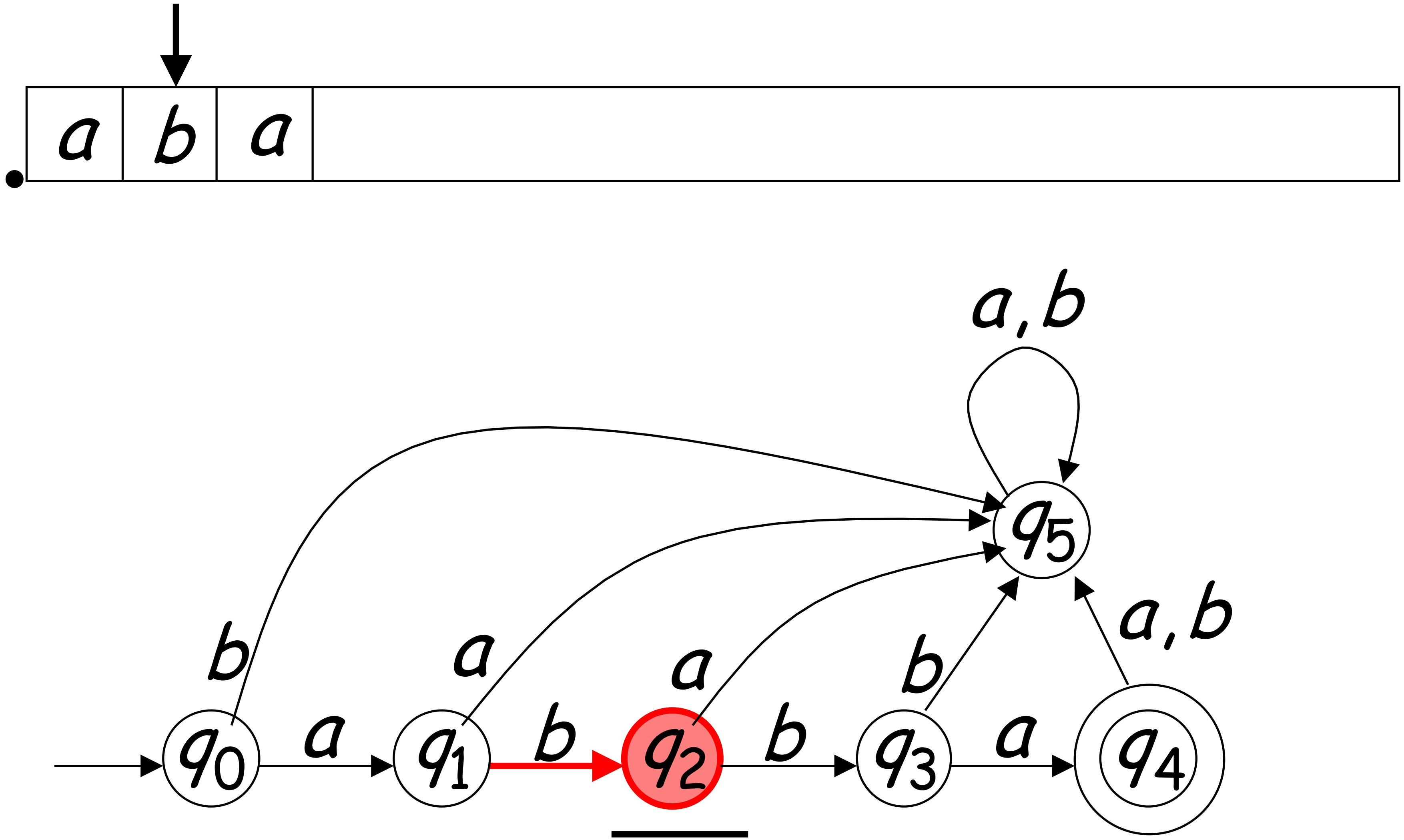


Output: “accept”

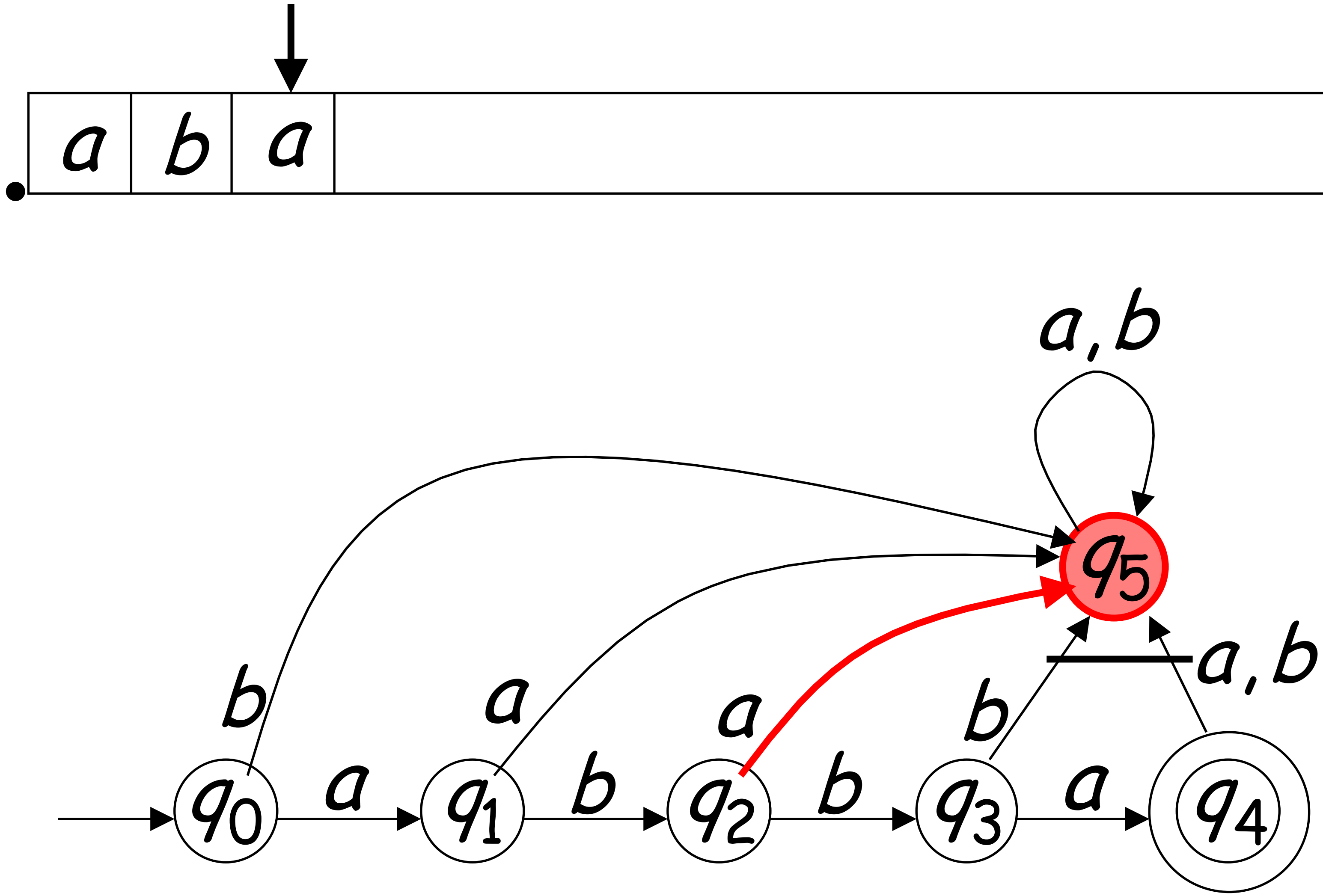
Rejection



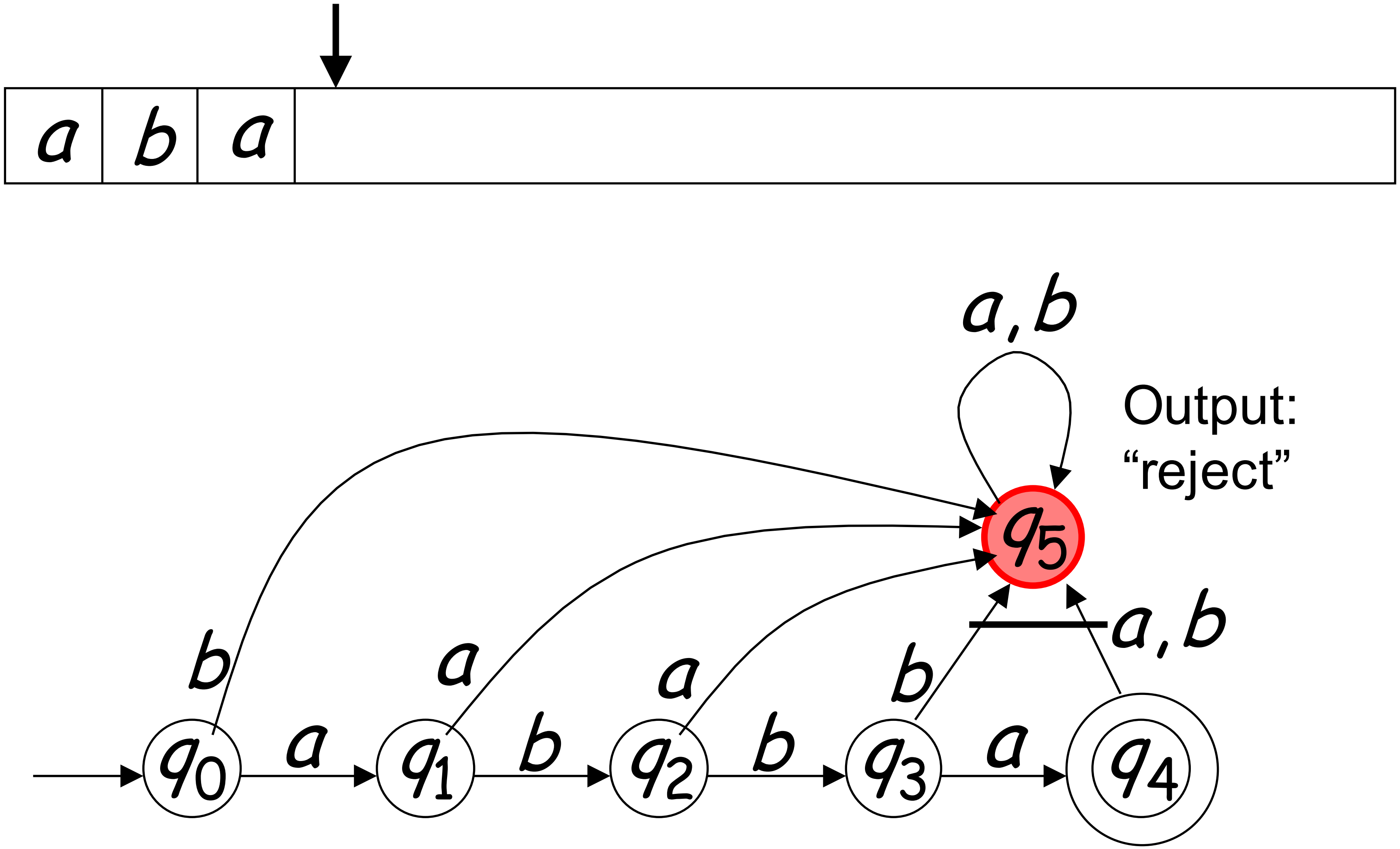

a
b
a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4



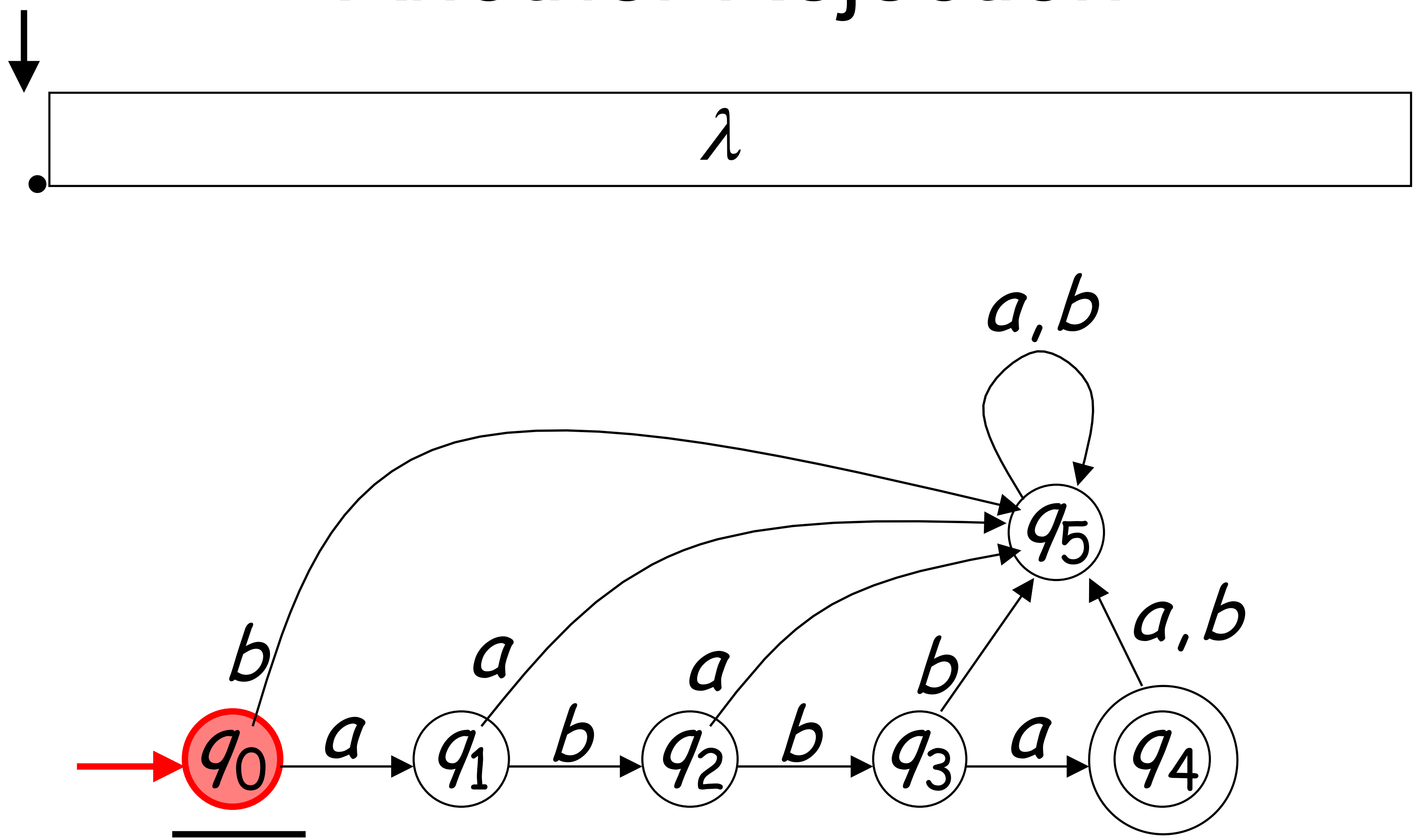
a
b
a
a,b
b
a
a
b
a,b
q_0
a
q_1
b
q_2
b
q_3
a
q_4
q_5

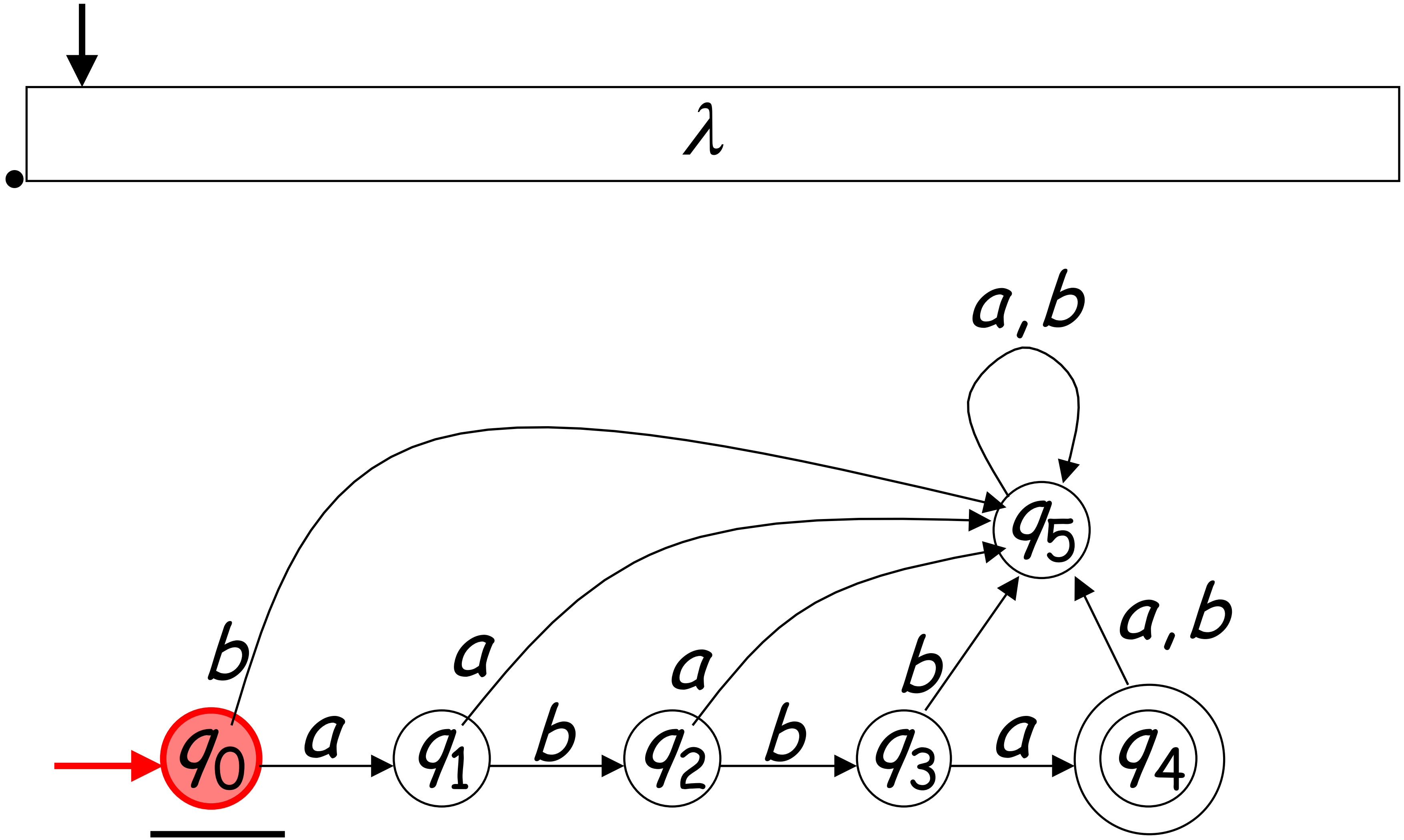

a
b
a
a,b
b
a
a
b
a,b
q5
q0
a
q1
b
q2
b
q3
a
q4

Input finished



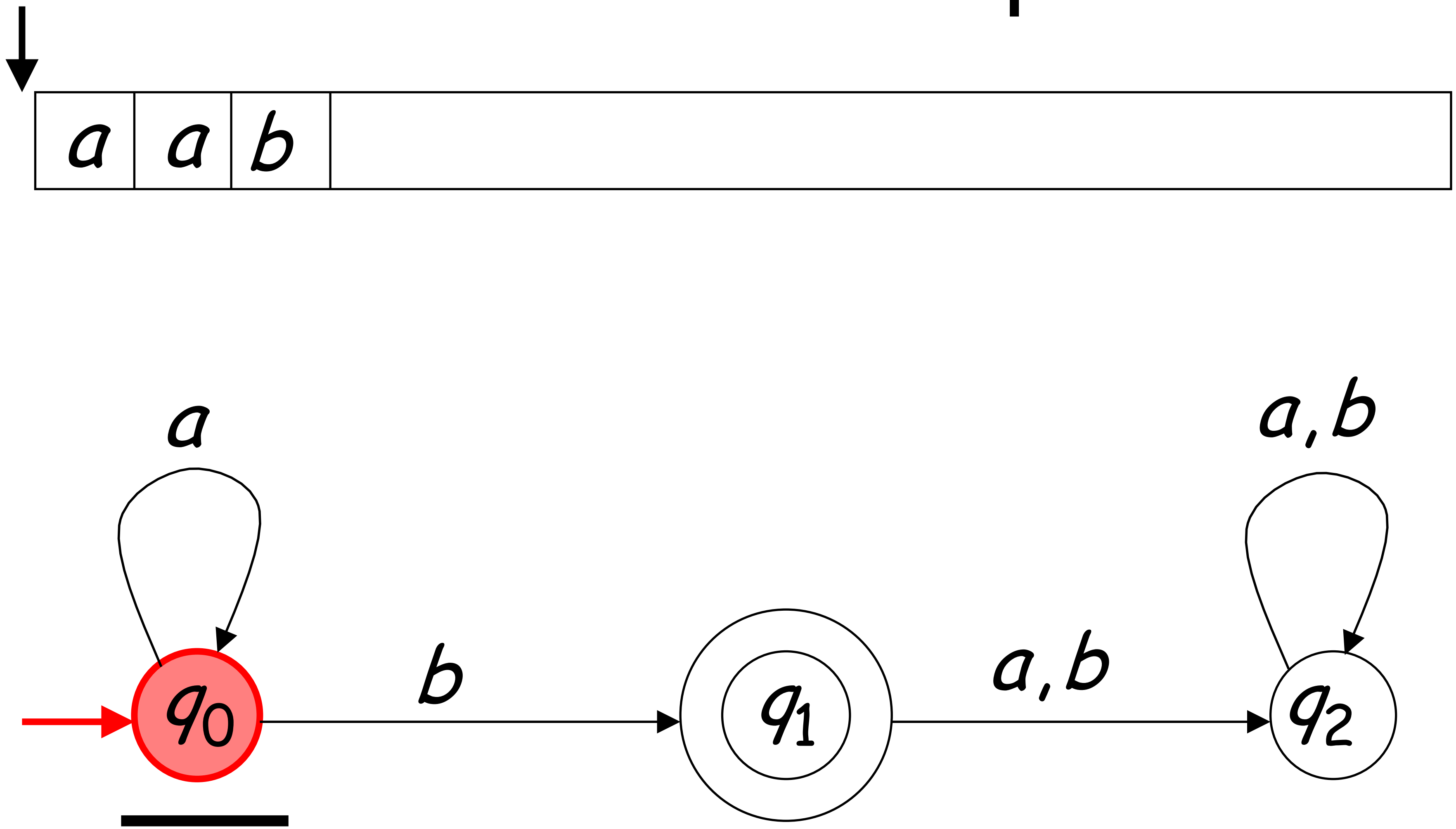
Another Rejection

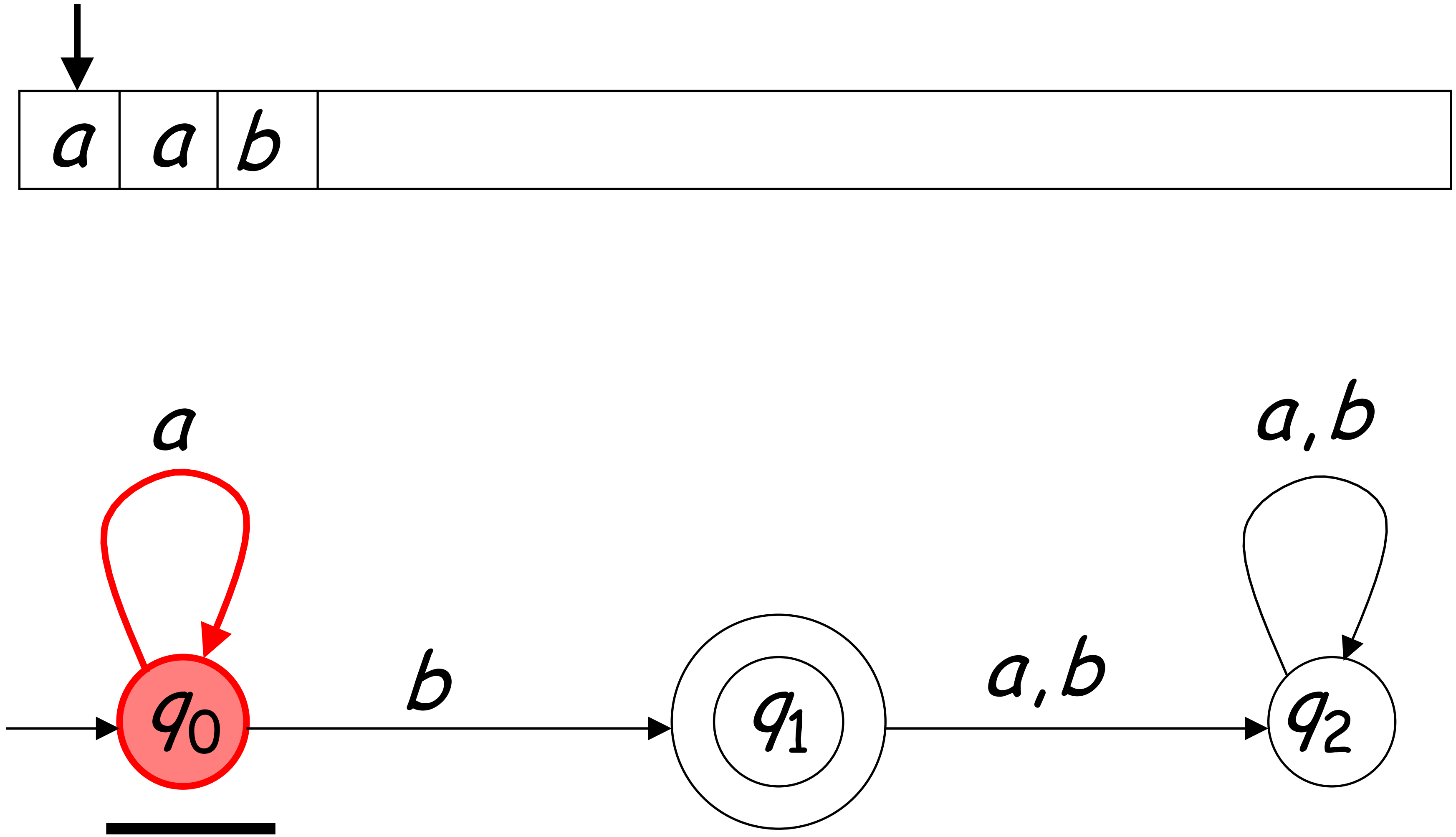


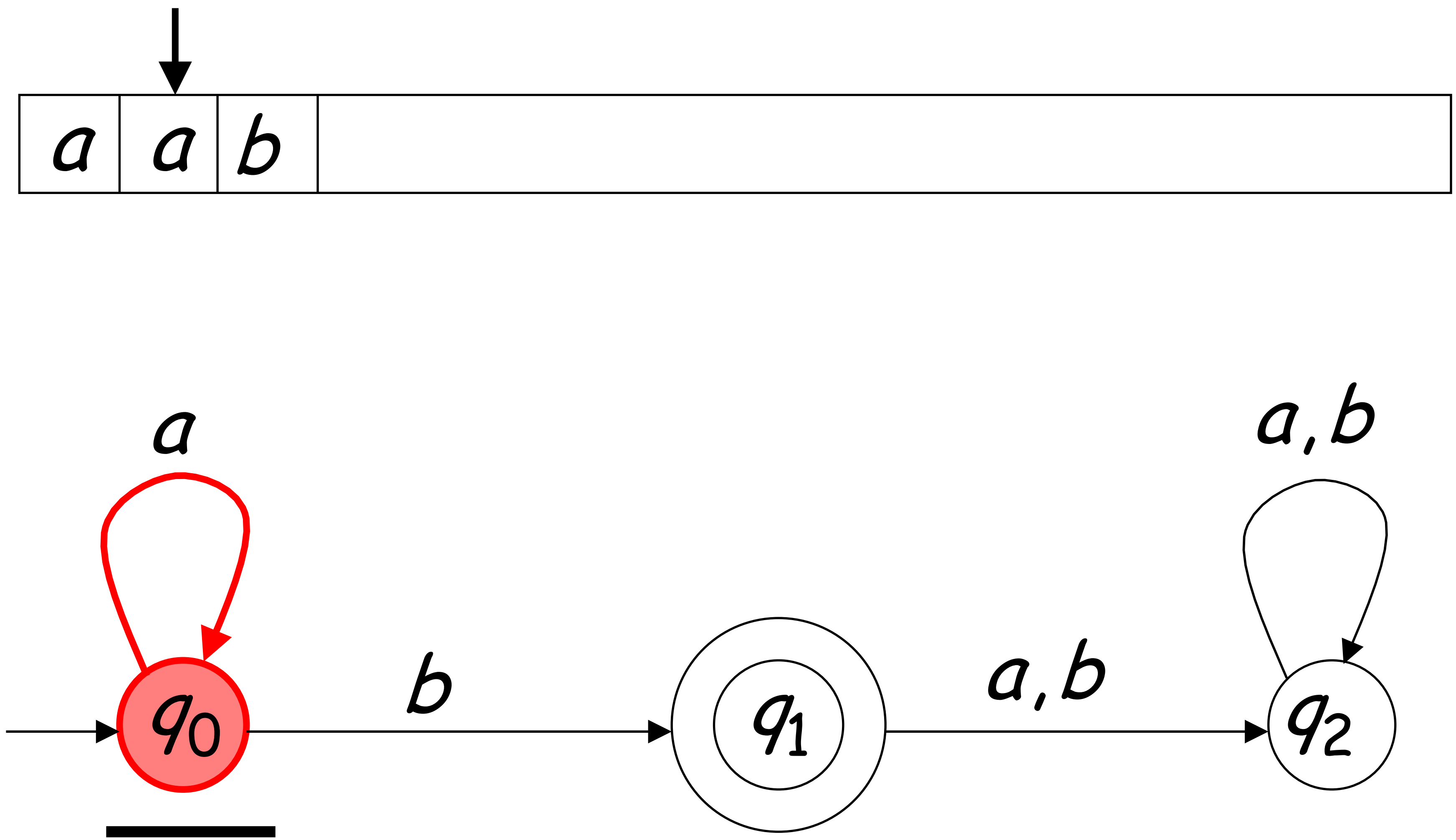


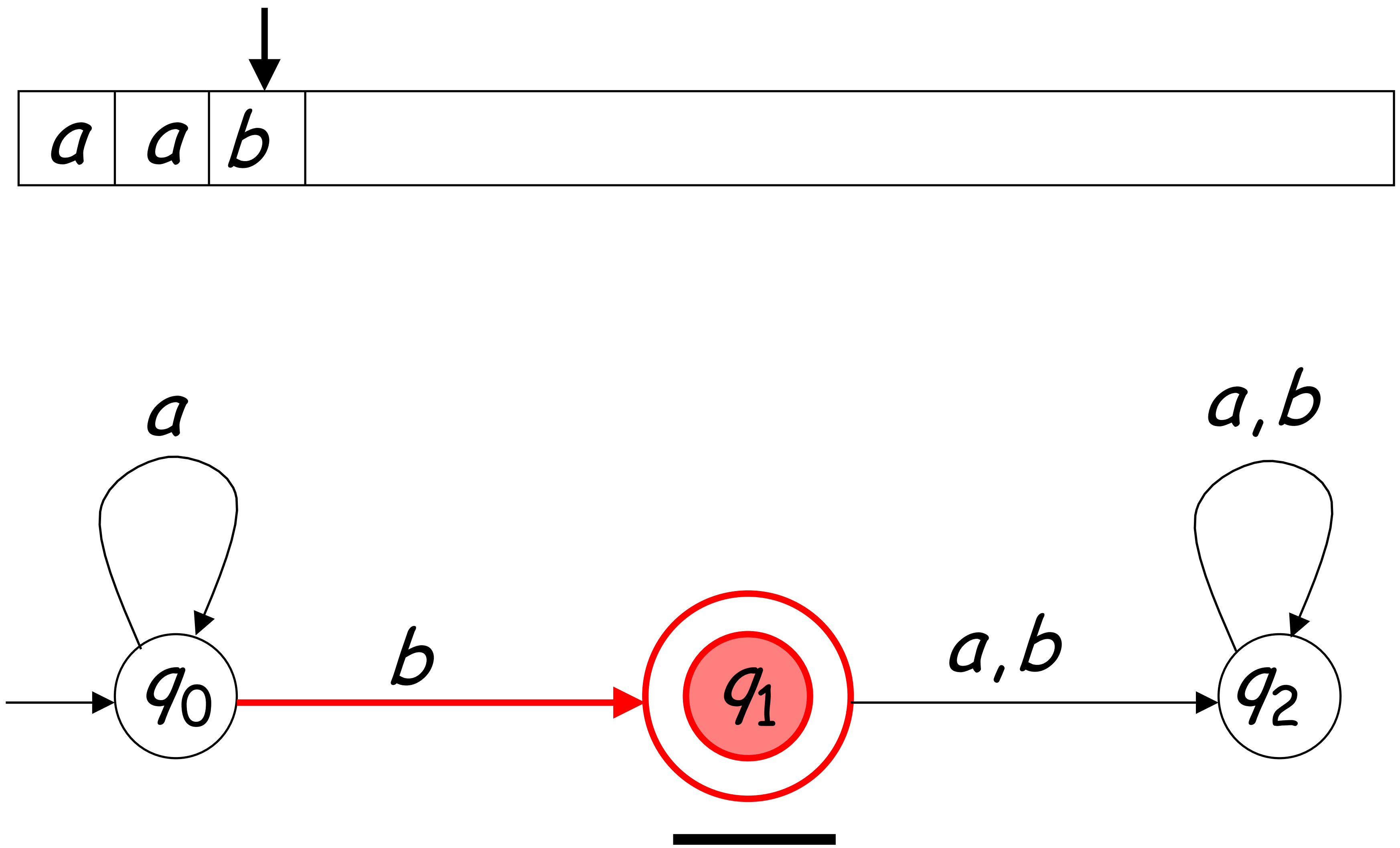
Output:
"reject"

Another Example

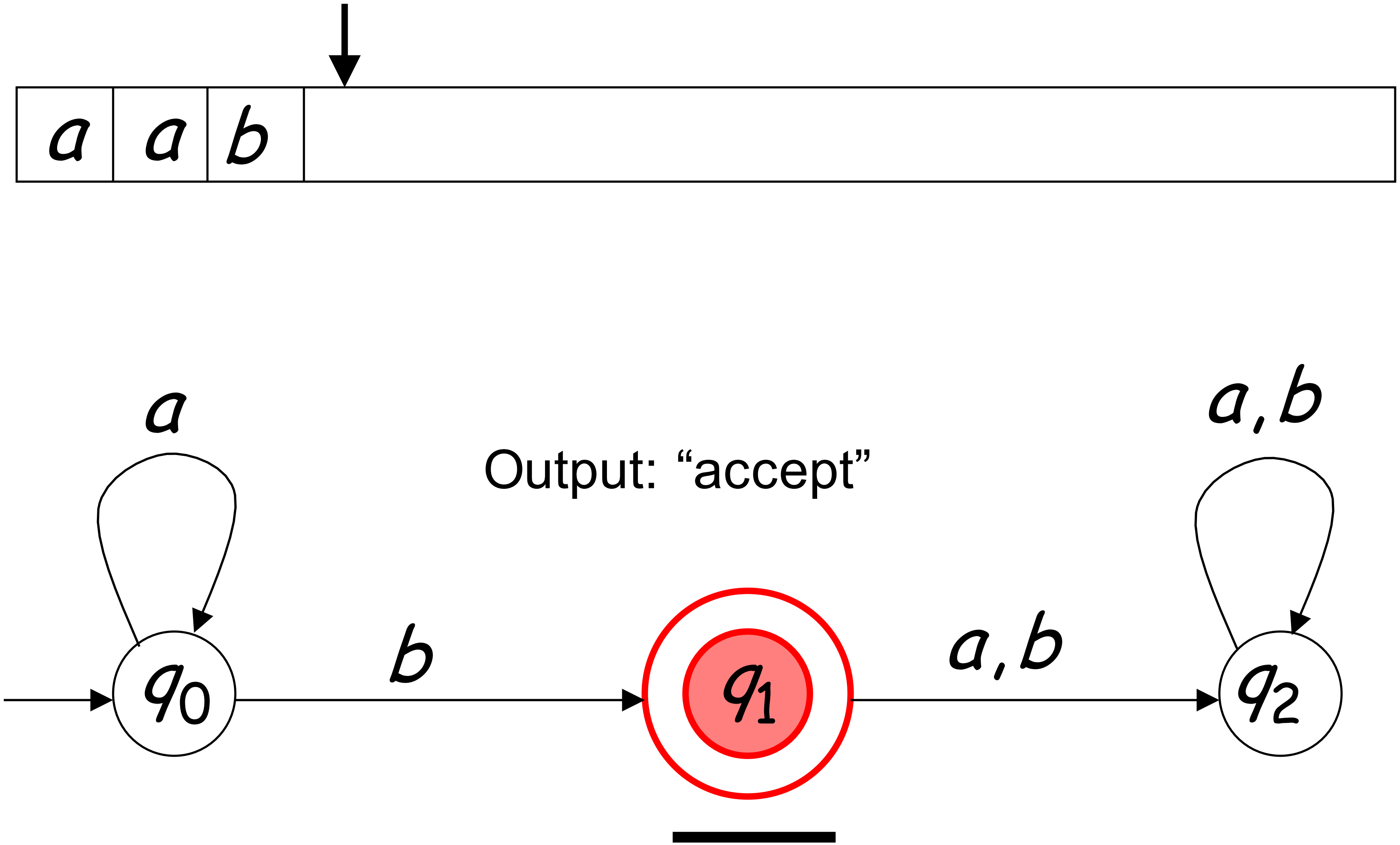



a
a
b
a
a,b
b
a,b
q_0
q_1
q_2


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

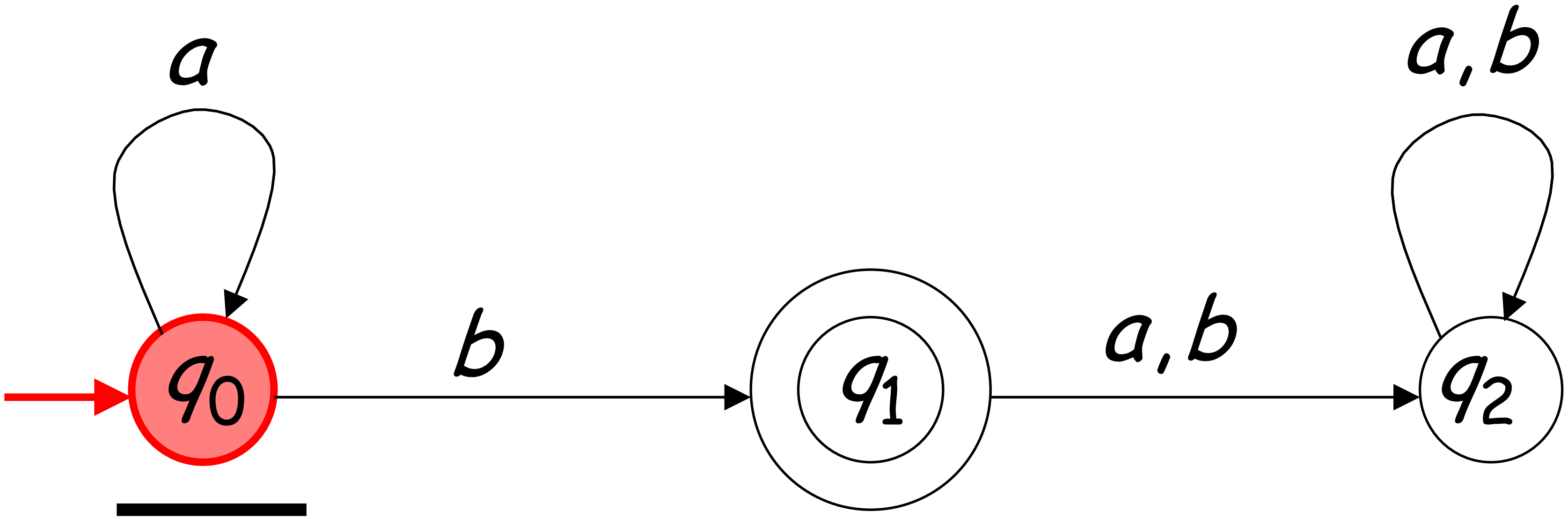

a a b
a
a,b
b
a,b
q_0
q_1
q_2

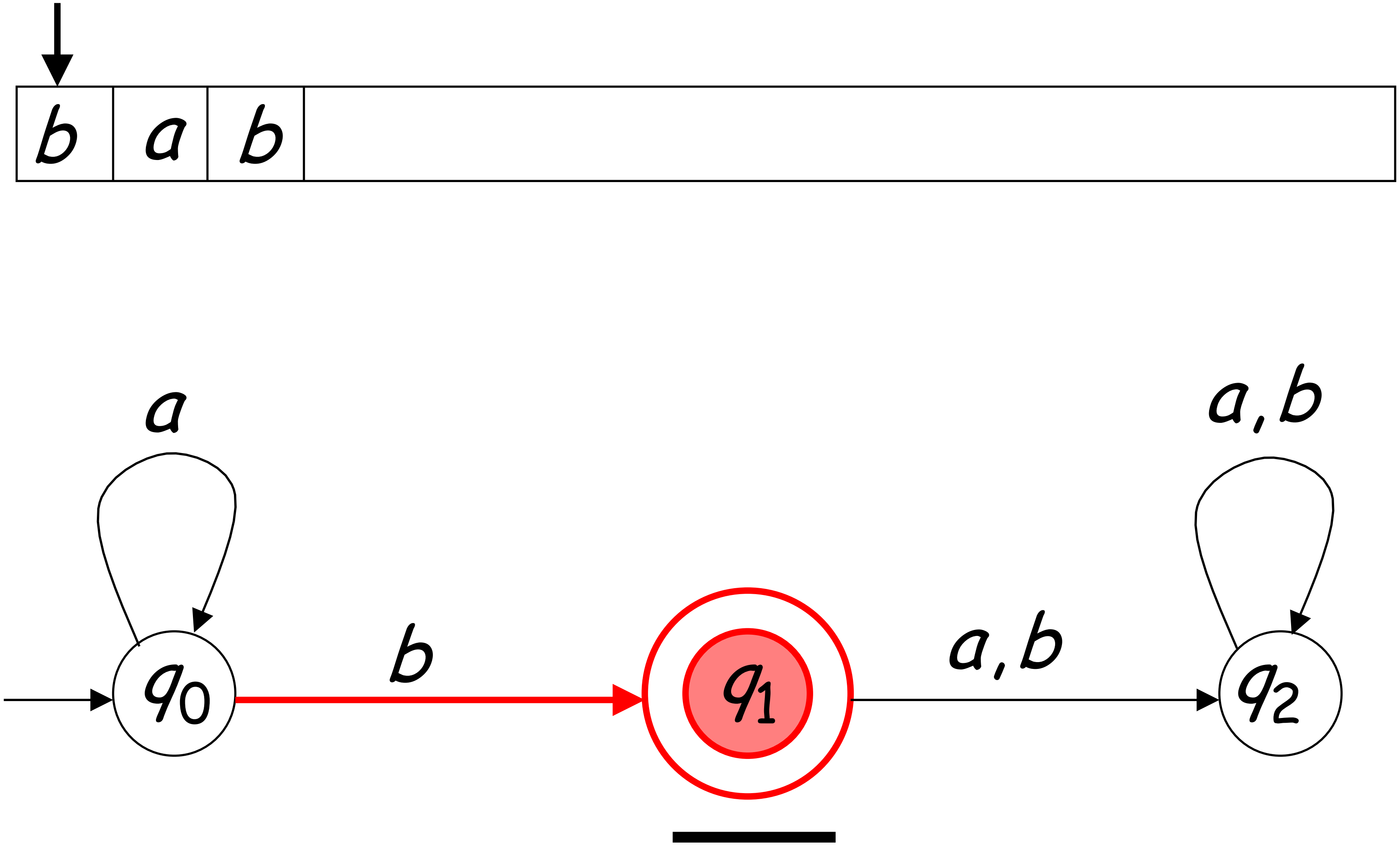
Input finished

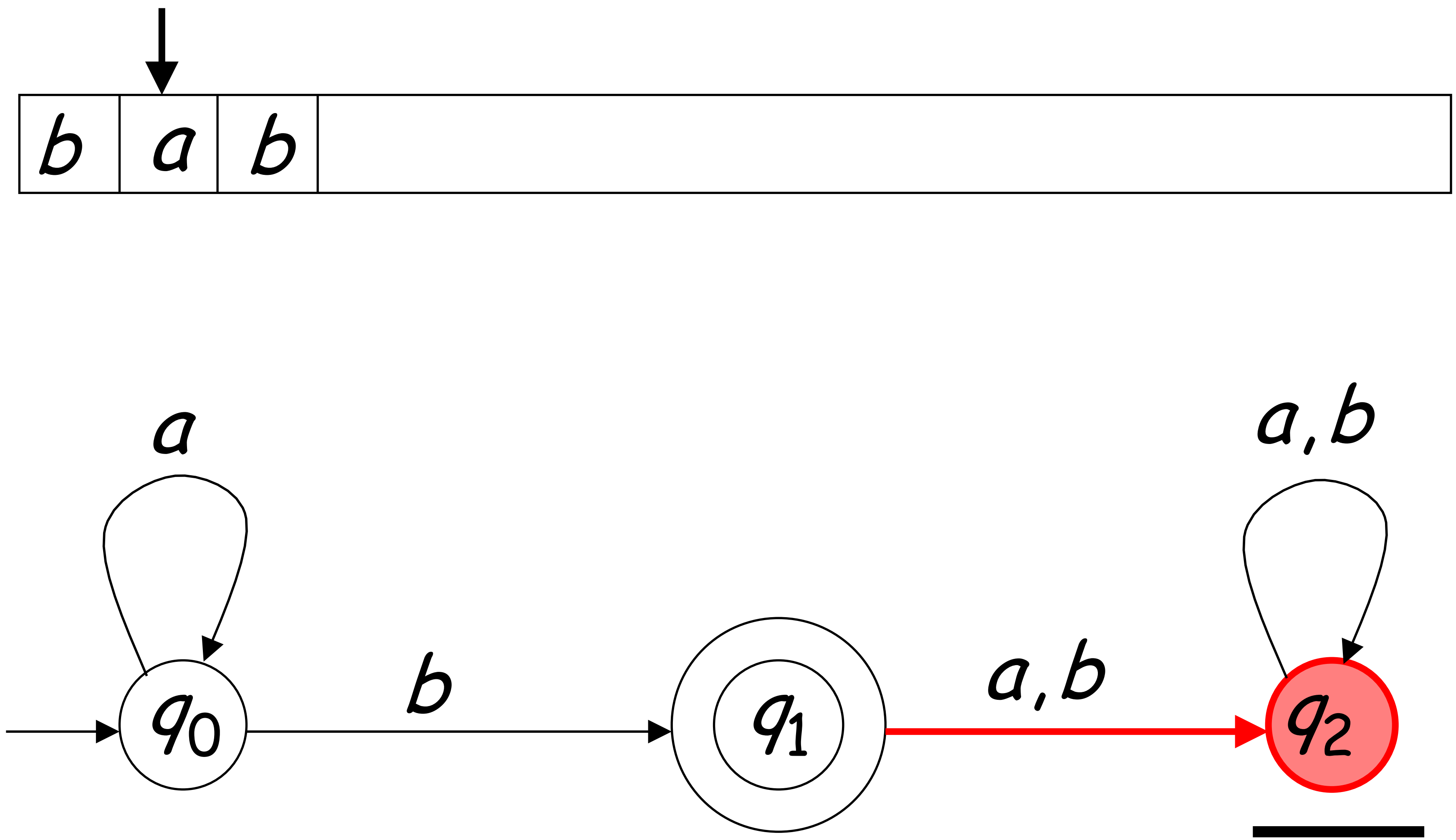


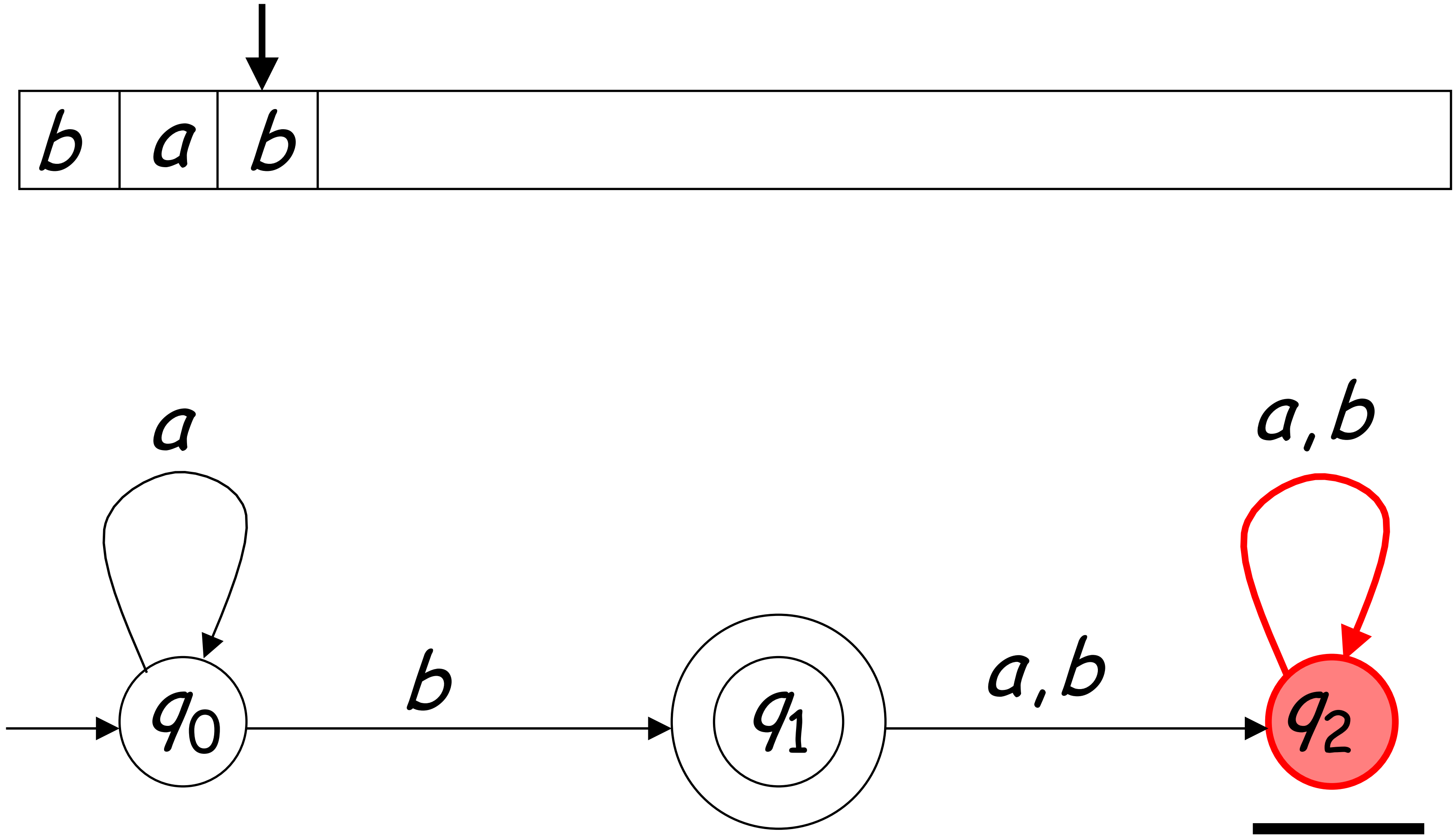
Output: “accept”

Rejection

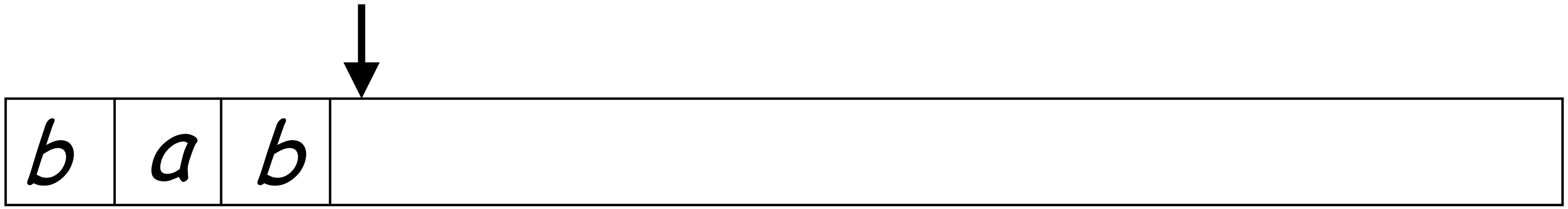


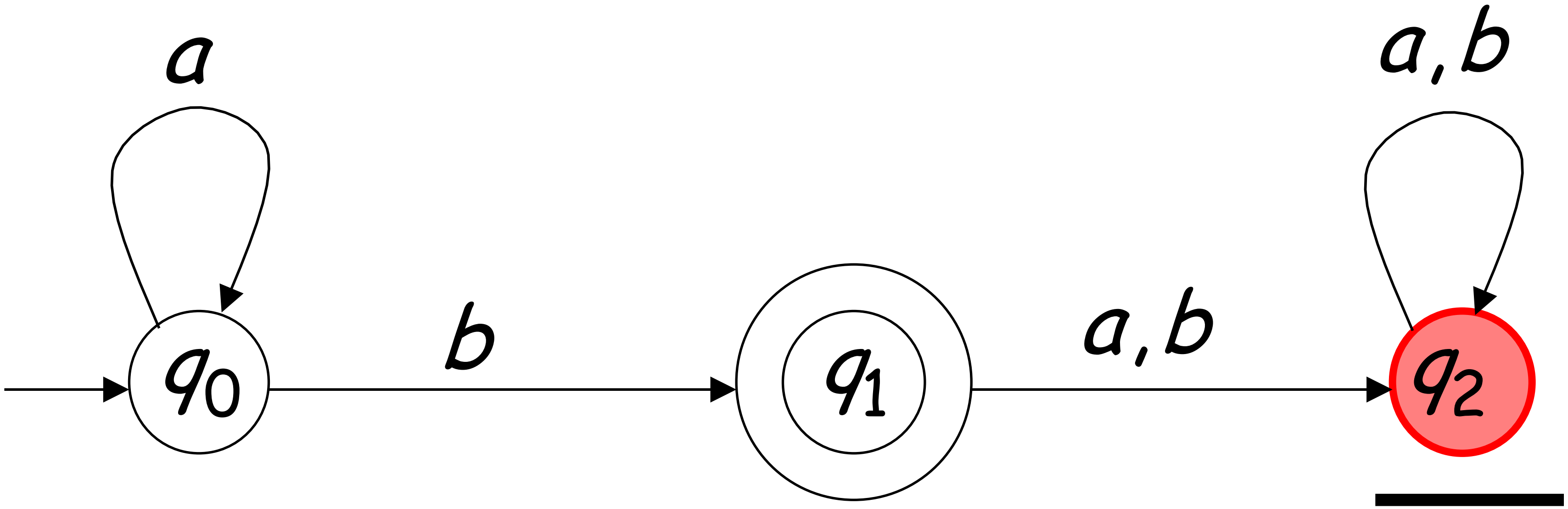

b a b
a
a,b
b
a,b
q_0
q_1
q_2


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


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

Input finished





Output: “reject”

Trap state

Definition 2.1

Deterministic Finite Accepter (DFA) is define by the 5-tuple

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

Q : a finite set of **internal states**

Σ : a finite set of symbols called **input alphabet**

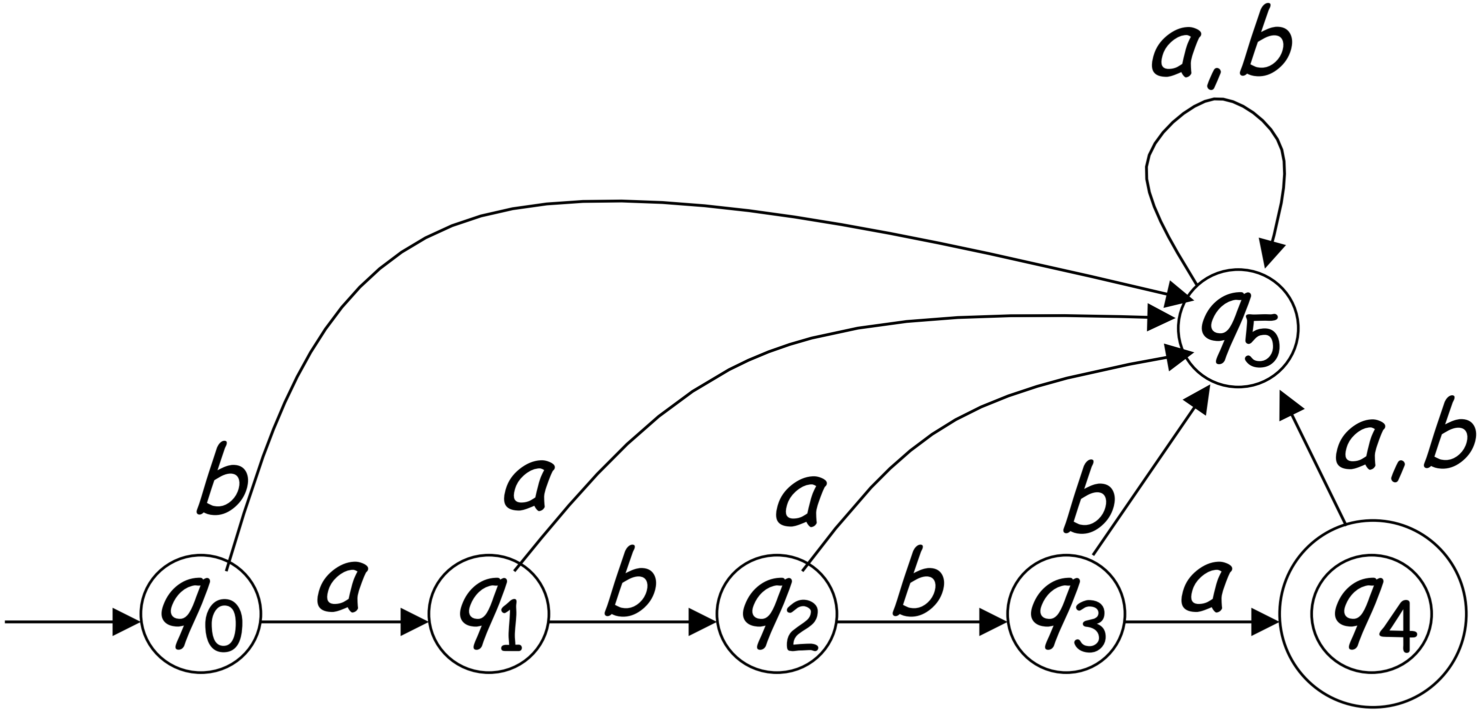
δ : $Q \times \Sigma \rightarrow Q$ called **transition function** (Total function)

q_0 : $q_0 \in Q$ is the **initial state**

F : $F \subseteq Q$ is a set of **final states**

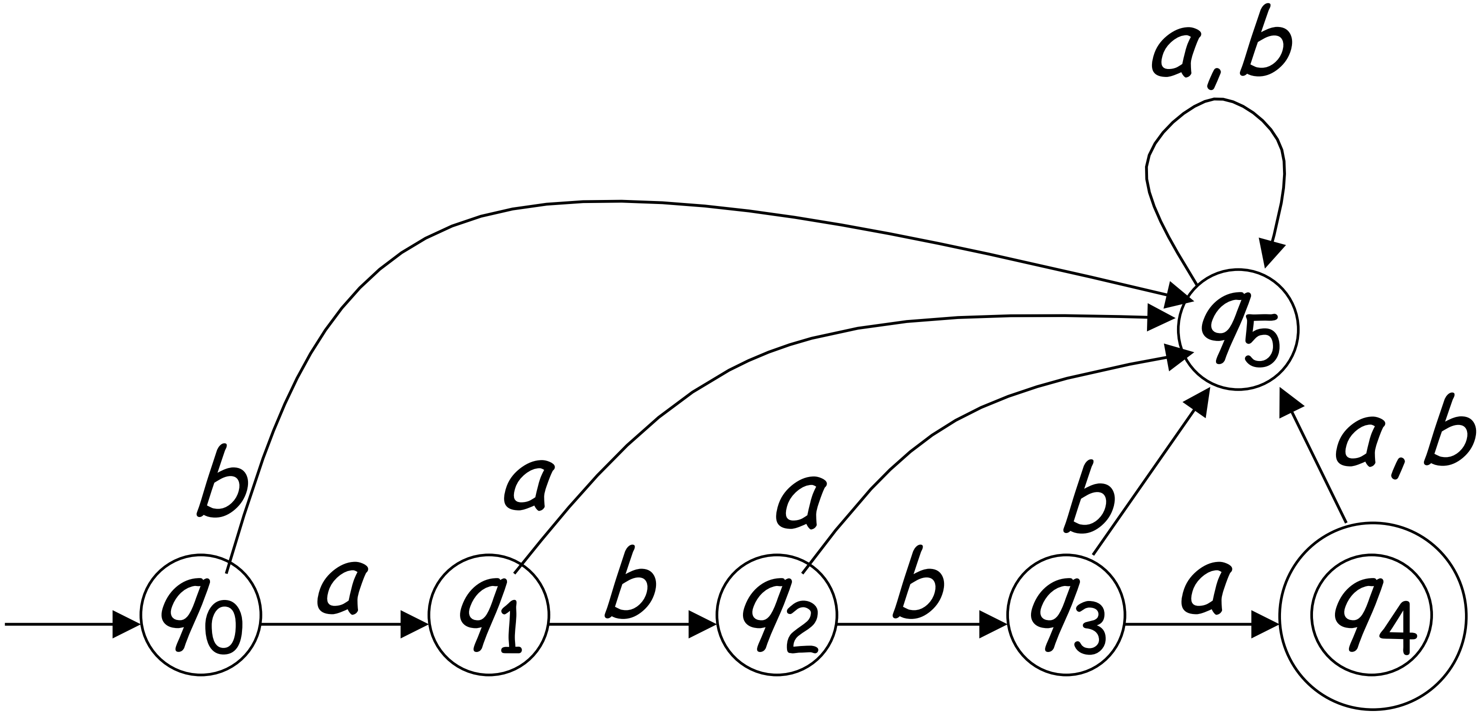
Input Alphabet Σ

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

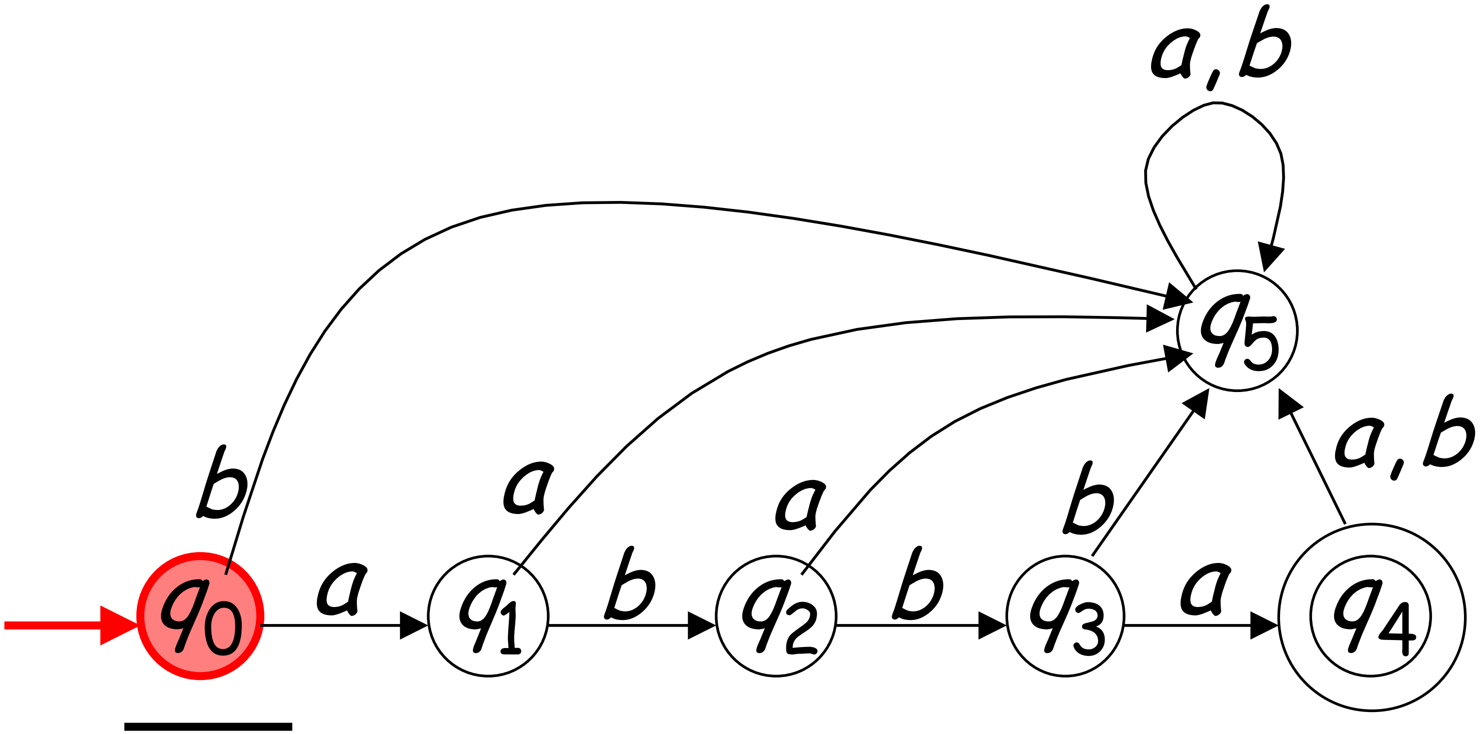


Set of States Q

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

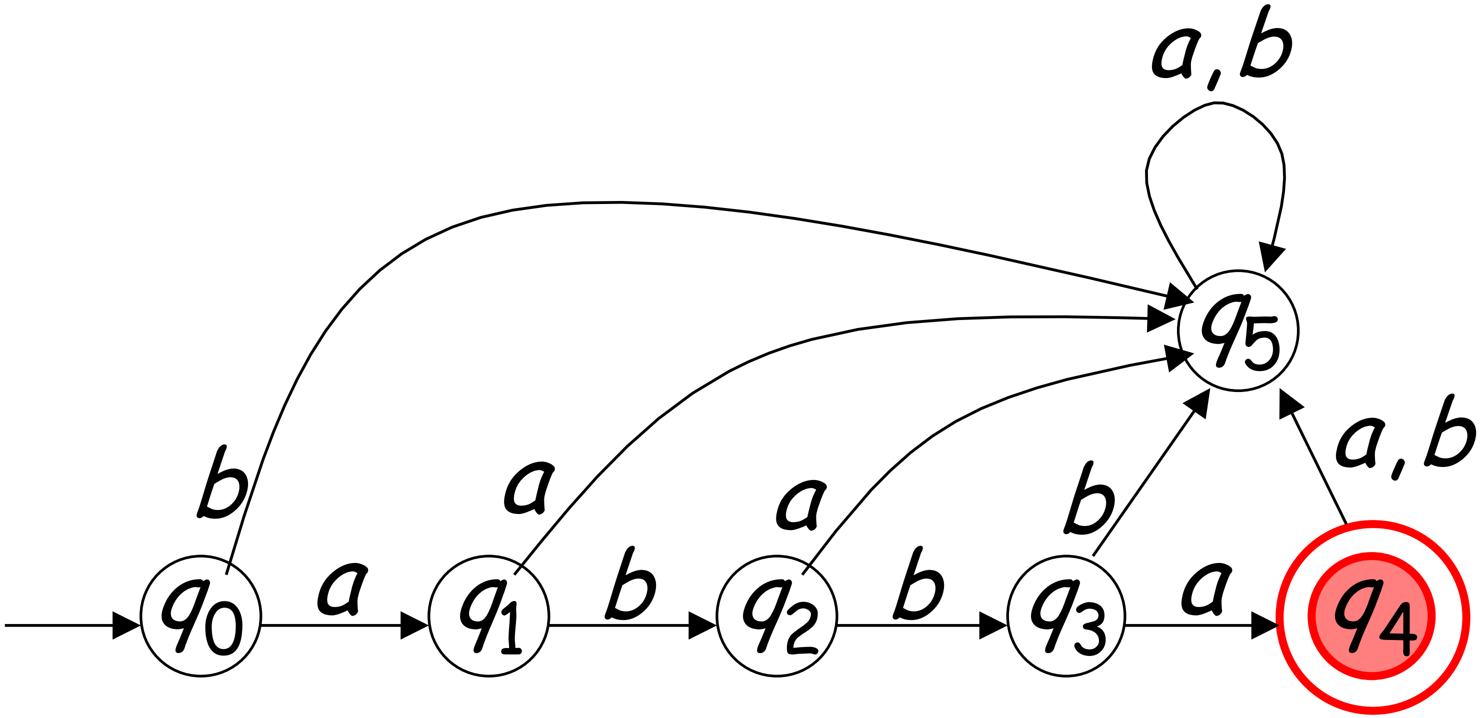


Initial State q_0



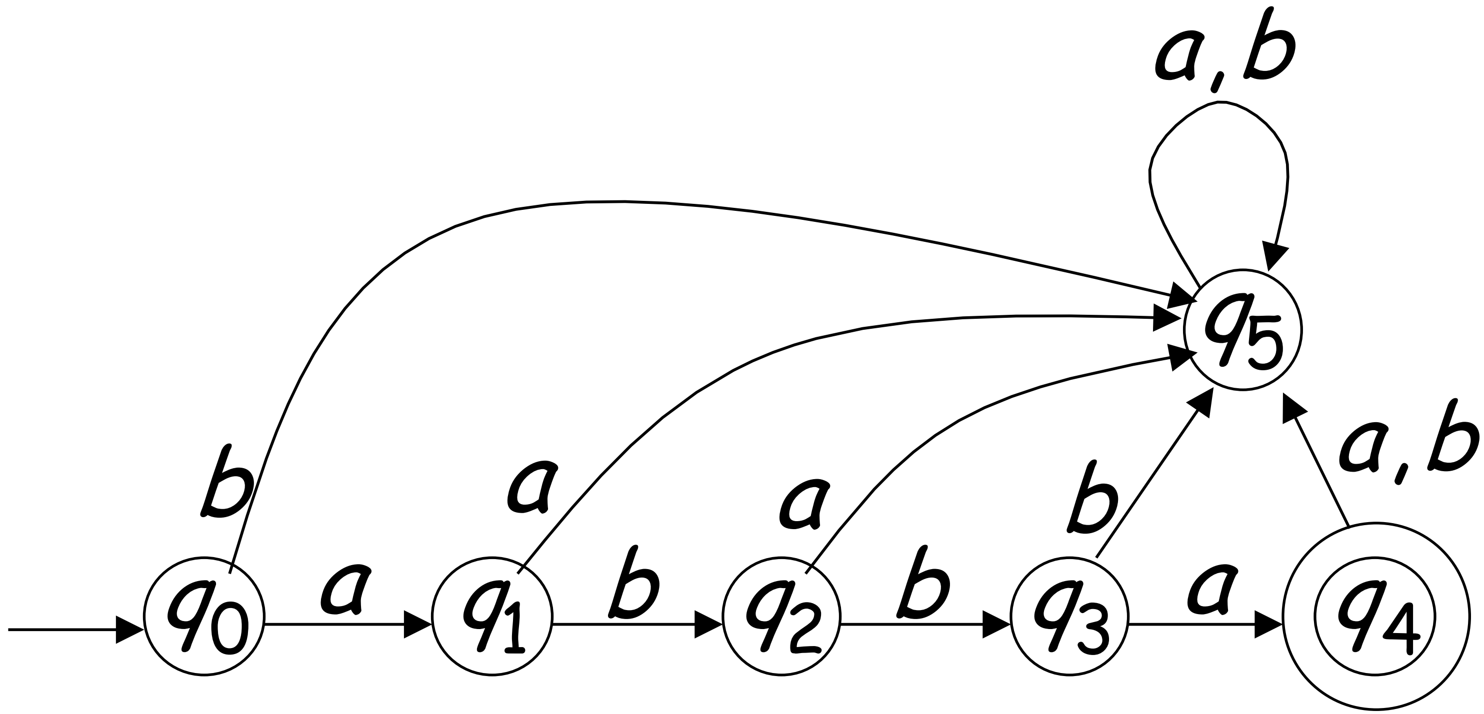
Set of Final States F

$$F = \{q_4\}$$

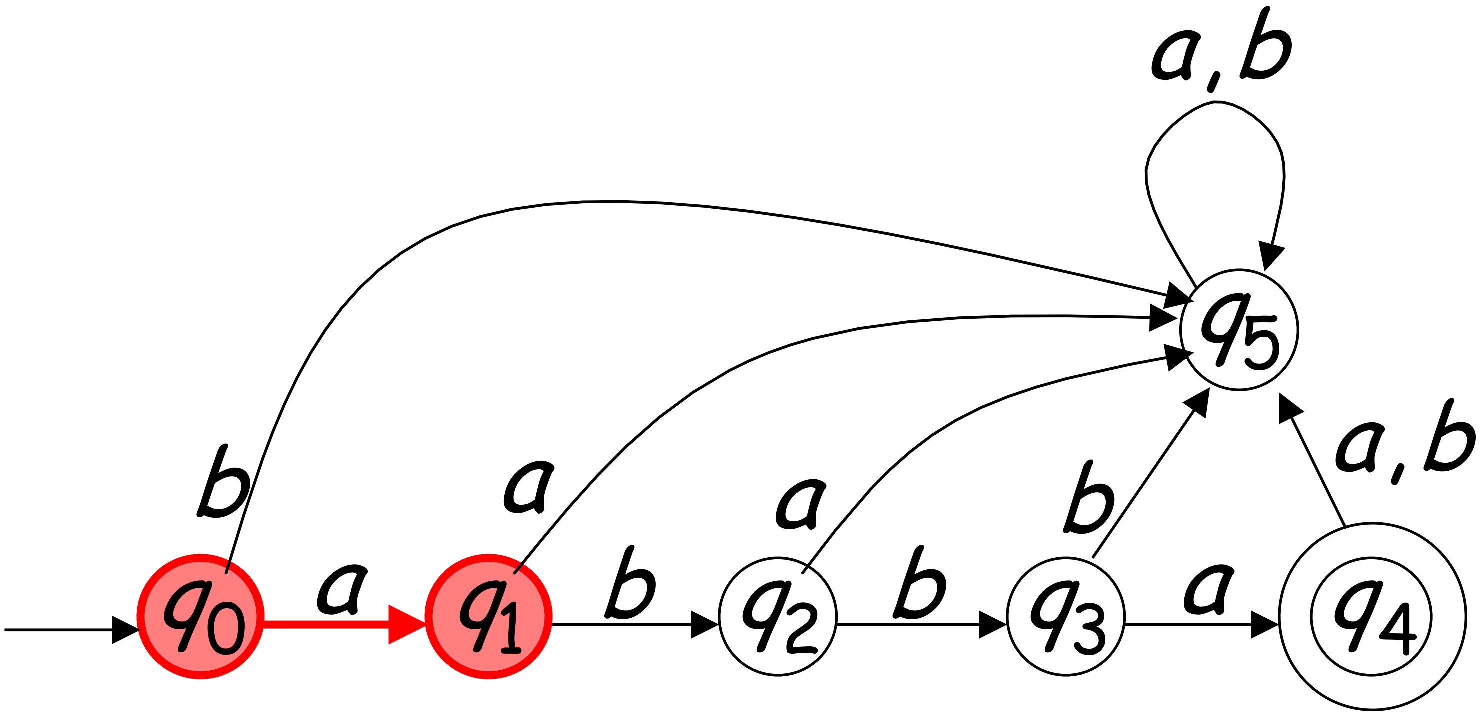


Transition Function δ

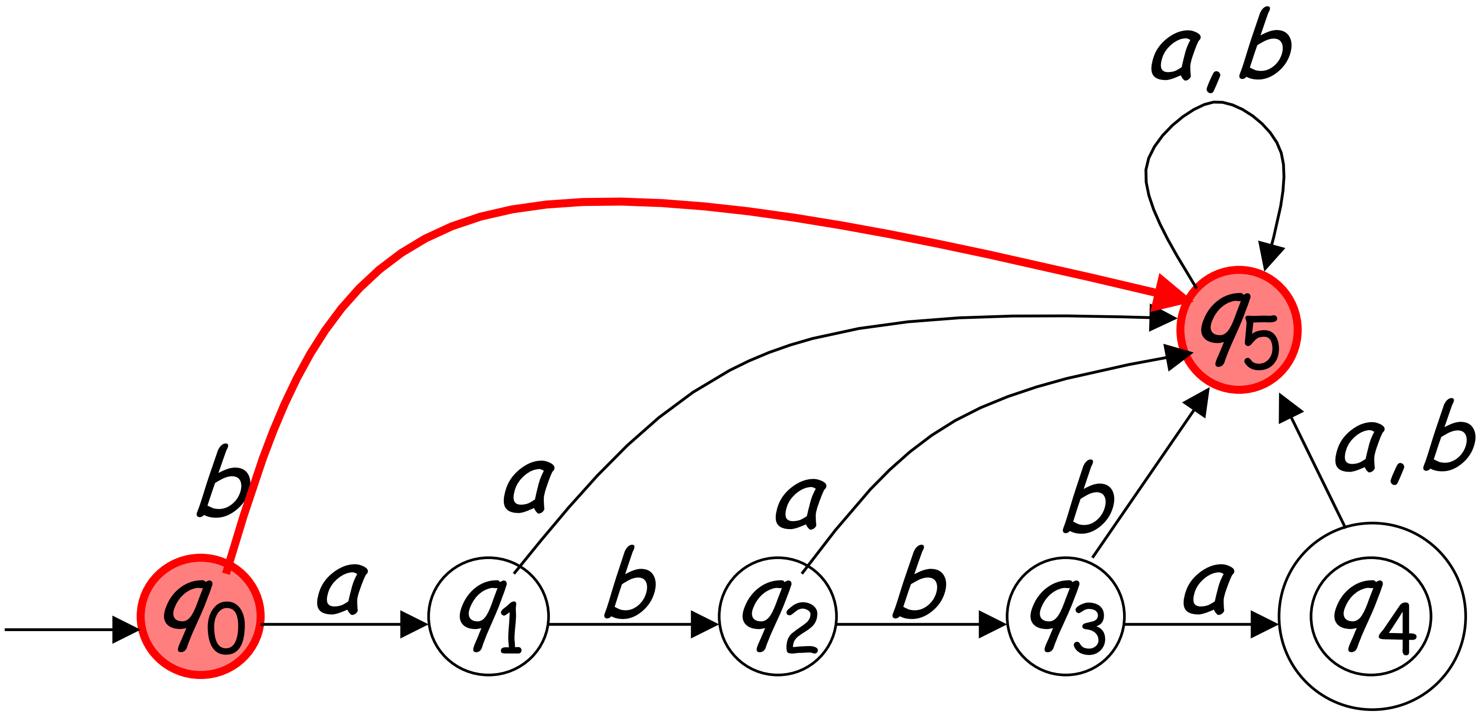
$$\delta : Q \times \Sigma \rightarrow Q$$



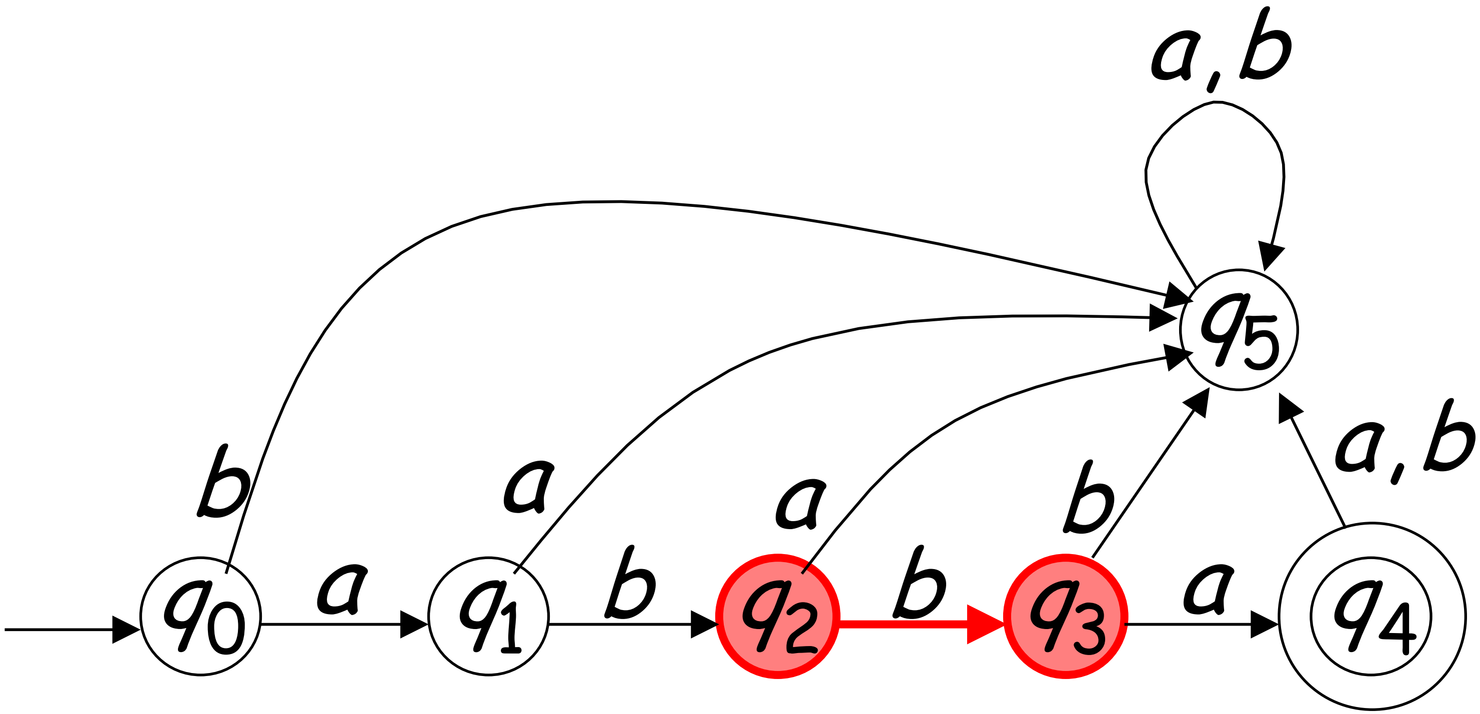
$$\delta(q_0, a) = q_1$$



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

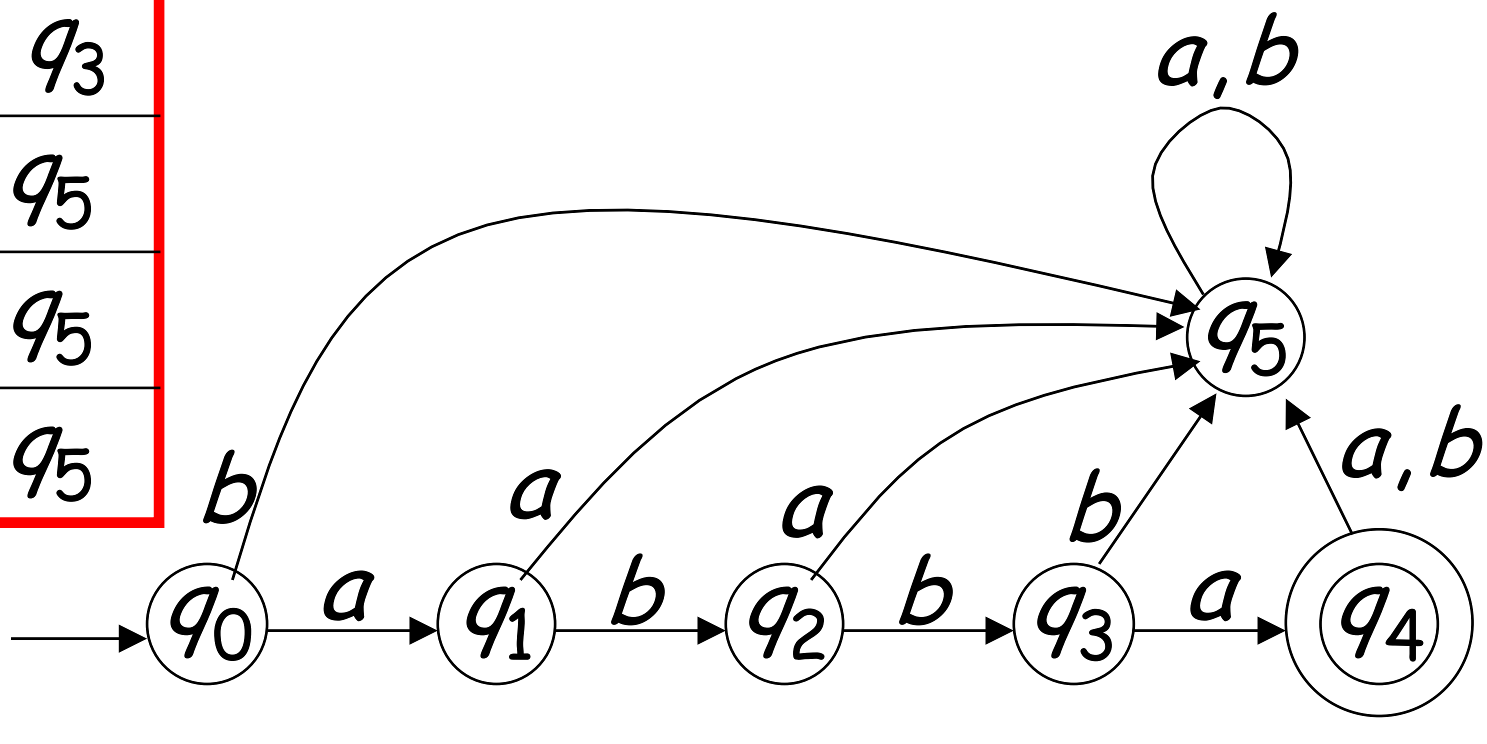


$$\delta(q_2, b) = q_3$$



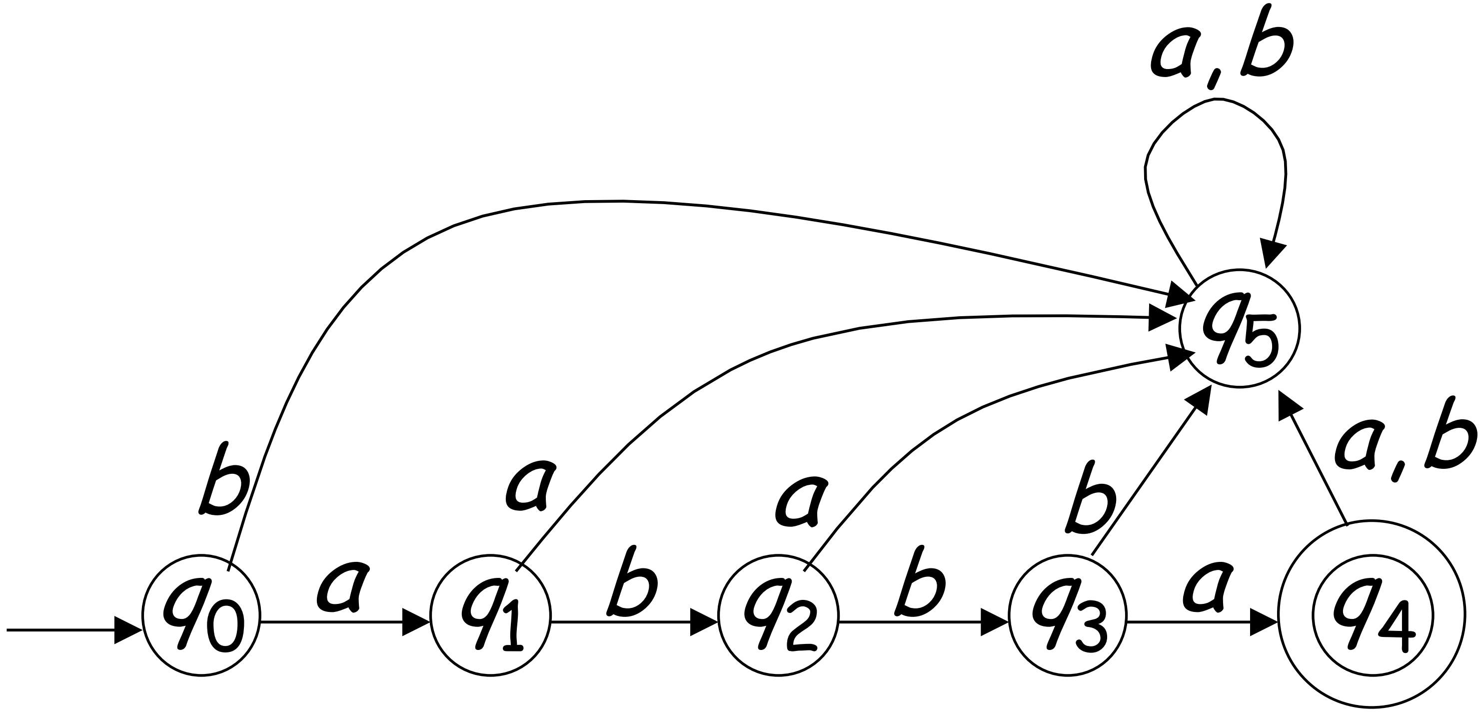
Transition Function δ

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

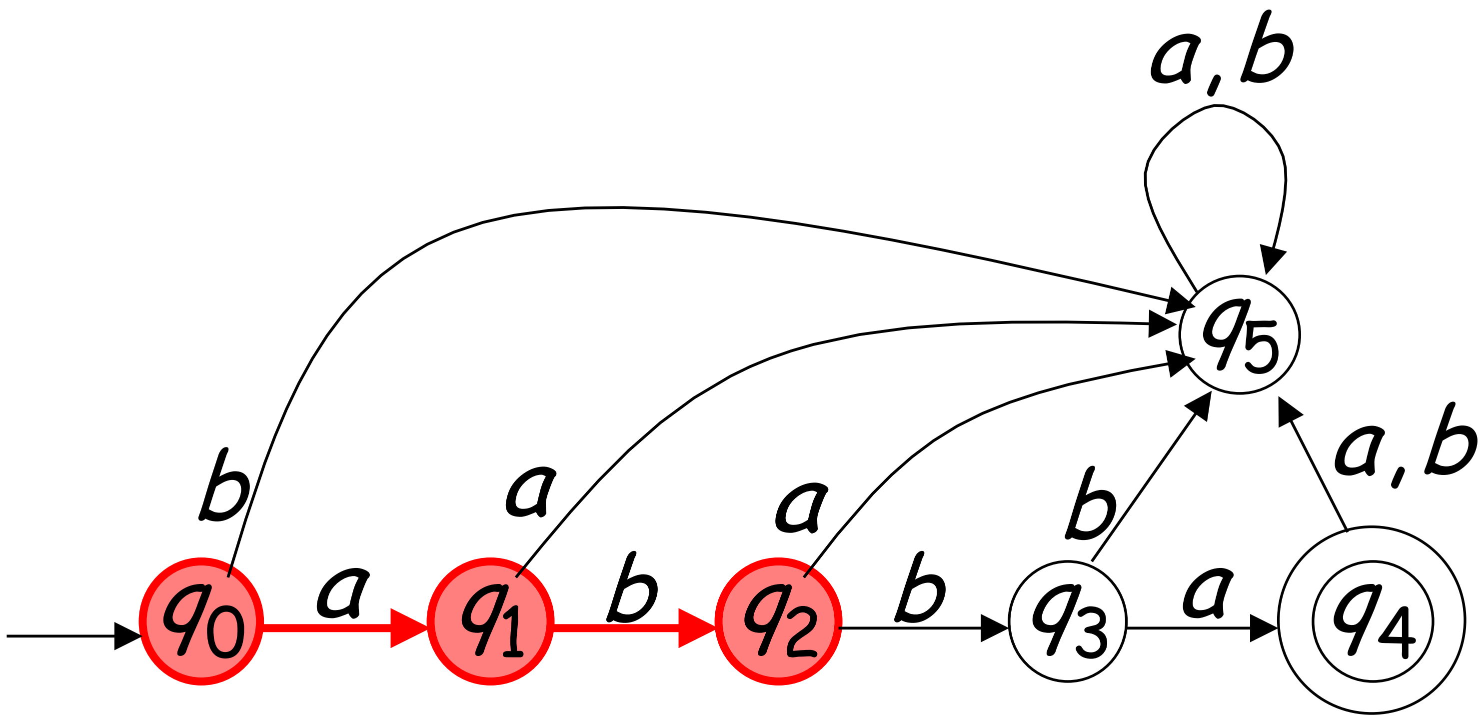


Extended Transition Function δ^*

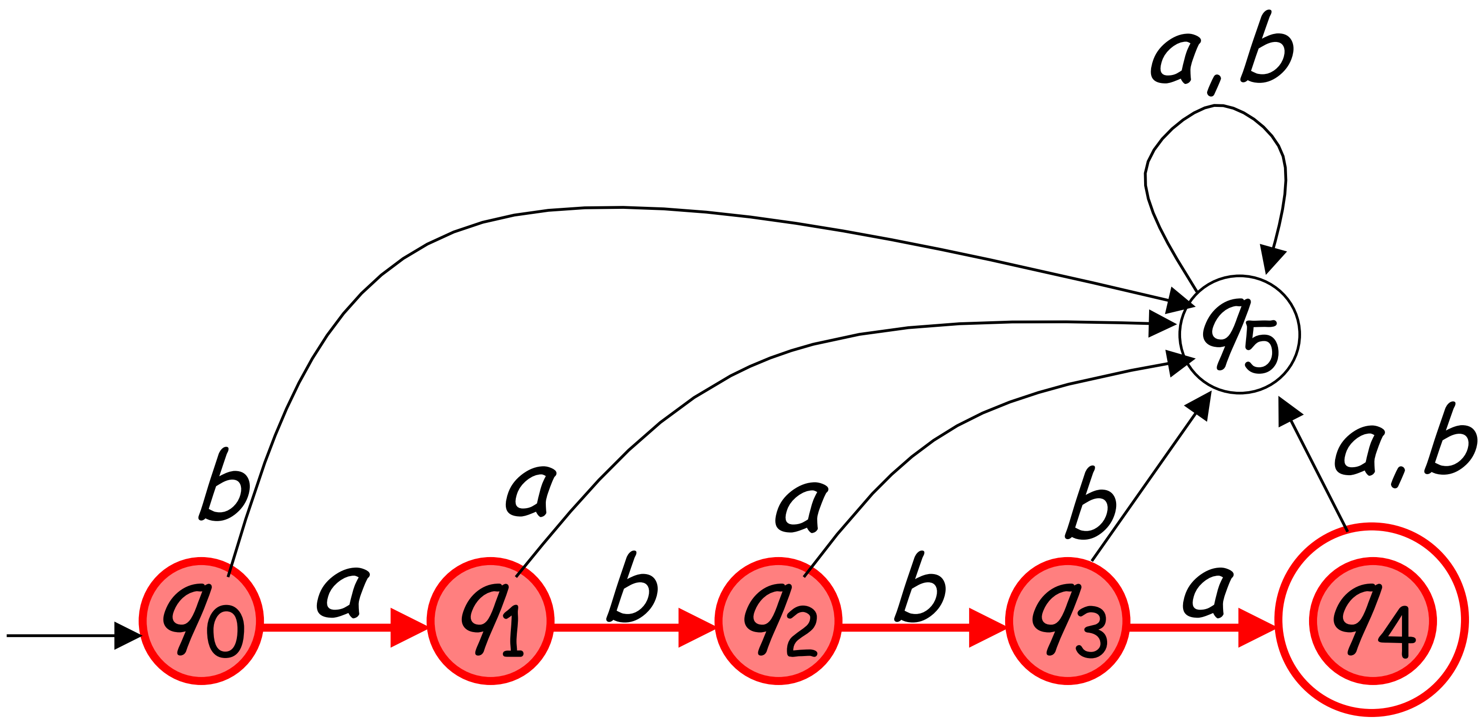
$$\delta^* : Q \times \Sigma^* \to Q$$



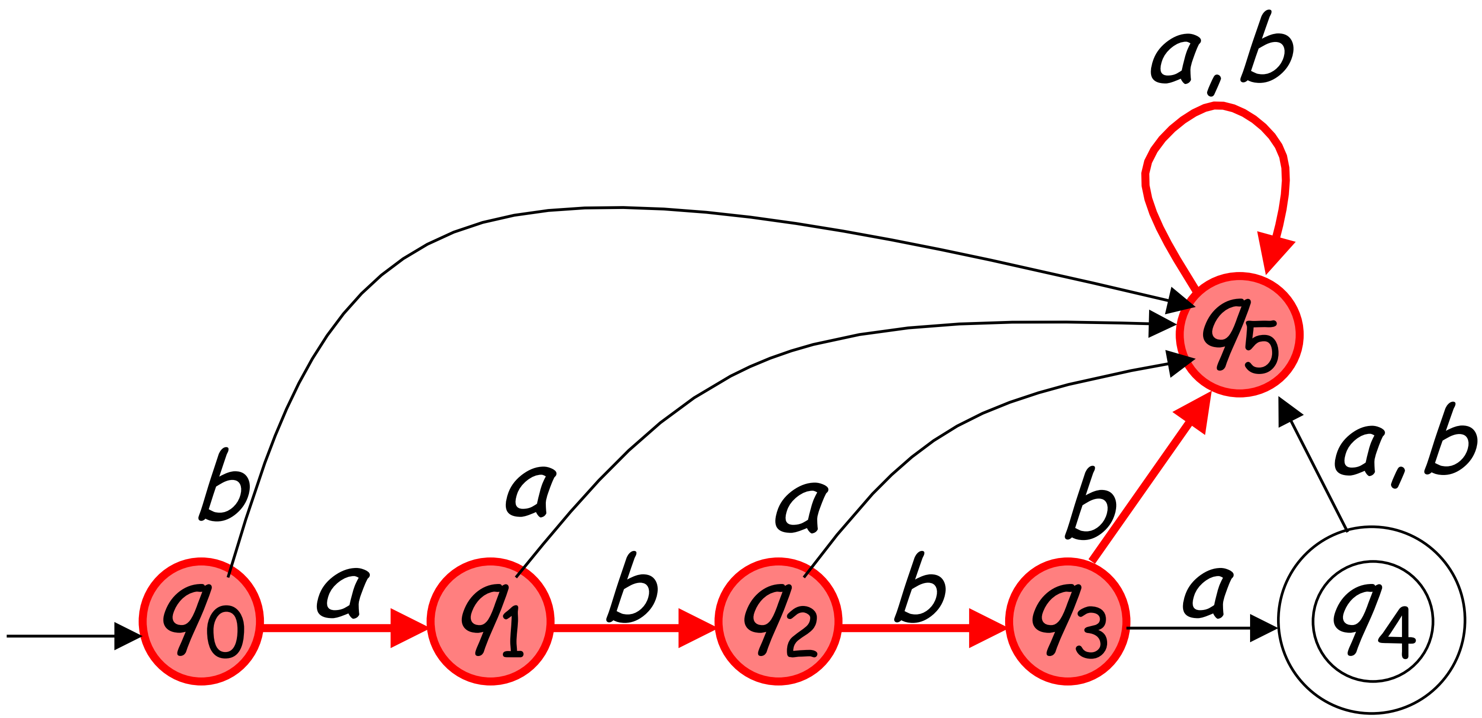
$$\delta^*(q_0, ab) = q_2$$



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



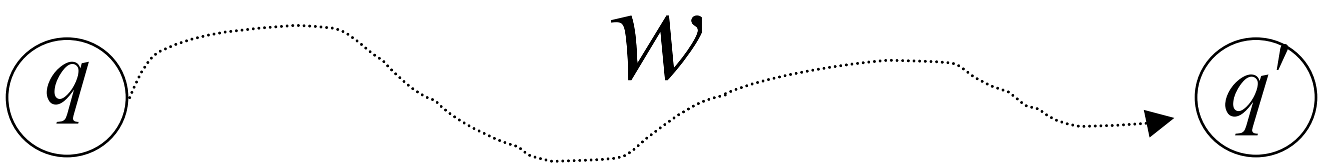
$$\delta^*(q_0, abbbaa) = q_5$$



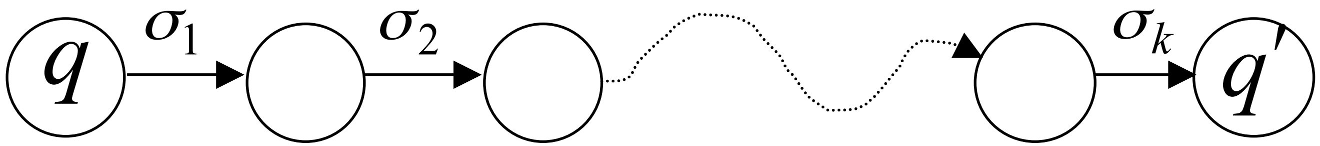
Observation: If there is a walk from q to q' with label w

■ Theorem 2.1

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

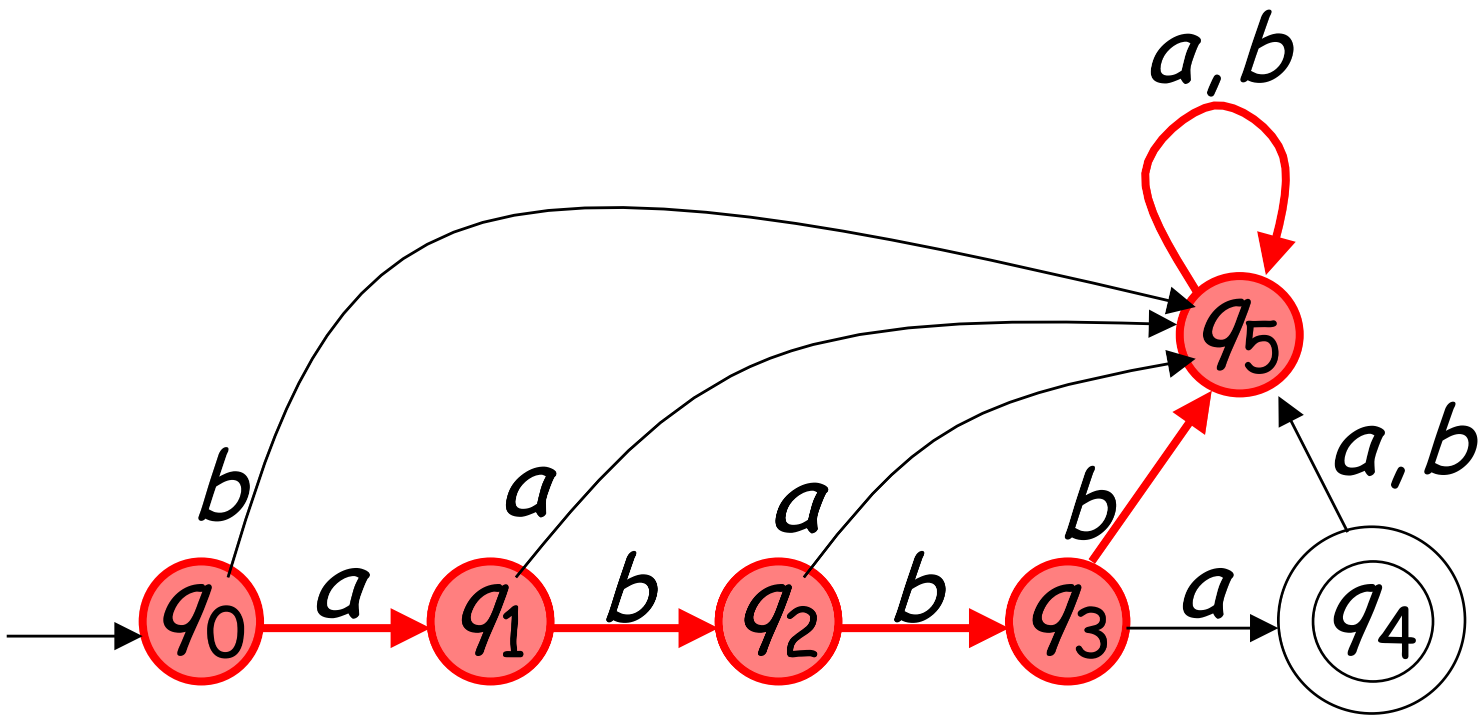


$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



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

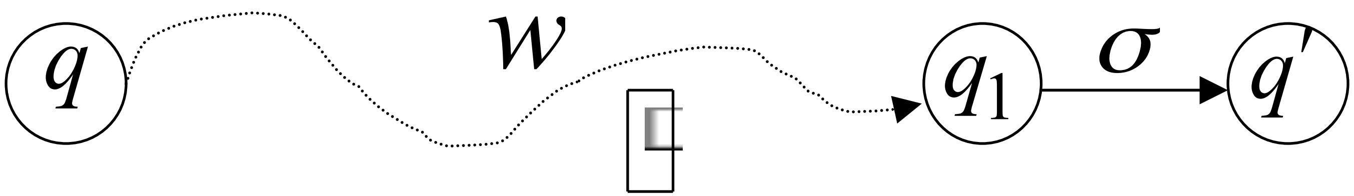
$$\delta^*(q_0, abbbaa) = q_5$$



Recursive Definition

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

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



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

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

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

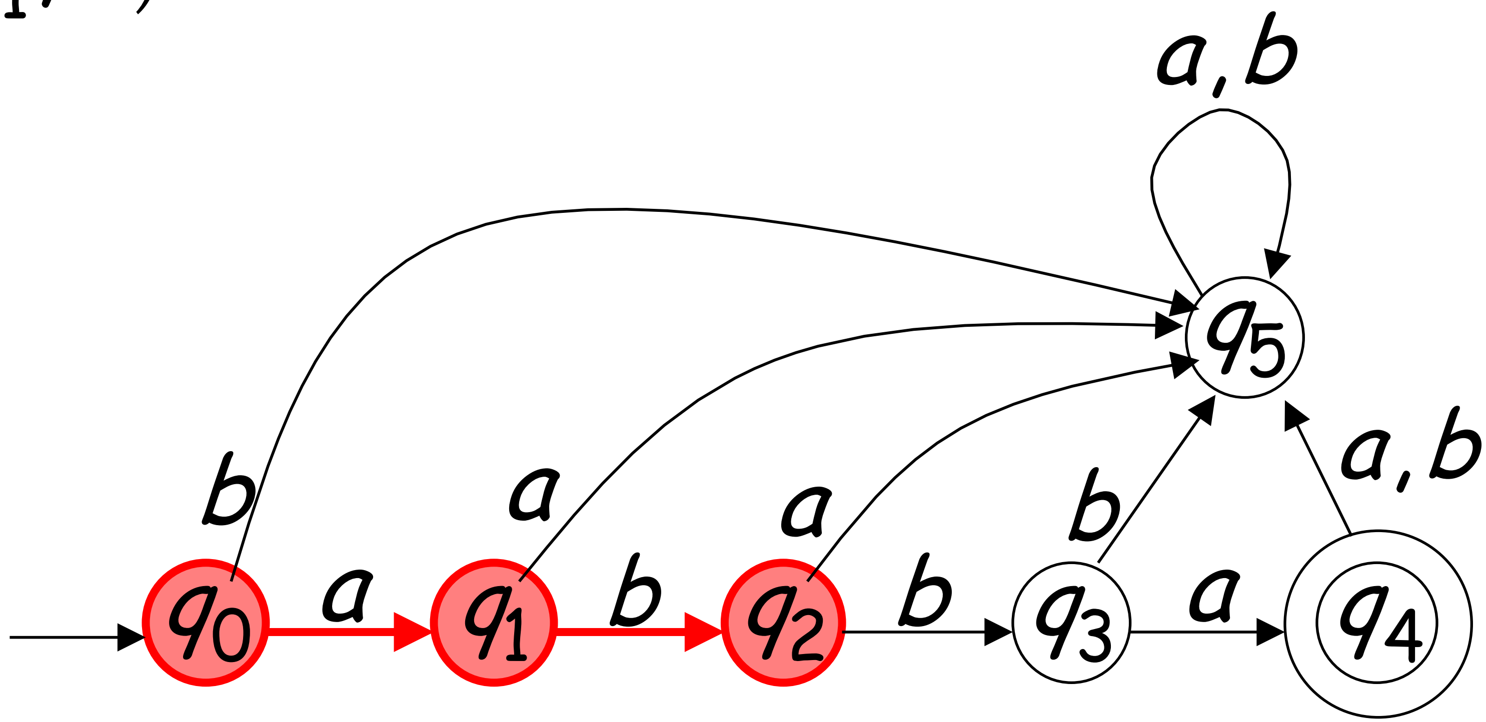
$$\delta(\delta^*(q_0, a), b) =$$

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

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

$$\delta(q_1, b) =$$

$$q_2$$



Languages Accepted by DFAs

Take DFA M

Definition:

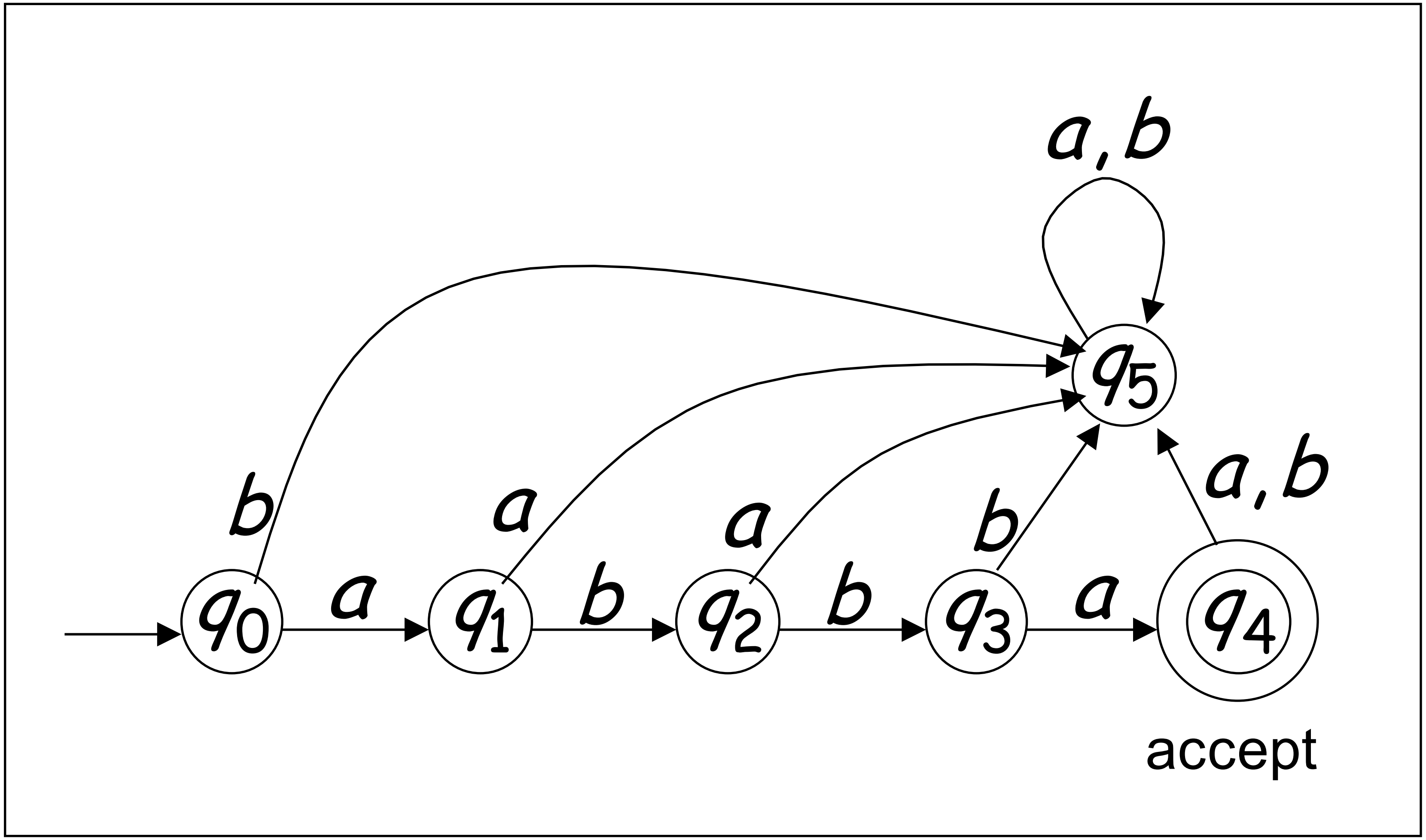
– The language $L(M)$ contains all input strings accepted by M

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

Example

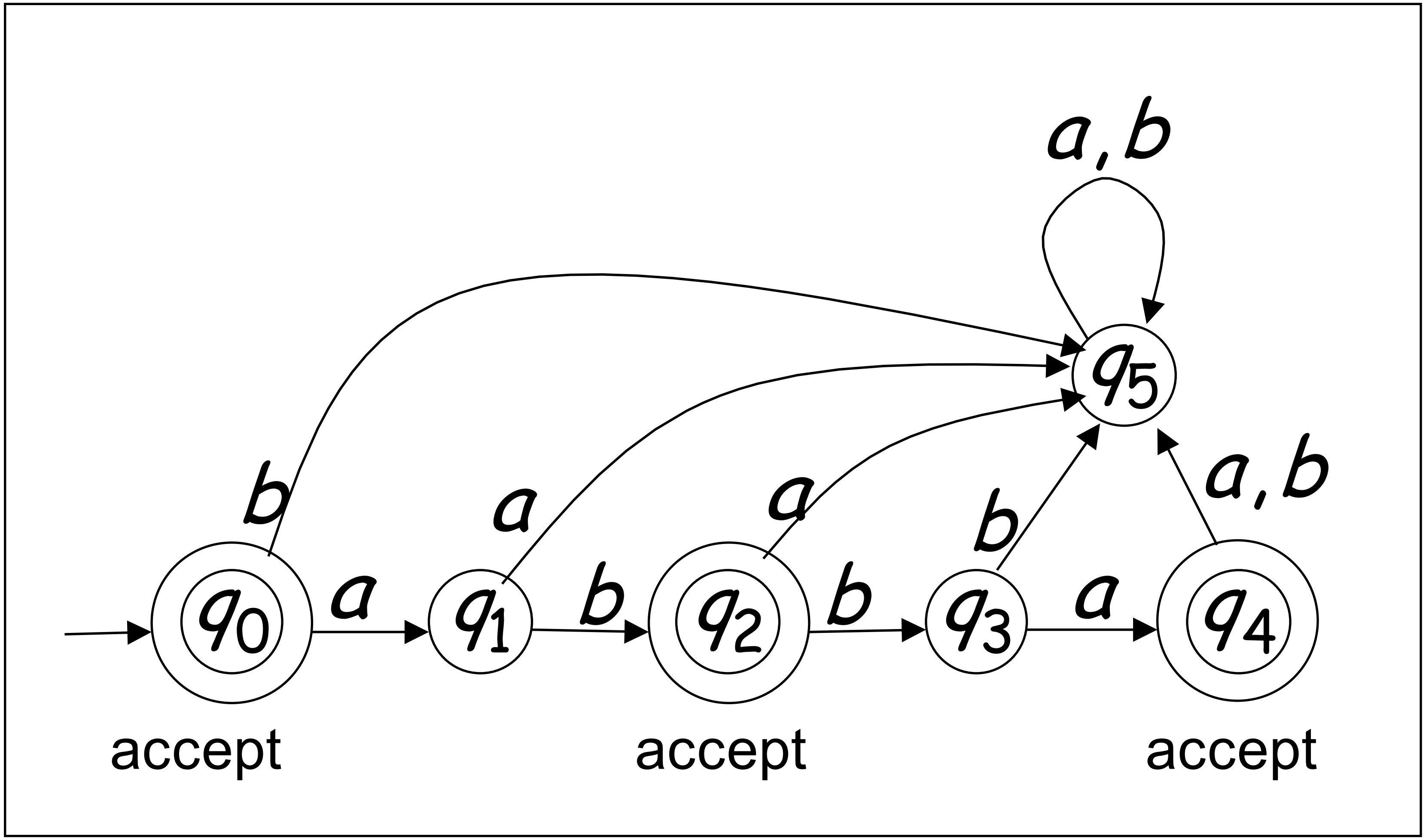
$L(M) = \{abba\}$

M



Another Example

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



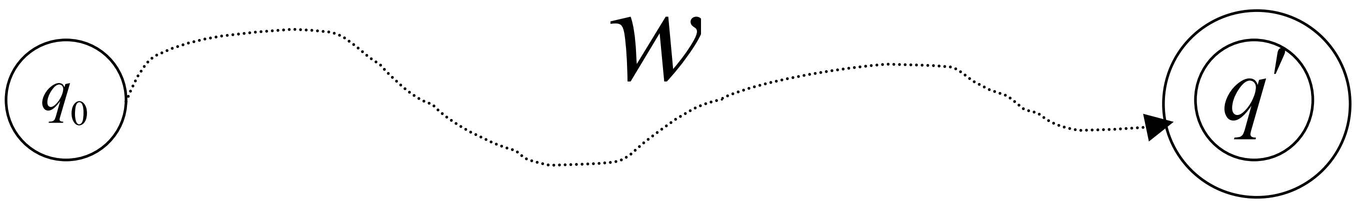
Formally

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 :

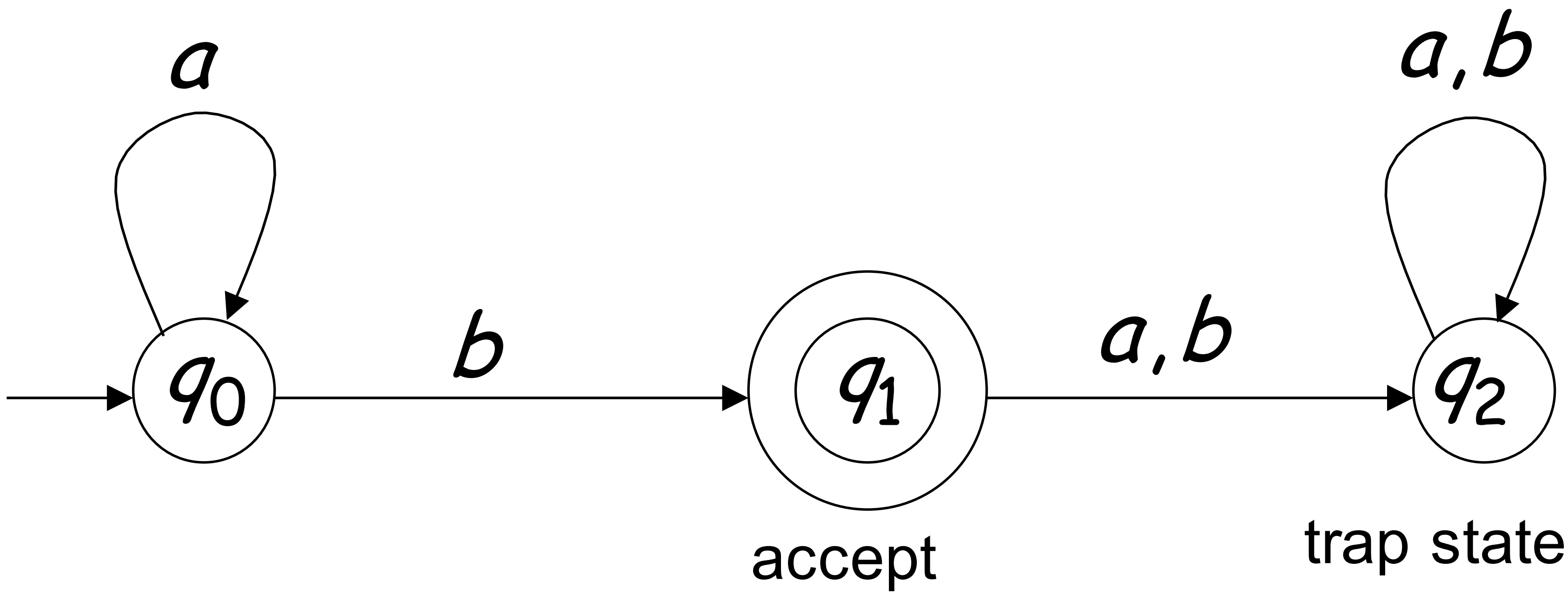
$$\overline{L(M)} = \{ w \in \Sigma^* : \delta^*(q_0, w) \notin F \}$$

$$q_0 \xrightarrow{w} q' \qquad q' \notin F$$

Example 2.2

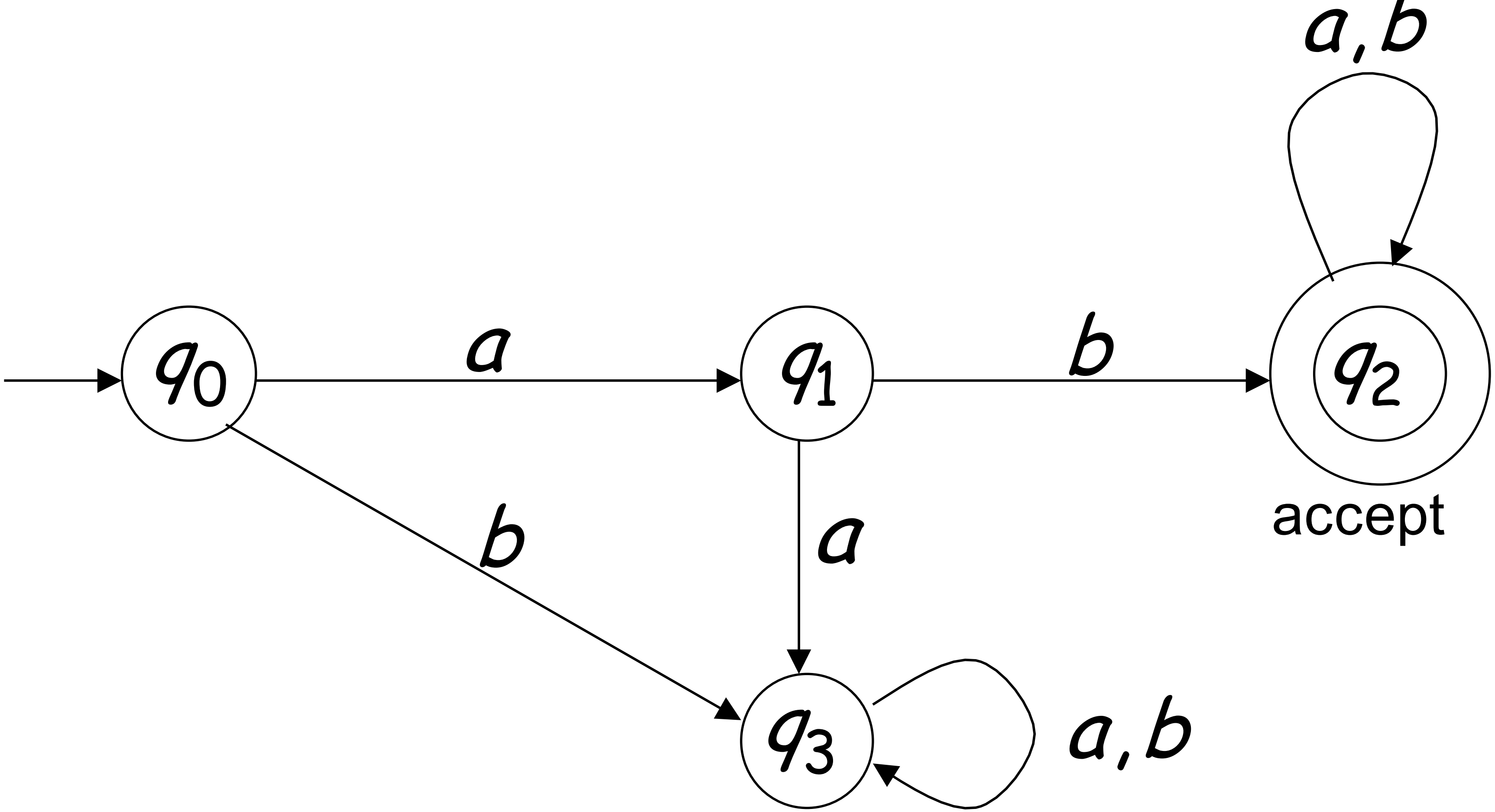
$$M = (Q, \Sigma, \delta, q_0, F)$$

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



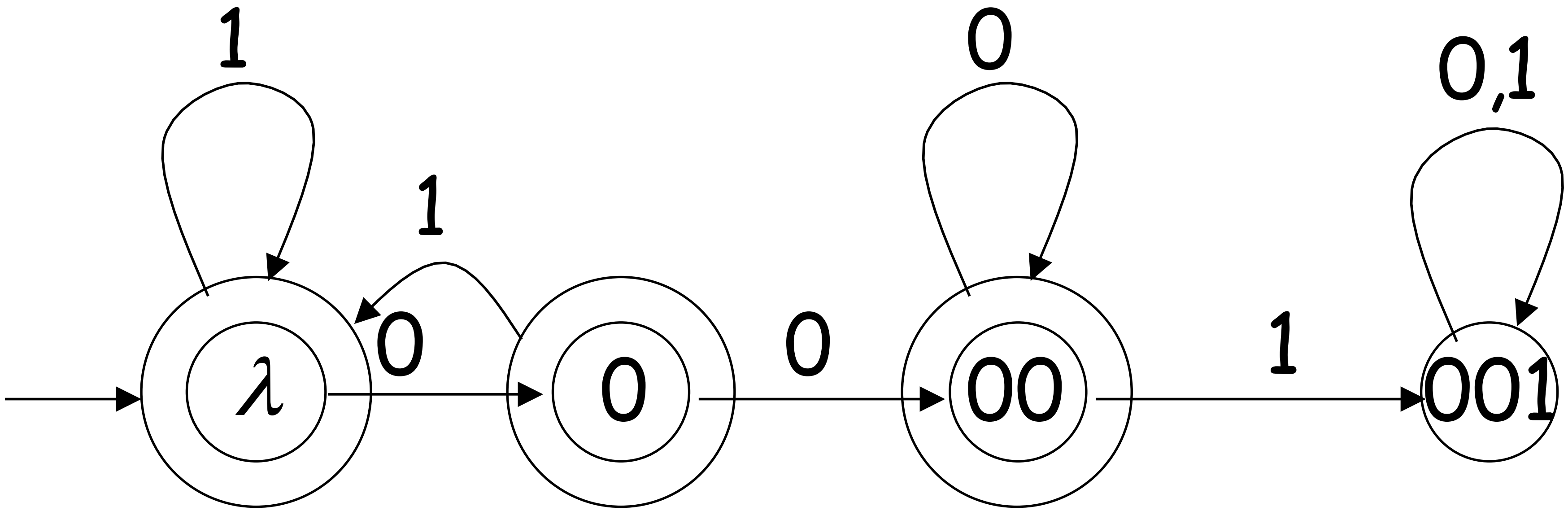
Example 2.3

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



Example 2.4

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



Regular Languages

A language L is regular iff there exists some 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 }

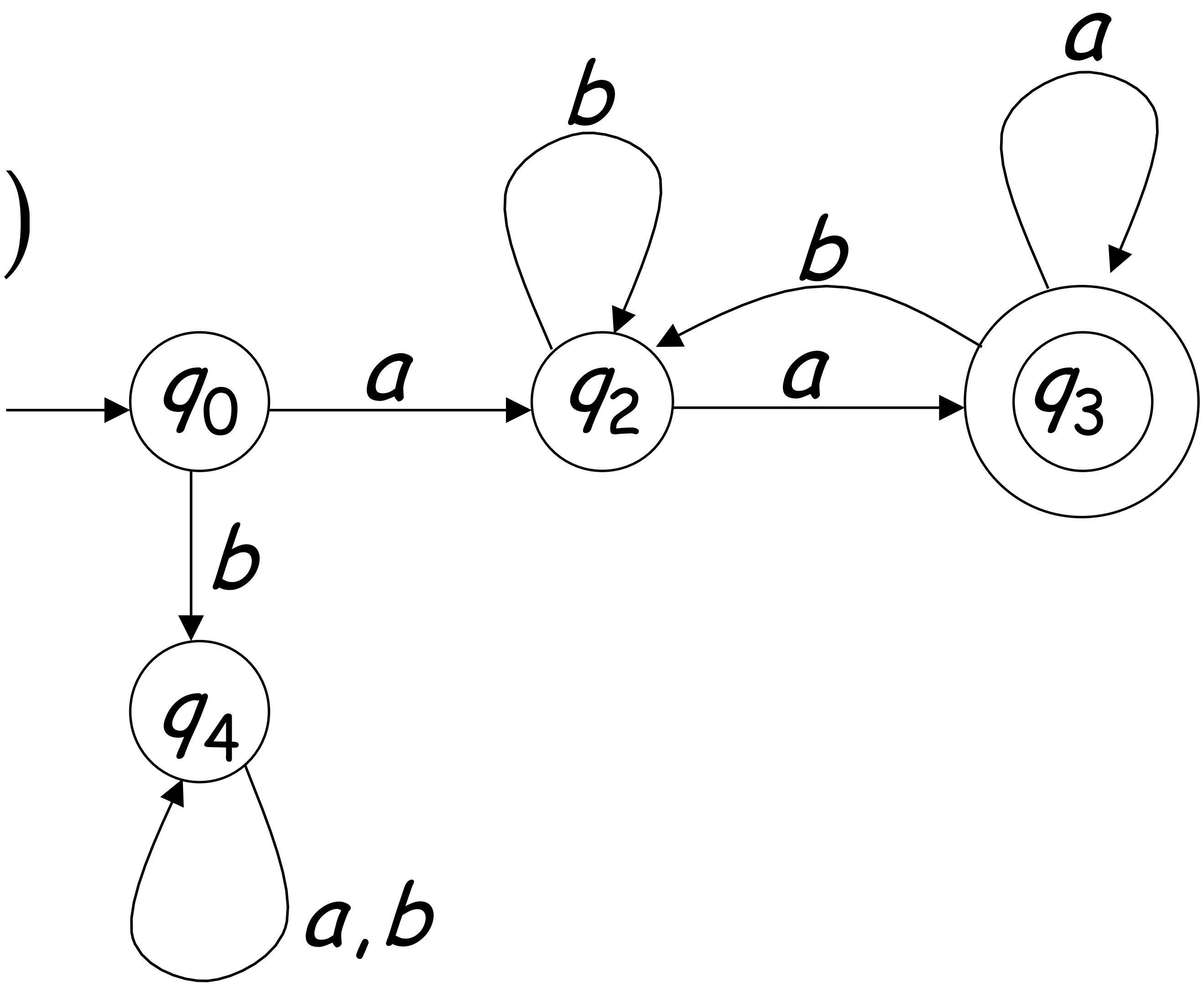
{ all strings without substring 001 }

There exists DFA that accept these Languages

Example 2.5

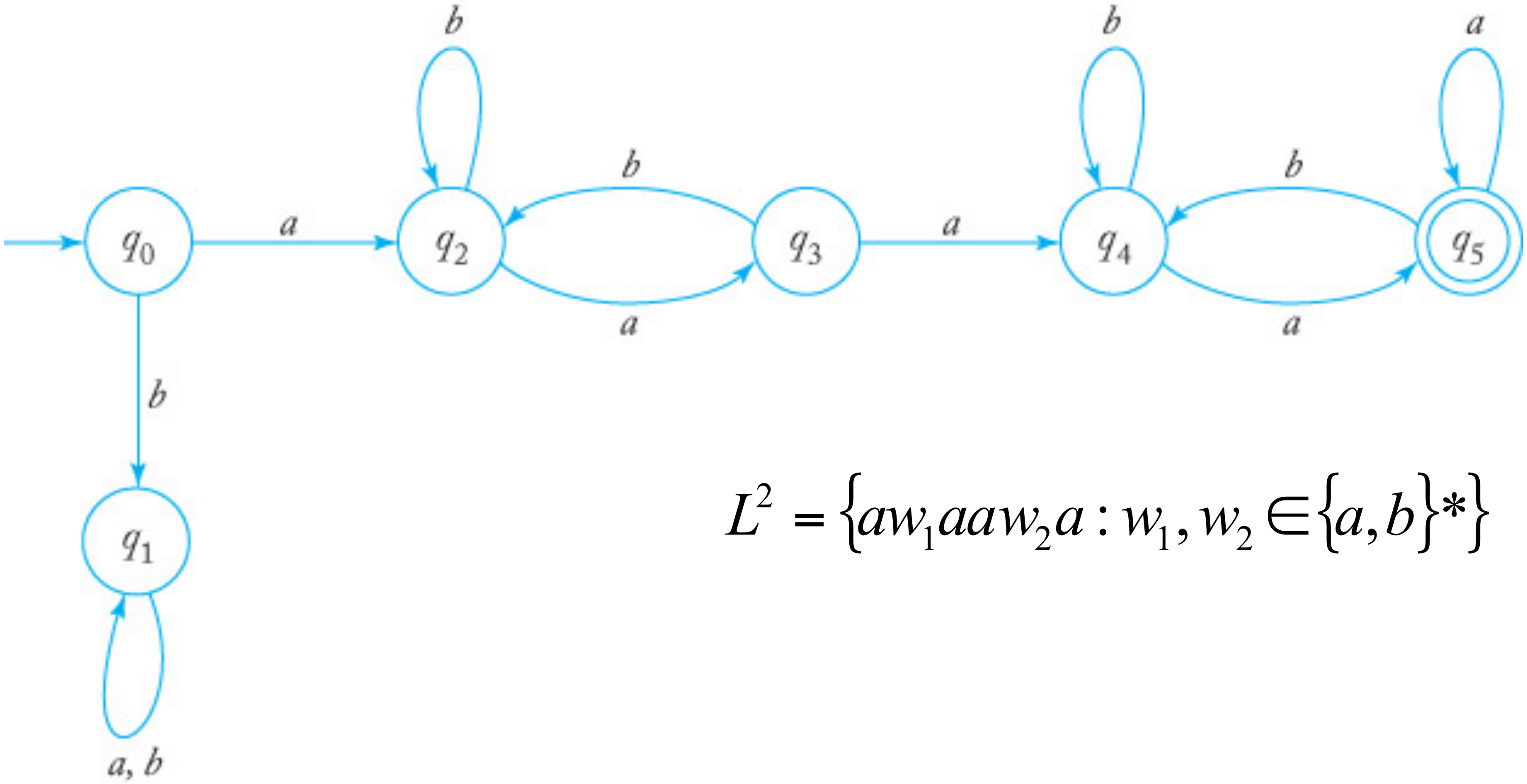
The language $L = \{awa : w \in \{a,b\}^*\}$ is regular:

$L = L(M)$



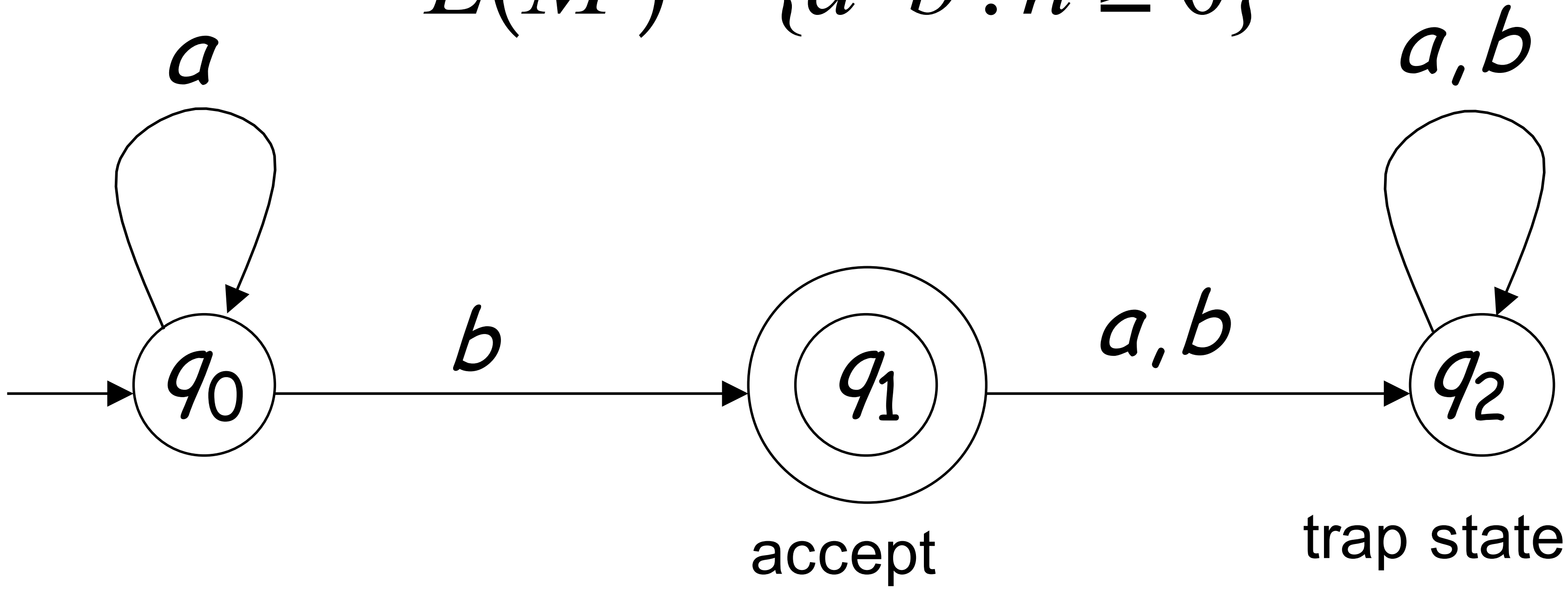
Example 2.6

The language $L = \{awa : w \in \{a,b\}^*\}$
is regular, how about L^2?



$$L^2 = \{aw_1aaw_2a : w_1, w_2 \in \{a,b\}^*\}$$

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

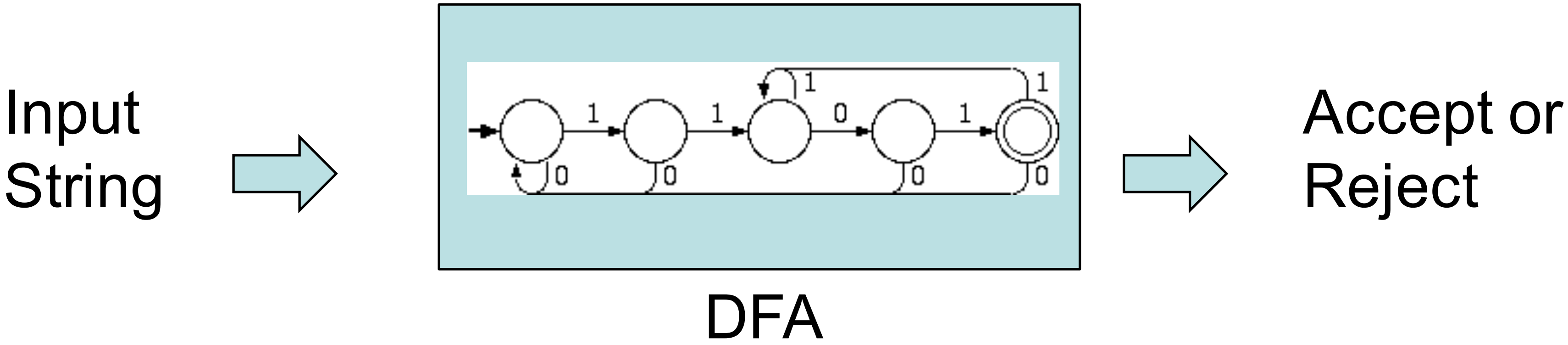


$$\boxed{L = \{a^n b^n : n \geq 0\}} \; ?$$

There exist languages which are not Regular:

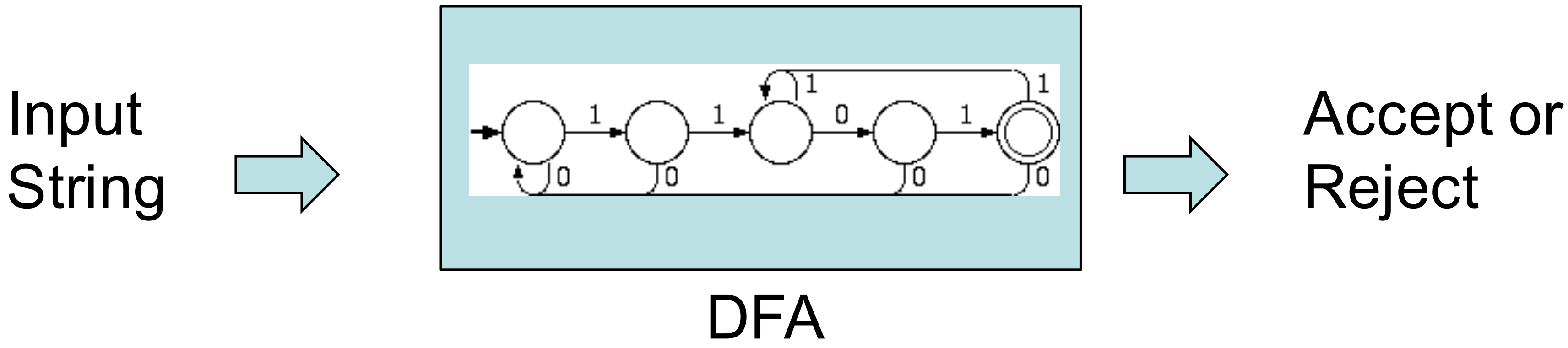
There is no DFA that accepts such a language
(we will prove this later in the class)

DFA Recap



- A machine with finite number of states, some states are accepting states, others are rejecting states
- At any time, it is in one of the states
- It reads an input string, one character at a time

DFA Recap



- After reading each character, it moves to another state depending on what is read and what is the current state
- If reading all characters, the DFA is in an accepting state, the input string is accepted.
- Otherwise, the input string is rejected.

Definition 2.1

Deterministic Finite Accepter (DFA) is define by the 5-tuple

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

Q : a finite set of **internal states**

Σ : a finite set of symbols called **input alphabet**

δ : $Q \times \Sigma \rightarrow Q$ called **transition function**

q_0 : $q_0 \in Q$ is the **initial state**

F : $F \subseteq Q$ is a set of **final states**

Regular Languages

A language L is regular iff there exists some DFA M such that $L = L(M)$

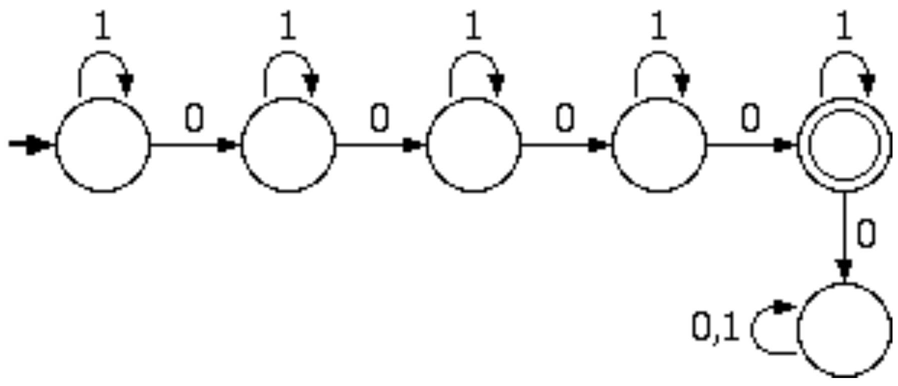
Definition:

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

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

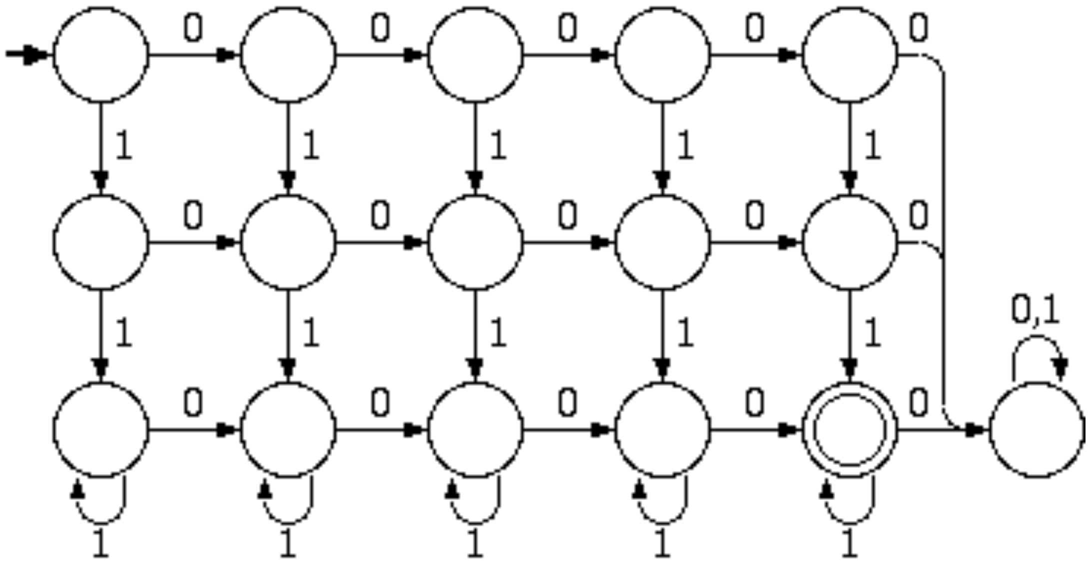
More Examples

- All strings that contain exactly 4 "0"s.
- All strings containing exactly 4 "0" s and at least 2 "1" s.
- All strings of length at most five.
- All strings ending in "1101".
- All strings whose binary interpretation is divisible by 5.
- All strings that contain the substring 0101.
- All strings that don't contain the substring 110.



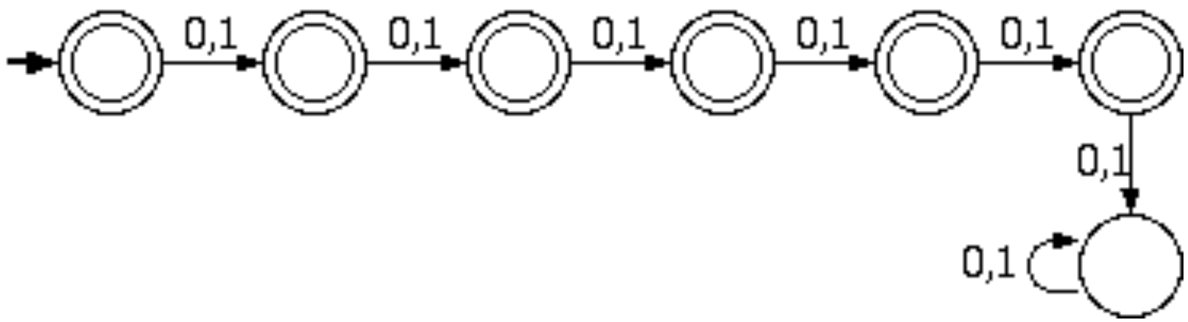
More Examples

- All strings that contain exactly 4 "0"s.
- All strings containing exactly 4 "0" s and at least 2 "1" s.
- All strings of length at most five.
- All strings ending in "1101".
- All strings whose binary interpretation is divisible by 5.
- All strings that contain the substring 0101.
- All strings that don’t contain the substring 110.



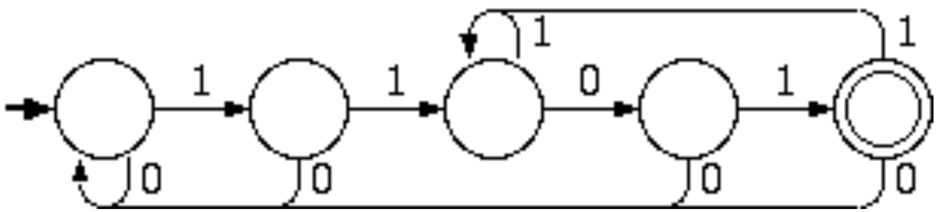
More Examples

- All strings that contain exactly 4 "0"s.
- All strings containing exactly 4 "0" s and at least 2 "1" s.
- All strings of length at most five.
- All strings ending in "1101".
- All strings whose binary interpretation is divisible by 5.
- All strings that contain the substring 0101.
- All strings that don’t contain the substring 110.



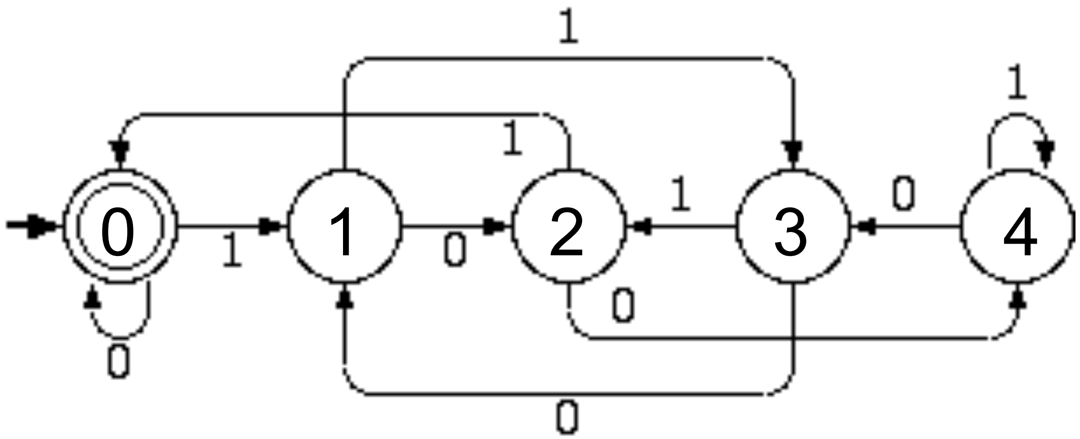
More Examples

- All strings that contain exactly 4 "0"s.
- All strings containing exactly 4 "0" s and at least 2 "1" s.
- All strings of length at most five.
- All strings ending in "1101".
- All strings whose binary interpretation is divisible by 5.
- All strings that contain the substring 0101.
- All strings that don’t contain the substring 110.



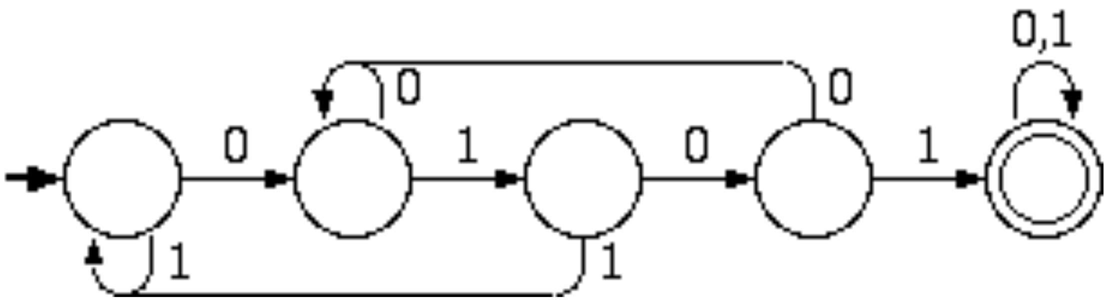
More Examples

- All strings that contain exactly 4 "0"s.
- All strings containing exactly 4 "0" s and at least 2 "1" s.
- All strings of length at most five.
- All strings ending in "1101".
- All strings whose binary interpretation is divisible by 5.
- All strings that contain the substring 0101.
- All strings that don’t contain the substring 110.



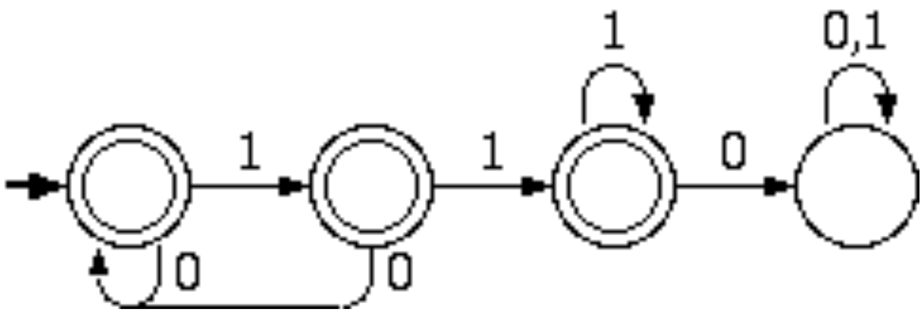
More Examples

- All strings that contain exactly 4 "0"s.
- All strings containing exactly 4 "0" s and at least 2 "1" s.
- All strings of length at most five.
- All strings ending in "1101".
- All strings whose binary interpretation is divisible by 5.
- All strings that contain the substring 0101.
- All strings that don’t contain the substring 110.



More Examples

- All strings that contain exactly 4 "0"s.
- All strings containing exactly 4 "0" s and at least 2 "1" s.
- All strings of length at most five.
- All strings ending in "1101".
- All strings whose binary interpretation is divisible by 5.
- All strings that contain the substring 0101.
- All strings that don’t contain the substring 110.

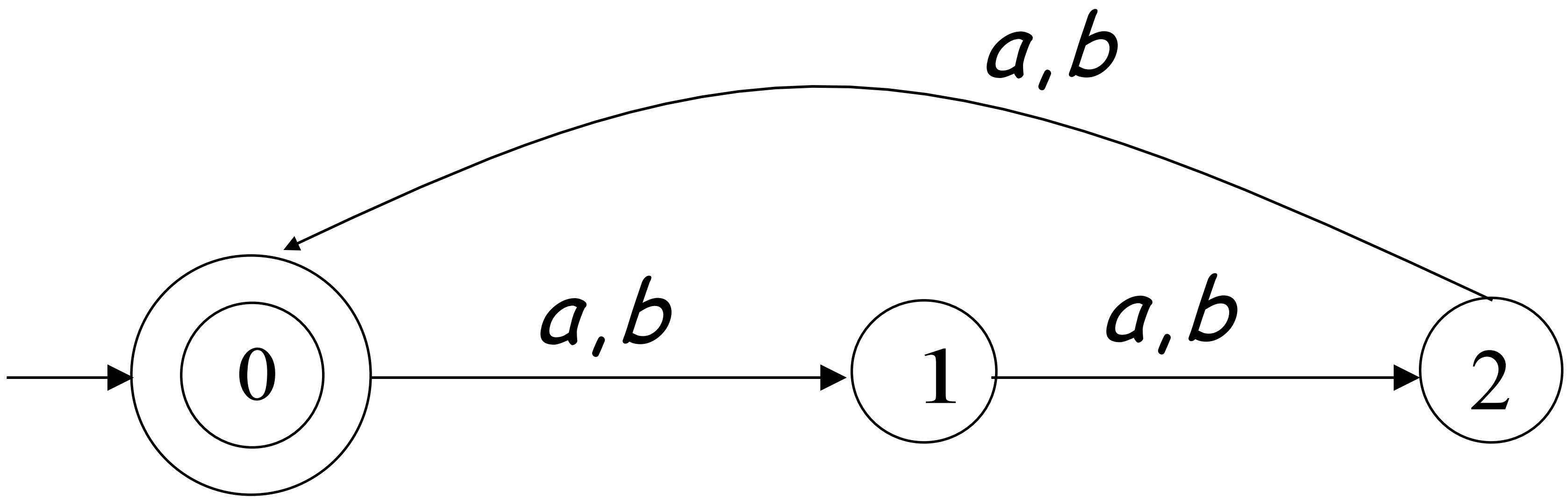


Exercise 2.1.7

Find DFAs for the following languages on $\Sigma=\{a,b\}$

(a) $L = \{w: |w| \bmod 3 = 0\}$

(b) $L = \{w: n_a(w) \bmod 3 > n_b(w) \bmod 3\}$

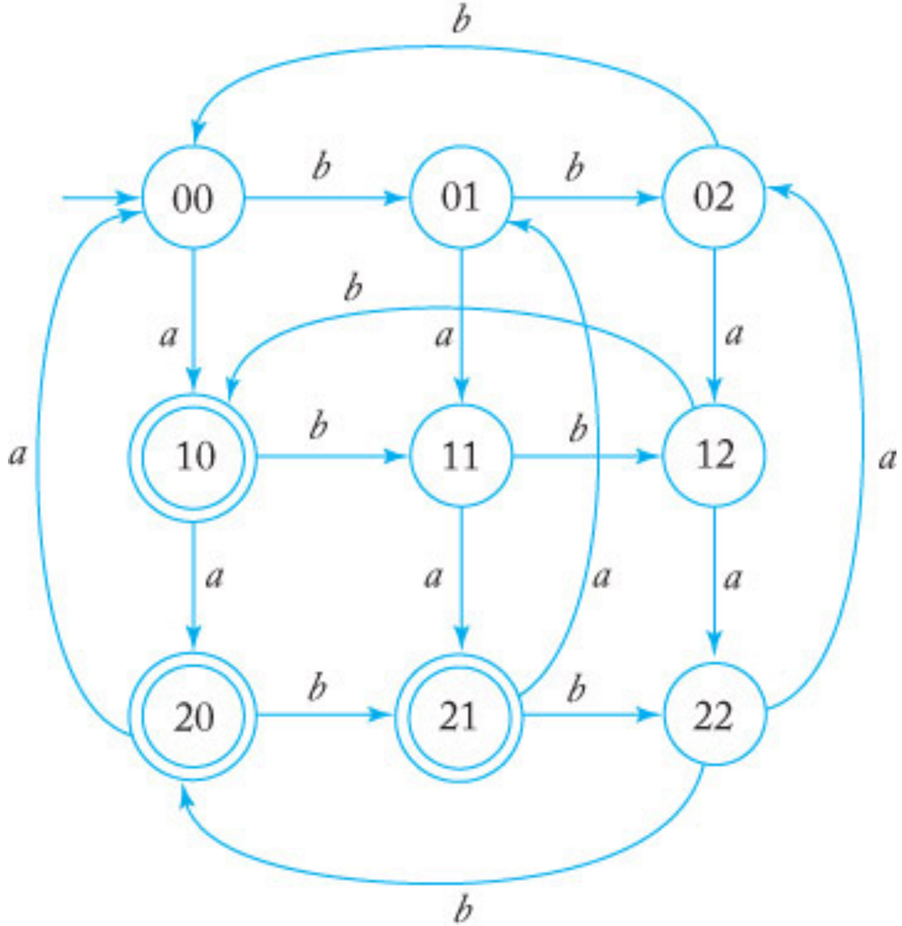


Exercise 2.1.7

Find DFAs for the following languages on $\Sigma=\{a,b\}$

(a) $L = \{w: |w| \bmod 3 = 0\}$

(b) $L = \{w: n_a(w) \bmod 3 > n_b(w) \bmod 3\}$



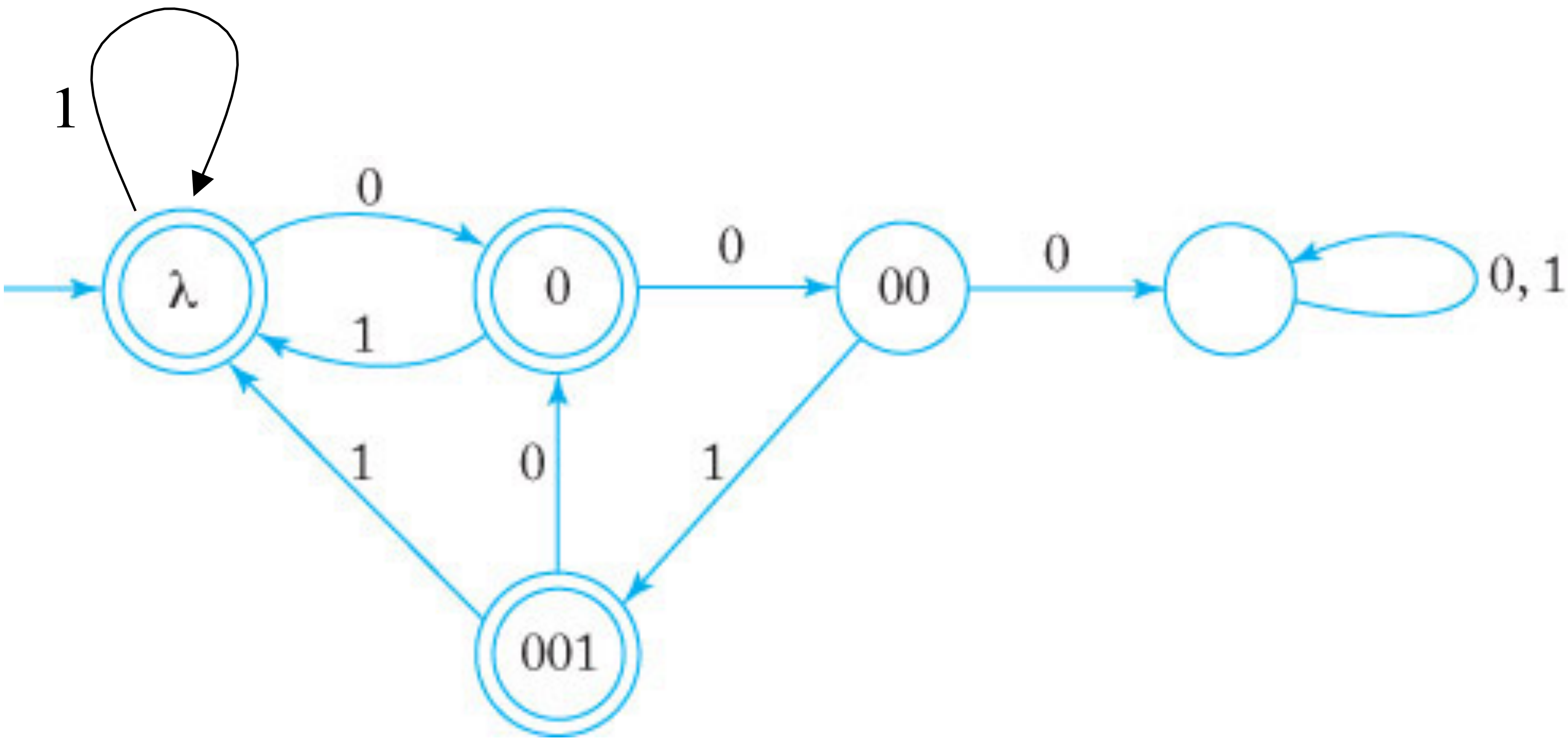
Exercise 2.1.9

(a)Every 00 is followed immediately by a 1.

Ex: 101, 0010,0010011001 ϵ L

0001 and 00100 ∉ L

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



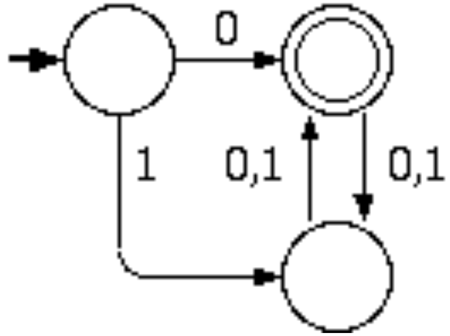
Questions?

Short Quiz

- All strings that start with 0 and have odd length or start with 1 and have even length.
- All strings where every odd position is a 1.

Short Quiz

- All strings that start with 0 and have odd length or start with 1 and have even length.



- All strings where every odd position is a 1.