23MT2014

THEORY OF COMPUTATION

Topic:

DFA AND ACCEPTANCE OF REGULAR LANGUAGE

Session - 3

AIM OF THE SESSION

The aim of the Finite Automata course is to introduce students to the theory and applications of finite automata, enabling them to understand the fundamental concepts and techniques related to automata theory.

INSTRUCTIONAL OBJECTIVES

This Session is designed to:

1. To familiarize students with the components and structure of finite automata, including states, alphabet, transitions, and accepting states.
2. To enable students to analyze and construct deterministic finite automata (DFAs) and non-deterministic finite automata (NFAs).

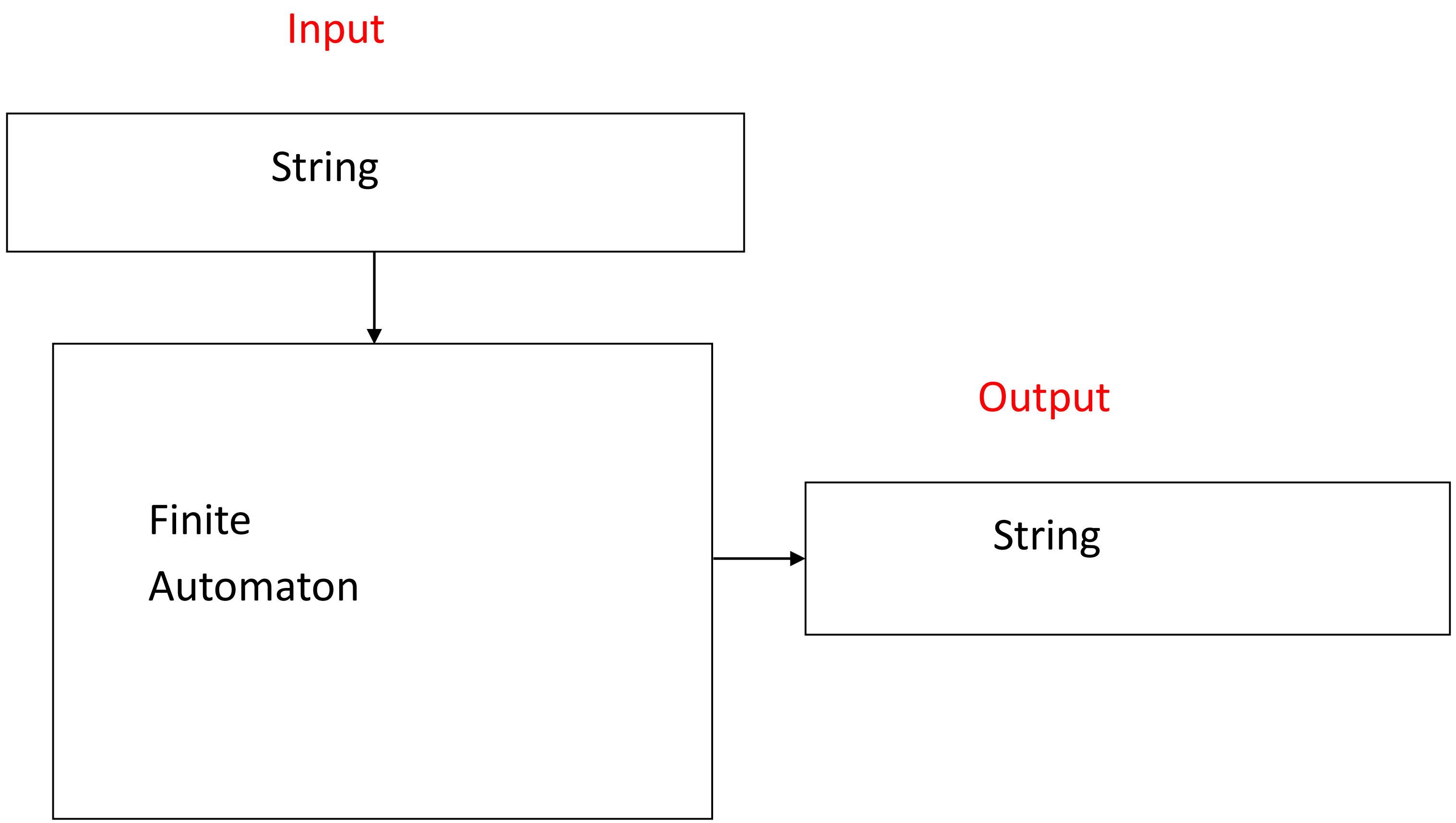
LEARNING OUTCOMES

At the end of this session, you should be able to:

1. Understand the components and structure of finite automata, including states, alphabet, transitions, and accepting states.
2. Analyze and construct deterministic finite automata (DFAs) and non-deterministic finite automata (NFAs).

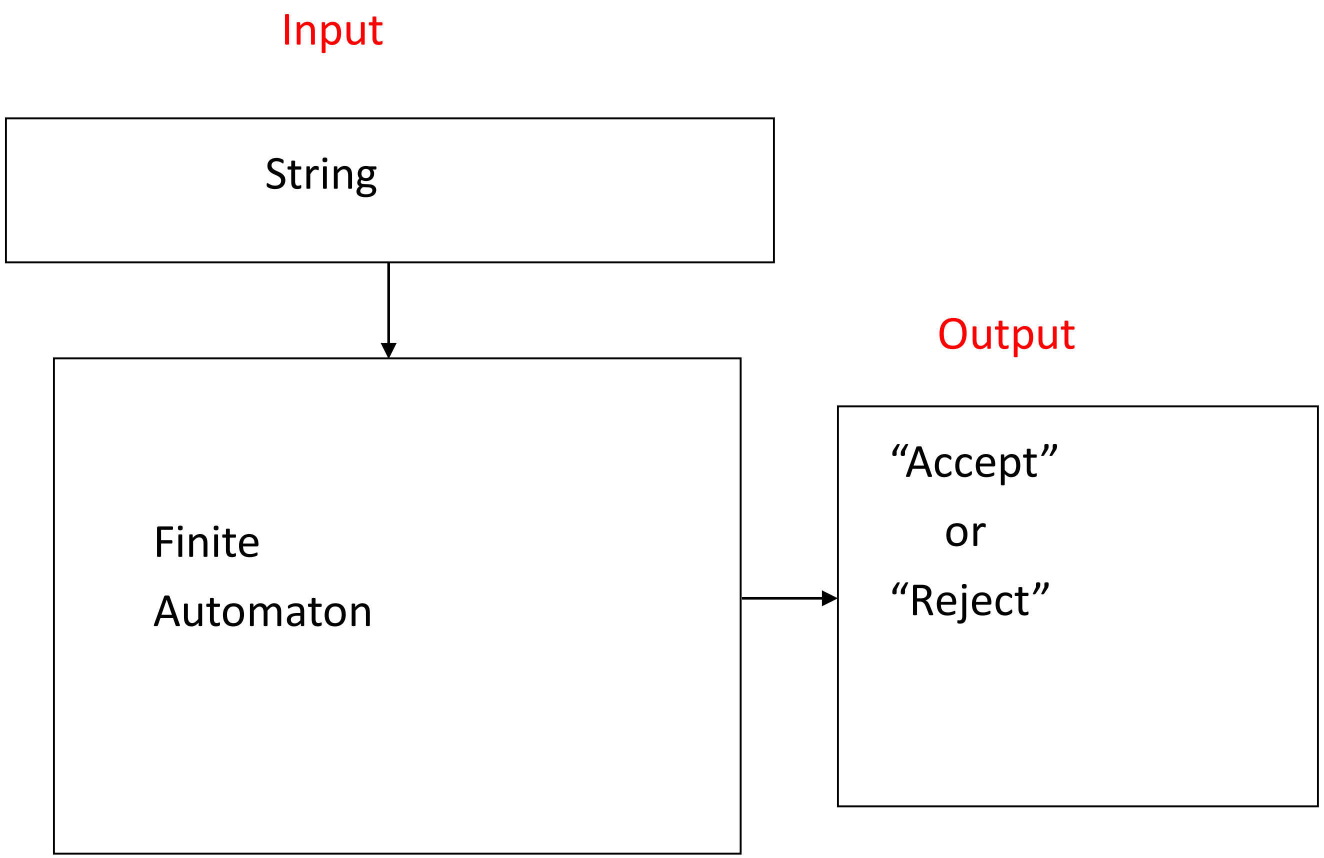
Finite Automaton

•



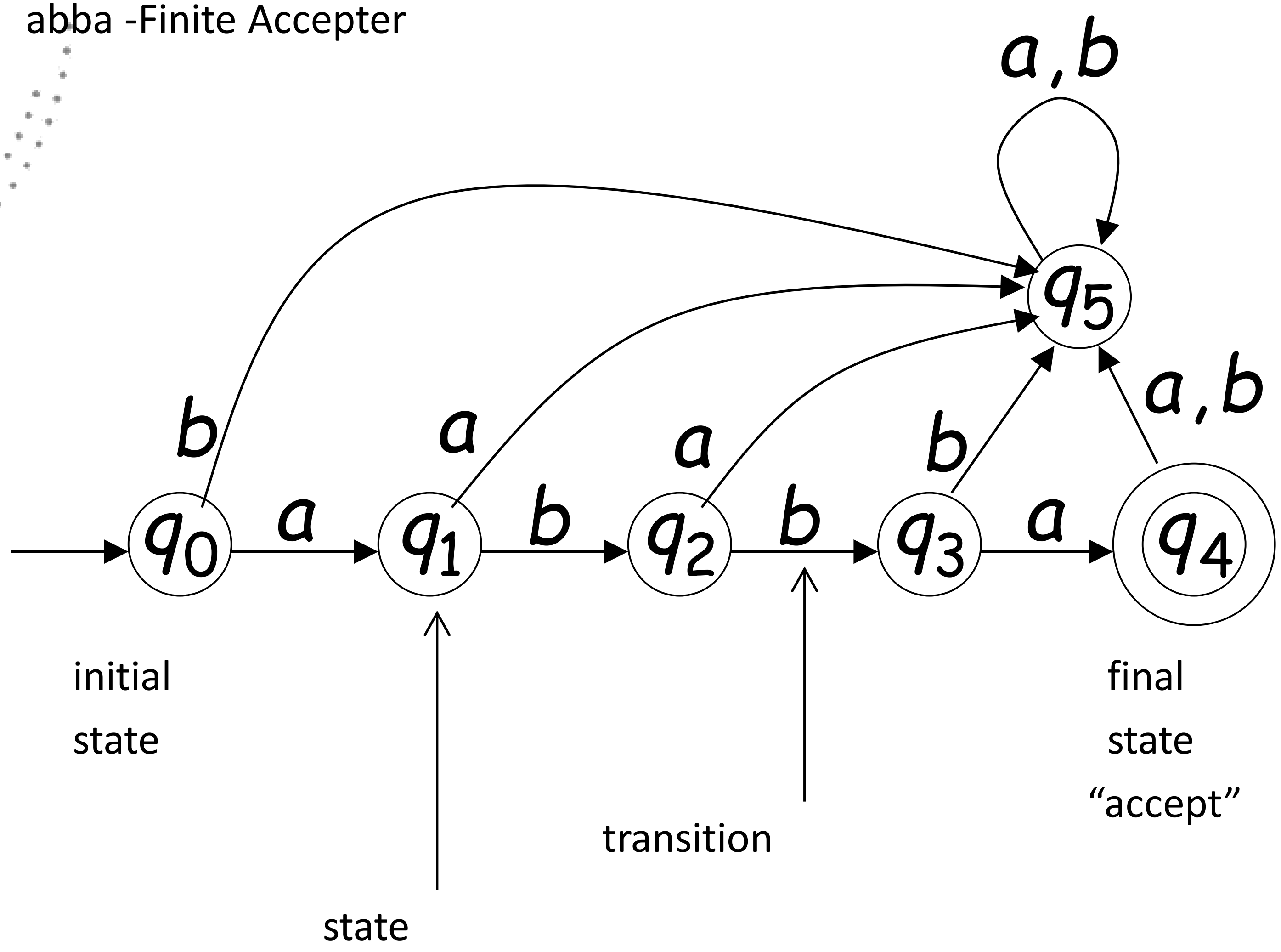
Finite Accepter

•

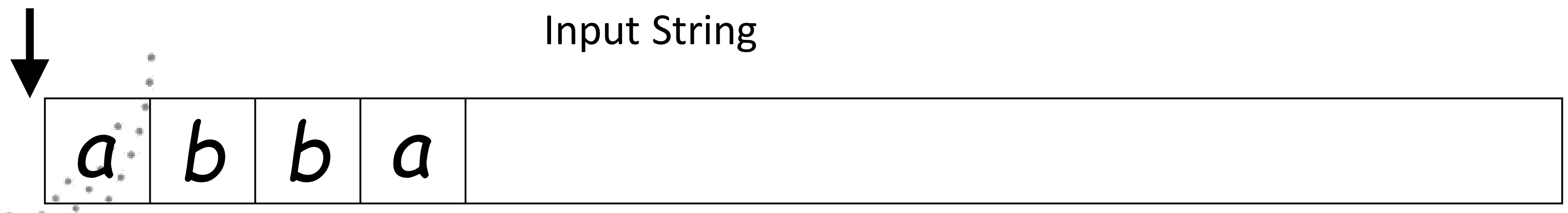


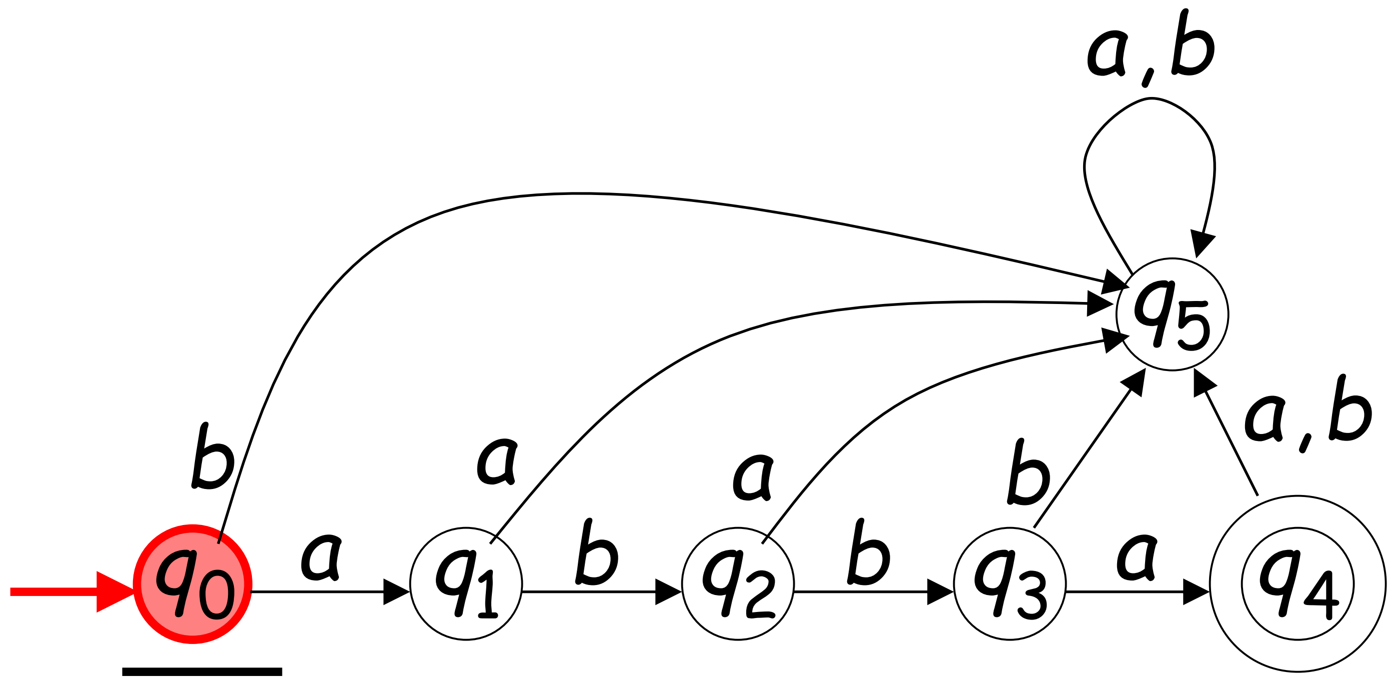
Transition Graph

- abba -Finite Accepter

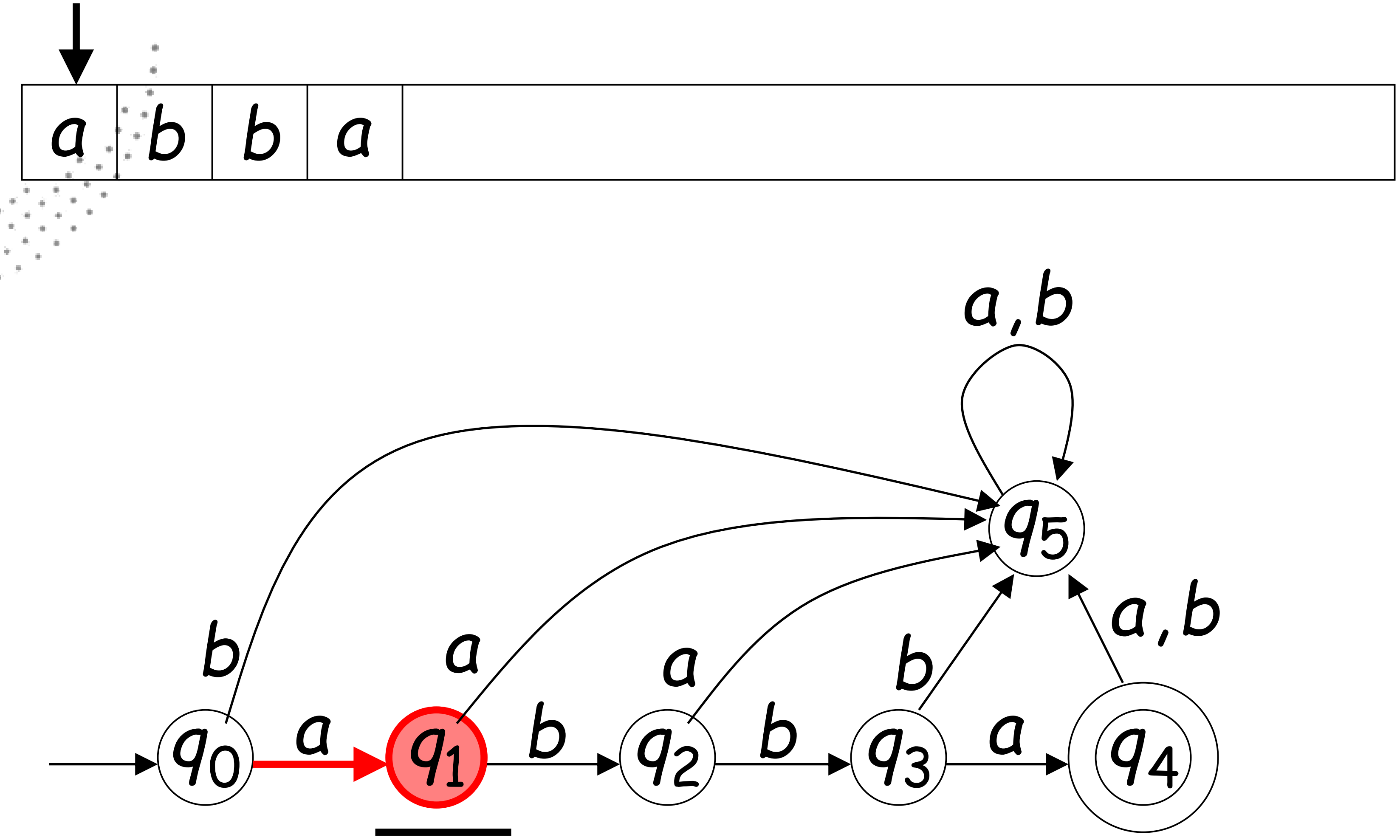


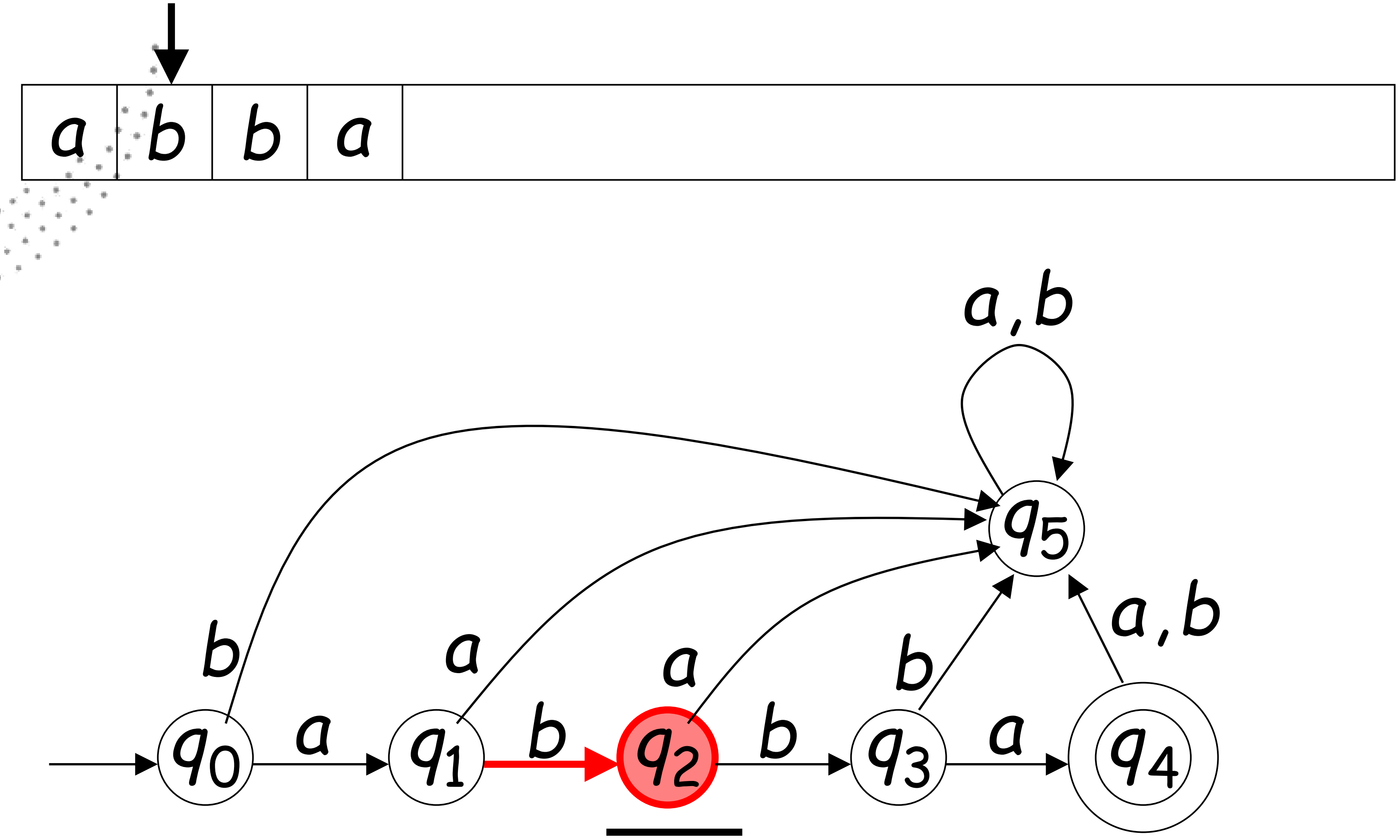
Initial Configuration

- 

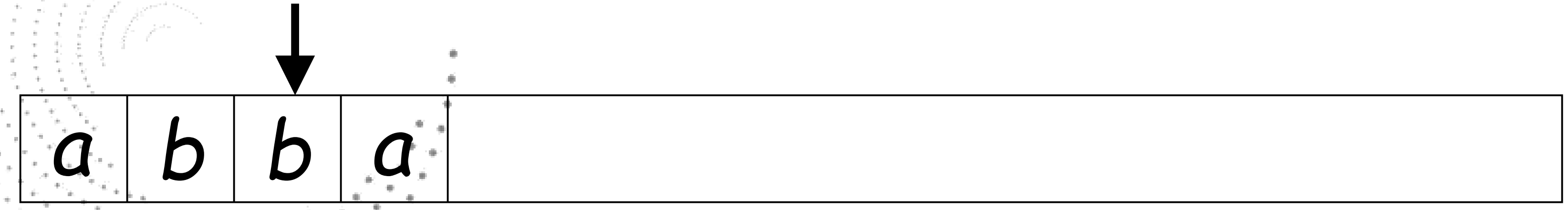


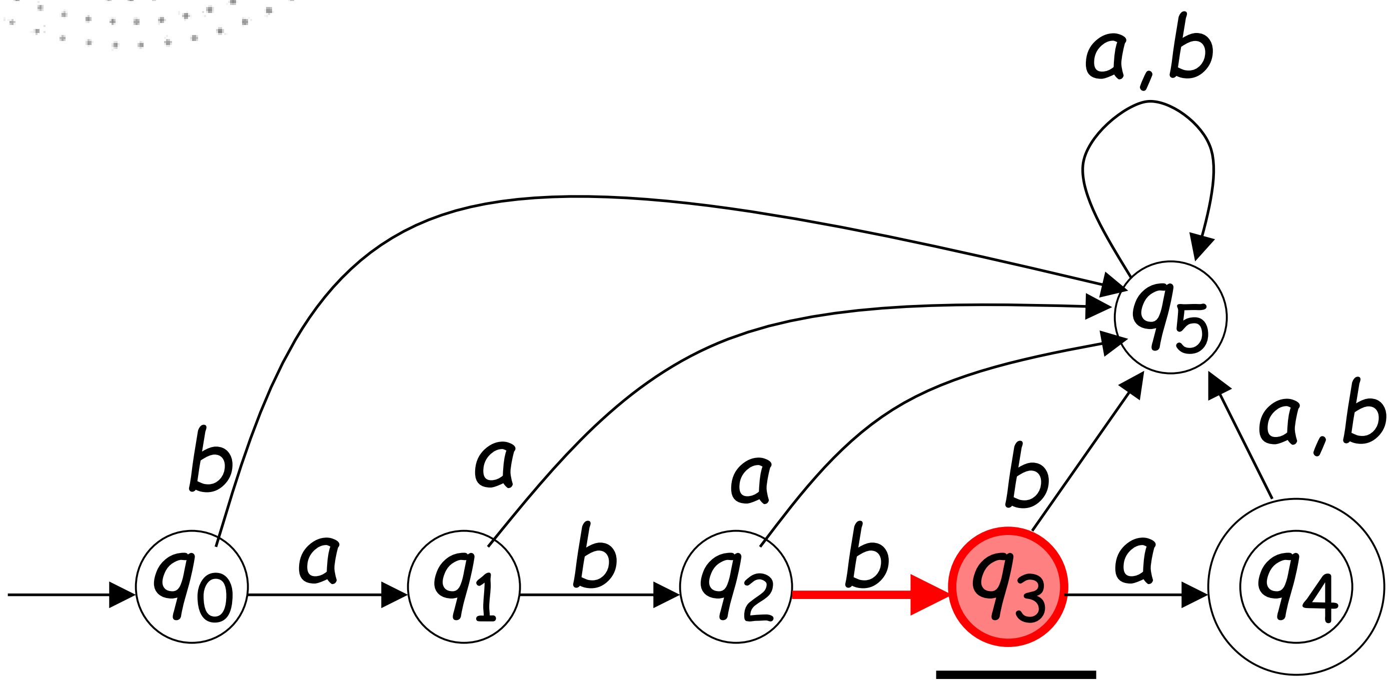
Reading the Input

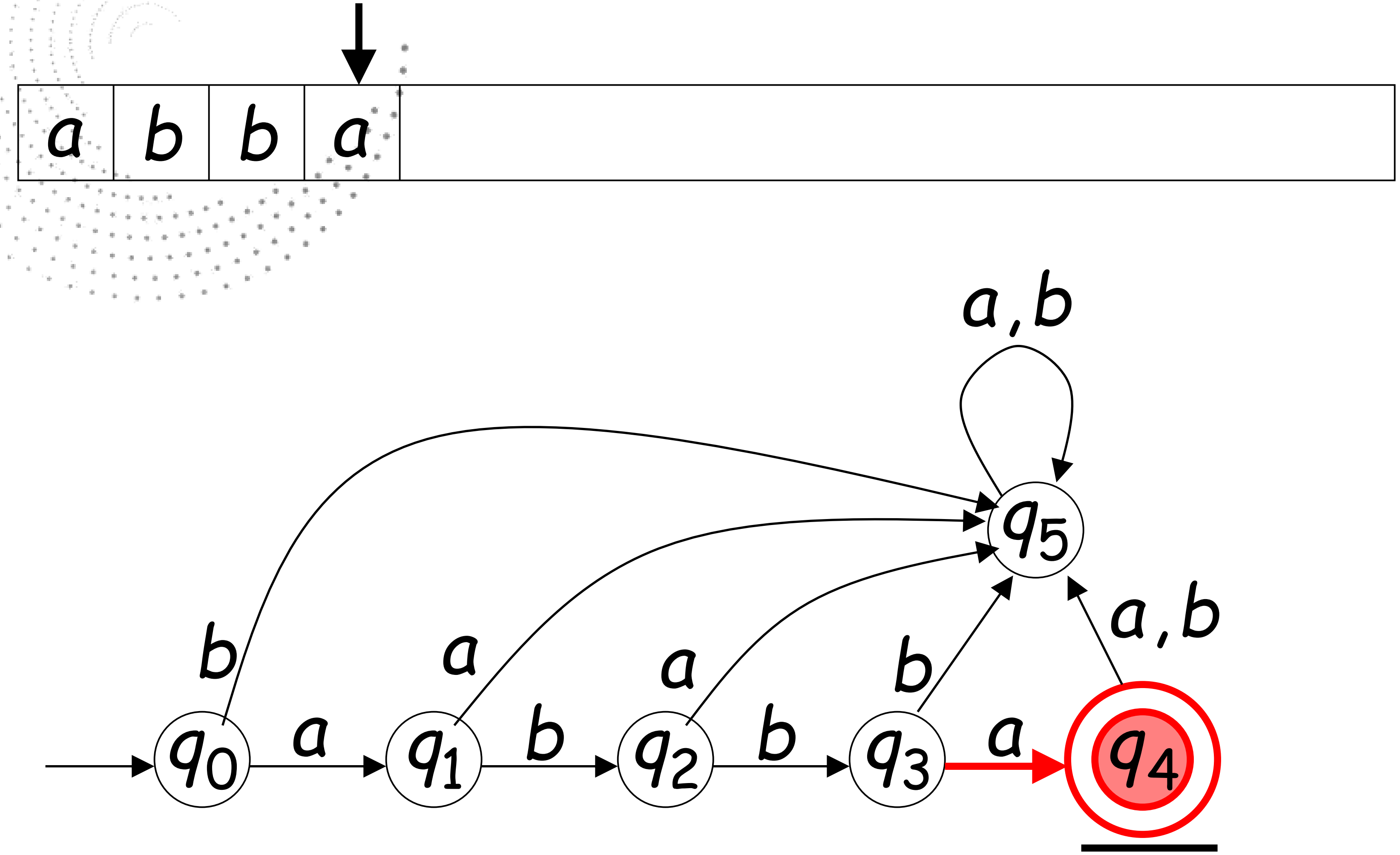
- 


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

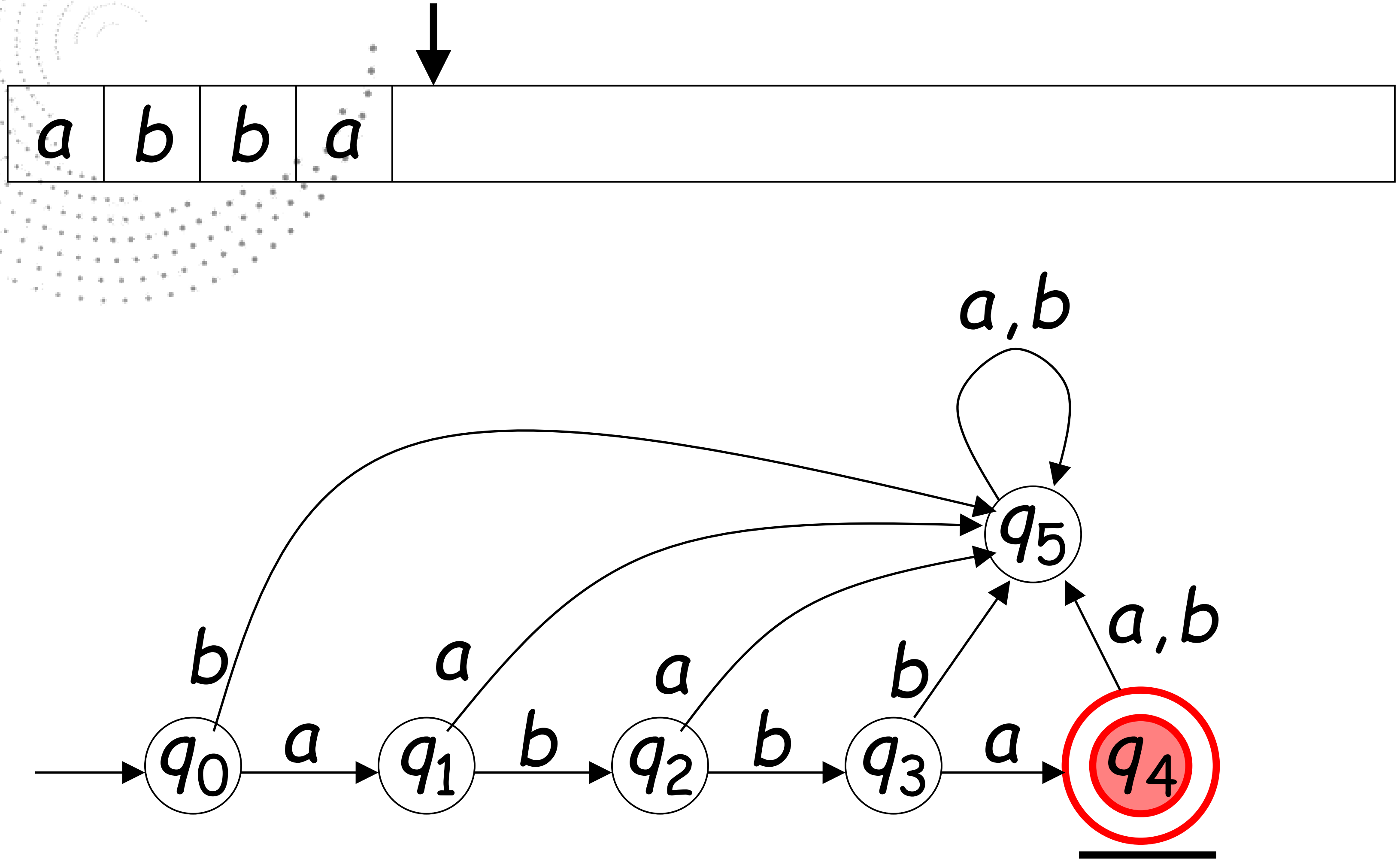
-


a
b
b
a


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

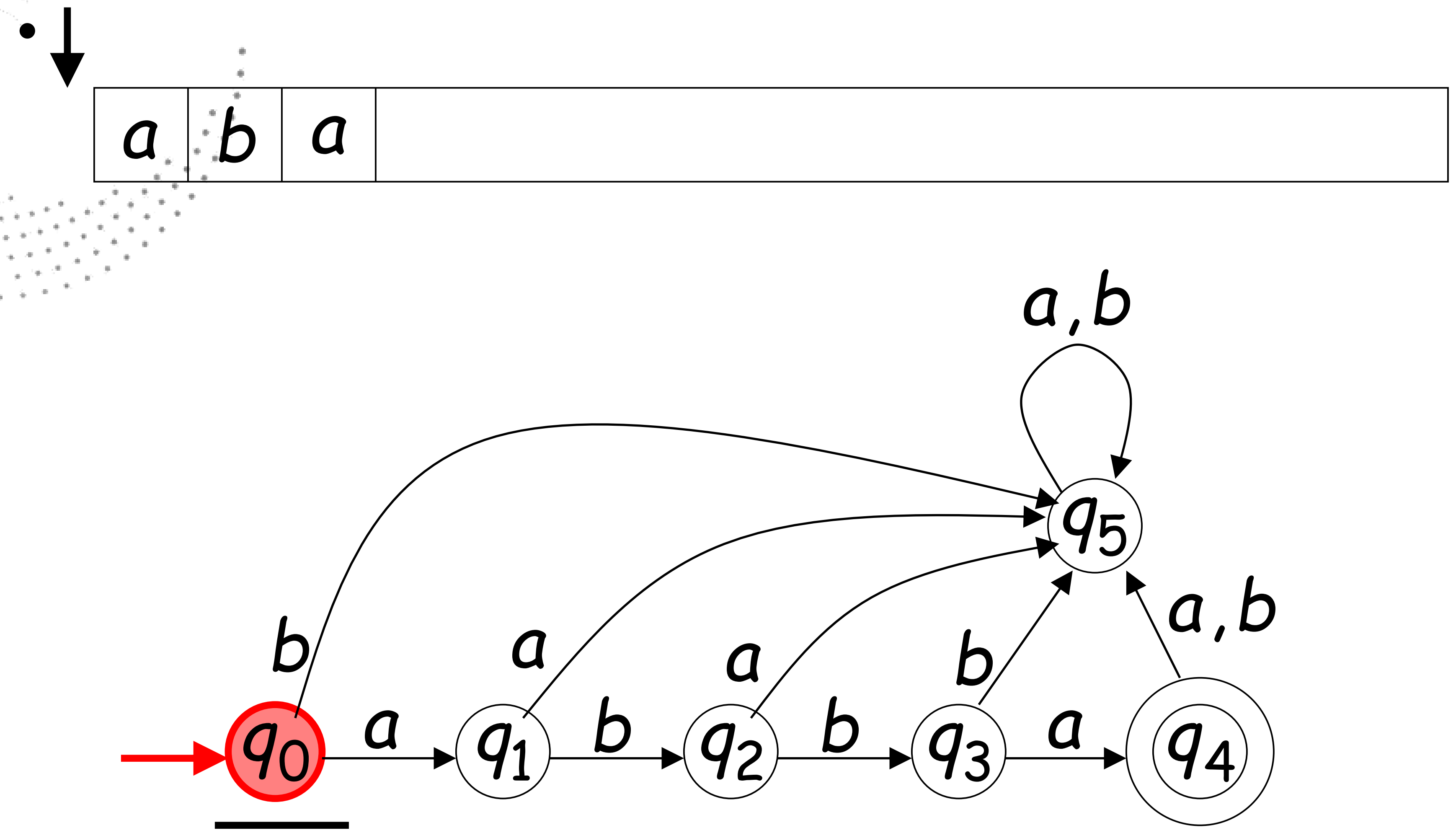

a
b
b
a
a,b
a,b
b
a
a
b
b
a
b
b
a
q0
q1
q2
q3
q4
q5

Input finished

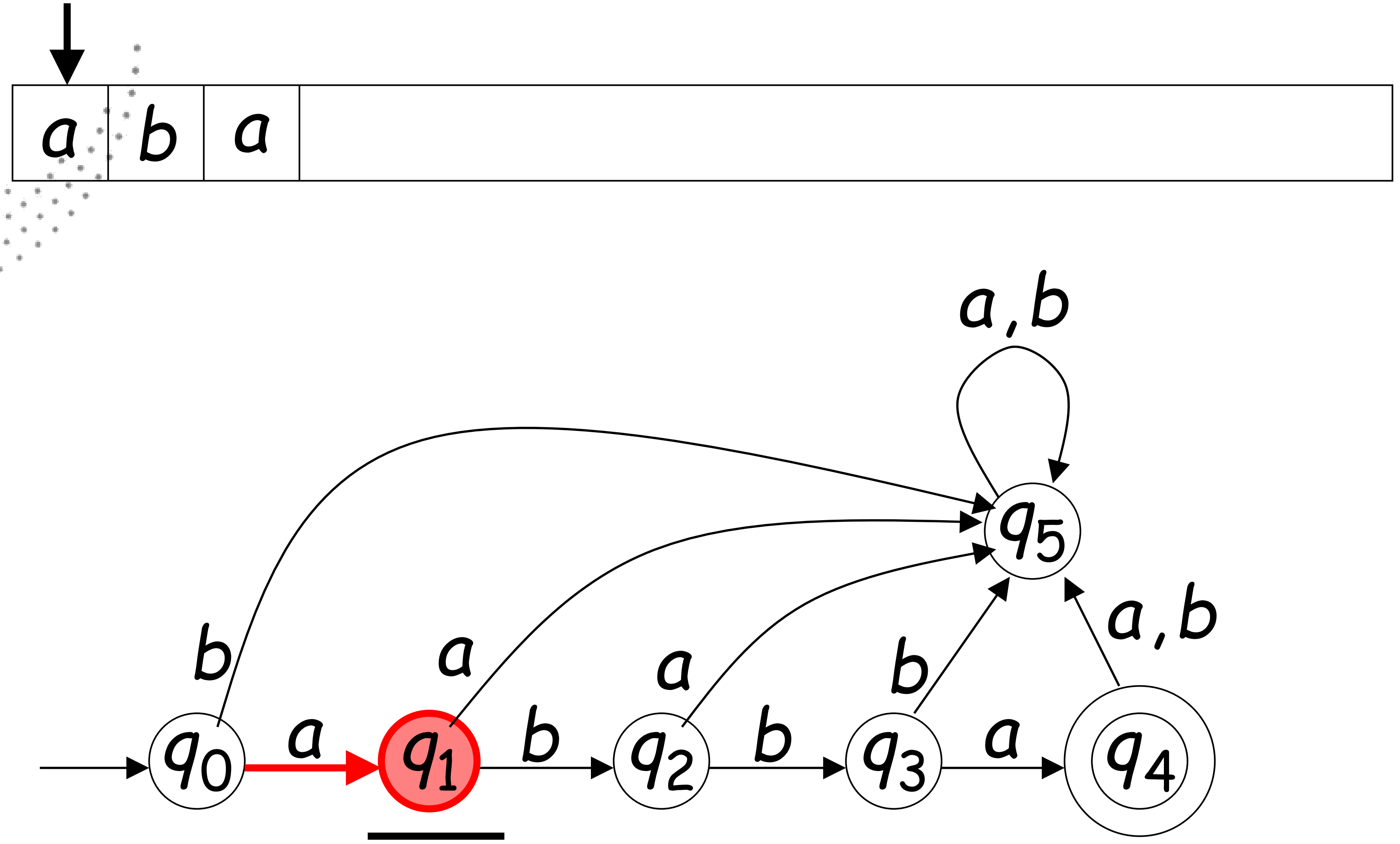


Output: "accept"

Rejection

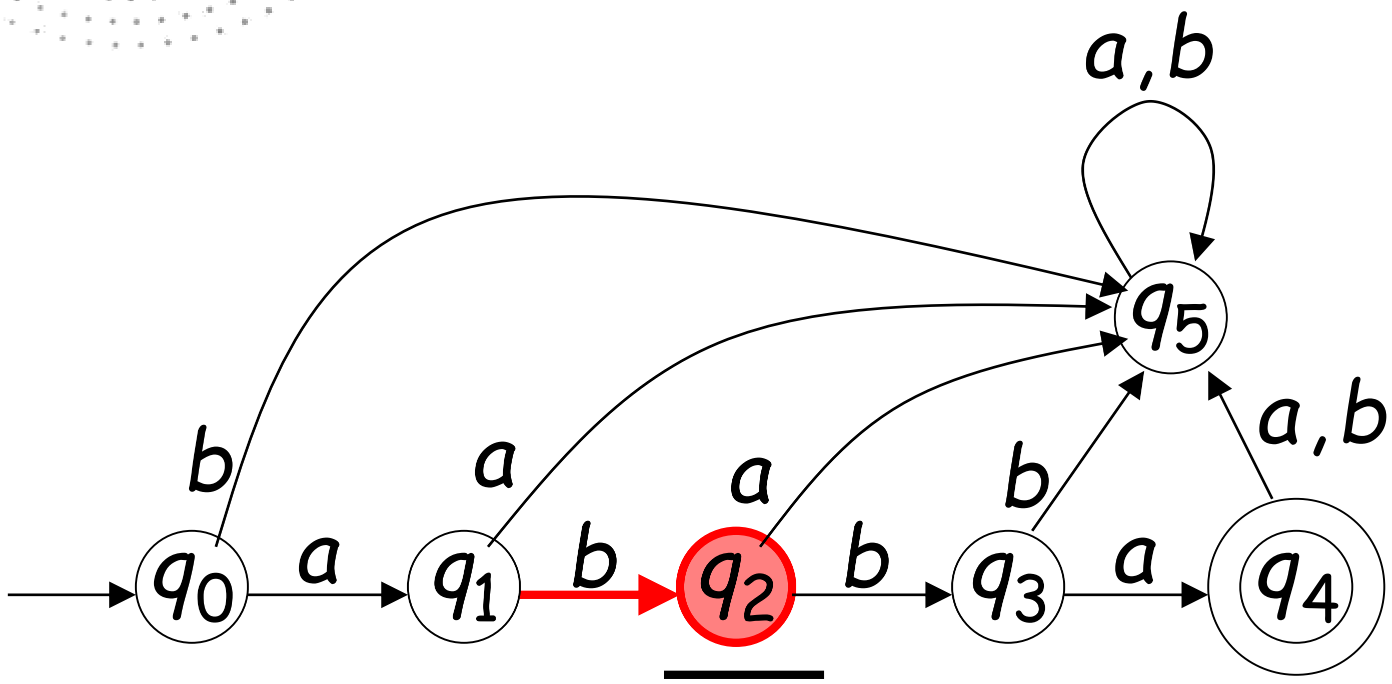


•

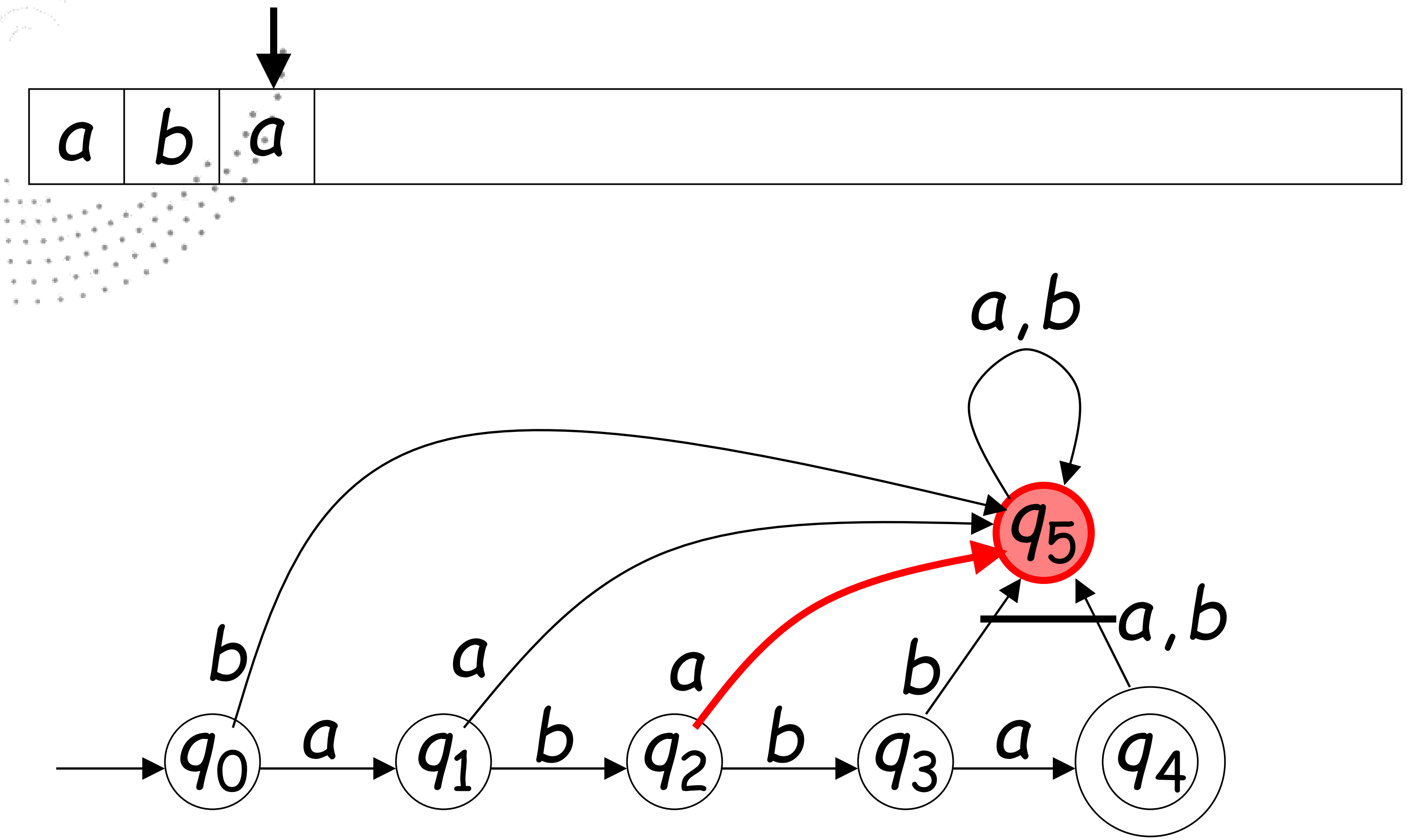

a
b
a
a,b
b
a
a
b
a,b
a
b
b
a
q_0
q_1
q_2
q_3
q_4
q_5

•

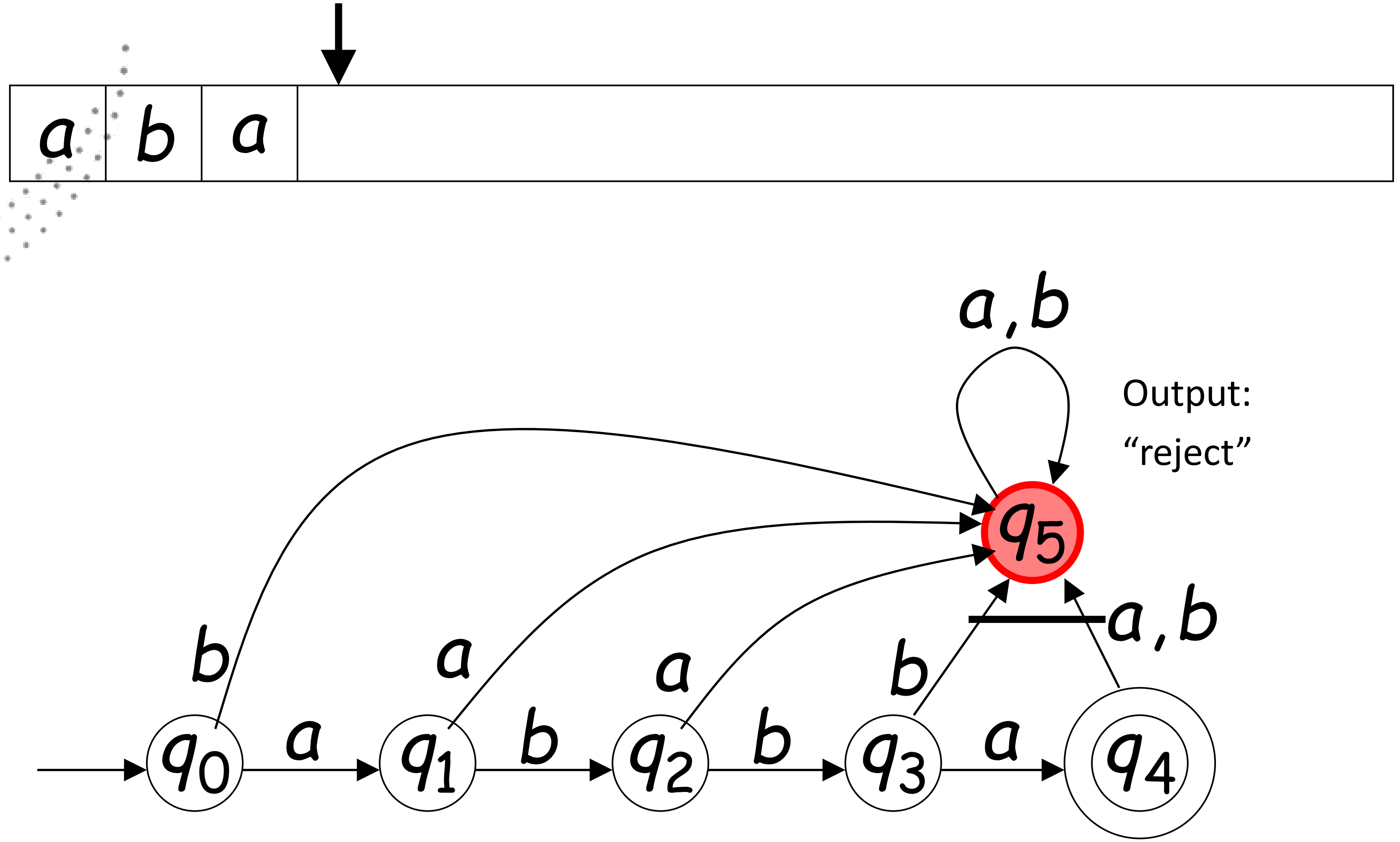

a
b
a


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

•

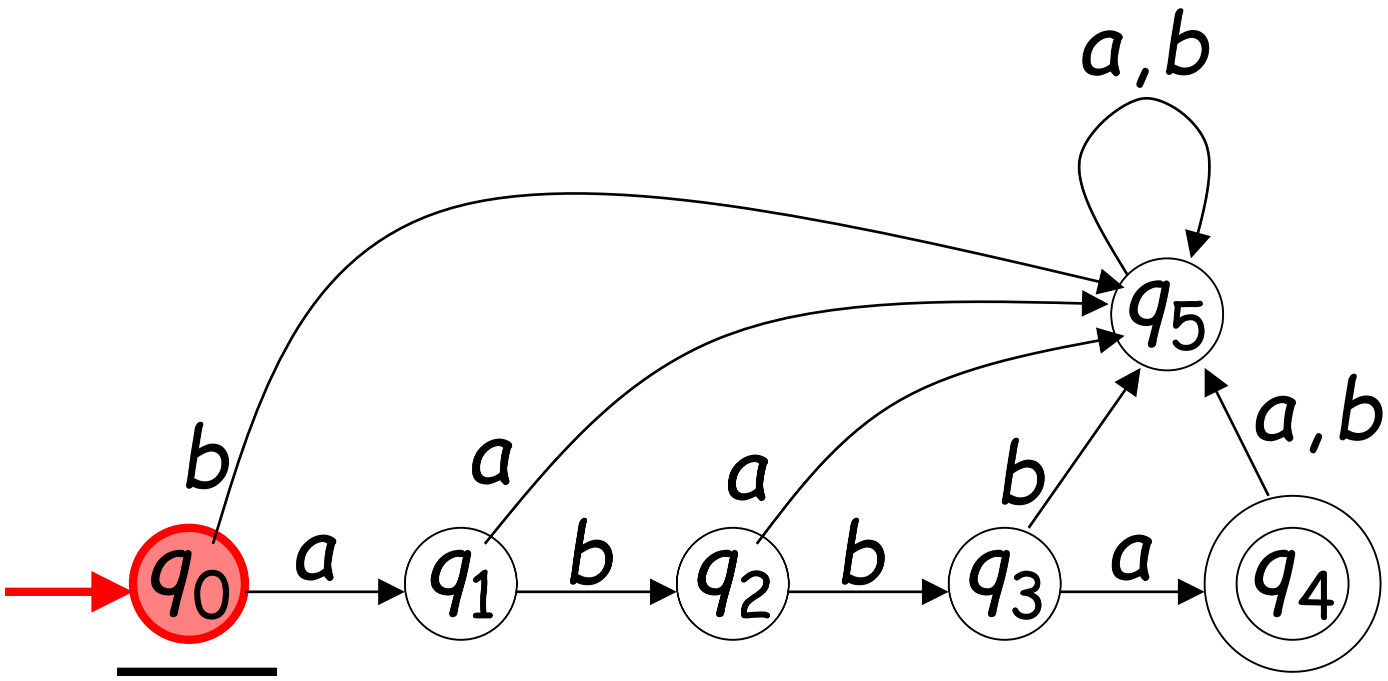

a
b
a
a,b
q5
a,b
b
a
a
b
a
b
b
a
q0
q1
q2
q3
q4

Input finished

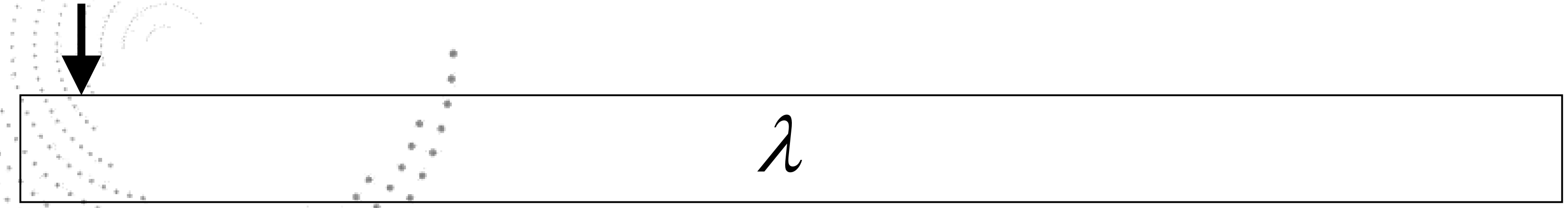


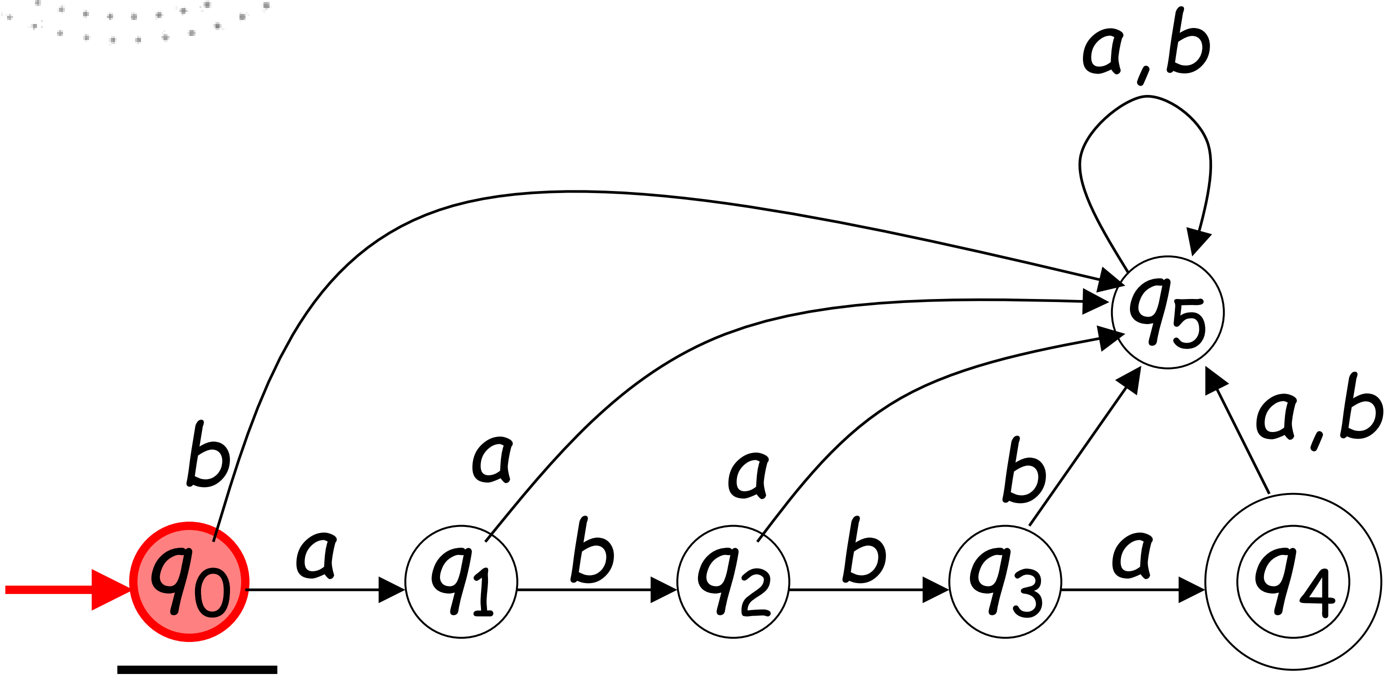
Another Rejection

- λ



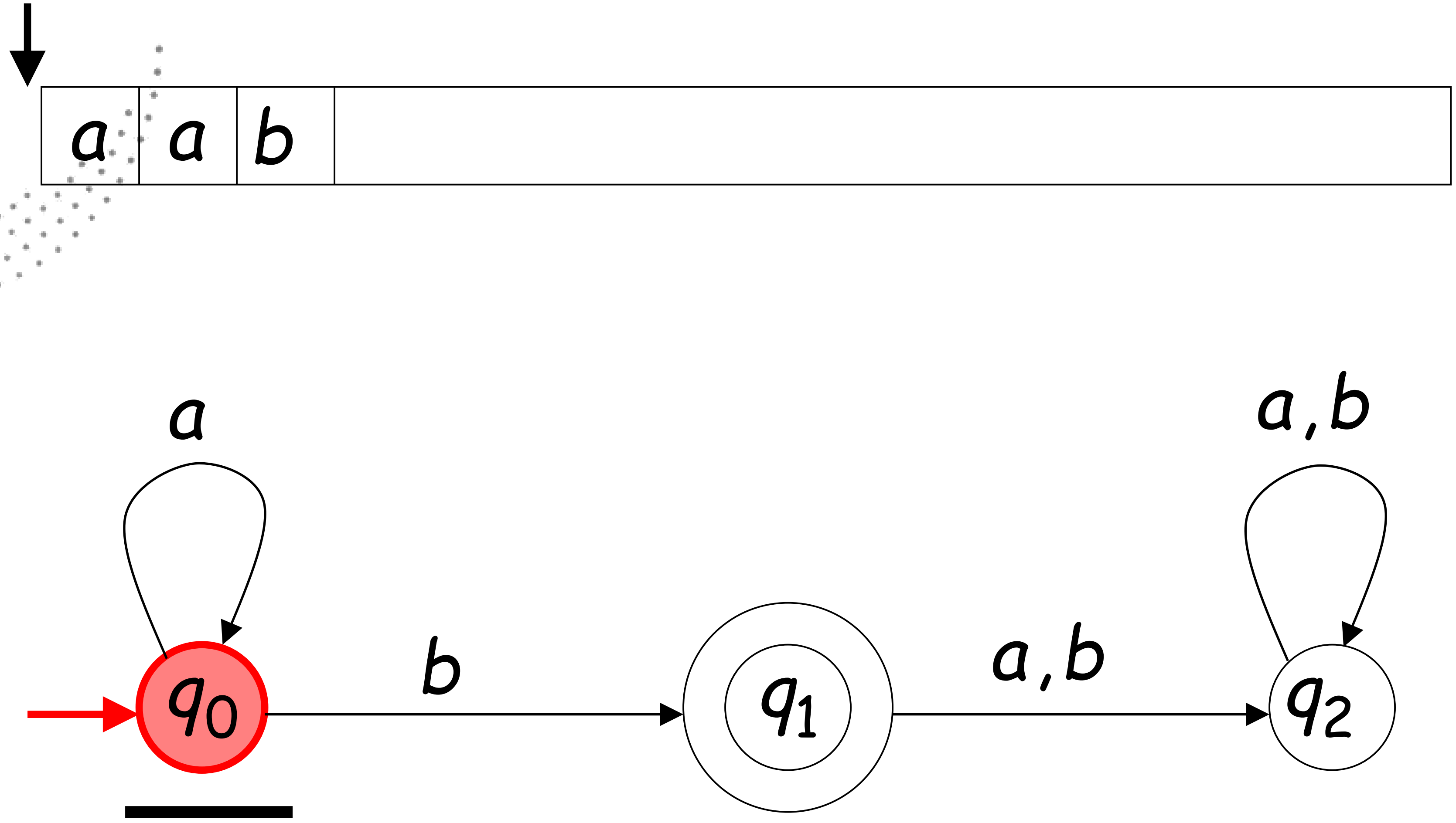
•

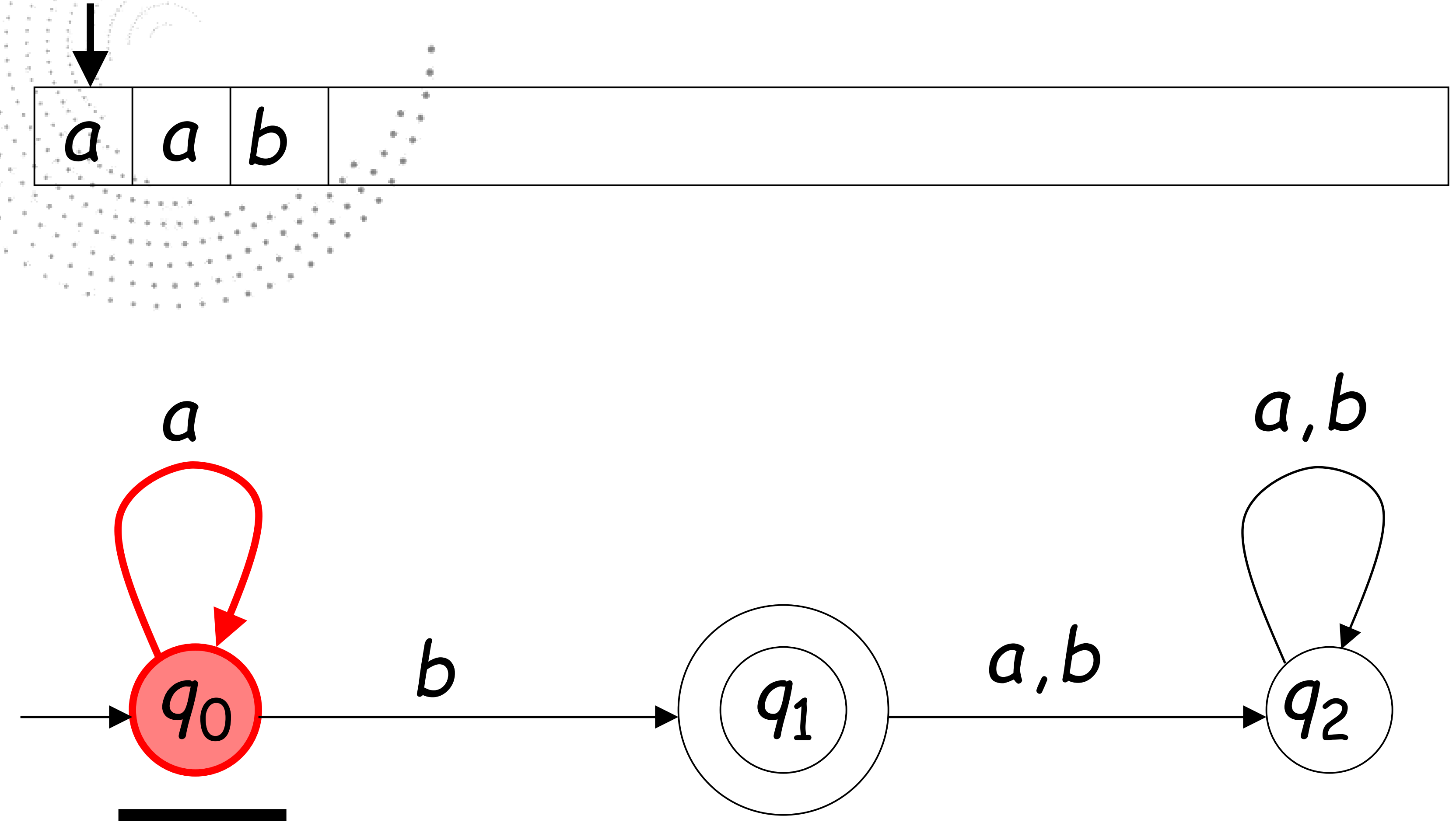


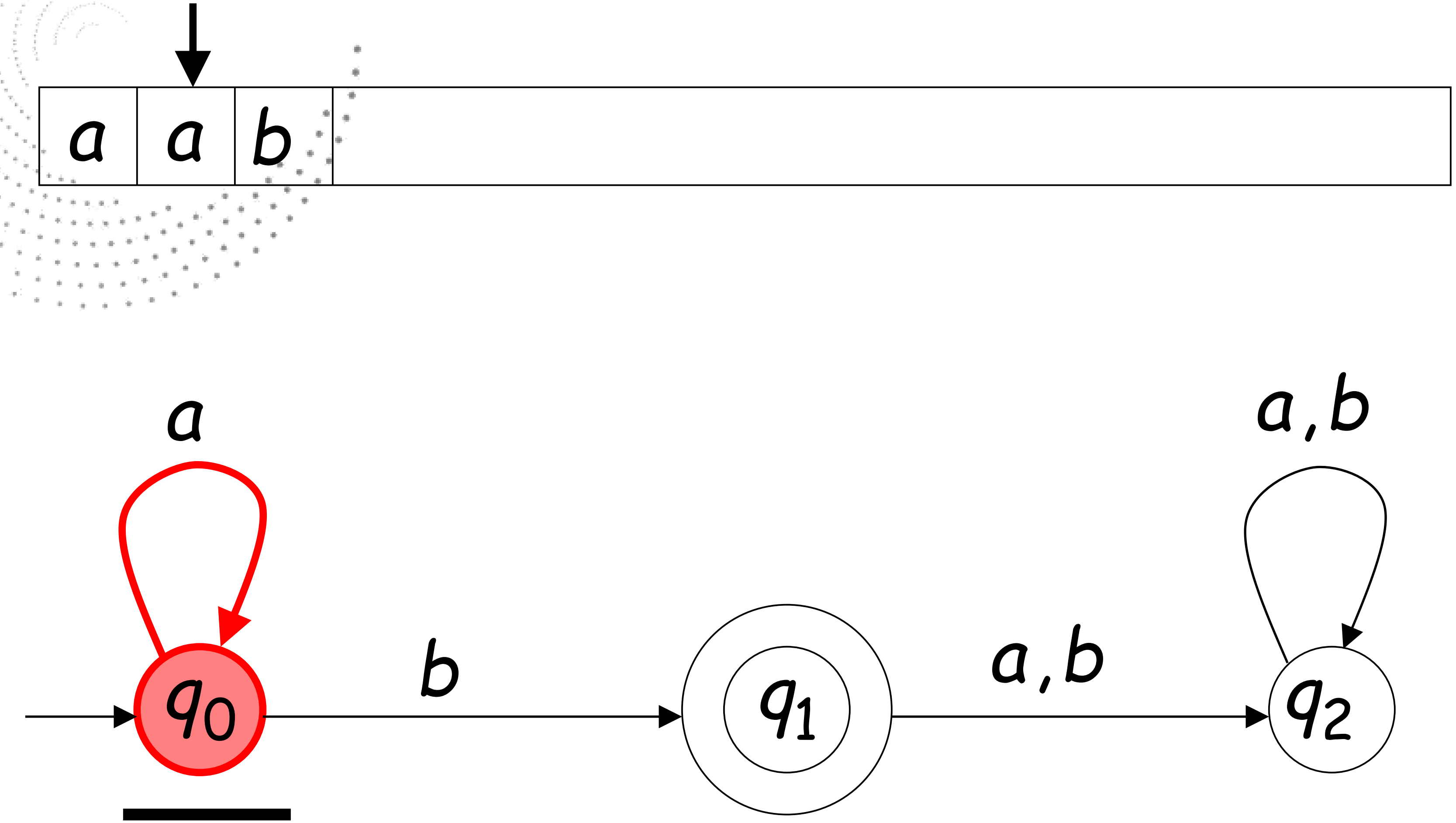


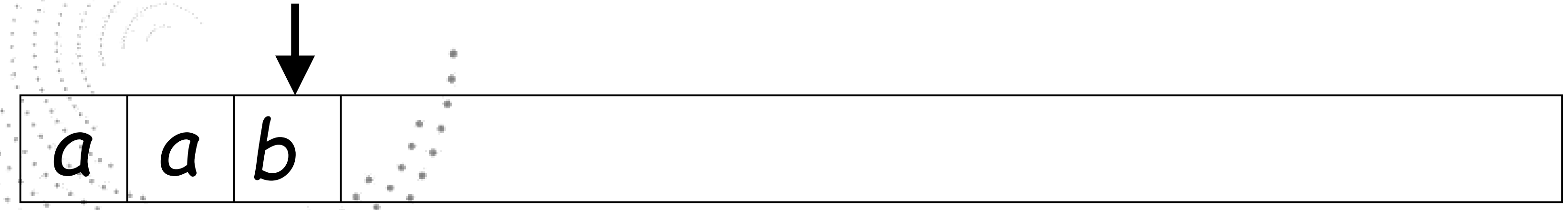
Output:
“reject”

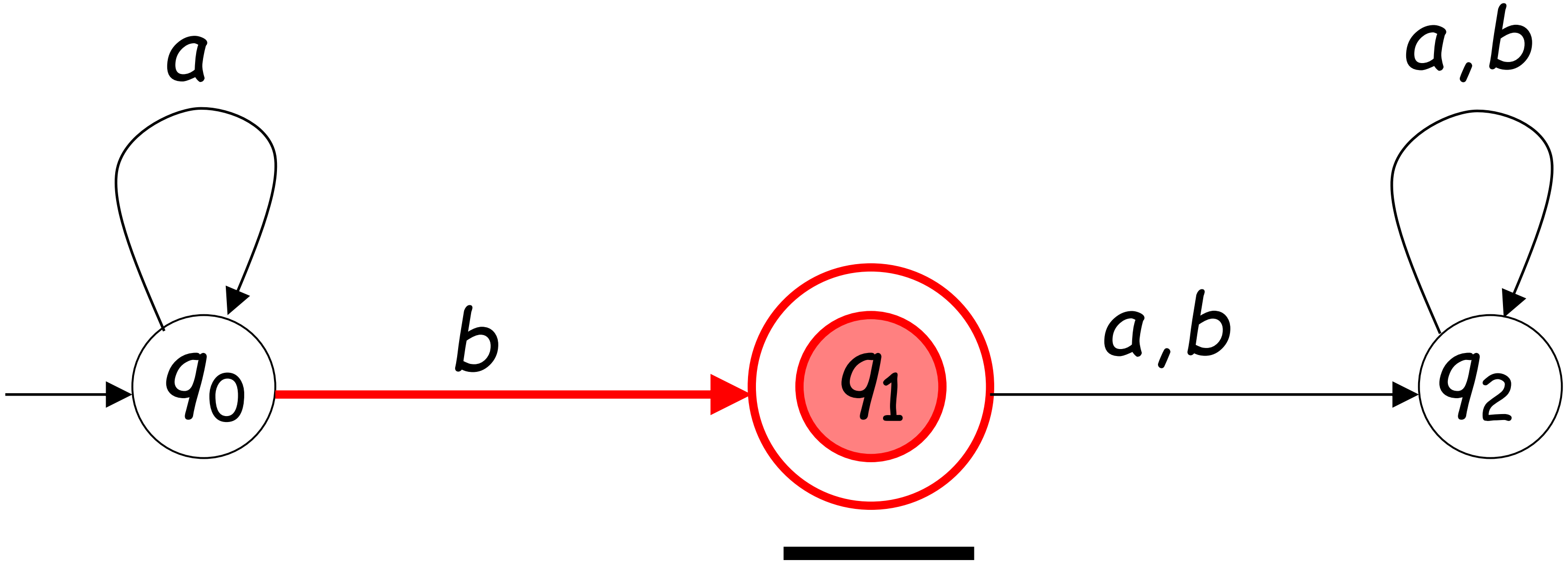
Another Example



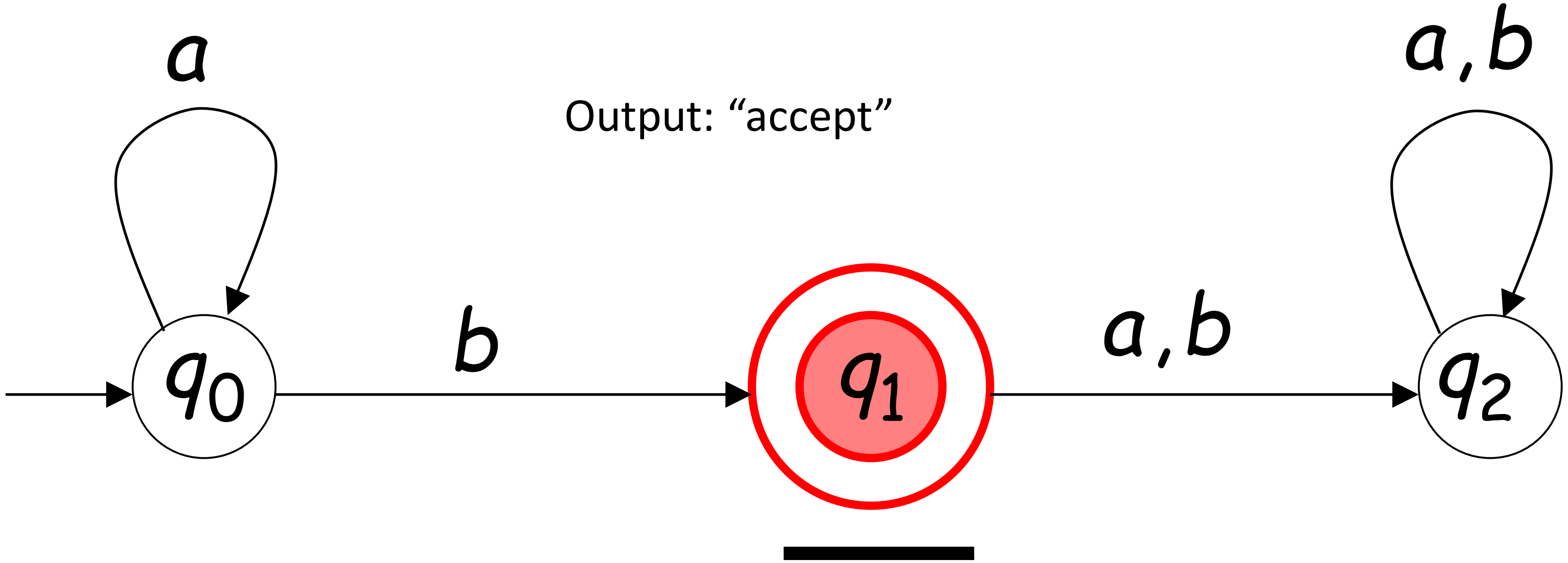

a
a
b
a
b
a,b
a,b
q_0
q_1
q_2


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

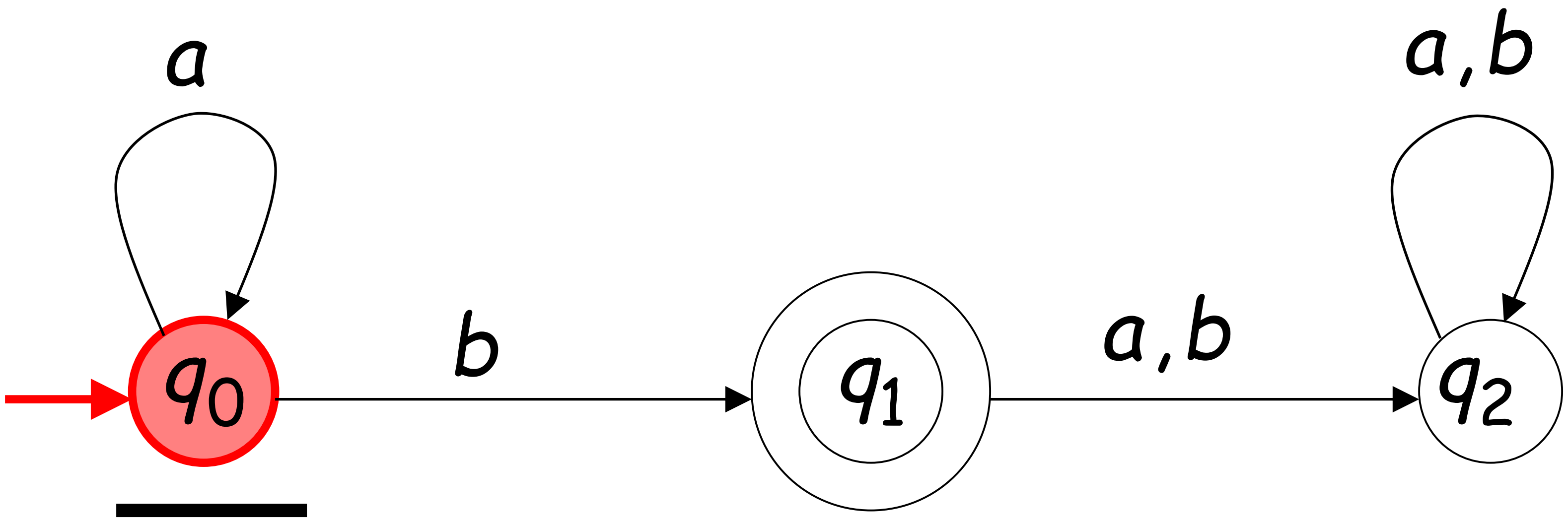

a
a
b

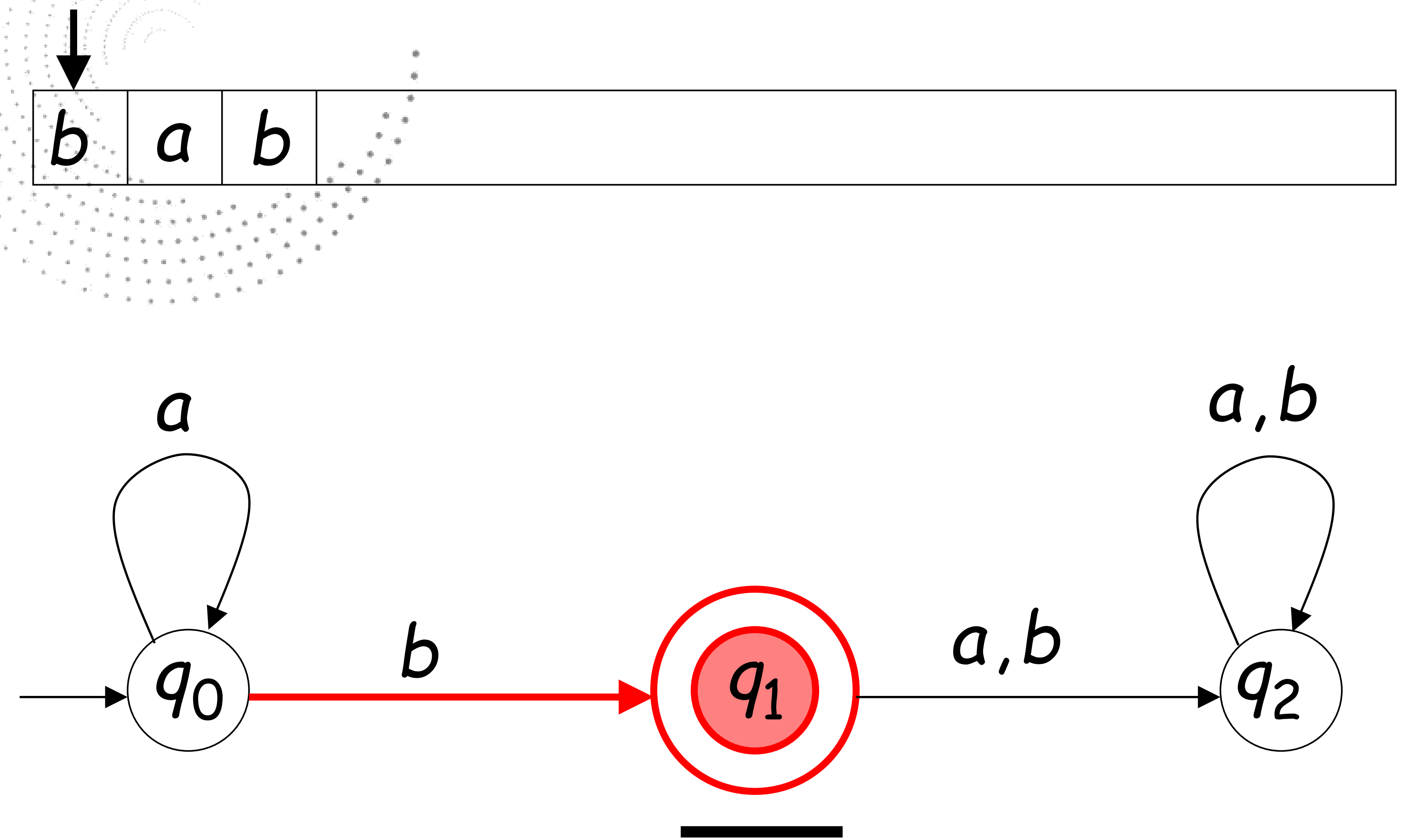

a
a,b
b
a,b
q_0
q_1
q_2

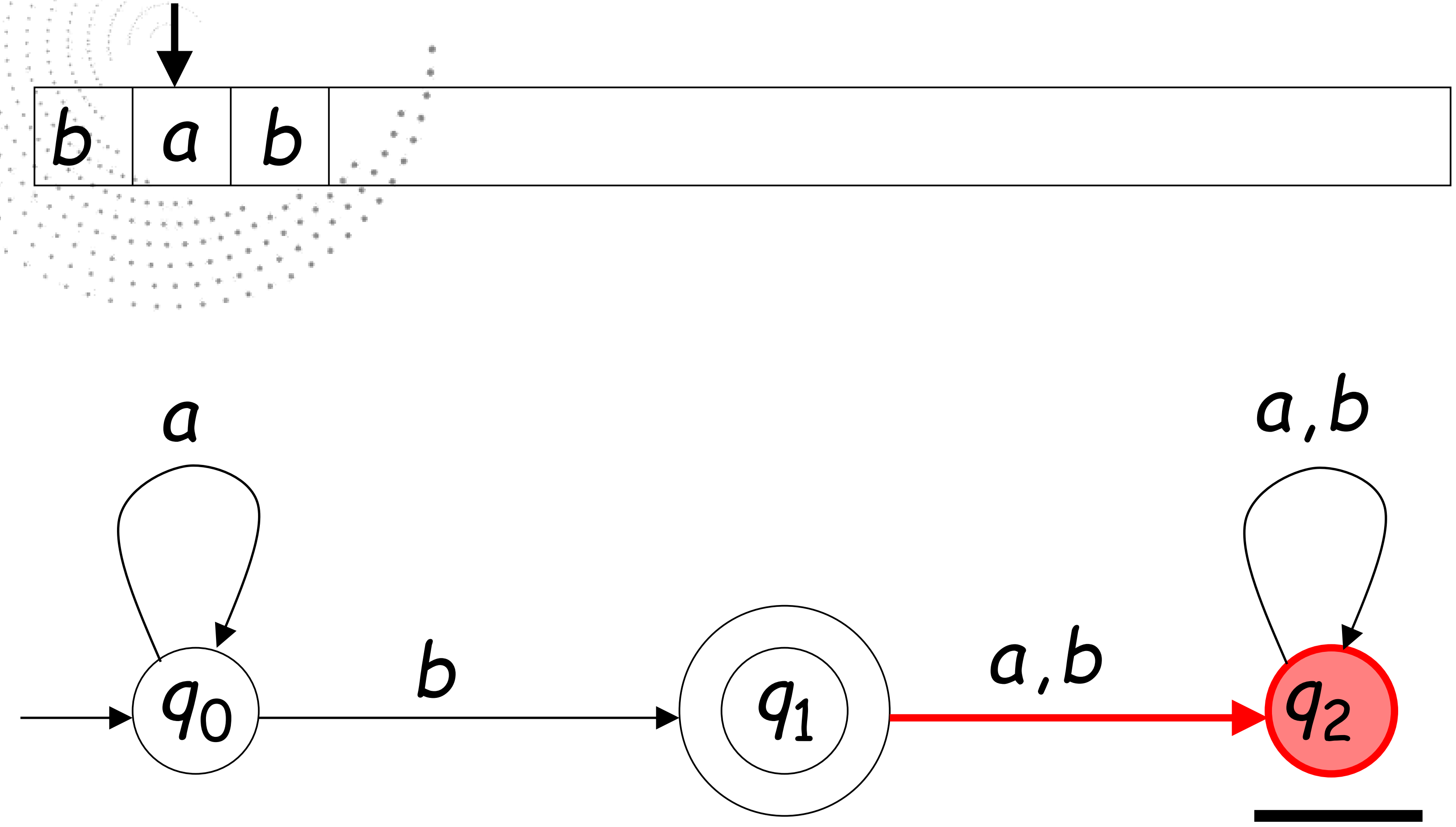
Input finished

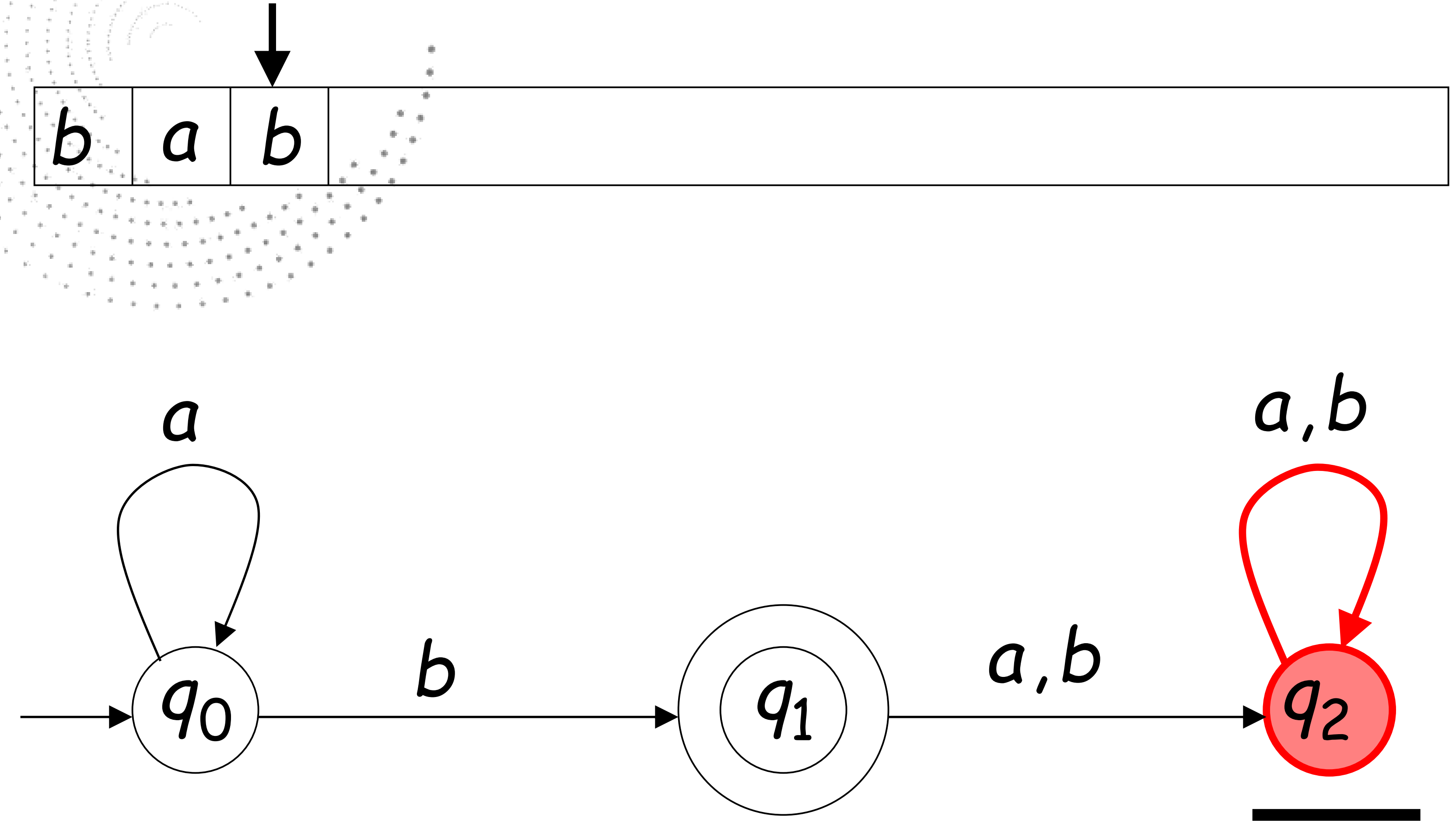


Rejection

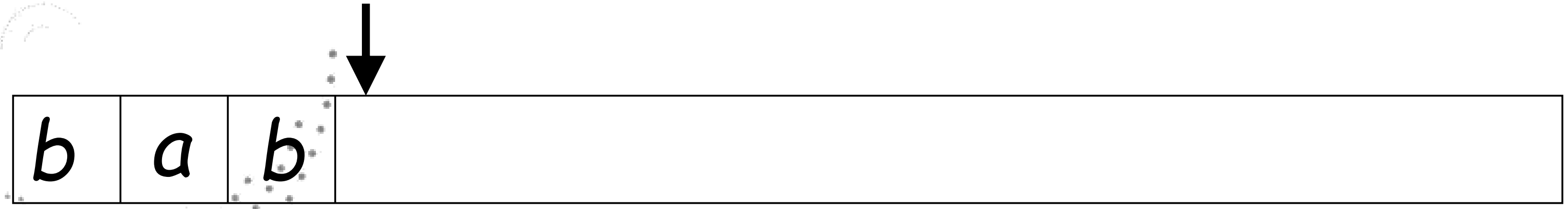


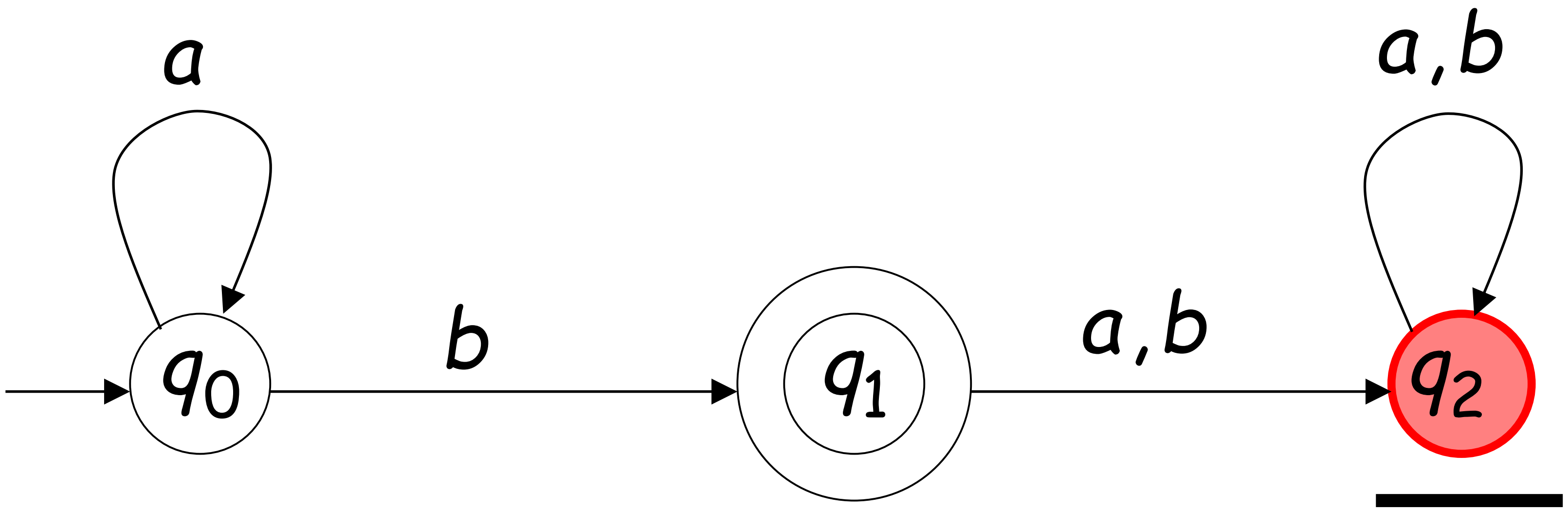

b
a
b
a
a,b
b
a,b
q0
q1
q2


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”

Formalities

- Deterministic Finite Accepter (DFA)

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

Q : set of states

Σ : input alphabet

δ : transition function

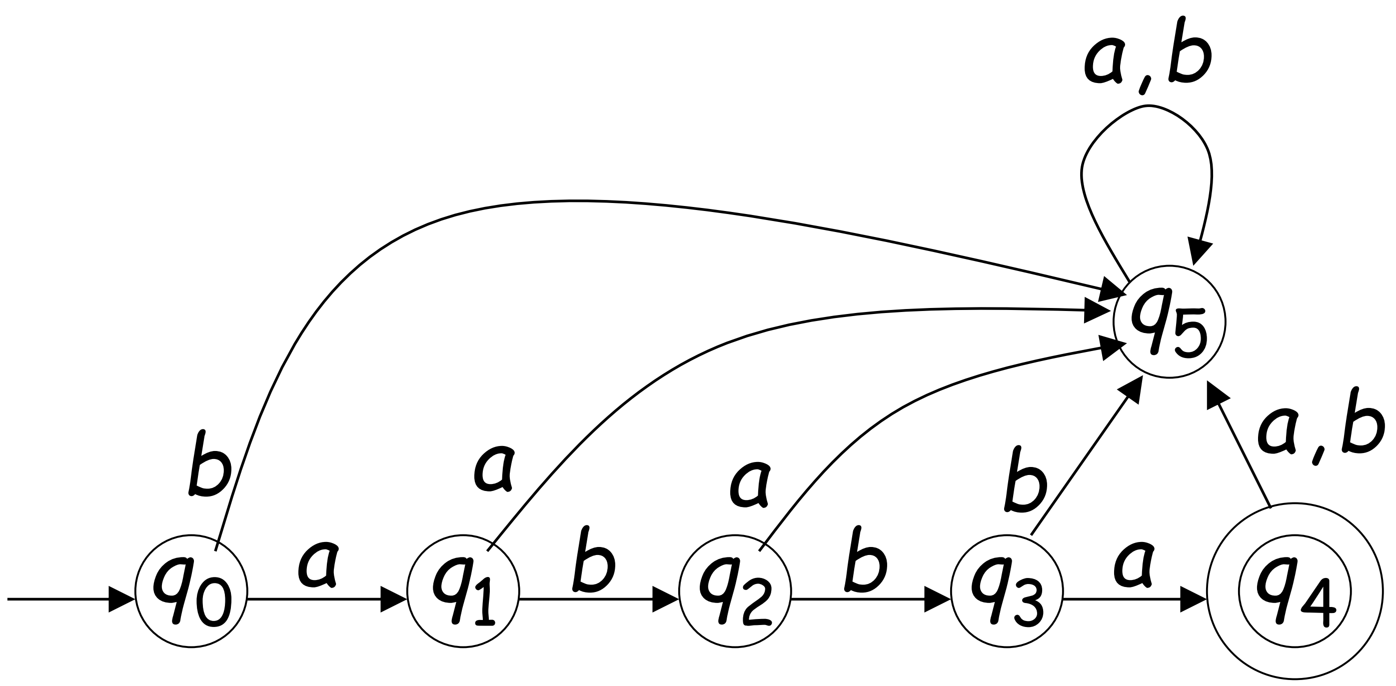
q_0 : initial state $\qquad q_0 \in Q$

F : set of final states $\qquad F \subseteq Q$

Input Alphabet Σ

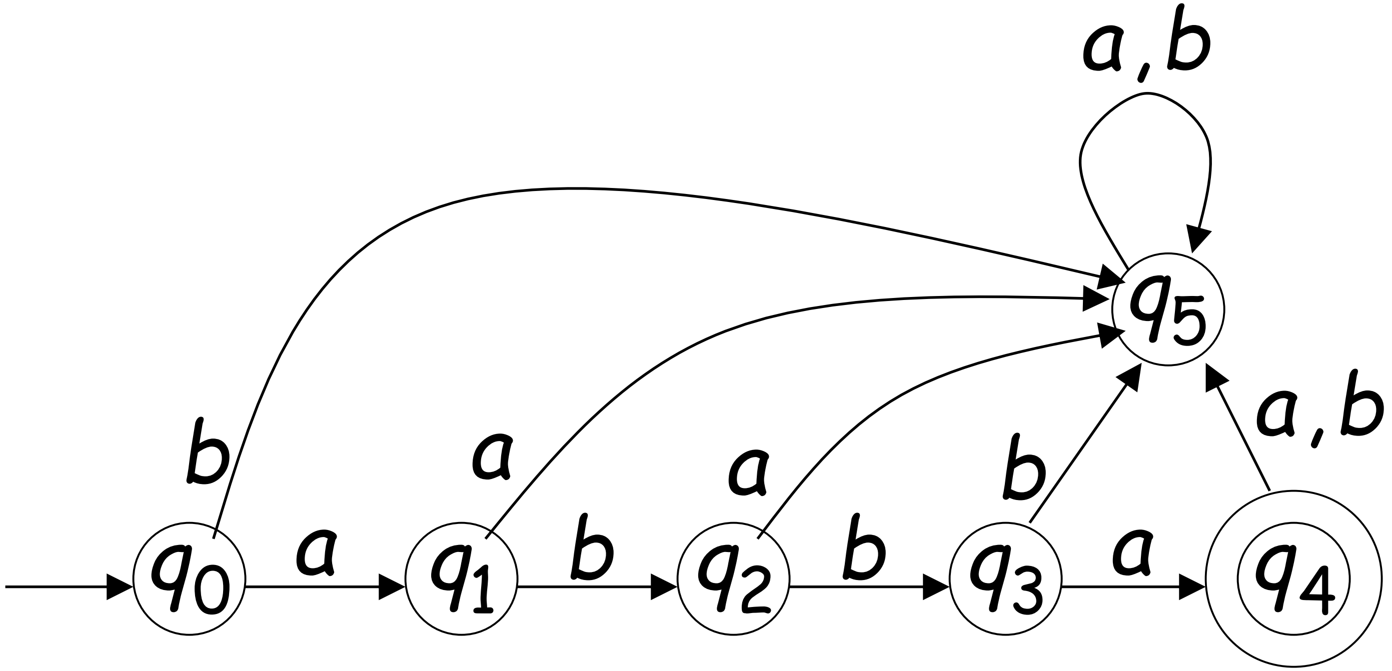
-

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



Set of States Q

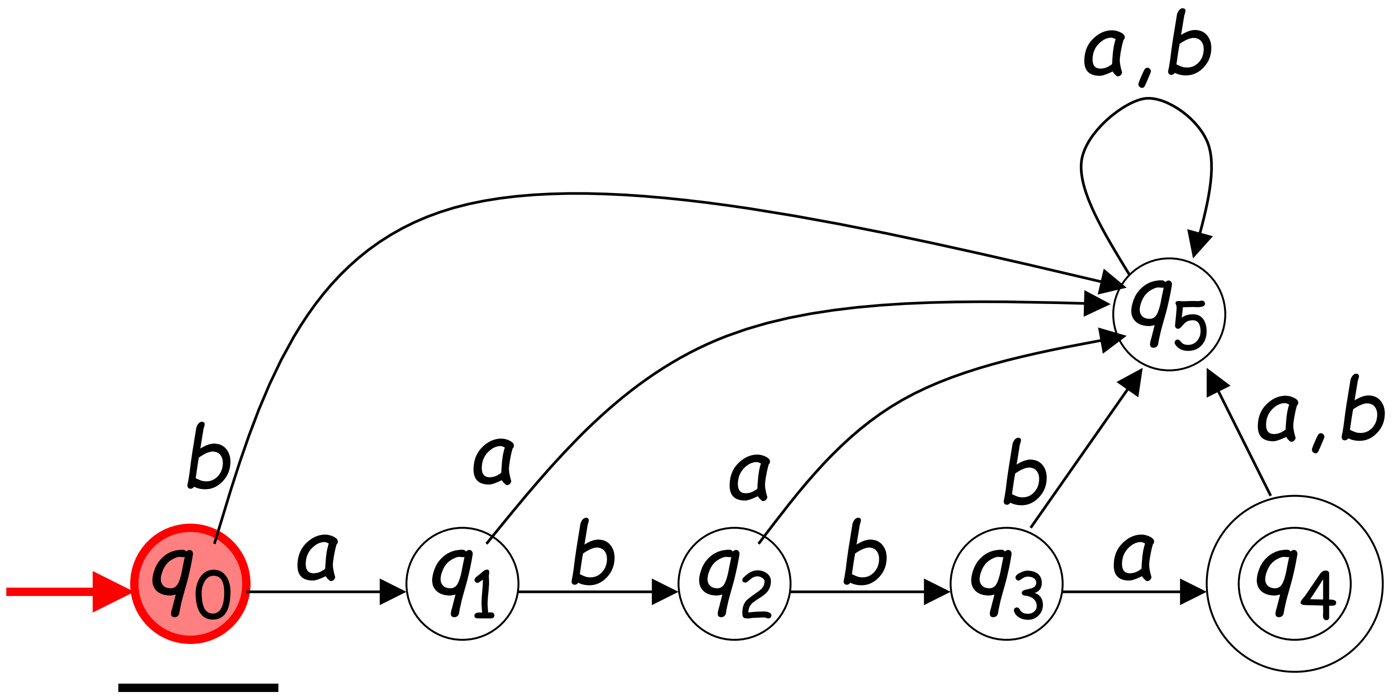
- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$



Initial State

q_0

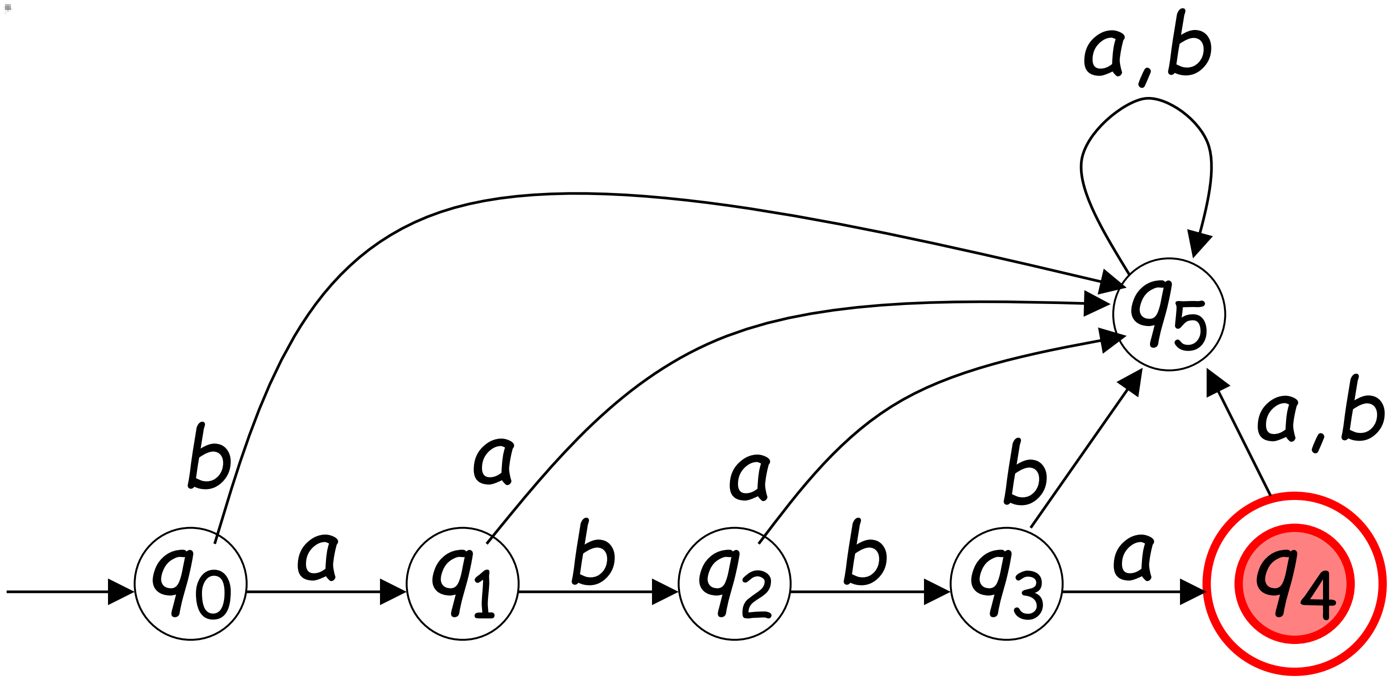
•



Set of Final States

F

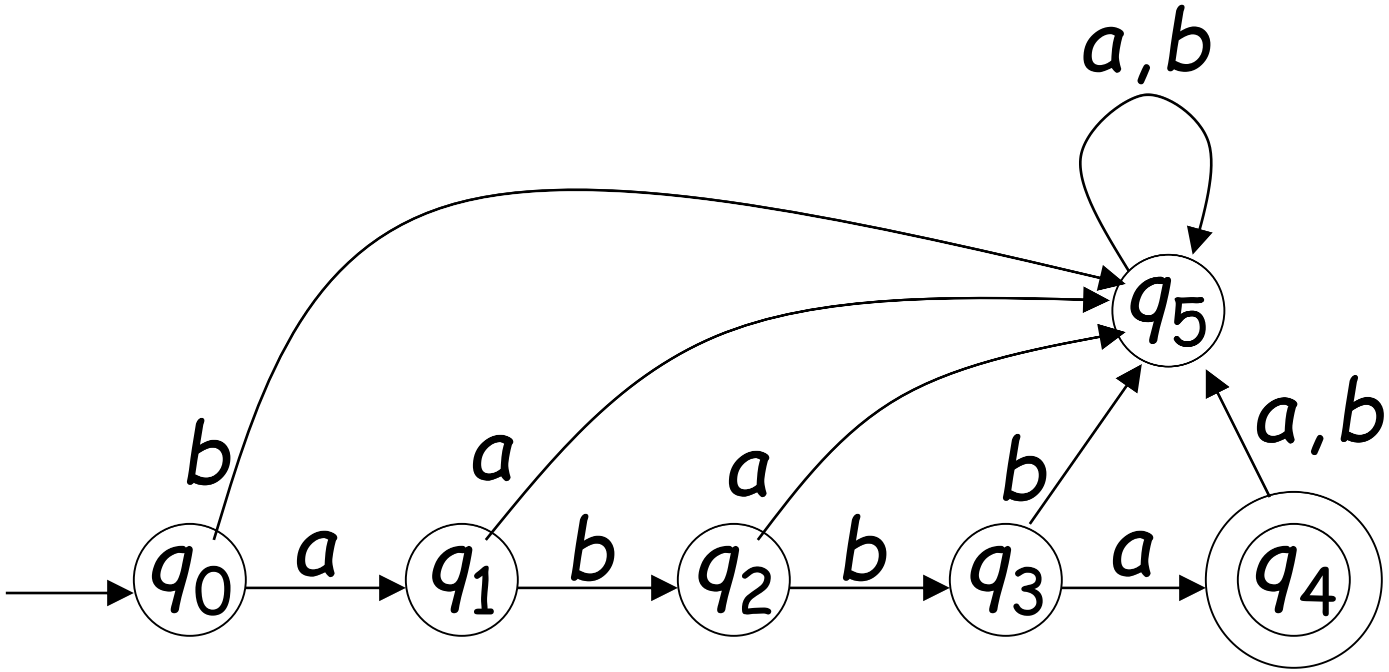
- $$F = \{q_4\}$$



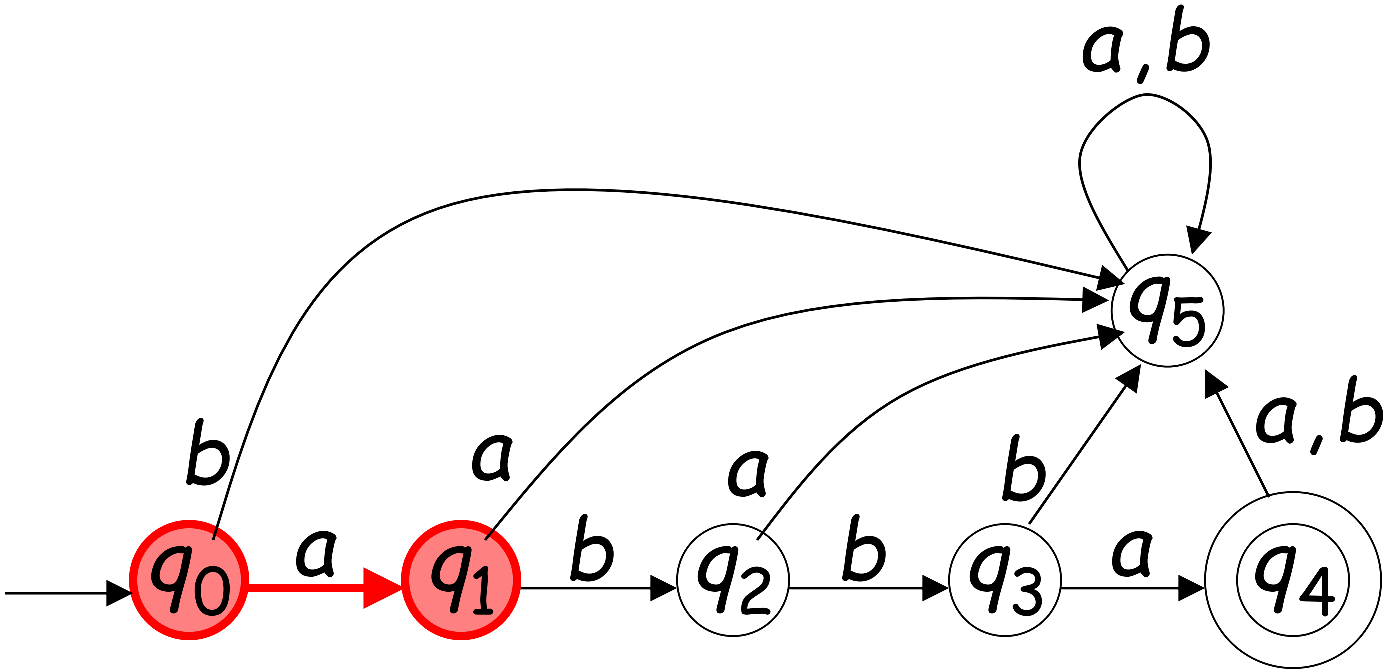
Transition Function δ

•

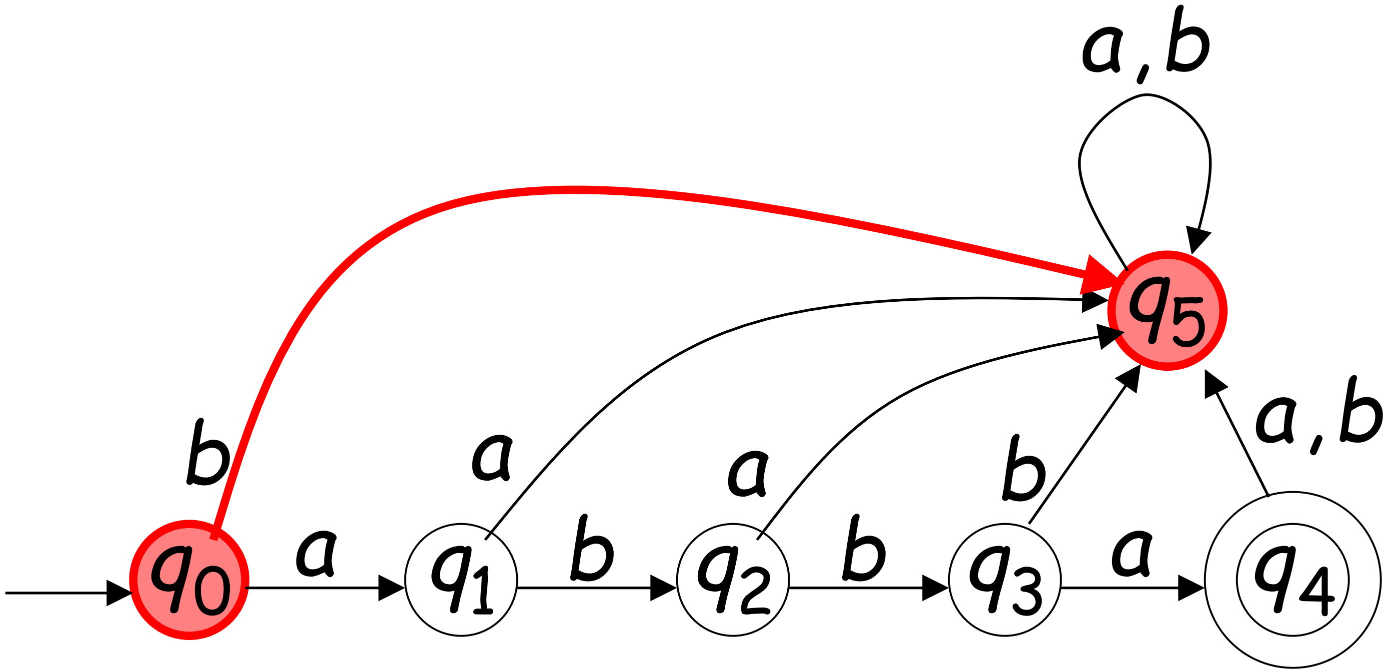
$$\delta : Q \times \Sigma \rightarrow Q$$



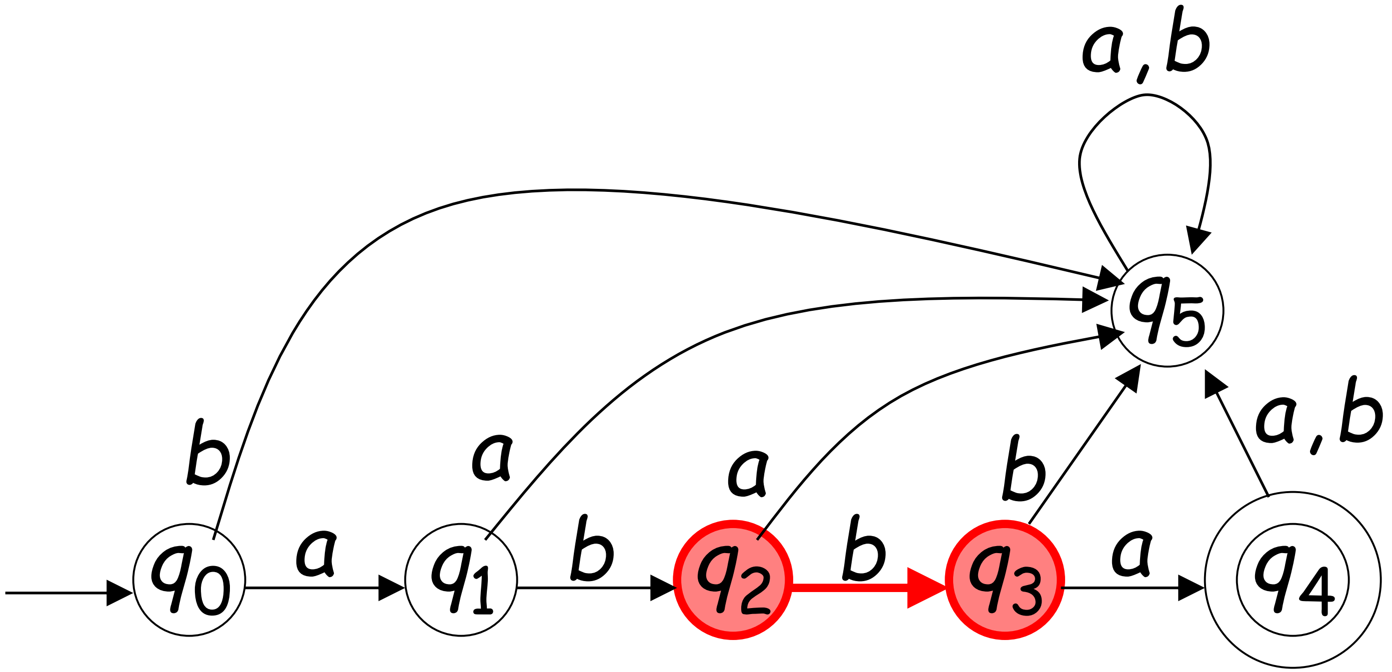
$$\delta(q_0, a) = q_1$$



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

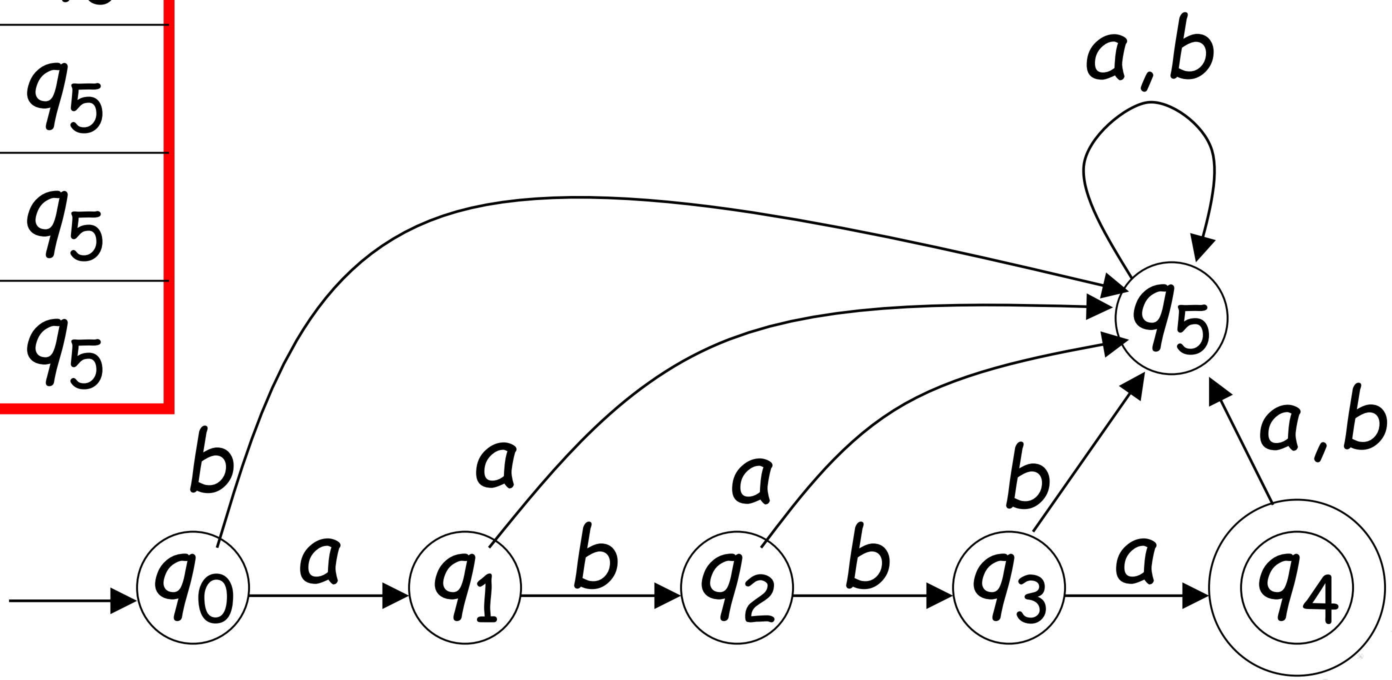


$$\delta(q_2, b) = q_3$$



Transition Function δ

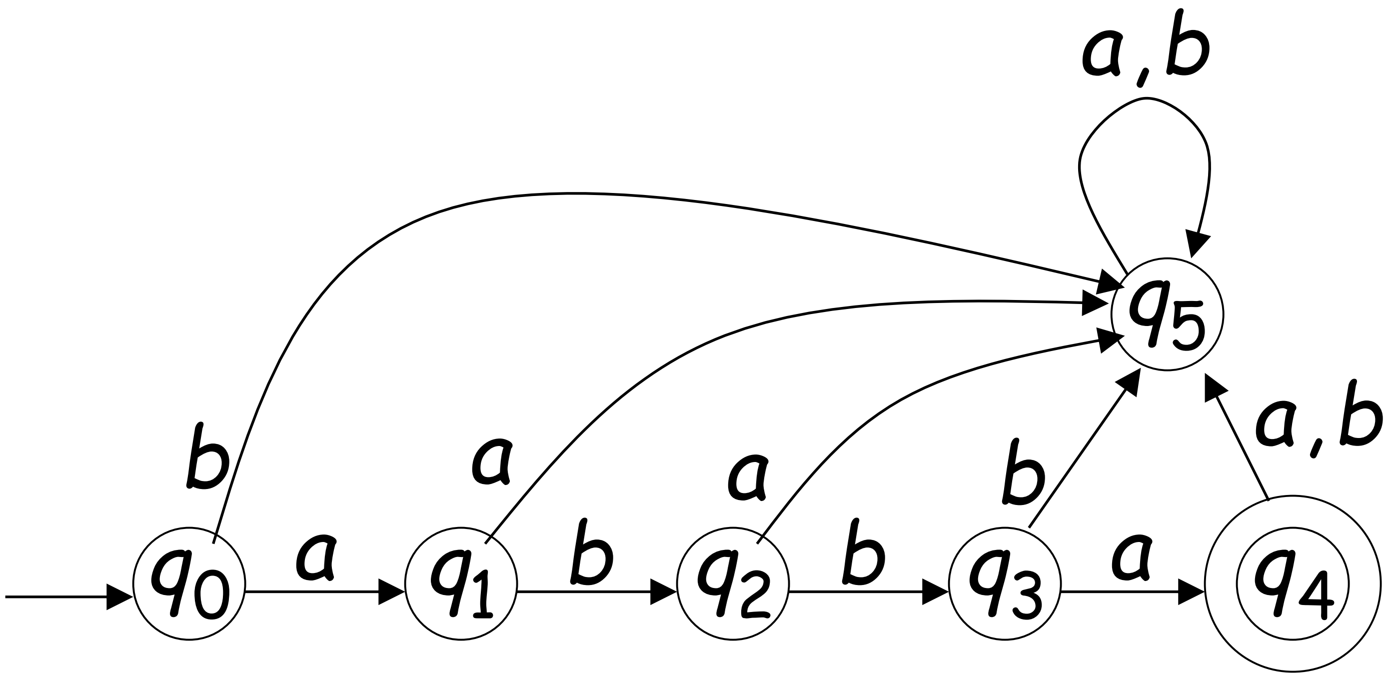
δ	a	b
q_0	q_1	q_5
q_1	q_5	q_2
q_2	q_5	q_3
q_3	q_4	q_5
q_4	q_5	q_5
q_5	q_5	q_5



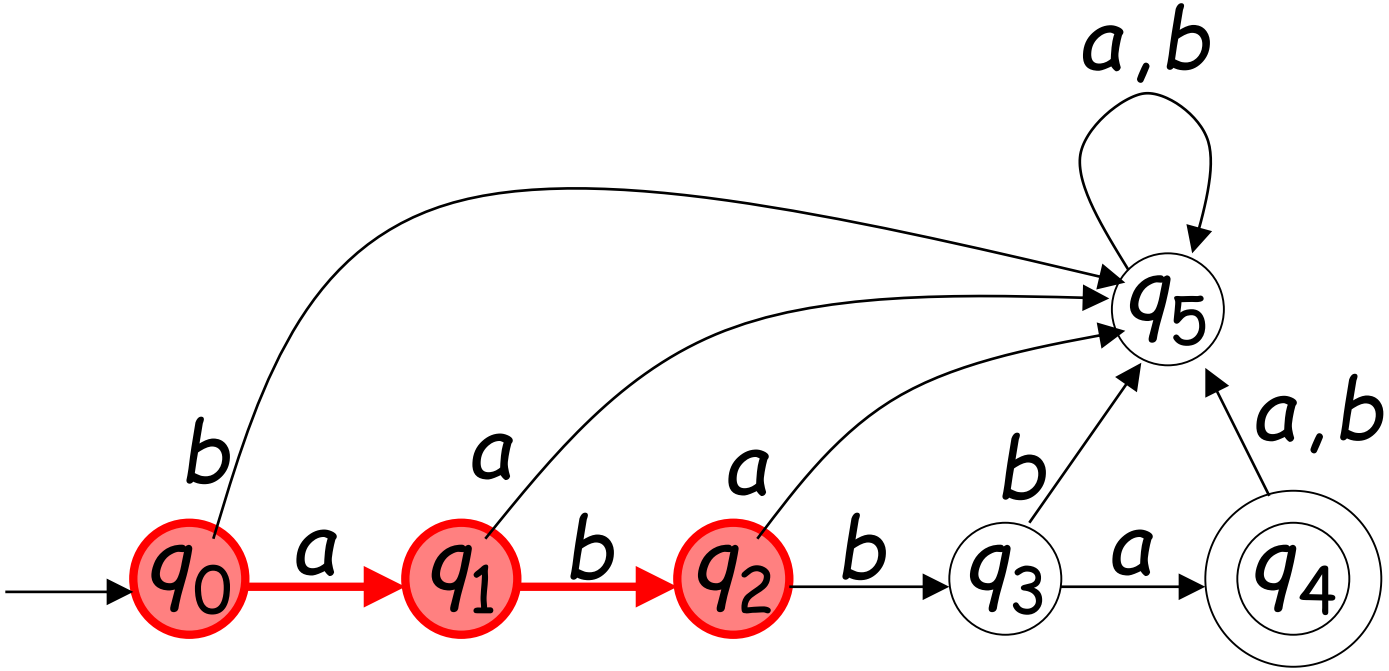
Extended Transition Function

- $$\delta * or \ \hat{\delta}$$

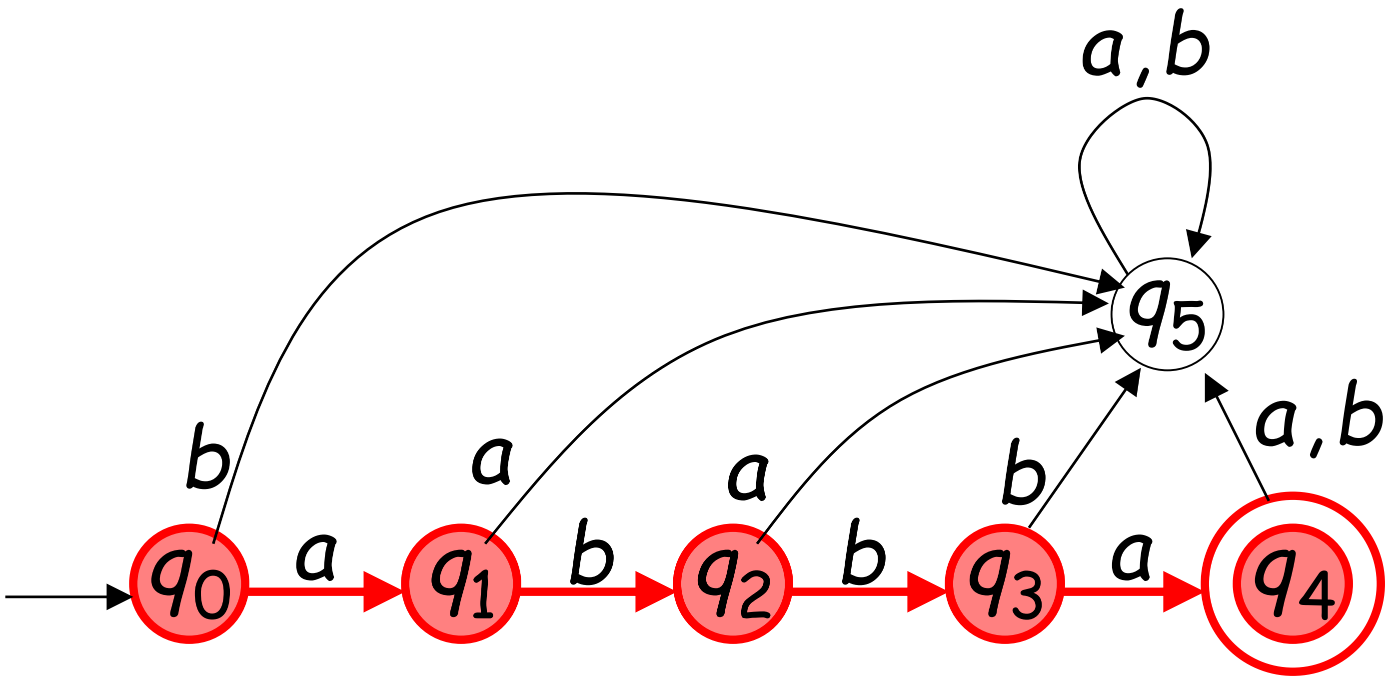
$$\delta^* : Q \times \Sigma^* \to Q$$



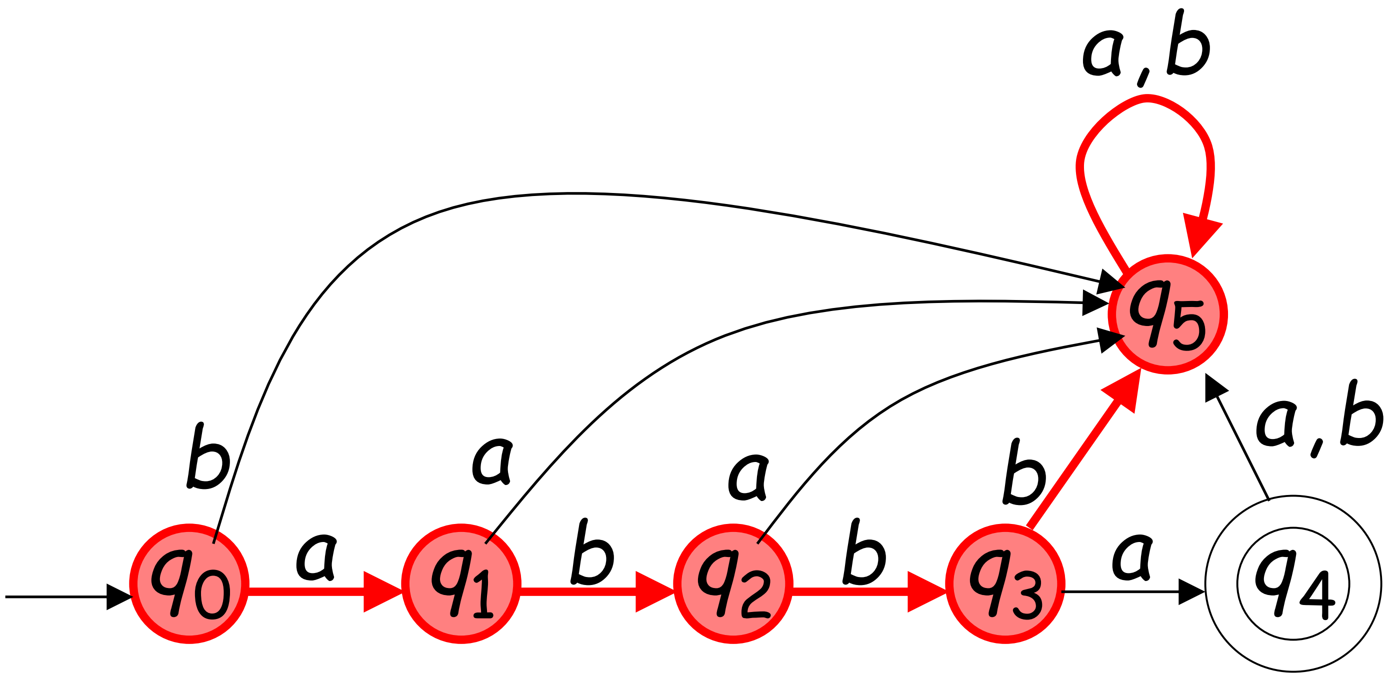
$$\delta^*(q_0, ab) = q_2$$



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

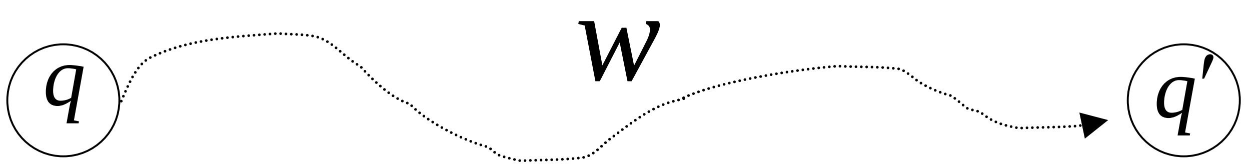


$$\delta^*(q_0, abbbaa) = q_5$$

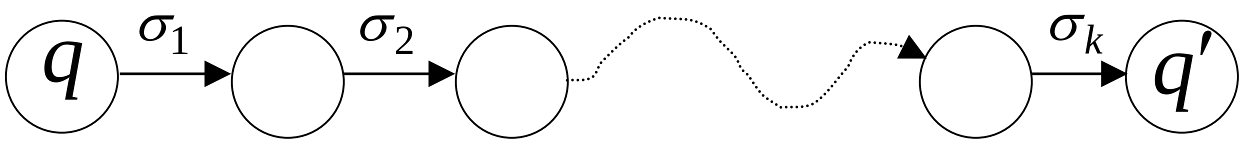


Observation: There is a walk from q to q' with label w

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

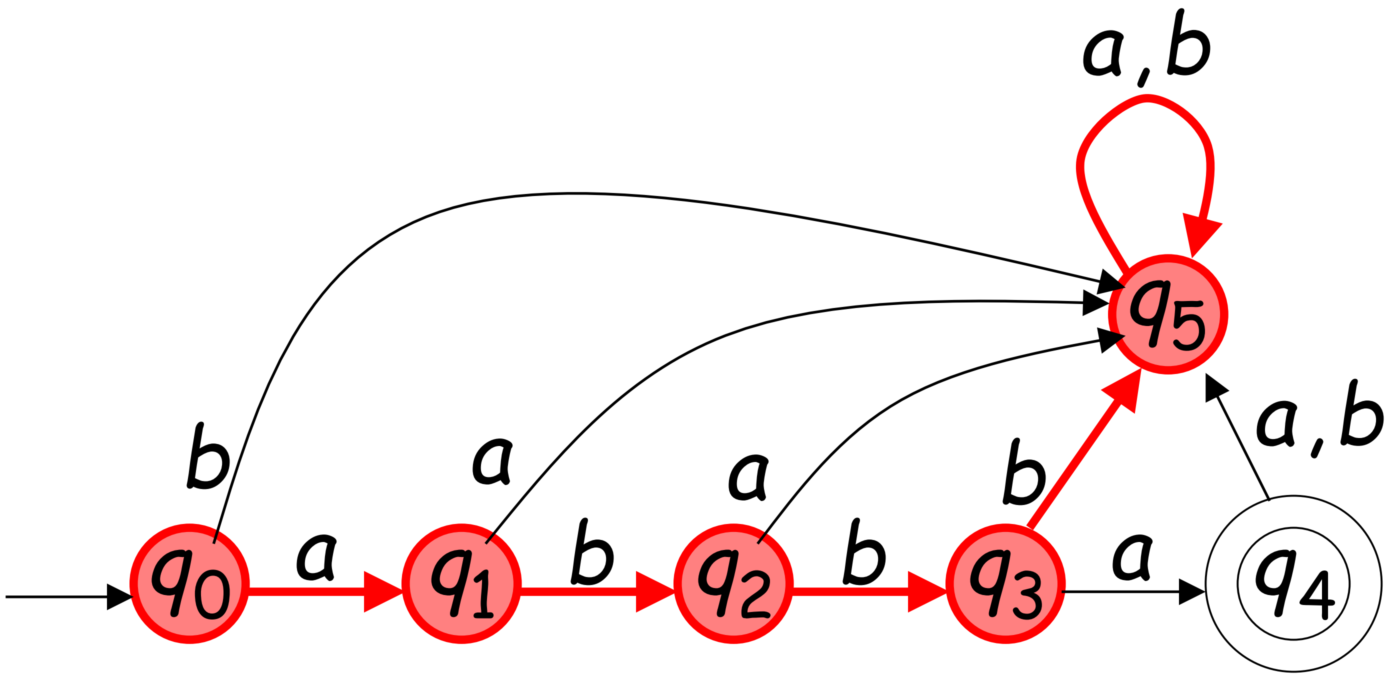


$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



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

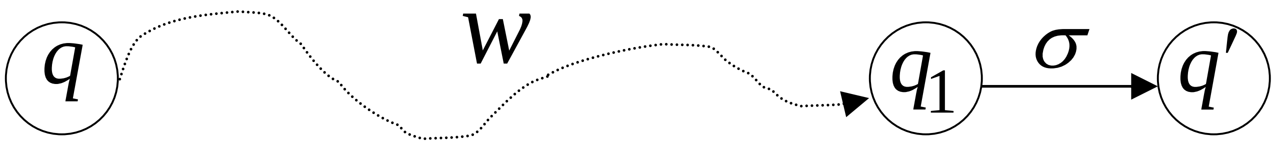
$$\delta^*(q_0, abbbaa) = q_5$$



Inductive Definition

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

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



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

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

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

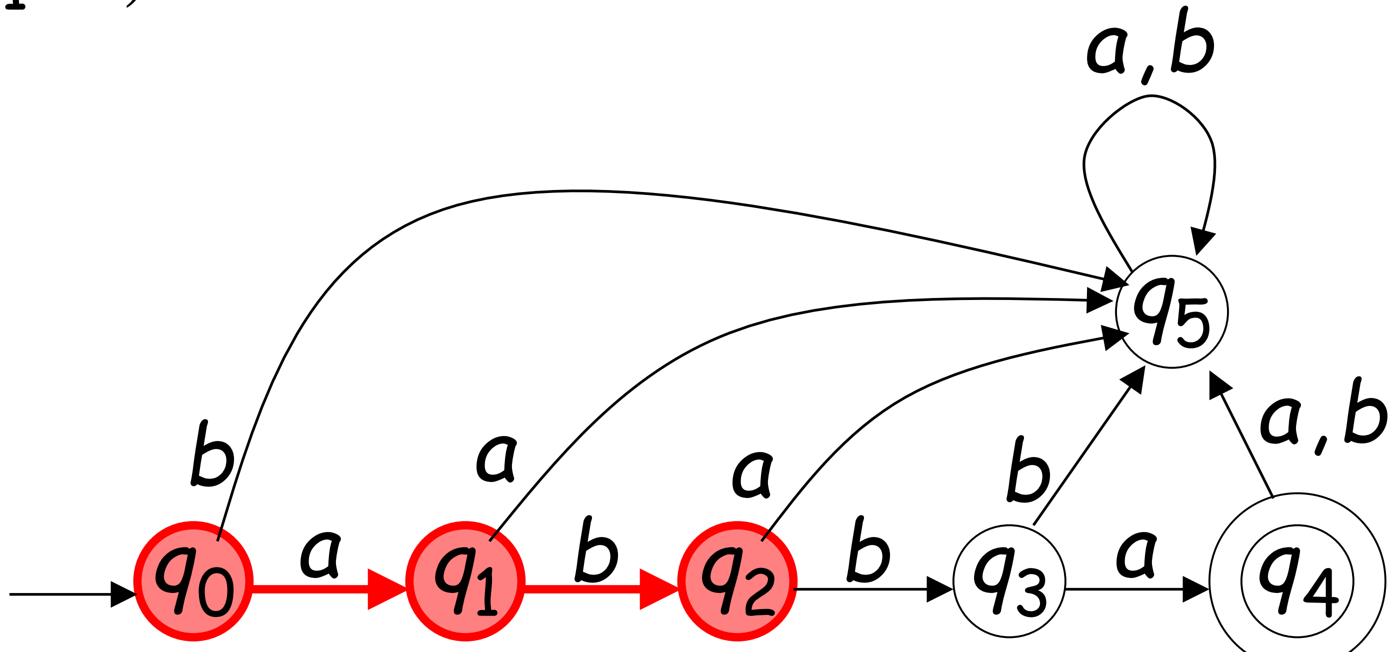
$$\delta(\delta^*(q_0, a), b) =$$

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

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

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

$$q_2$$



Languages Accepted by DFAs

- Take DFA M

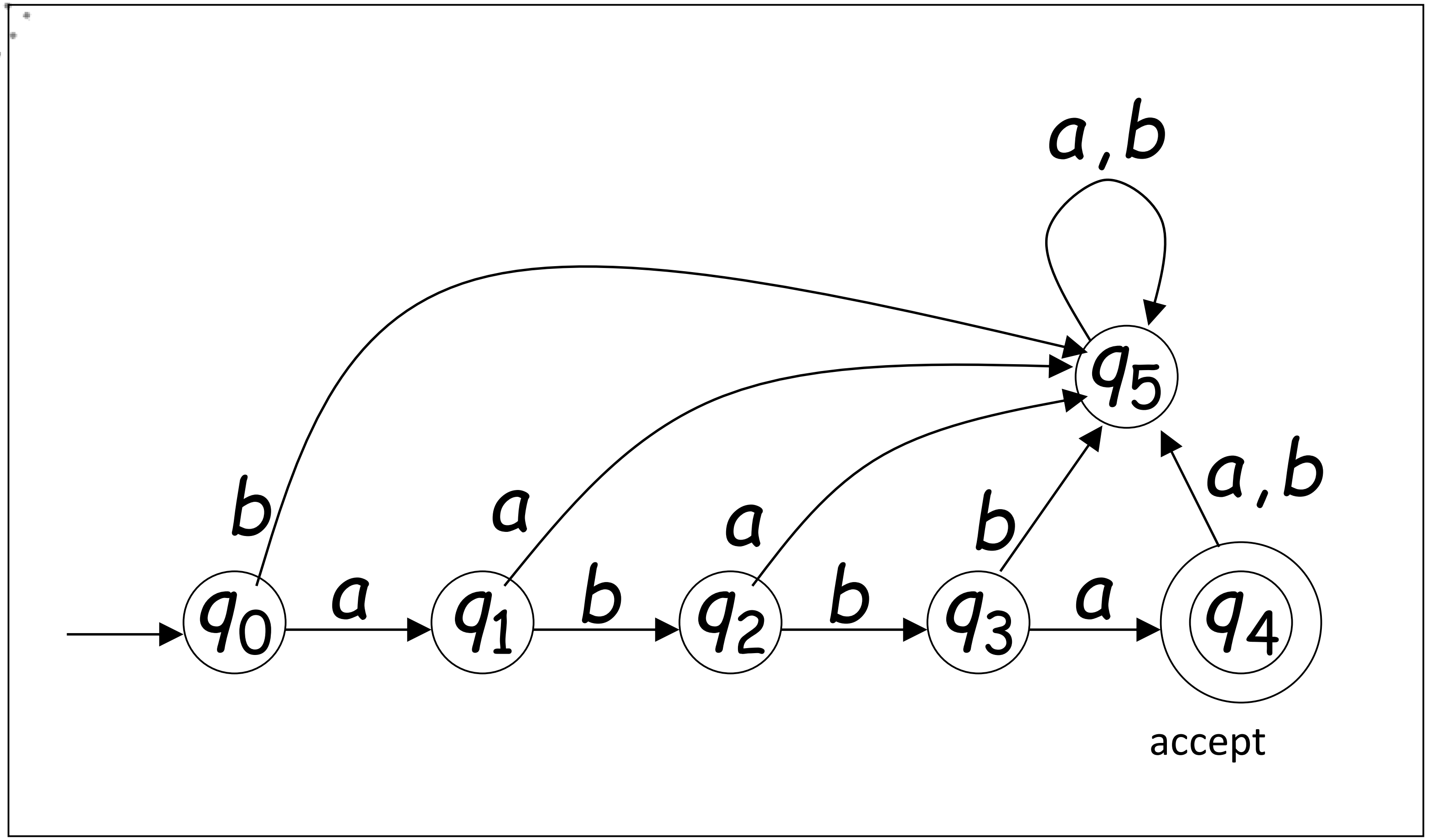
- Definition:
  - The language $L(M)$ contains
  - all input strings accepted by M

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

Example

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

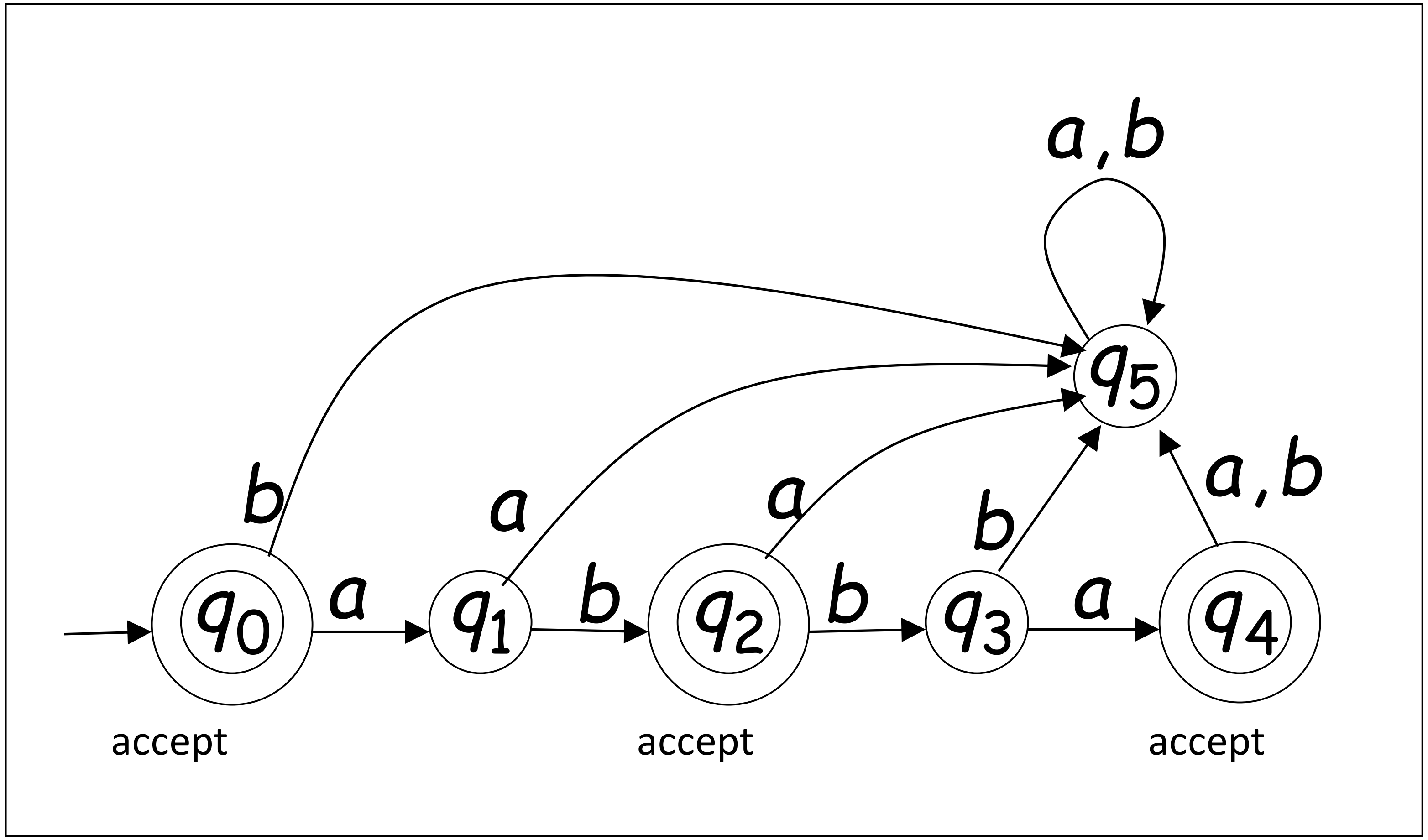
M



Another Example

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

M

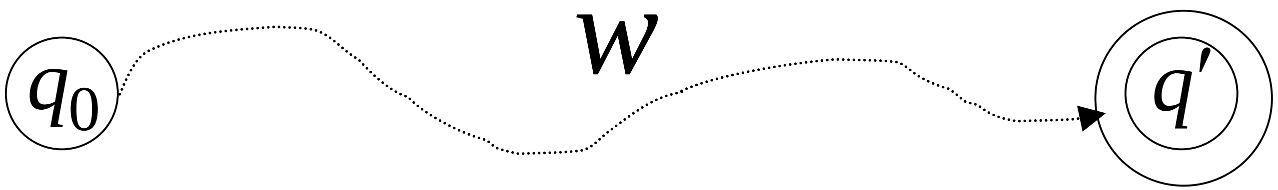


Formally

- 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 :

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

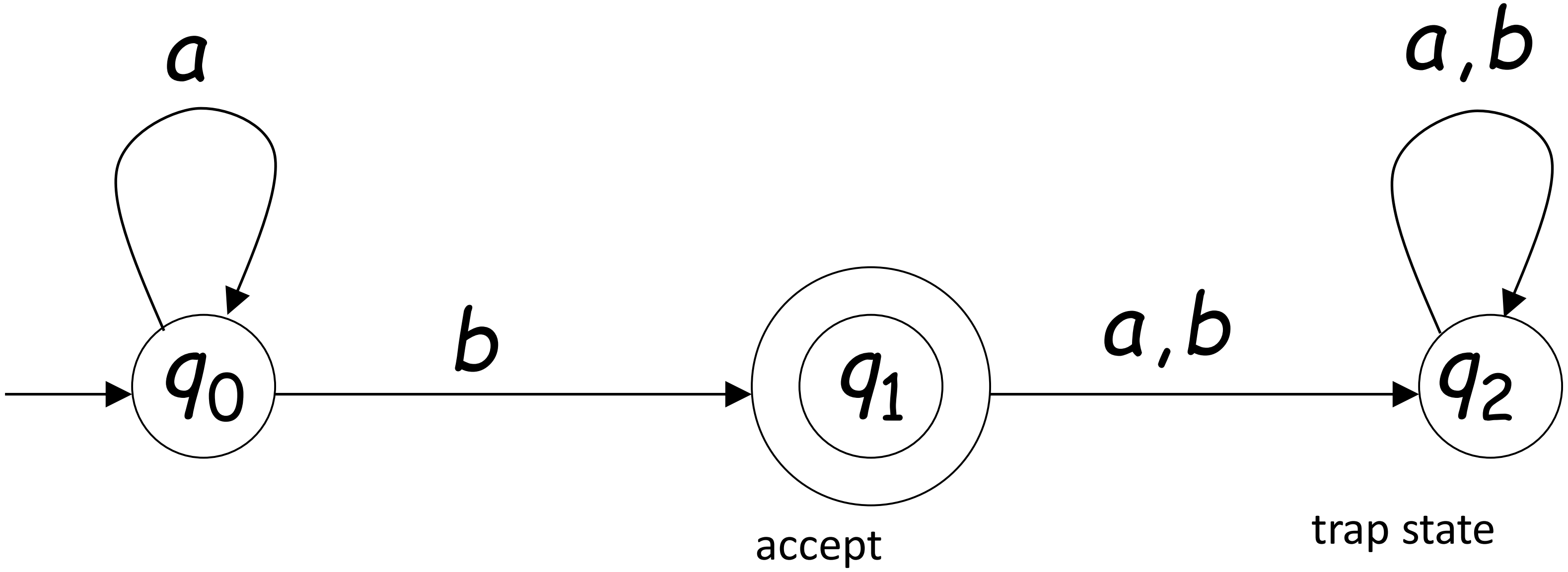
$$q_0 \xrightarrow{w} q' \qquad q' \notin F$$

More Examples

- Design a DFA that accepts the following Language:

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

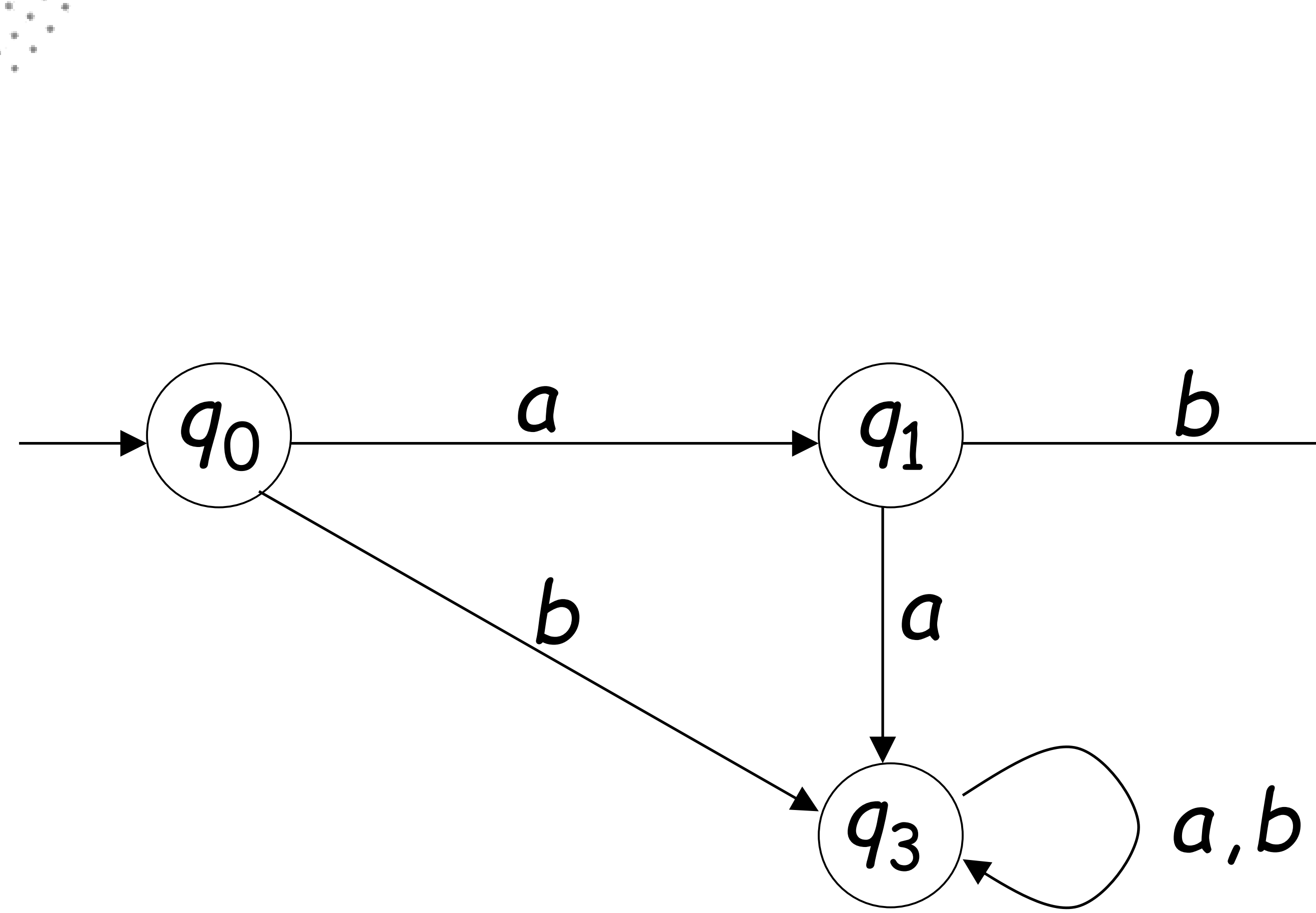
$$L(M) = \{a^n b : n \geq 0\}$$



- Desing a DFA that accepts the following language:

L(M)= { all strings with prefix “ab”}

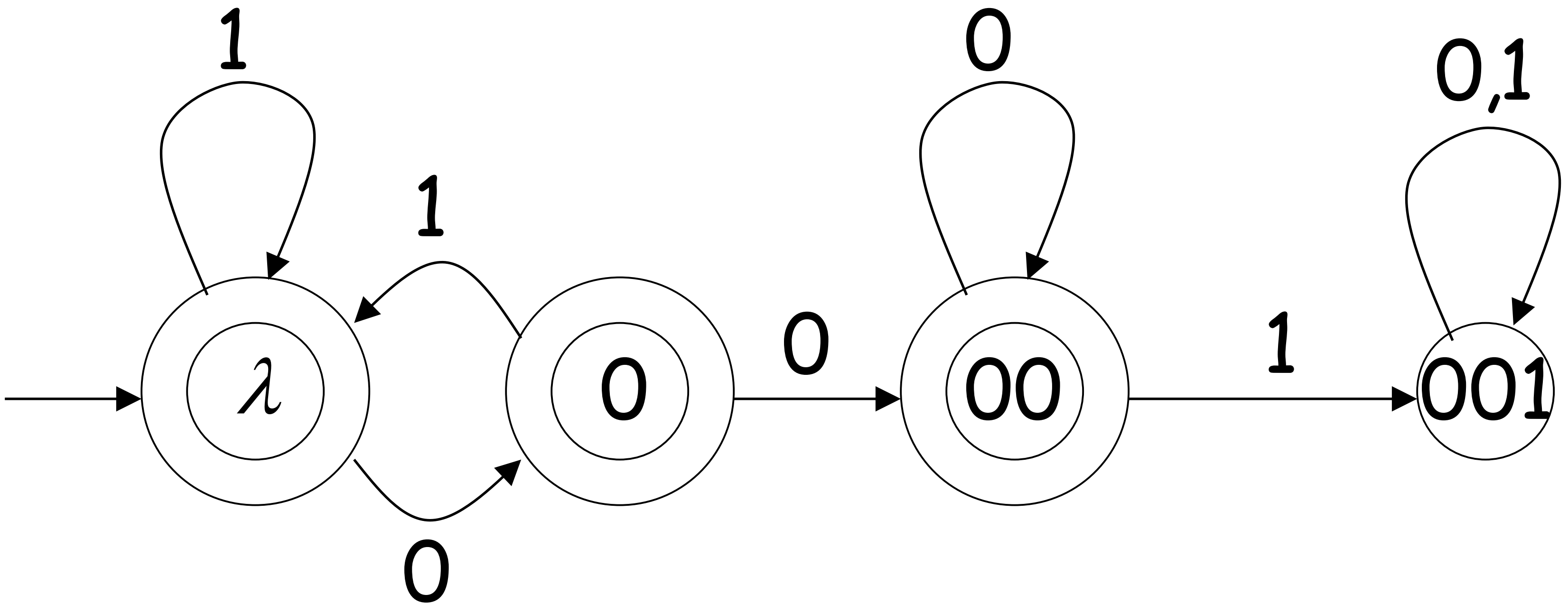
L(M)= { all strings with prefix “ab”}



Design a DFA which accepts the following language.

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

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 with prefix ab }

{ all strings without substring 001 }

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

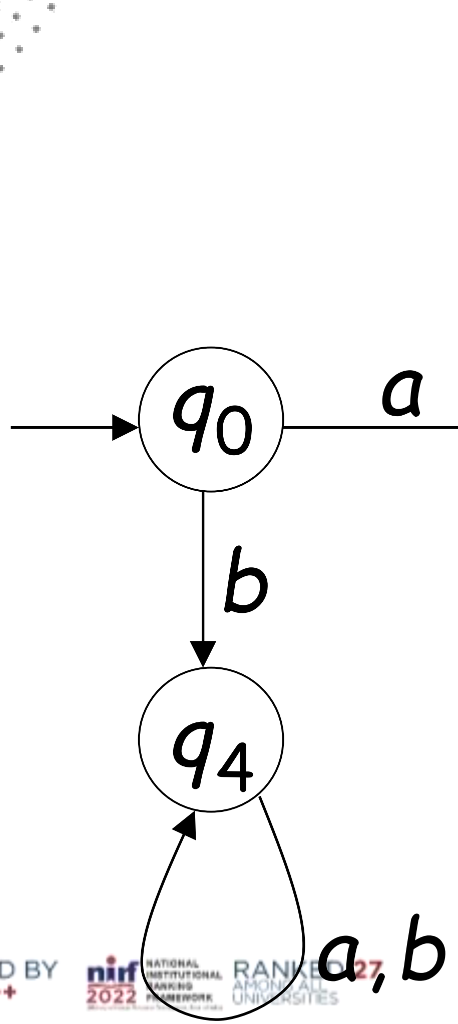
Another Example

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

-

$$L = L(M)$$

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



- Design a DFA that accpets the following language:

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

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)

QUIZ TIME

1. Which of the following is the correct definition of a finite automaton?

a) A machine with an infinite number of states

b) A machine with a finite number of states

c) A machine that can process an infinite input sequence

d) A machine that can process only regular languages

Answer: b) A machine with a finite number of states

QUIZ TIME

1.Which of the following is the correct definition of a finite automaton?

a) A machine with an infinite number of states

b) A machine with a finite number of states

c) A machine that can process an infinite input sequence

d) A machine that can process only regular languages

QUIZ TIME

1.Which of the following is NOT a component of a finite automaton?

a) Alphabet

b) Input tape

c) Transition function

d) Initial state

Answer: b) Input tape

QUIZ TIME

1. Which of the following is NOT a component of a finite automaton?

a) Alphabet

b) Input tape

c) Transition function

d) Initial state

QUIZ TIME

What is the purpose of a transition function in a finite automaton?

a) It defines the set of all possible inputs
b) It determines the initial state of the automaton
c) It specifies the set of final states
d) It describes the next state based on the current state and input symbol

Answer: d) It describes the next state based on the current state and input symbol

QUIZ TIME

What is the purpose of a transition function in a finite automaton?

a) It defines the set of all possible inputs

b) It determines the initial state of the automaton

c) It specifies the set of final states

d) It describes the next state based on the current state and input symbol

QUIZ TIME

Which of the following languages can be recognized by a deterministic finite automaton (DFA)?

a) All regular languages

b) All context-free languages

c) All context-sensitive languages

d) All recursive enumerable languages

Answer: a) All regular languages

QUIZ TIME

Which of the following languages can be recognized by a deterministic finite automaton (DFA)?

a) All regular languages

b) All context-free languages

c) All context-sensitive languages

d) All recursive enumerable languages

QUIZ TIME

1. What is the key difference between a nondeterministic finite automaton (NFA) and a deterministic finite automaton (DFA)?

a) NFAs have an infinite number of states, while DFAs have a finite number of states

b) b) NFAs can recognize non-regular languages, while DFAs can only recognize regular languages

c) NFAs have multiple possible next states for a given state and input, while DFAs have a unique next state

d) NFAs do not have a transition function, while DFAs have a transition function

QUIZ TIME

1.What is the key difference between a nondeterministic finite automaton (NFA) and a deterministic finite automaton (DFA)?

a) NFAs have an infinite number of states, while DFAs have a finite number of states

b) b) NFAs can recognize non-regular languages, while DFAs can only recognize regular languages

c) NFAs have multiple possible next states for a given state and input, while DFAs have a unique next state

d) NFAs do not have a transition function, while DFAs have a transition function

QUIZ TIME

Which of the following best describes a finite automaton?

a) A computational model that can solve any problem

b) A mathematical model consisting of an input tape, head, and rules

c) A mathematical model consisting of states, input symbols, transitions, and a start state

d) A machine that can compute any function

Answer: c) A mathematical model consisting of states, input symbols, transitions, and a start state

What is the purpose of the transition function in a finite automaton?

a) To define the initial state of the automaton

b) To determine the input symbols accepted by the automaton

c) To specify the rules for moving between states based on input symbols

d) To terminate the execution of the automaton

Answer: c) To specify the rules for moving between states based on input symbols

QUIZ TIME

Which of the following best describes a finite automaton?

a) A computational model that can solve any problem

b) A mathematical model consisting of an input tape, head, and rules

c) A mathematical model consisting of states, input symbols, transitions, and a start state

d) A machine that can compute any function

What is the purpose of the transition function in a finite automaton?

a) To define the initial state of the automaton

b) To determine the input symbols accepted by the automaton

c) To specify the rules for moving between states based on input symbols

d) To terminate the execution of the automaton

Which of the following describes a language recognized by a finite automaton?

a) A set of strings that the automaton can generate

b) A set of strings that the automaton can accept as input

c) A set of strings that the automaton can compute

d) A set of strings that the automaton can print as output

Answer: b) A set of strings that the automaton can accept as input

What is the key difference between a deterministic finite automaton (DFA) and a non-deterministic finite automaton (NFA)?

a) DFAs can accept an infinite number of input strings, while NFAs can only accept a finite number of input strings.

b) DFAs can have multiple transitions for a state and input symbol, while NFAs have a single transition for each state and input symbol.

c) DFAs can recognize regular languages, while NFAs can recognize context-free languages.

d) DFAs can recognize non-regular languages, while NFAs can only recognize regular languages.

Answer: b) DFAs can have multiple transitions for a state and input symbol, while NFAs have a single transition for each state and input symbol.

Which of the following describes a language recognized by a finite automaton?

a) A set of strings that the automaton can generate

b) A set of strings that the automaton can accept as input

c) A set of strings that the automaton can compute

d) A set of strings that the automaton can print as output

What is the key difference between a deterministic finite automaton (DFA) and a non-deterministic finite automaton (NFA)?

a) DFAs can accept an infinite number of input strings, while NFAs can only accept a finite number of input strings.

b) NFAs can have multiple transitions for a state and input symbol, while DFAs have a single transition for each state and input symbol.

c) DFAs can recognize regular languages, while NFAs can recognize context-free languages.

d) DFAs can recognize non-regular languages, while NFAs can only recognize regular languages.

Terminal Questions

1. What is a Deterministic Finite Automaton (DFA)?

2. What are the components of a DFA?

3. How does a DFA accept or reject strings?

4. What is the difference between deterministic and nondeterministic automata?

5. What is the role of the transition function in a DFA?

6. Can a DFA recognize and accept non-regular languages?

7. How can you convert a regular expression into an equivalent DFA?

8. What is the significance of the start state in a DFA?

9. How can you prove that a language is regular using a DFA?

10. What are some real-world applications of DFAs and regular language acceptance?

THANK YOU

Team – TOC