Languages

Languages

A language is a set of strings

String: A sequence of letters

Examples: "cat", "dog", "house", ...

Defined over an alphabet:

$$\Sigma = \{a, b, c, \ldots, z\}$$

Alphabets and Strings

We will use small alphabets: $\Sigma = \{a, b\}$

Strings

$$a$$

$$ab \qquad\qquad u = ab$$

$$abba \qquad\qquad v = bbbaaa$$

$$baba \qquad\qquad w = abba$$

$$aaabbbaabab$$

String Operations

$$w = a_1 a_2 \cdots a_n \qquad abba$$

$$v = b_1 b_2 \cdots b_m \qquad bbbaaa$$

Concatenation

$$wv = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m \qquad abbabbbaaa$$

$$w = a_1 a_2 \cdots a_n$$

$ababaaabbb$

Reverse

$$w^R = a_n \cdots a_2 a_1$$

$bbbaaababa$

String Length

$$w = a_1 a_2 \cdots a_n$$

Length: $|w| = n$

Examples:

$$|abba| = 4$$

$$|aa| = 2$$

$$|a| = 1$$

Recursive Definition of Length

For any letter: $|a| = 1$

For any string *wa*: $|wa| = |w| + 1$

Example:

$$\begin{aligned} |abba| &= |abb| + 1 \\ &= |ab| + 1 + 1 \\ &= |a| + 1 + 1 + 1 \\ &= 1 + 1 + 1 + 1 \\ &= 4 \end{aligned}$$

Length of Concatenation

$$|uv| = |u| + |v|$$

Example: $u = aab, \quad |u| = 3$

$$v = abaab, \quad |v| = 5$$

$$|uv| = |aababaab| = 8$$

$$|uv| = |u| + |v| = 3 + 5 = 8$$

Proof of Concatenation Length

Claim: $|uv| = |u| + |v|$

Proof: By induction on the length $|v|$

Induction basis: $|v| = 1$

From definition of length:

$$|uv| = |u| + 1 = |u| + |v|$$

Inductive hypothesis: $|uv| = |u| + |v|$

for $|v| = 1, 2, \ldots, n$

Inductive step: we will prove $|uv| = |u| + |v|$

for $|v| = n + 1$

Inductive Step

Write $v = wa$, where $|w| = n$, $|a| = 1$

From definition of length: $|uv| = |uwa| = |uw| + 1$

$$|wa| = |w| + 1$$

From inductive hypothesis: $|uw| = |u| + |w|$

Thus: $|uv| = |u| + |w| + 1 = |u| + |wa| = |u| + |v|$

Empty String

A string with no letters: λ

Observations: $|\lambda| = 0$

$$\lambda w = w\lambda = w$$

$$\lambda abba = abba\lambda = abba$$

Substring

Substring of string:

a subsequence of consecutive characters

String	Substring
$\underline{ab}bab$	ab
$\underline{abba}b$	$abba$
$ab\underline{b}ab$	b
$a\underline{bbab}$	$bbab$

Prefix and Suffix

$$abbab$$

Prefixes	Suffixes
λ	$abbab$
a	$bbab$
ab	bab
abb	ab
$abba$	b
$abbab$	λ

$$w = uv$$

prefix

suffix

Another Operation

$$w^n = \underbrace{ww\cdots w}_{n}$$

Example: $(abba)^2 = abbaabba$

Definition: $w^0 = \lambda$

$$(abba)^0 = \lambda$$

The * Operation

Σ^*: the set of all possible strings from alphabet Σ

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

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

The + Operation

Σ^+ : the set of all possible strings from alphabet Σ except λ

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

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

$$\Sigma^+ = \Sigma^* - \lambda$$

$$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

Language

A language is any subset of Σ^*

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

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, \ldots\}$$

Languages: $\{\lambda\}$

$$\{a, aa, aab\}$$

$$\{\lambda, abba, baba, aa, ab, aaaaaa\}$$

Another Example

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

$$\left.\begin{array}{l} \lambda \\ ab \\ aabb \\ aaaaabbbbb \end{array}\right\} \in L \qquad abb \notin L$$

Operations on Languages

The usual set operations

$$\{a, ab, aaaa\} \bigcup \{bb, ab\} = \{a, ab, bb, aaaa\}$$

$$\{a, ab, aaaa\} \bigcap \{bb, ab\} = \{ab\}$$

$$\{a, ab, aaaa\} - \{bb, ab\} = \{a, aaaa\}$$

Complement: $\overline{L} = \Sigma^* - L$

$$\overline{\{a, ba\}} = \{\lambda, b, aa, ab, bb, aaa, \ldots\}$$

Reverse

Definition: $L^R = \{w^R : w \in L\}$

Examples: $\{ab, aab, baba\}^R = \{ba, baa, abab\}$

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

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

Concatenation

Definition: $L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$

Example: $\{a, ab, ba\}\{b, aa\}$

$$= \{ab, aaa, abb, abaa, bab, baaa\}$$

Another Operation

Definition: $$L^n = \underbrace{LL\cdots L}_{n}$$

$$\{a,b\}^3 = \{a,b\}\{a,b\}\{a,b\} =$$
$$\{aaa,aab,aba,abb,baa,bab,bba,bbb\}$$

Special case: $$L^0 = \{\lambda\}$$

$$\{a,bba,aaa\}^0 = \{\lambda\}$$

More Examples

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

$$L^2 = \{a^n b^n a^m b^m : n, m \geq 0\}$$

$$aabbaaabbb \in L^2$$

Star-Closure (Kleene *)

Definition: $L^* = L^0 \cup L^1 \cup L^2 \cdots$

Example:

$$\{a, bb\}^* = \left\{\begin{array}{l} \lambda, \\ a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{array}\right\}$$

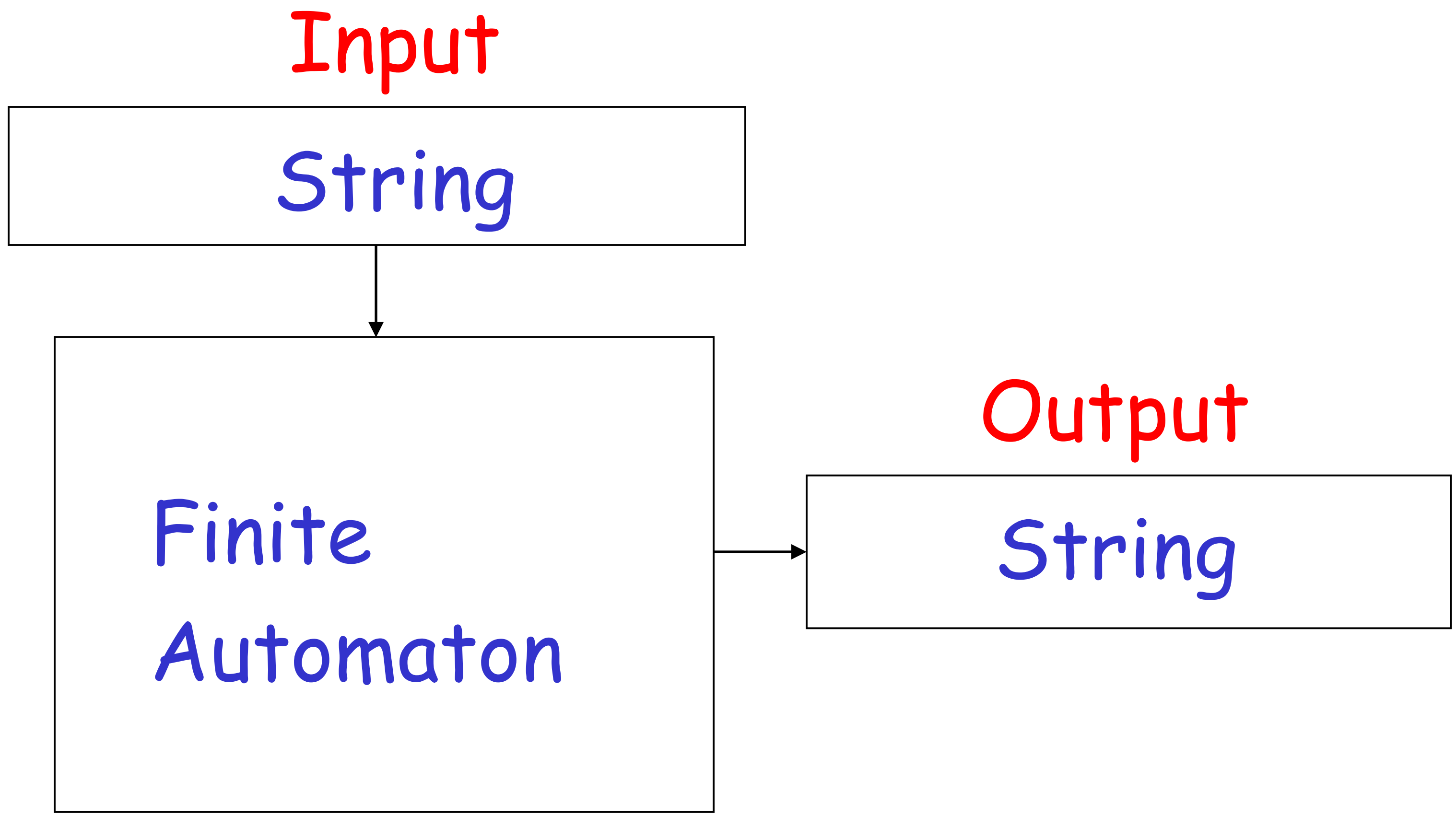
Positive Closure

Definition: $$L^{+} = L^{1} \bigcup L^{2} \bigcup \cdots$$
$$= L^{*} - \{\lambda\}$$

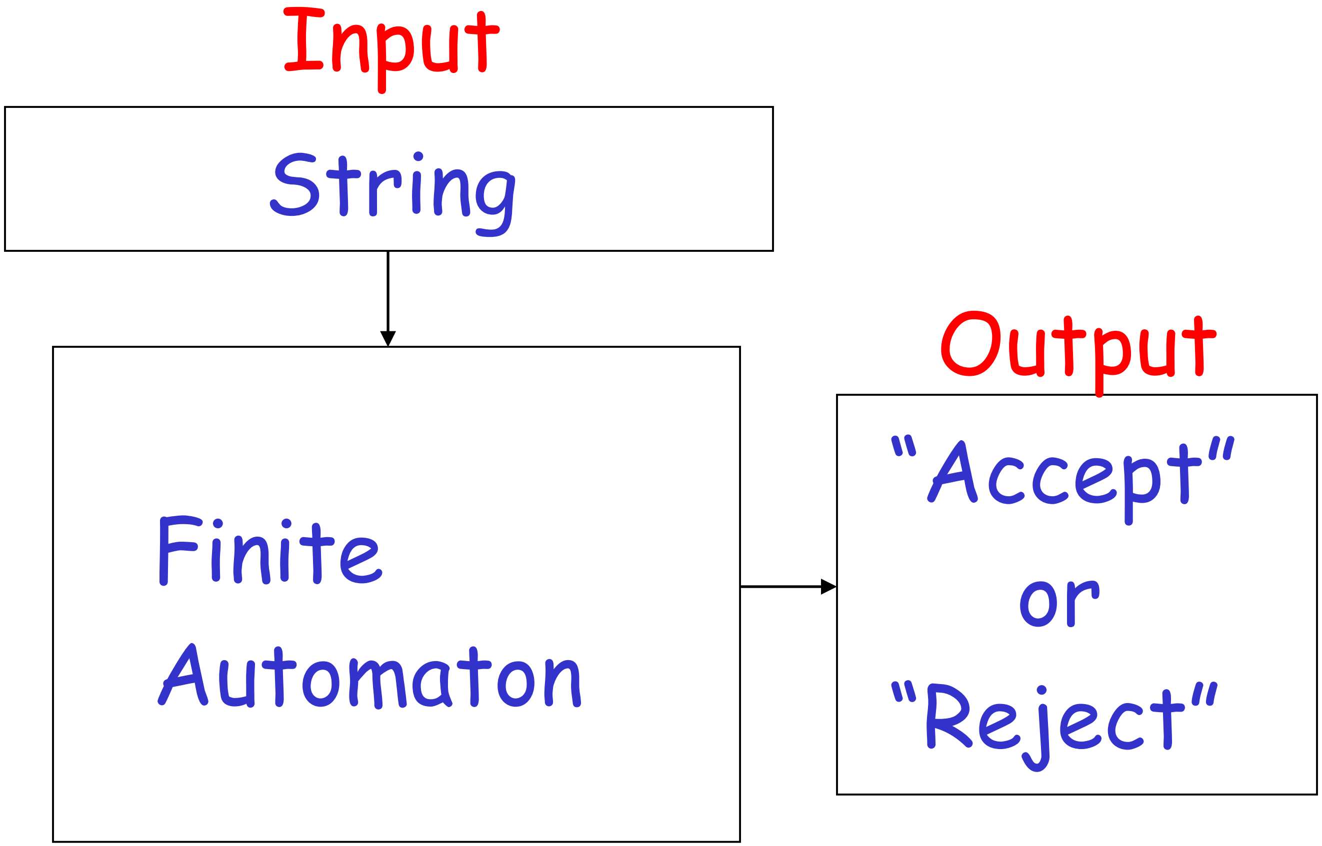
$$\{a, bb\}^{+} = \left\{ \begin{array}{l} a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{array} \right\}$$

Finite Automata

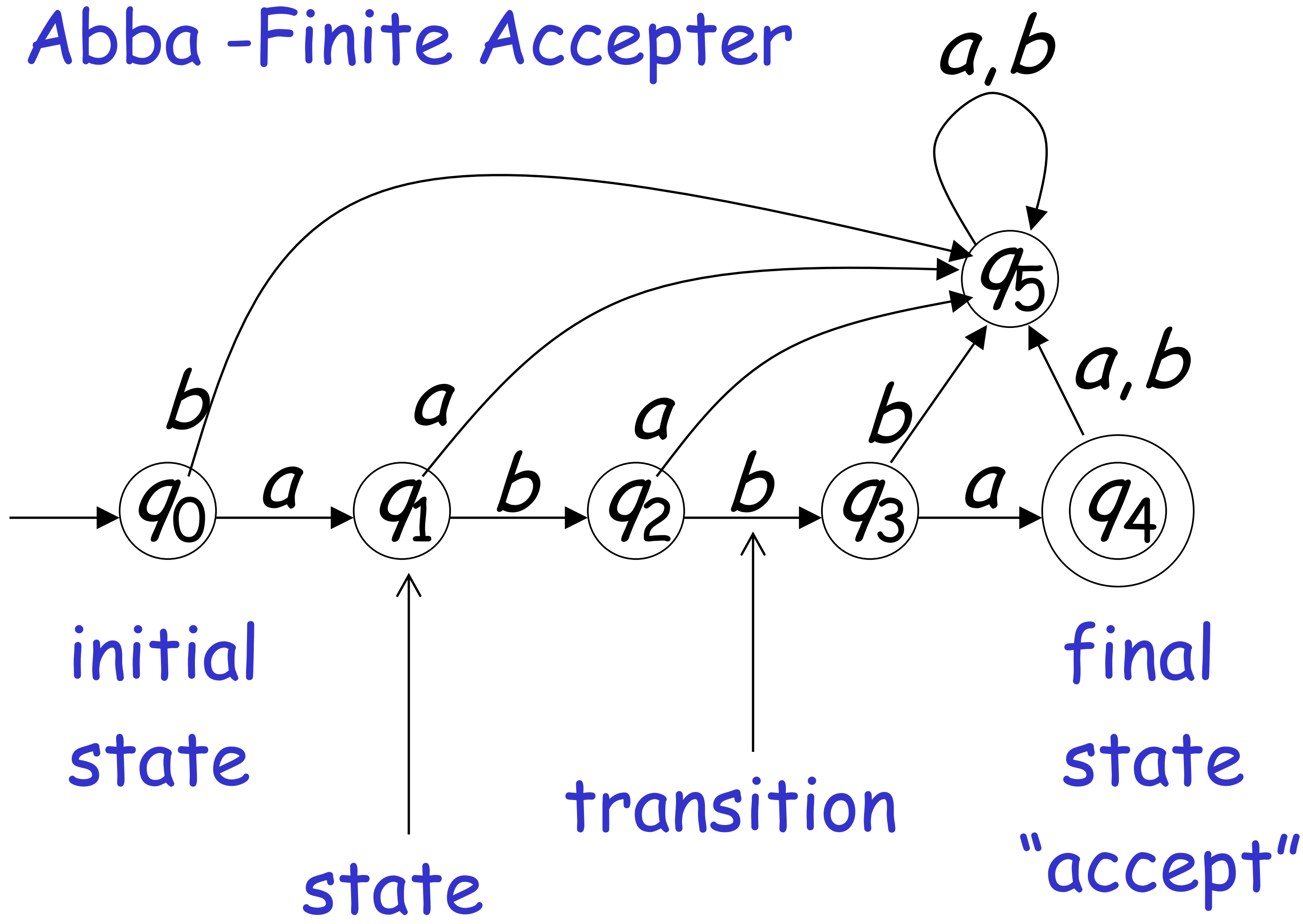
Finite Automaton



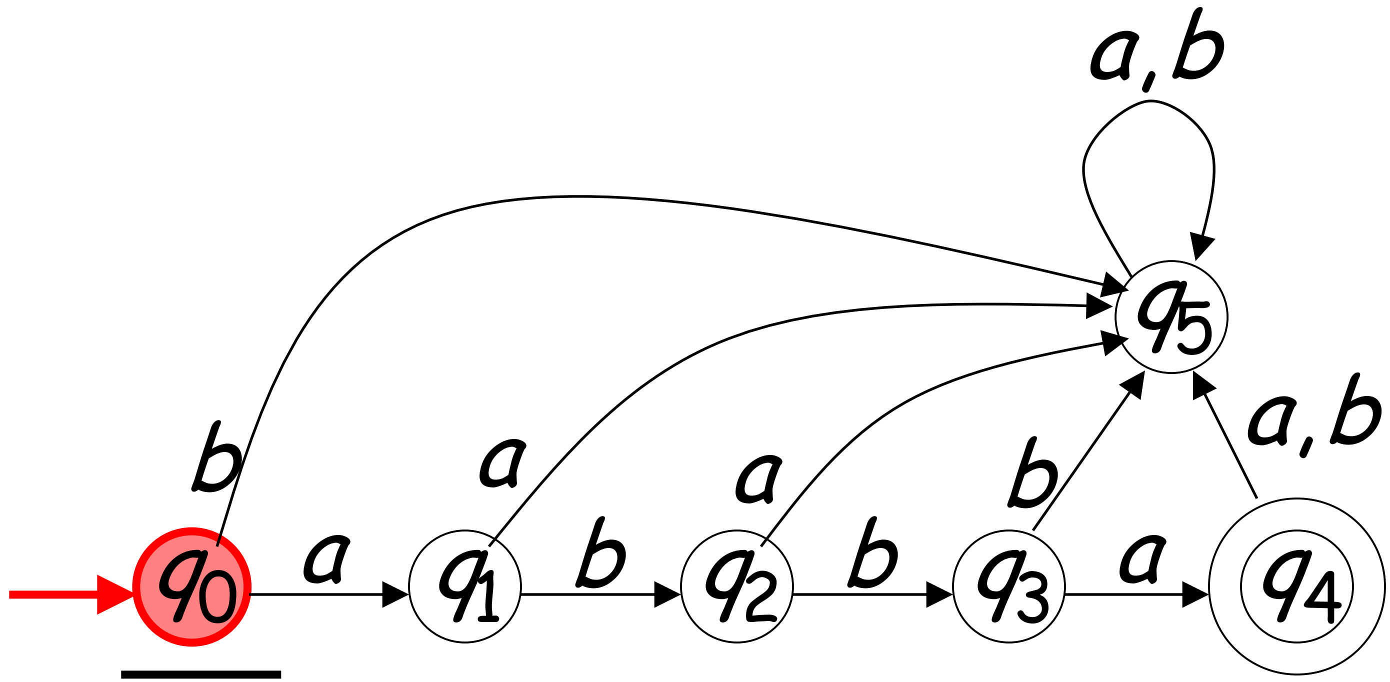
Finite Accepter



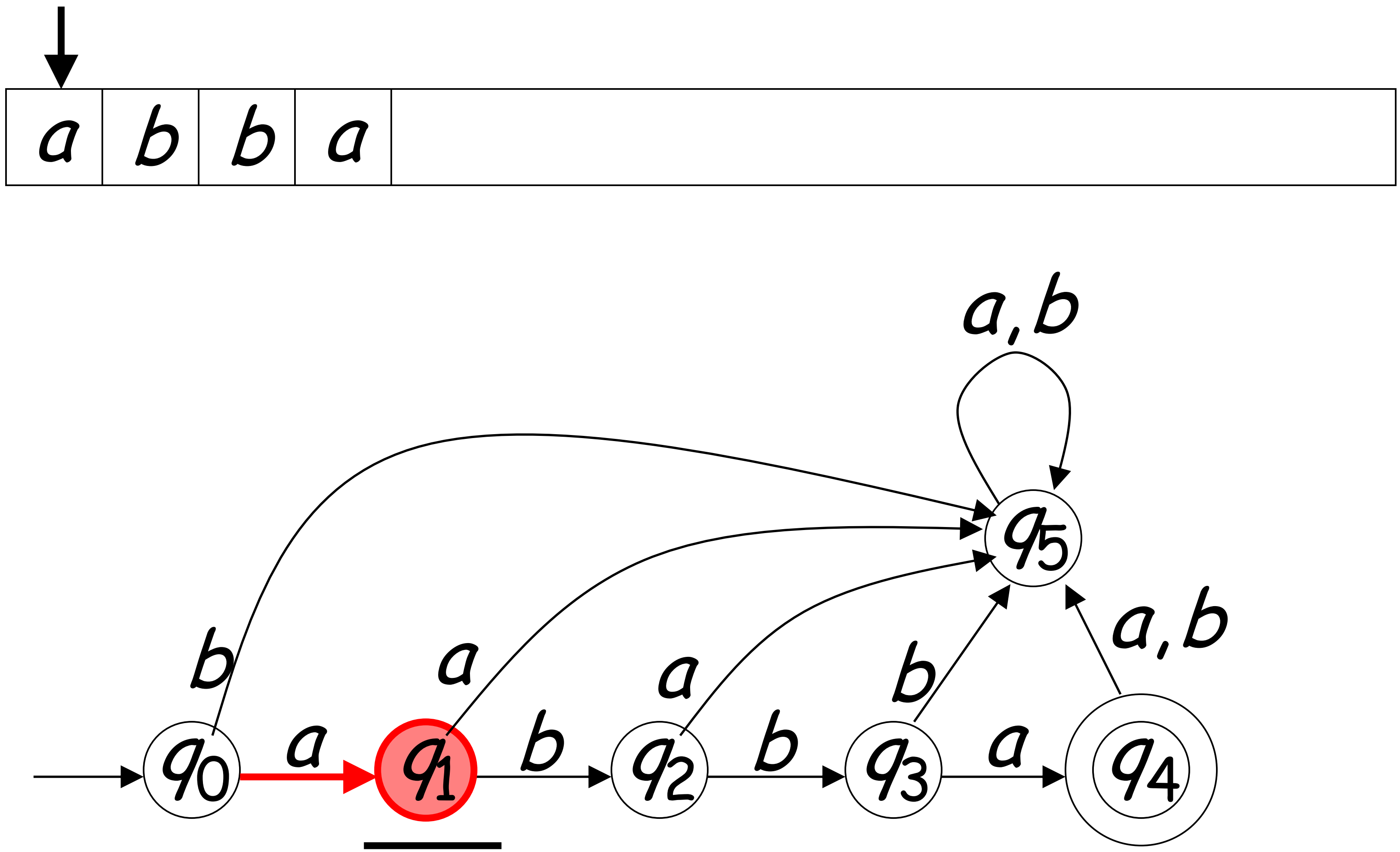
Transition Graph

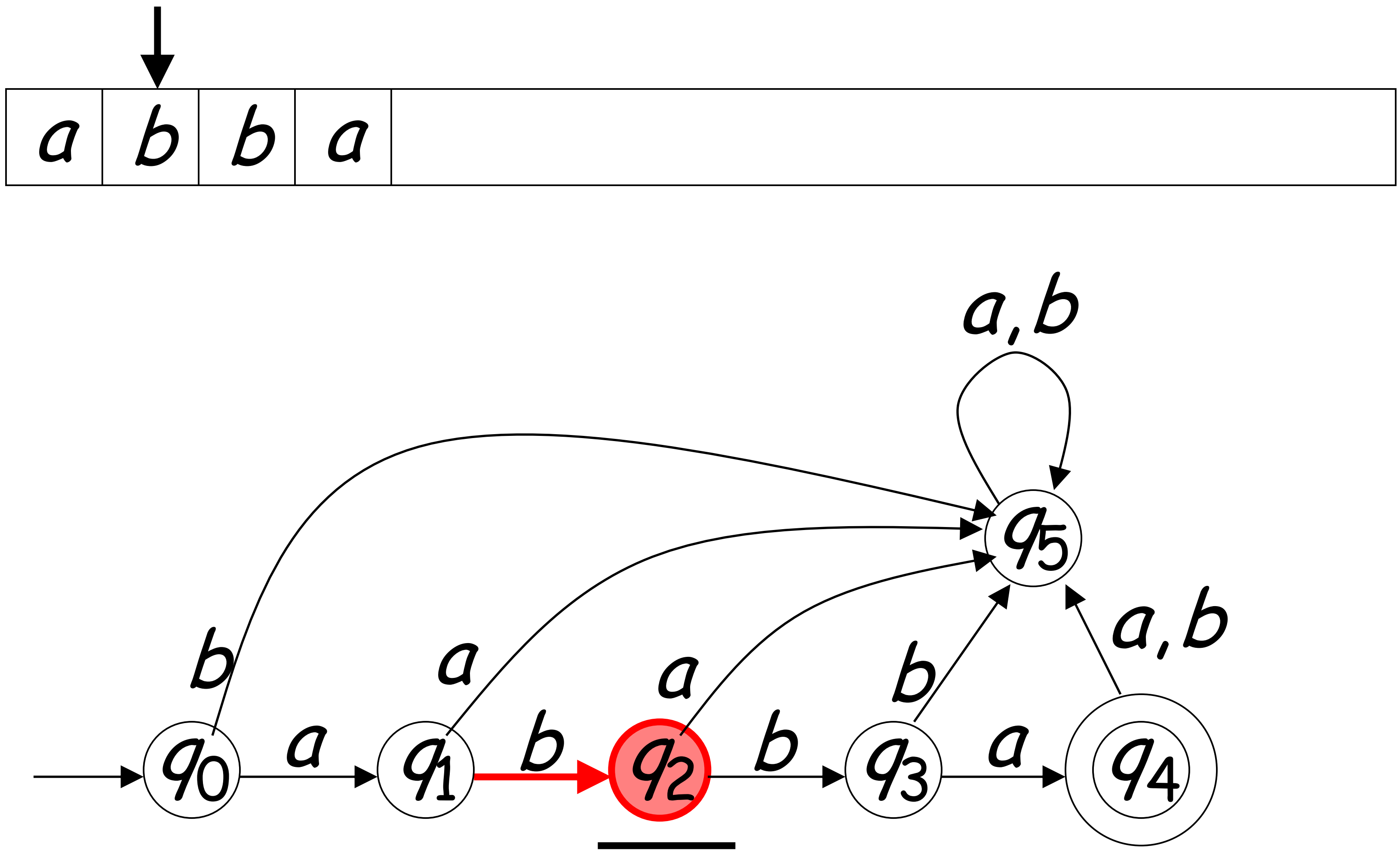


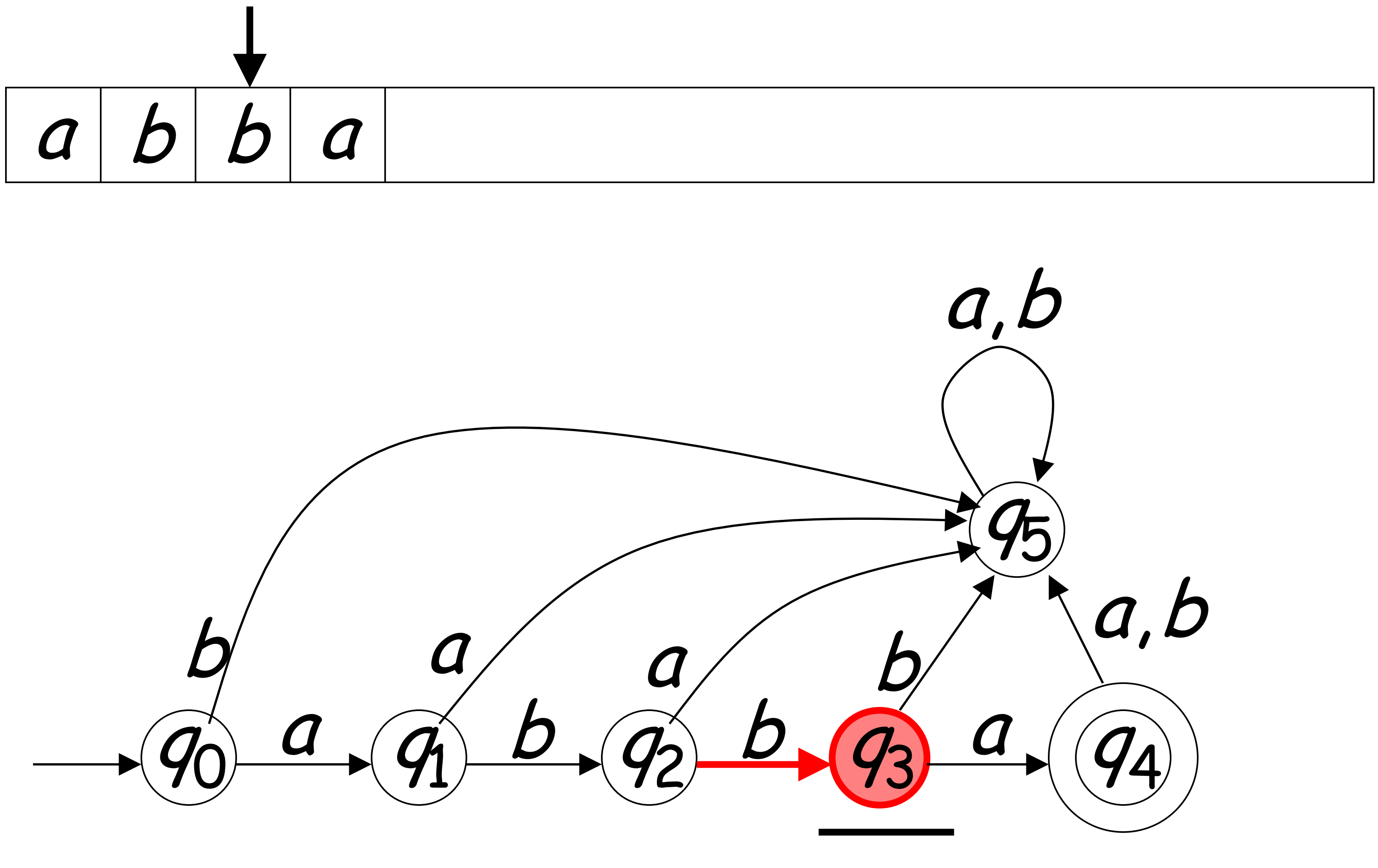
Initial Configuration

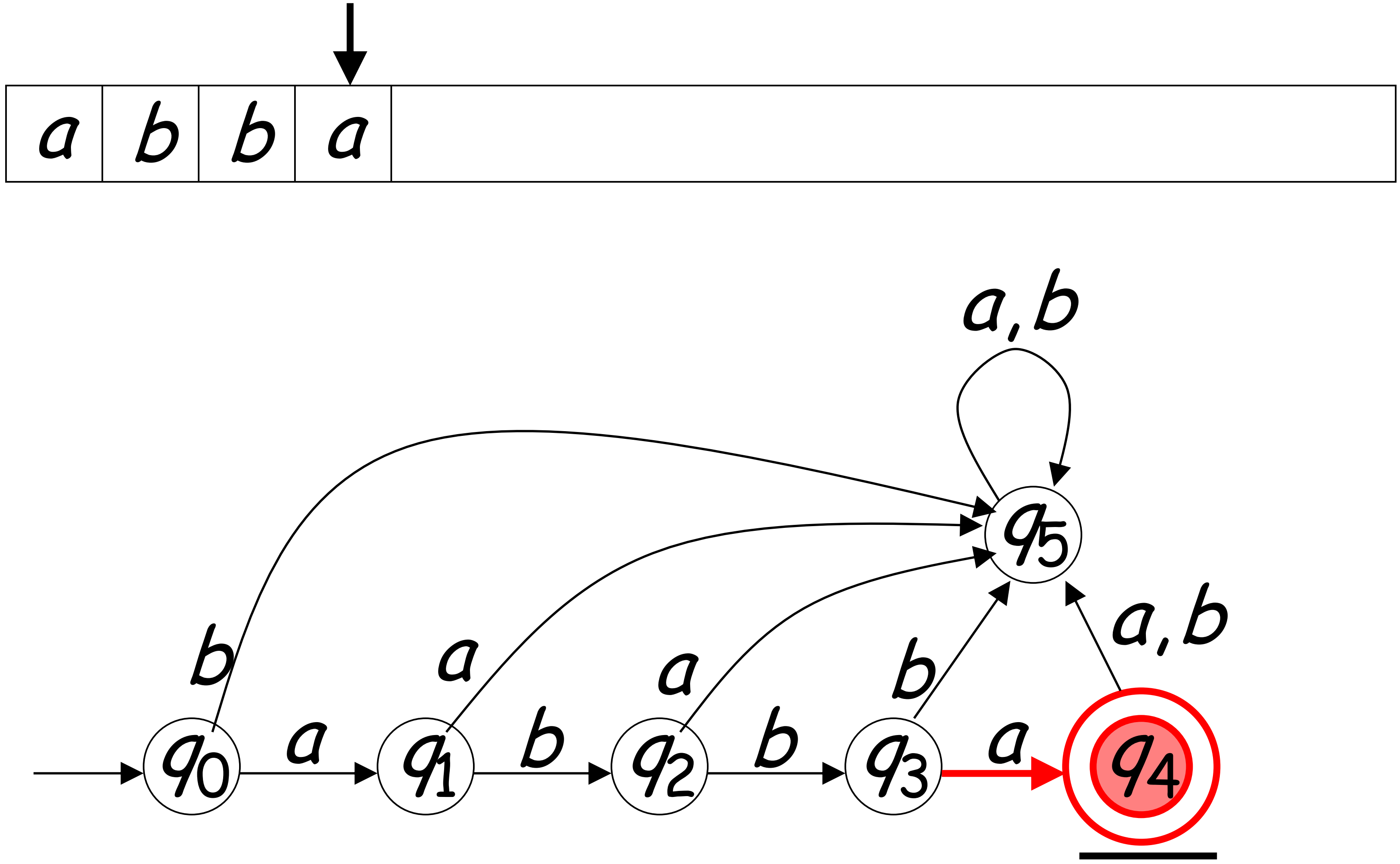


Reading the Input

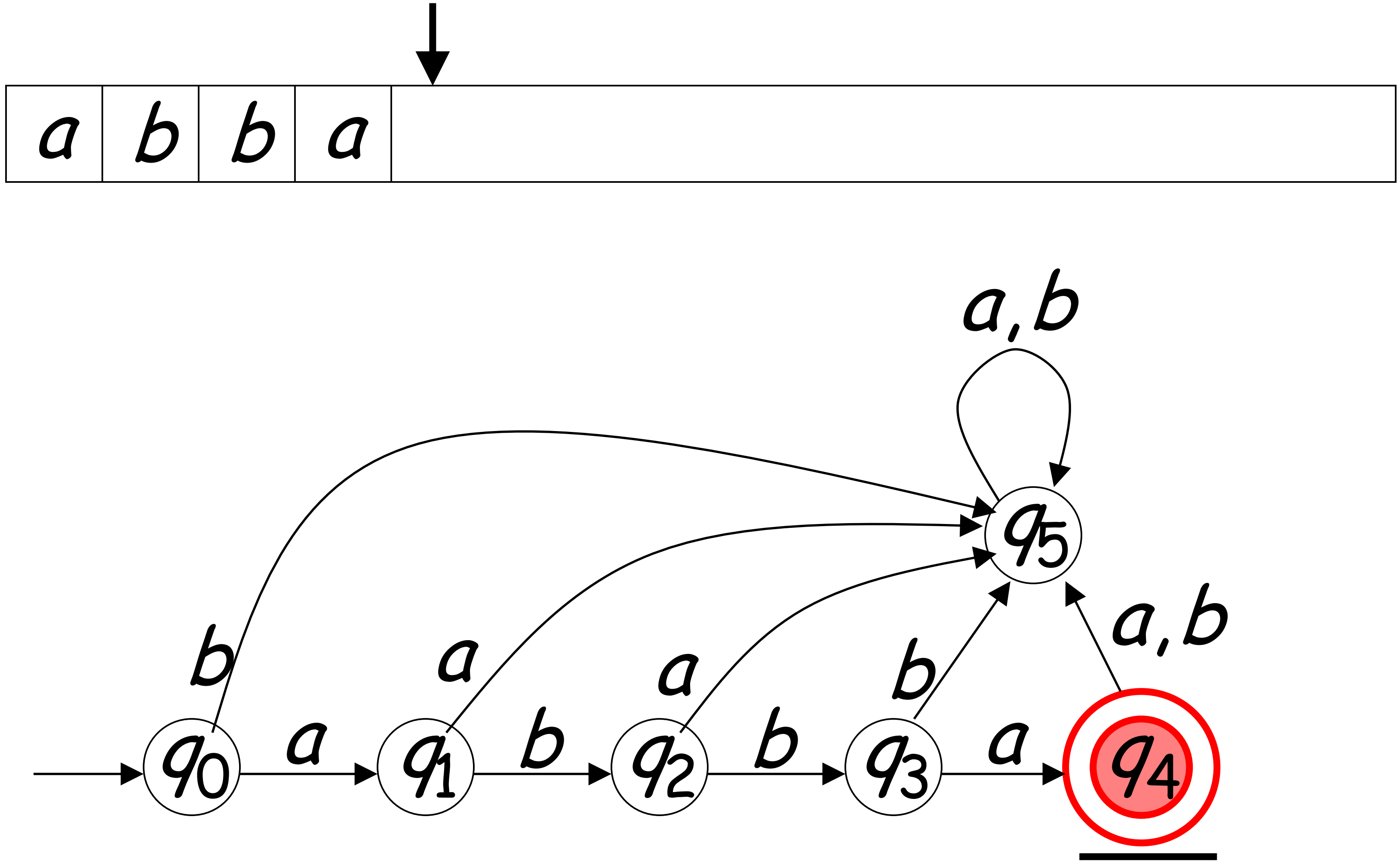



a
b
b
a
a,b
b
a
a
b
a
b
a
b
a
a,b
q_0
q_1
q_2
q_3
q_4
q_5


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

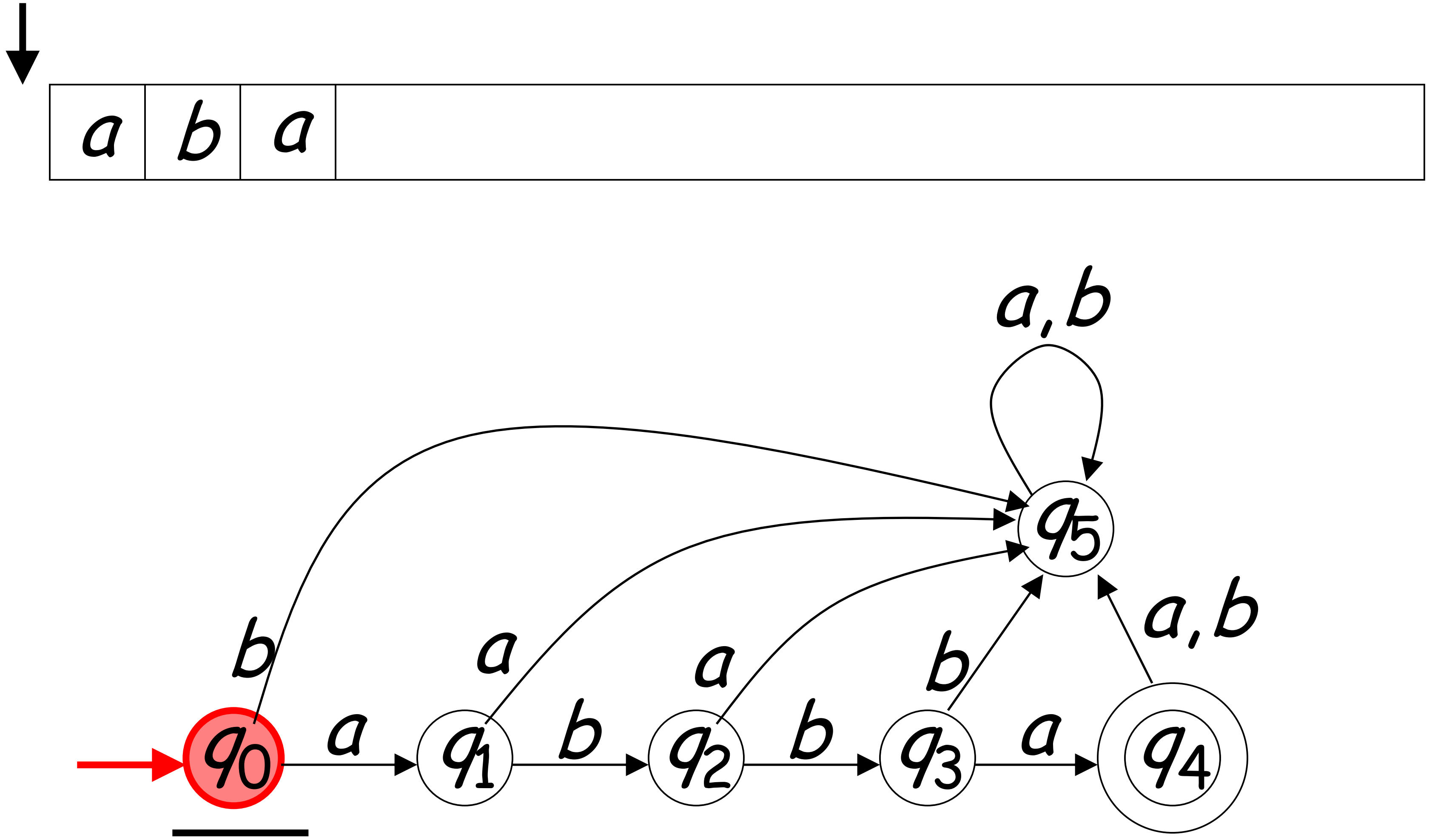

a b b a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4

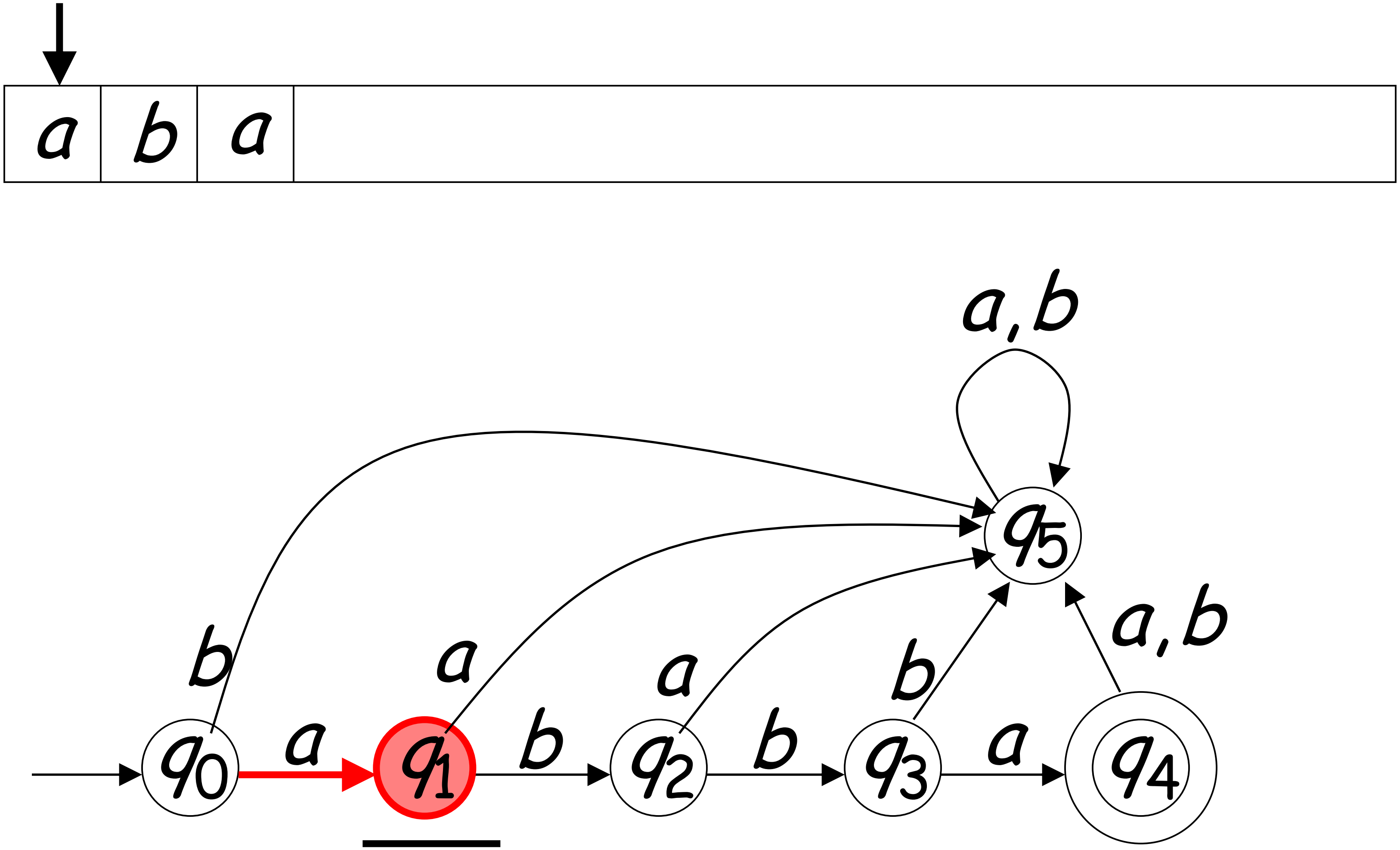
Input finished

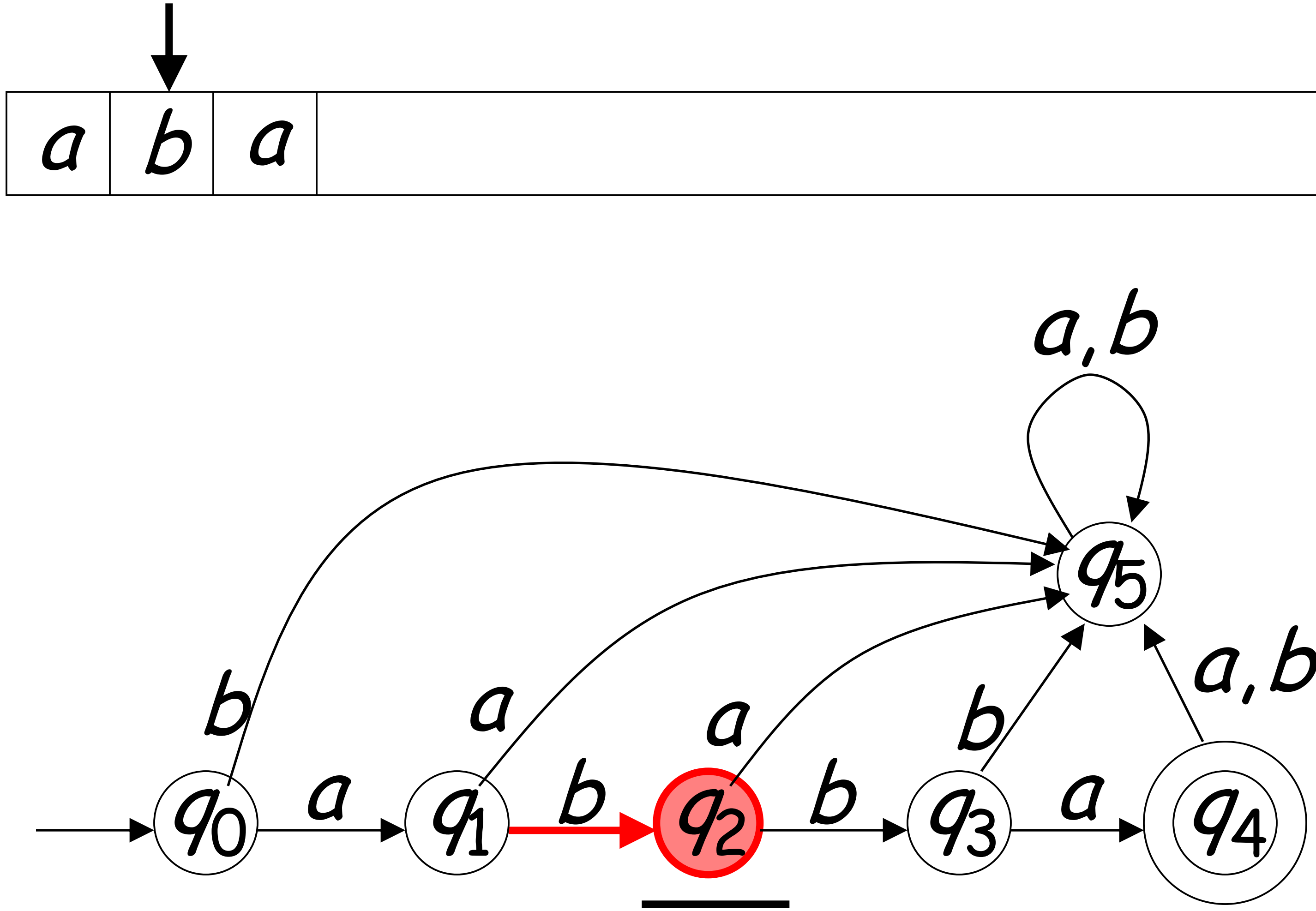


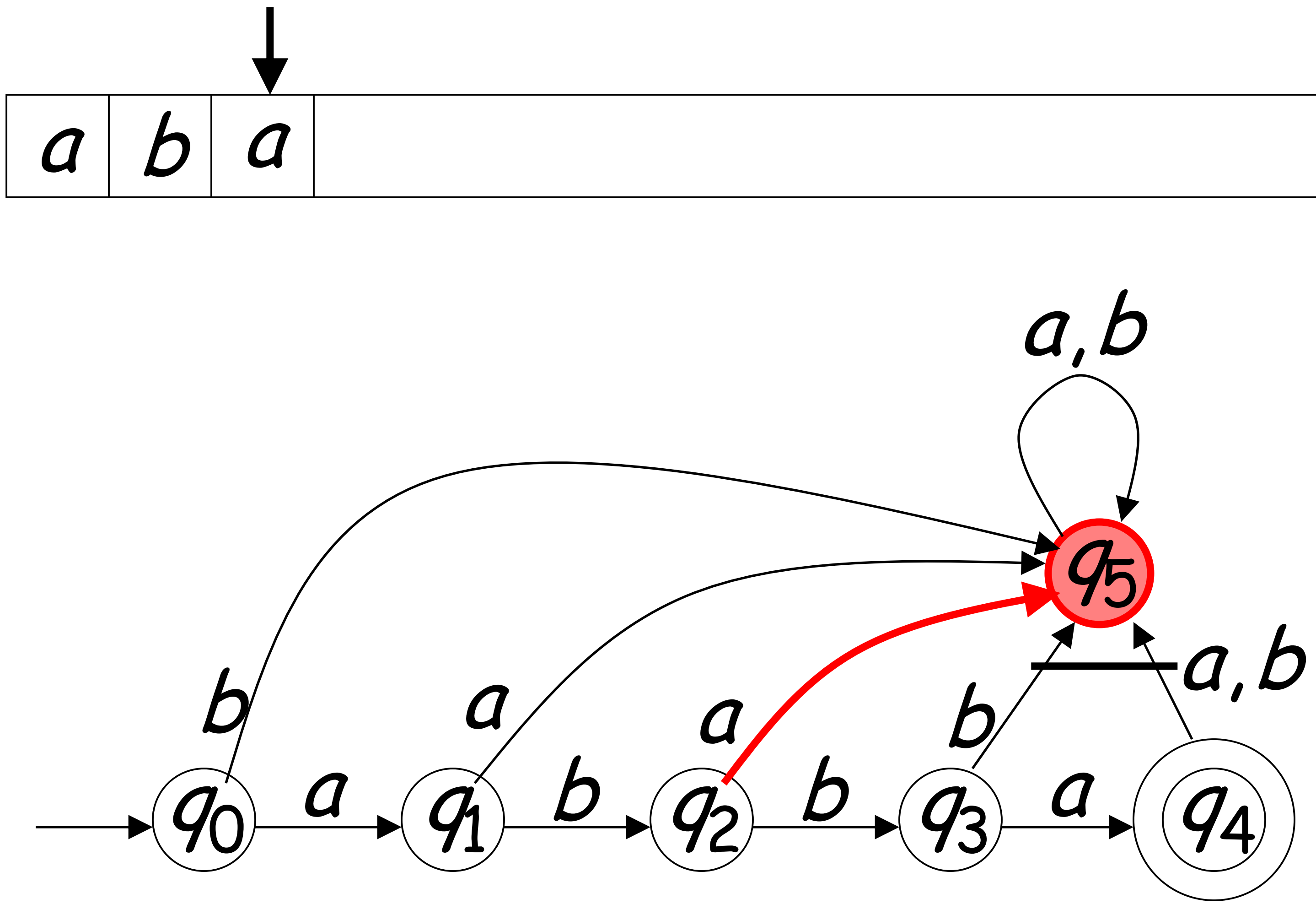
Output: “accept”

Rejection



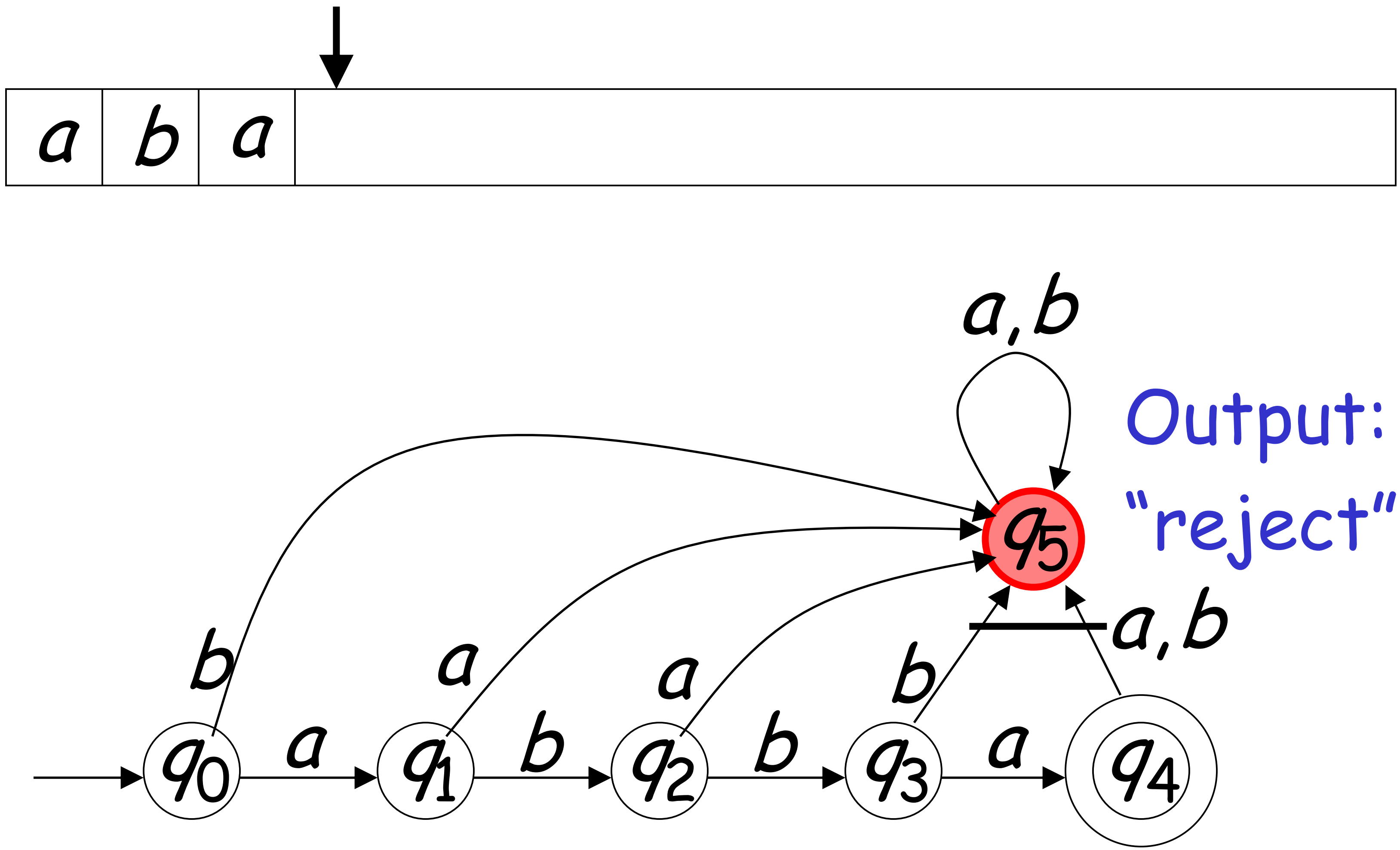

a
b
a
a,b
b
a
a
b
a
b
b
a
a,b
q_0
q_1
q_2
q_3
q_4
q_5


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

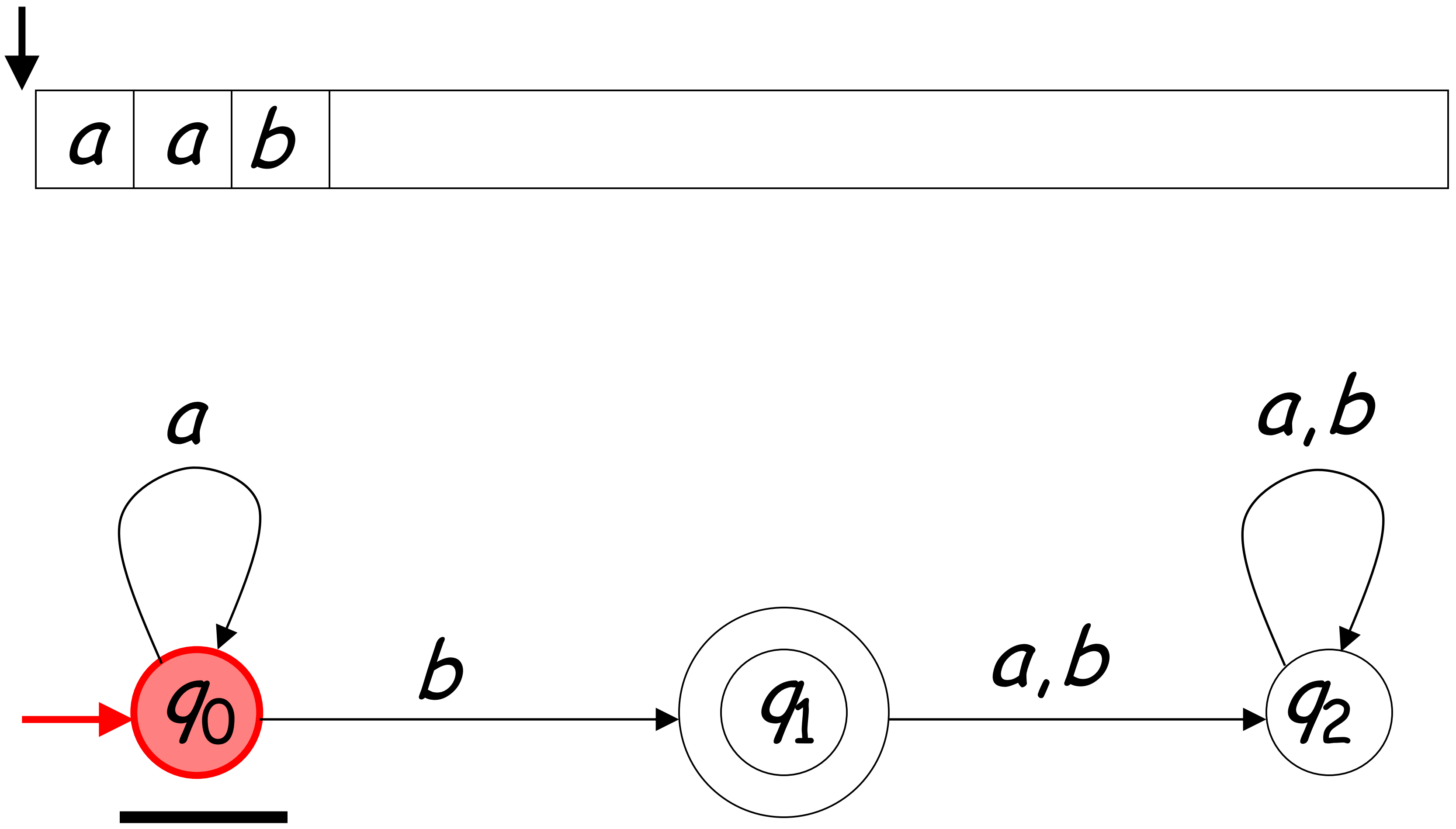


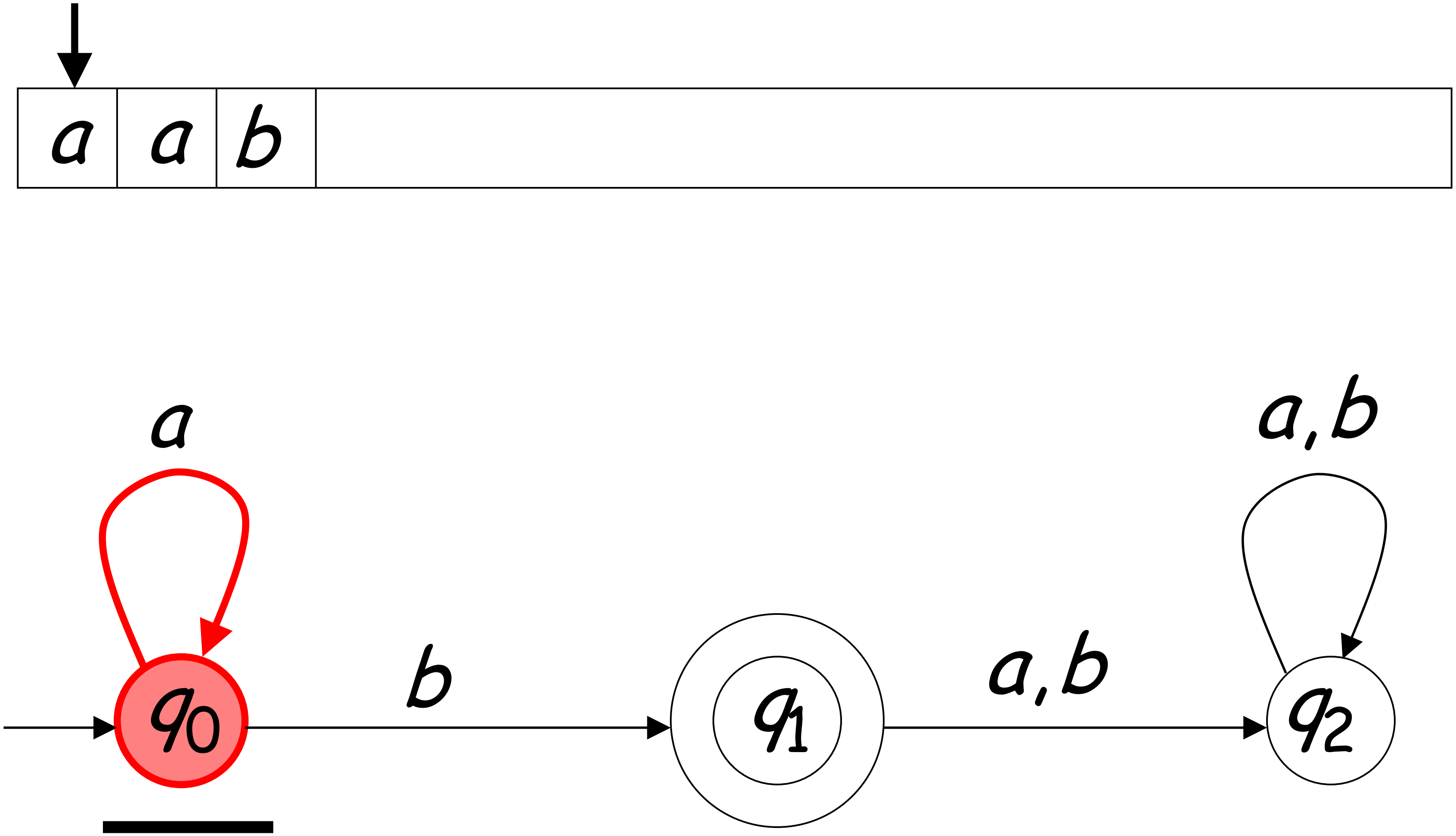
a
b
a
a,b
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4
q5

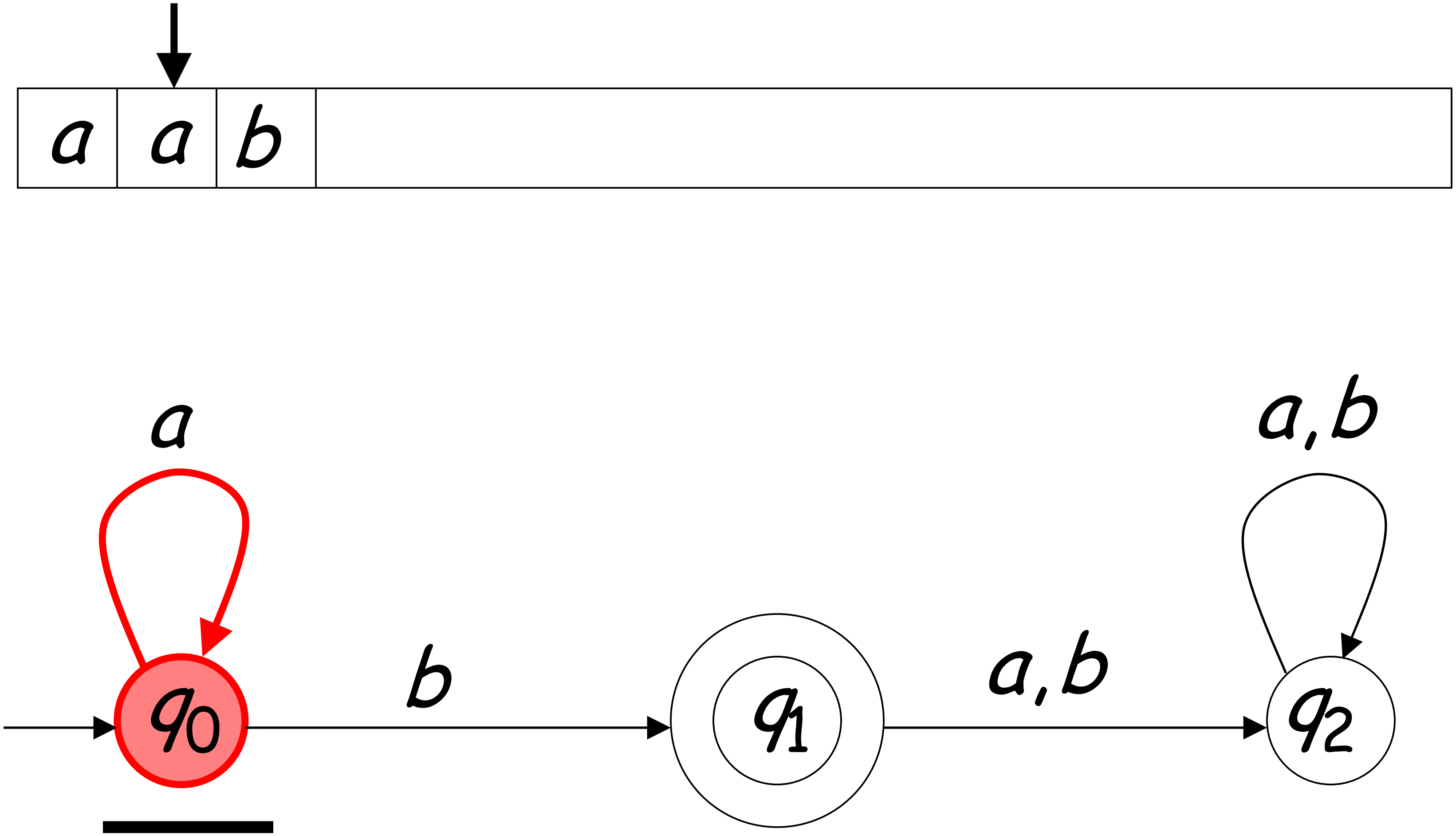
Input finished



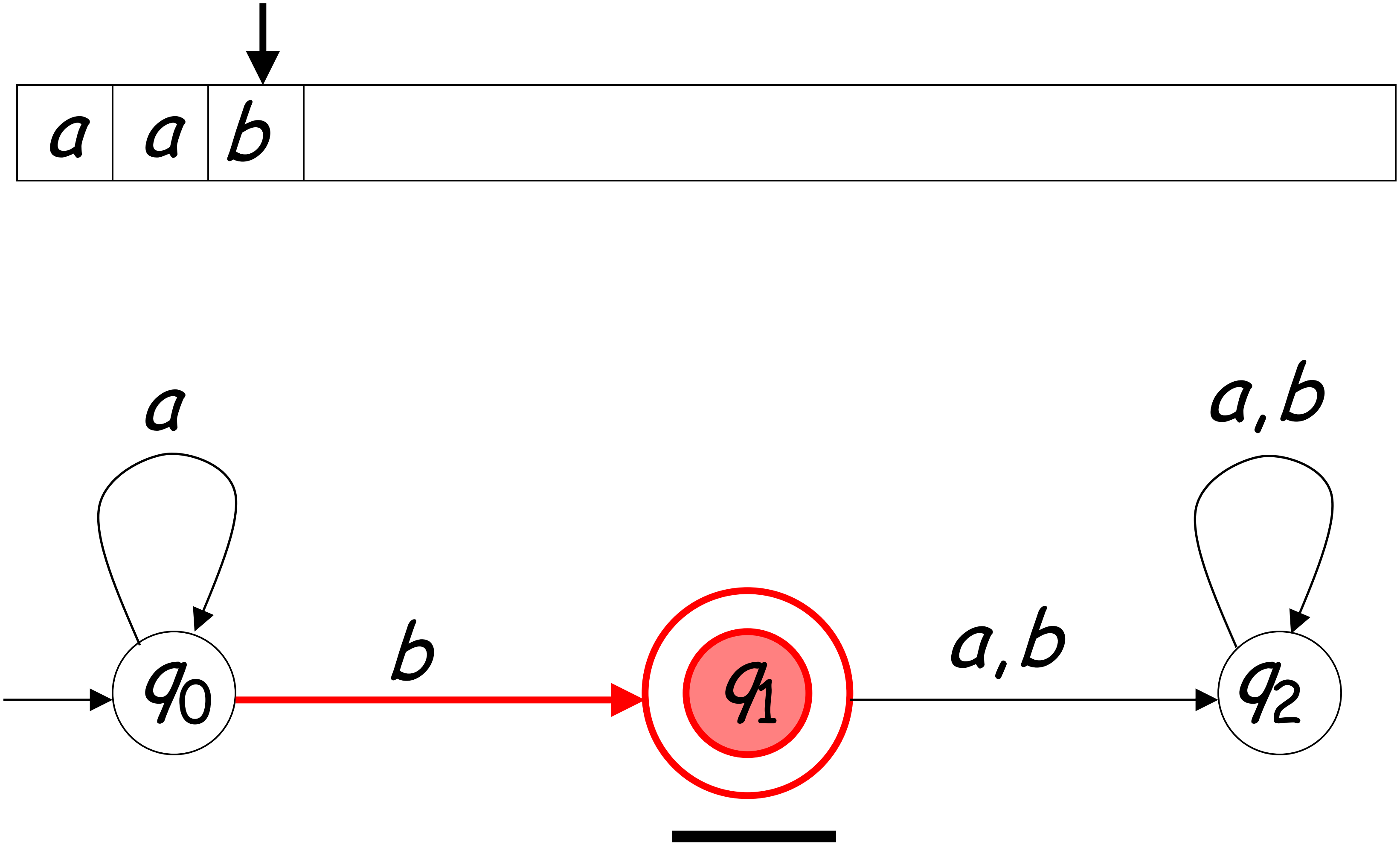
Another Example




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

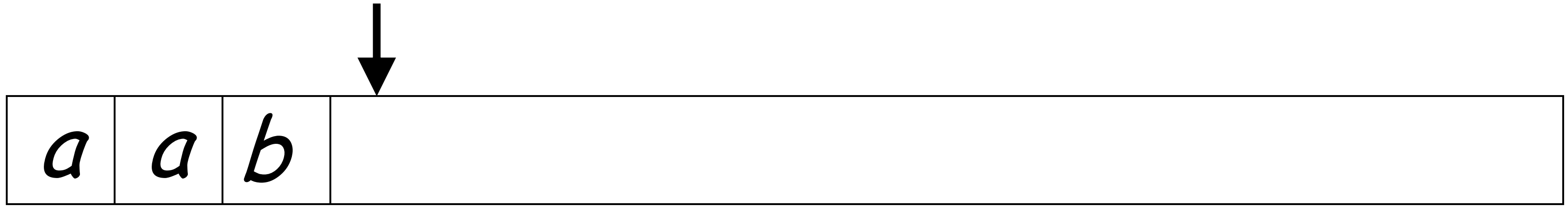


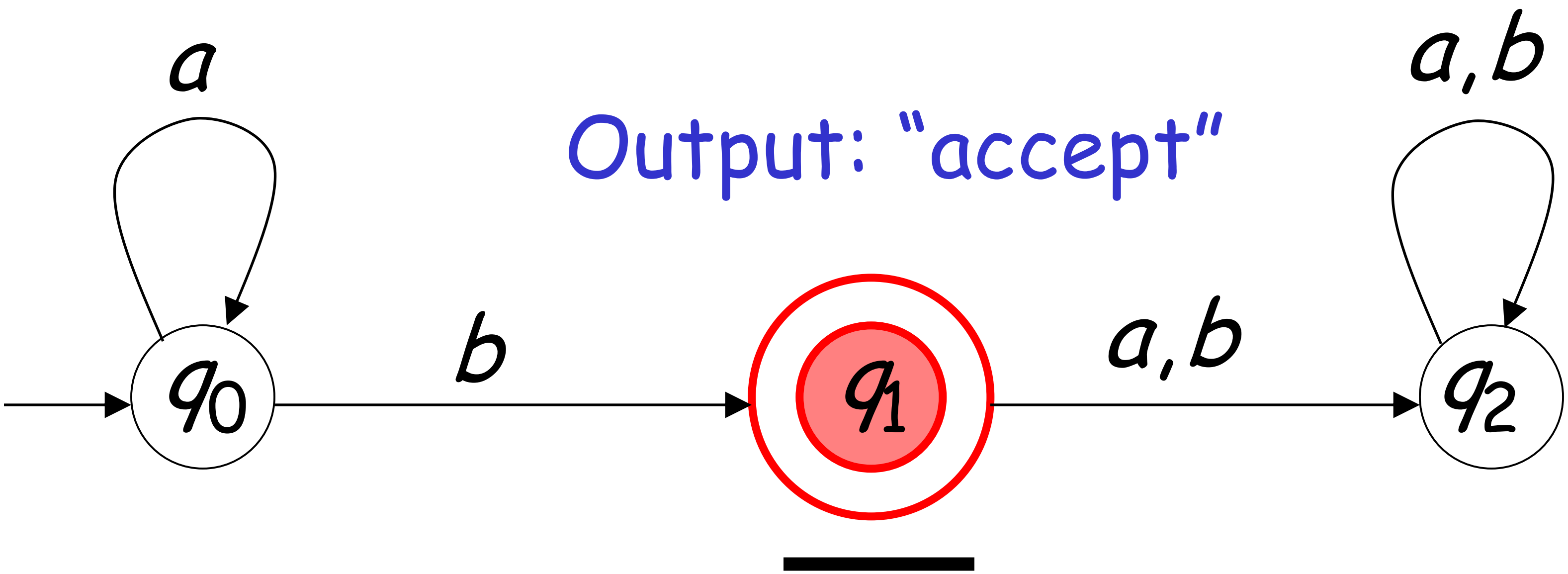
a
a
b
a
a,b
b
q_0
q_1
a,b
q_2



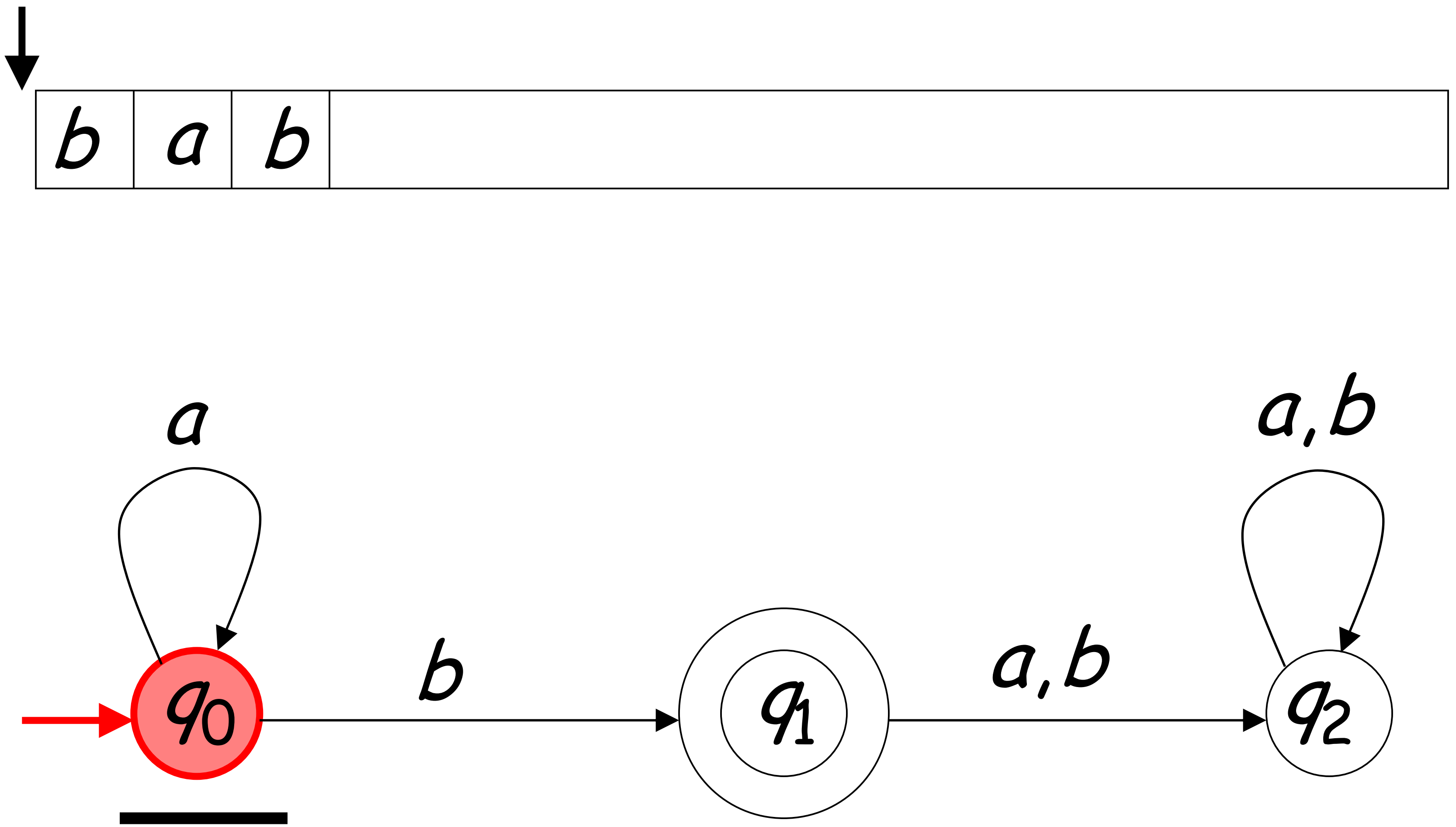
a
a
b
a
a,b
b
a,b
q_0
q_1
q_2

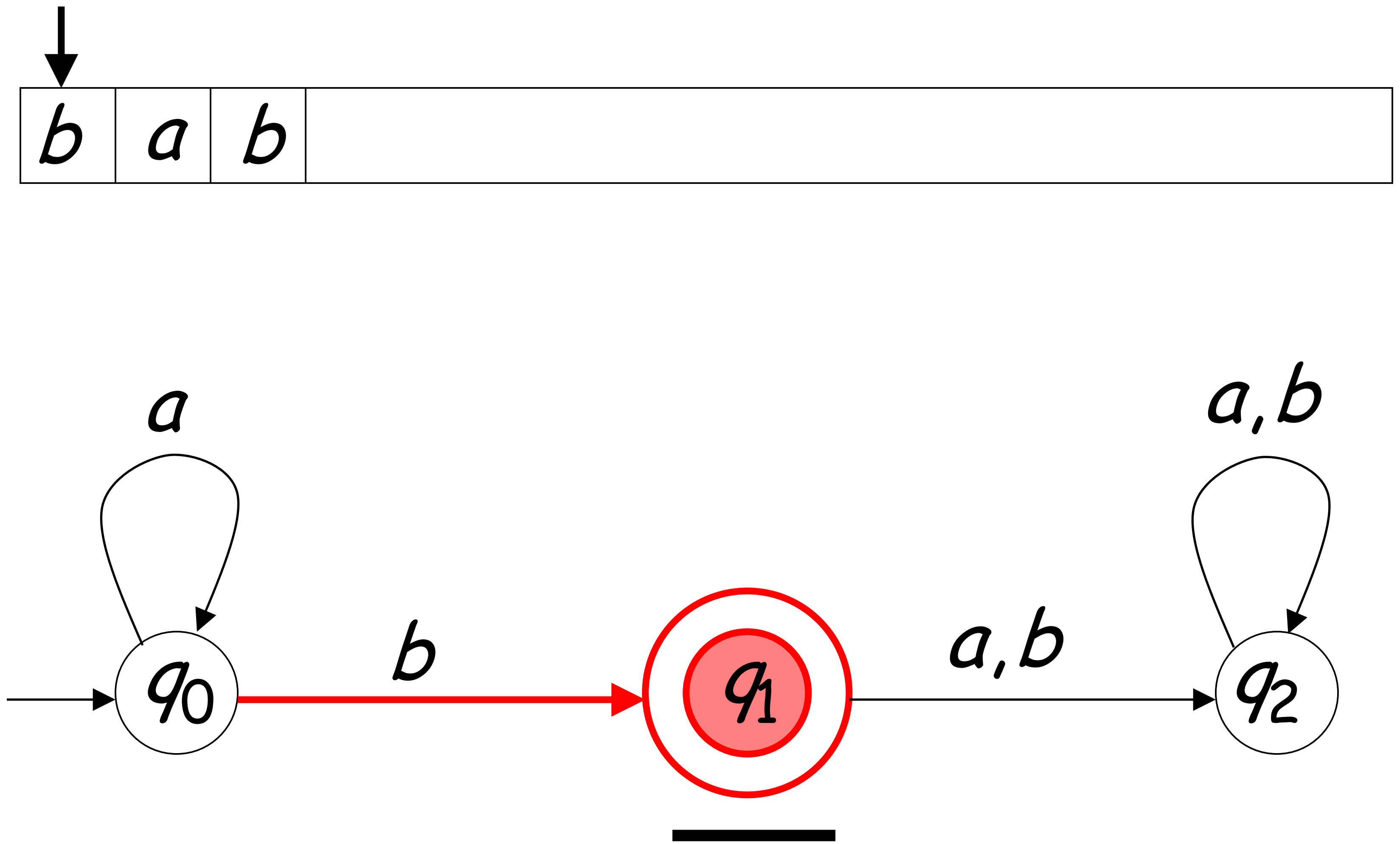
Input finished

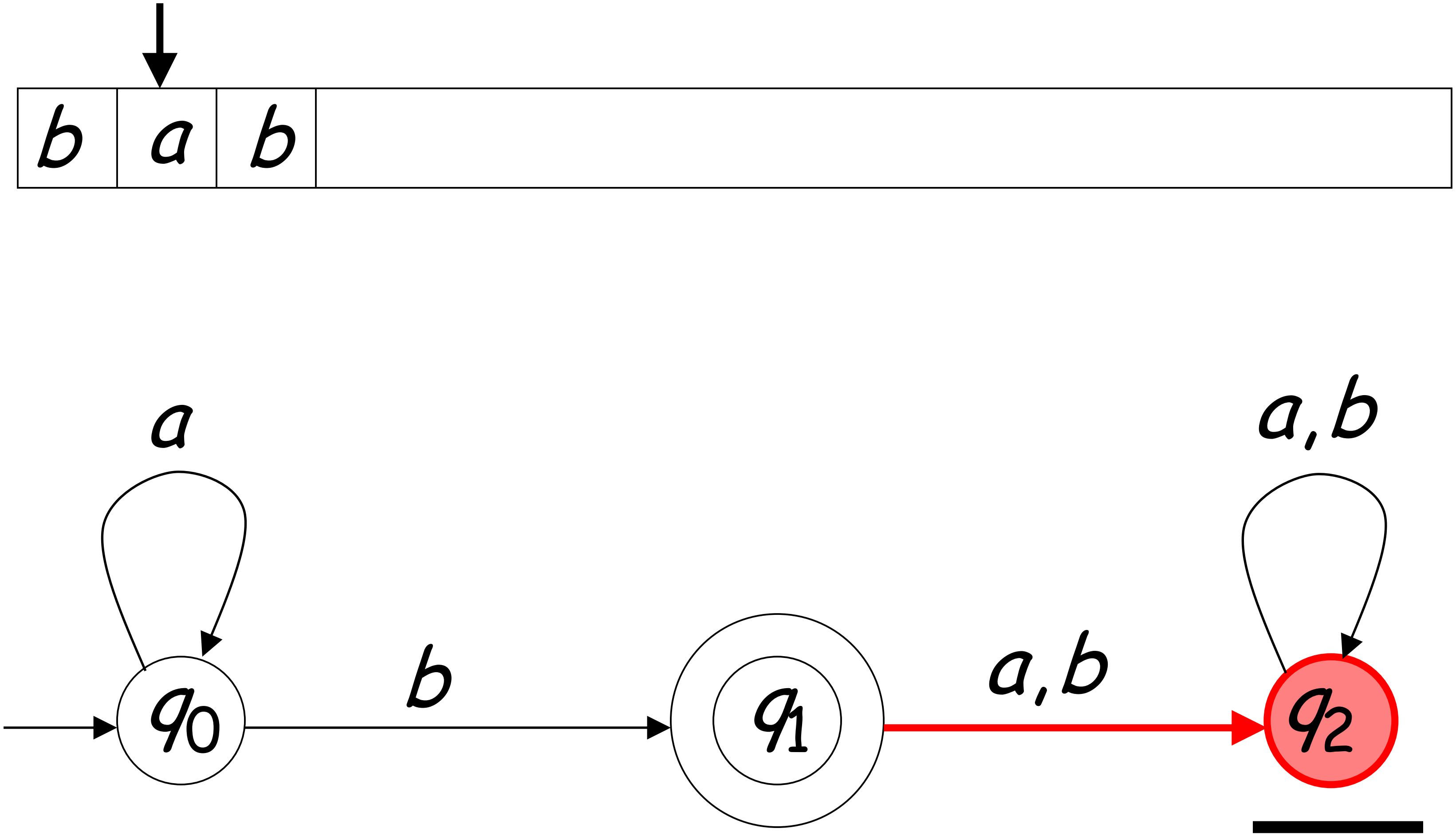


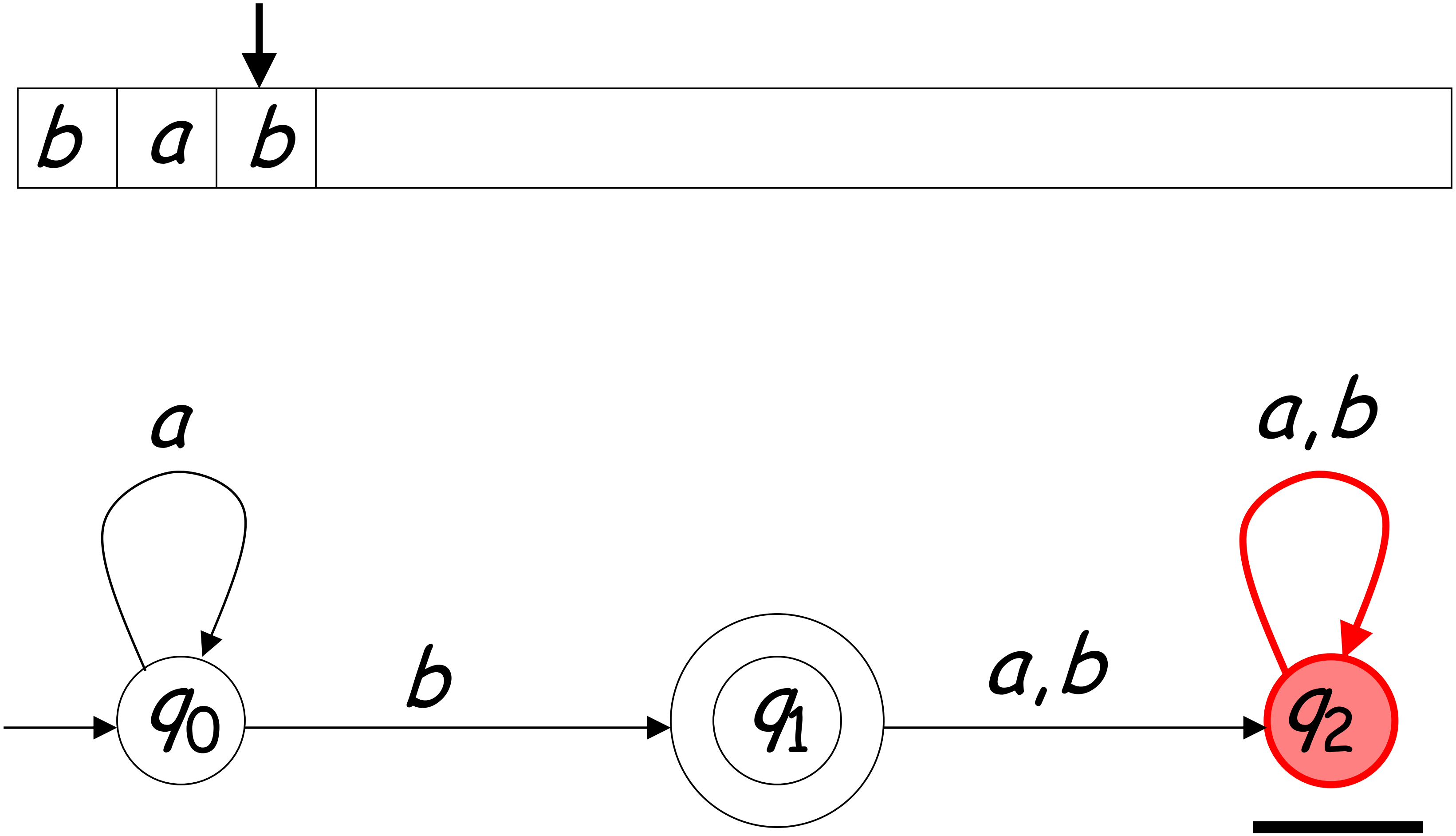


Rejection

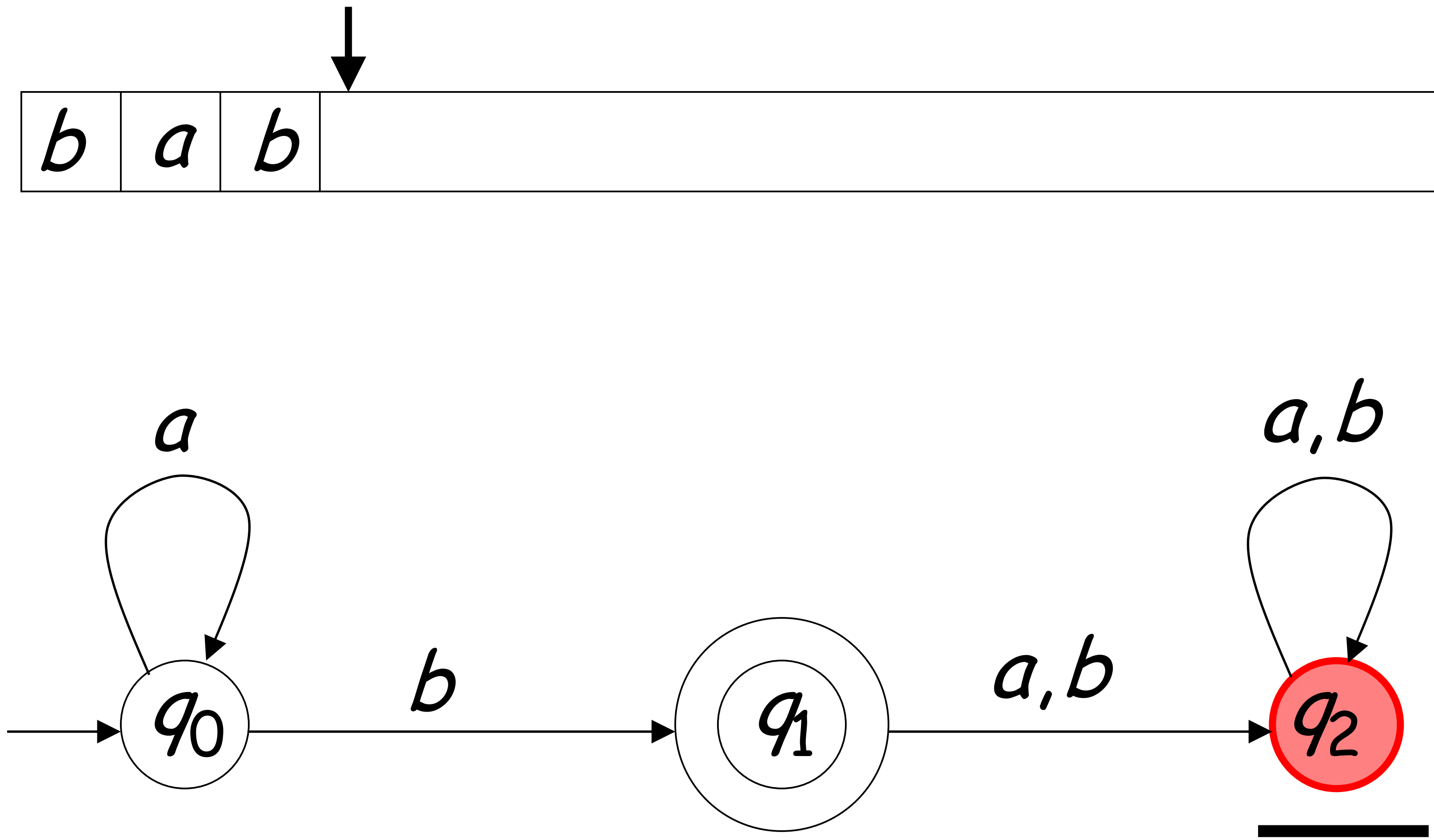



b a b
a
a,b
b
a,b
q_0
q_1
q_2


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


b
a
b
a
a,b
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

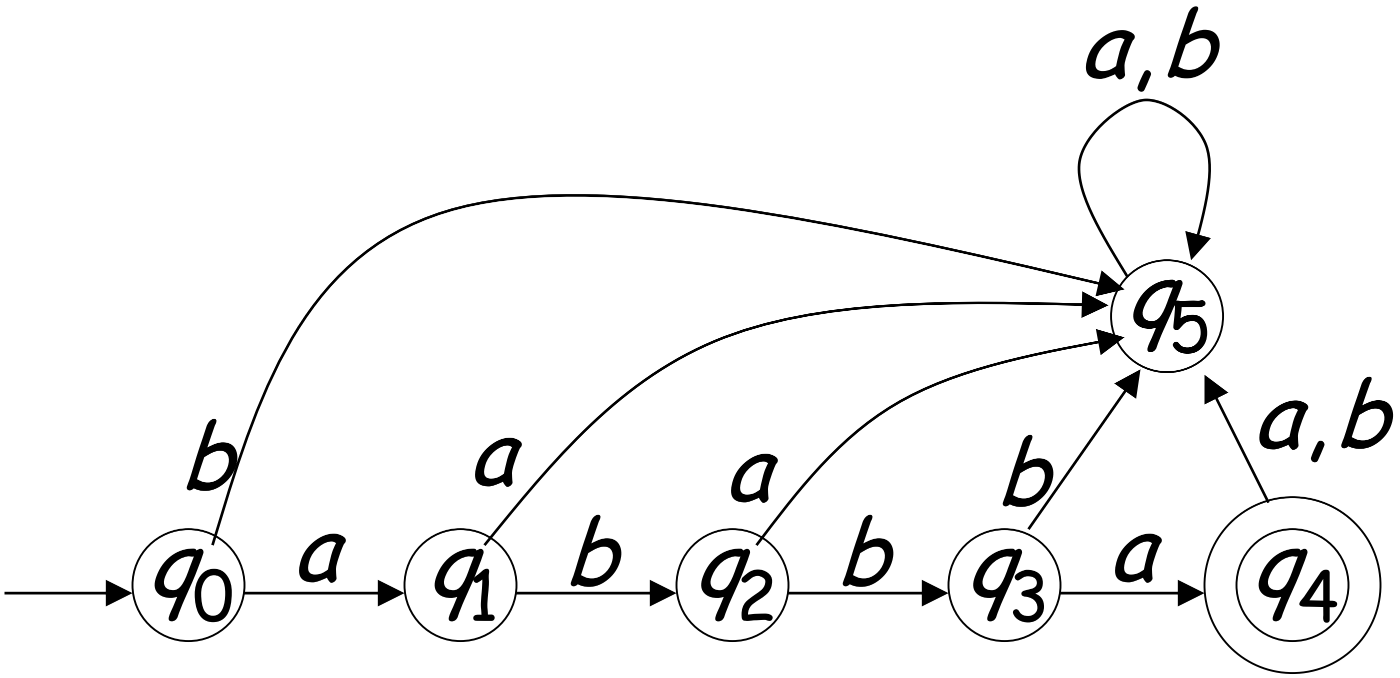
δ : transition function

q_0 : initial state

F : set of final states

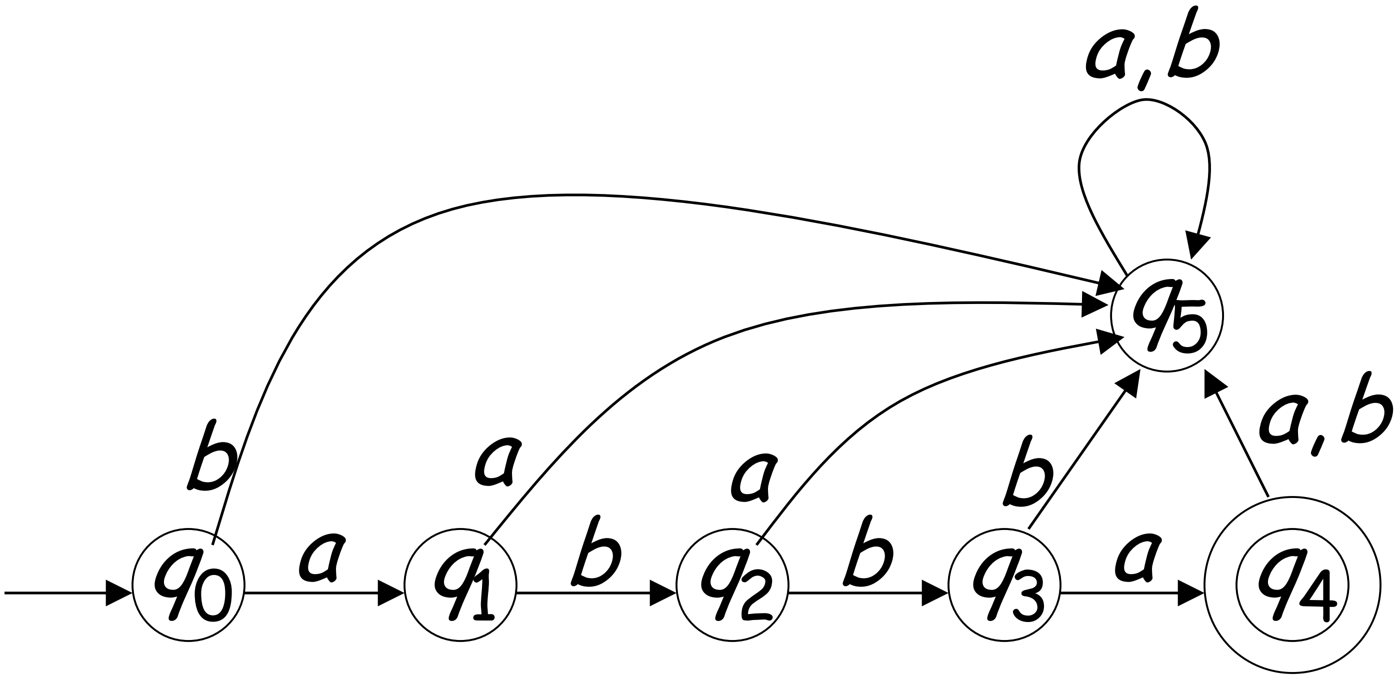
Input Alphabet Σ

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

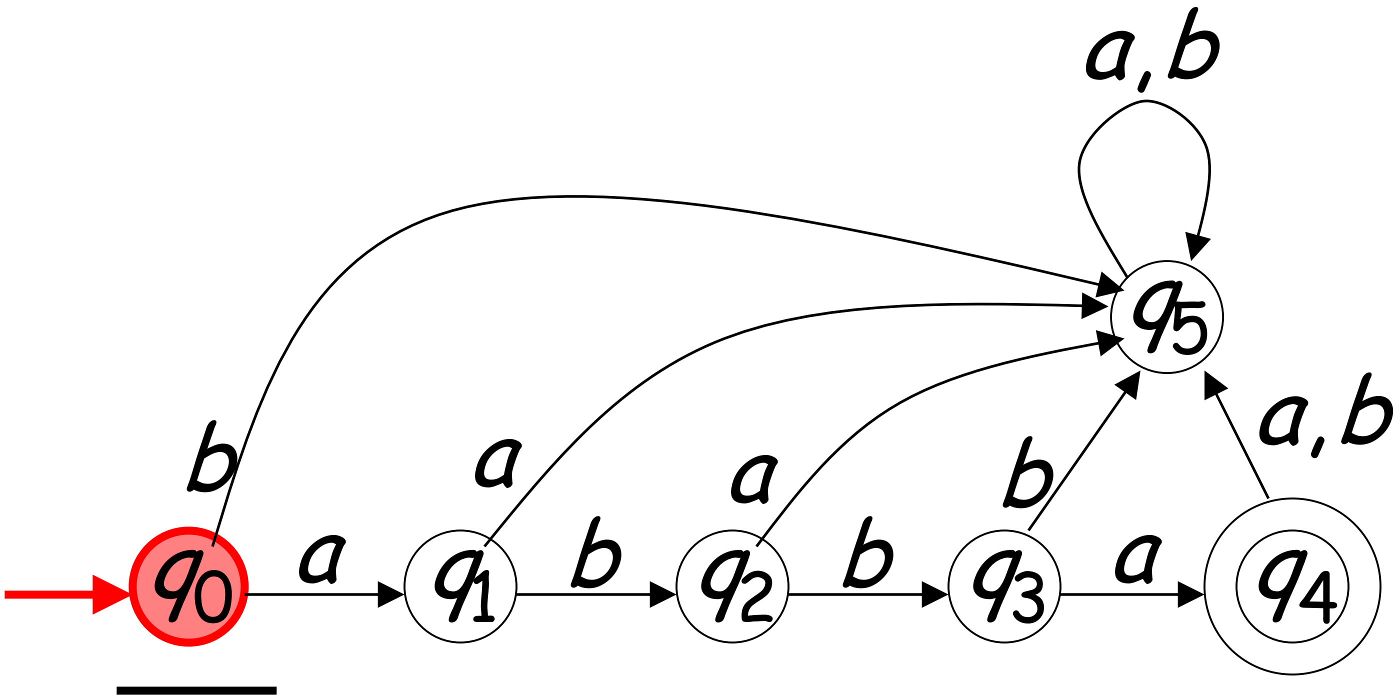


Set of States Q

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

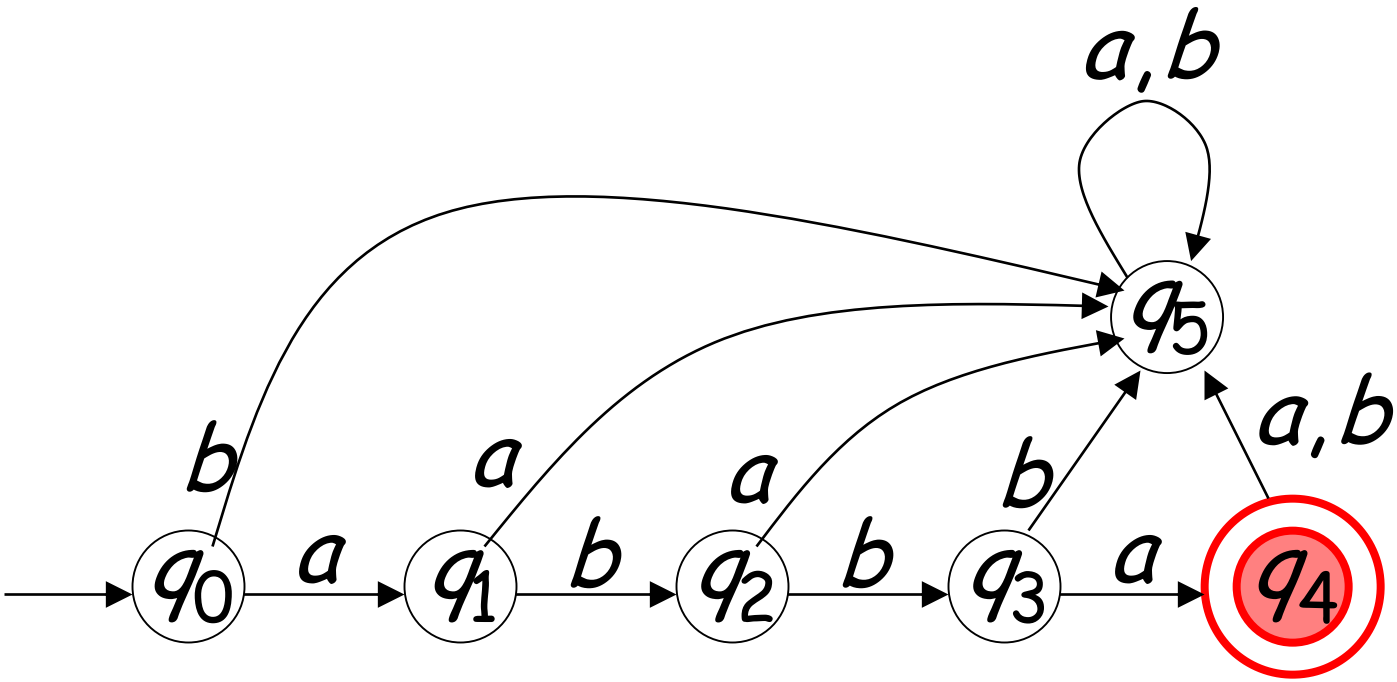


Initial State q_0



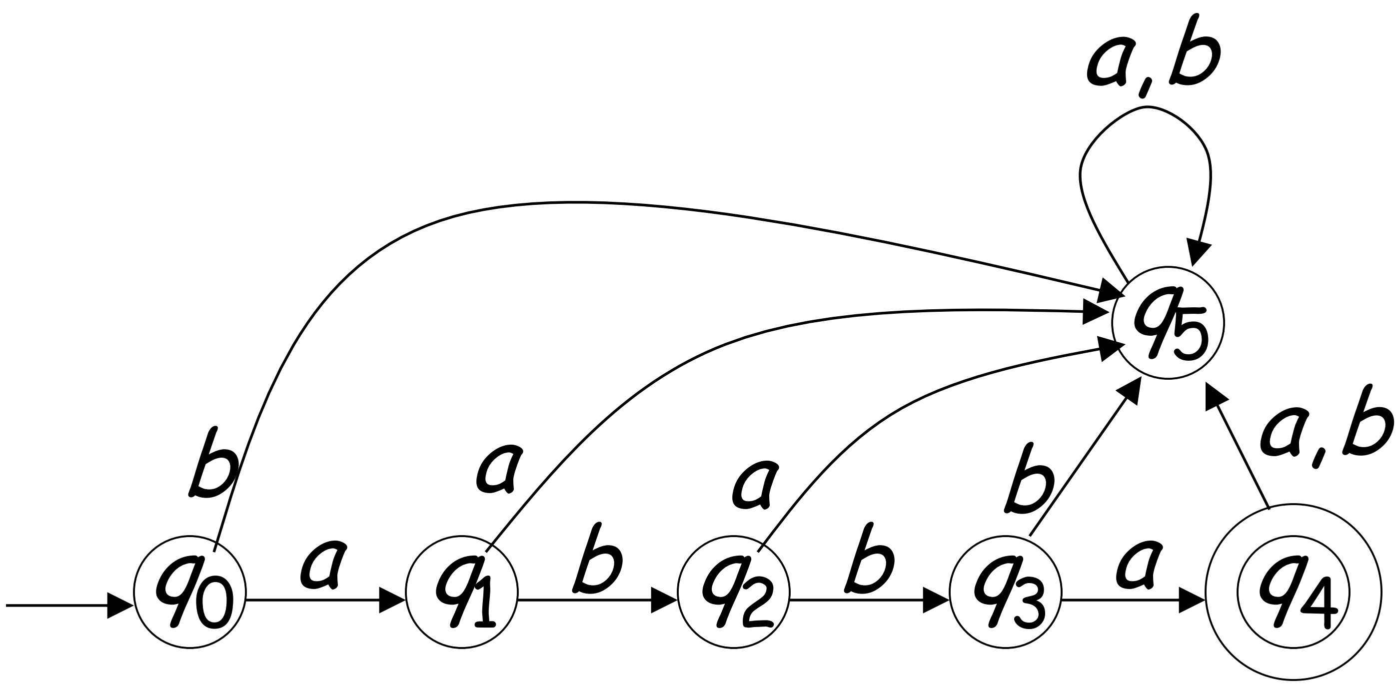
Set of Final States F

$$F = \{q_4\}$$

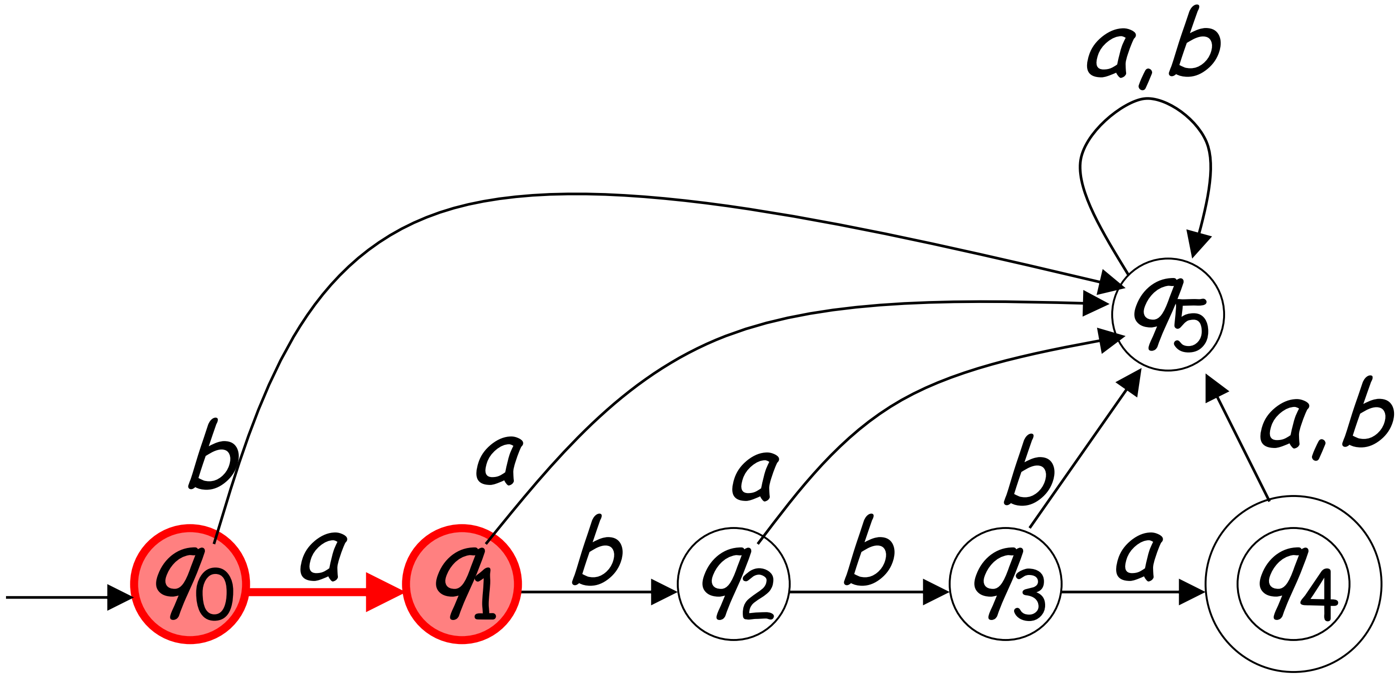


Transition Function δ

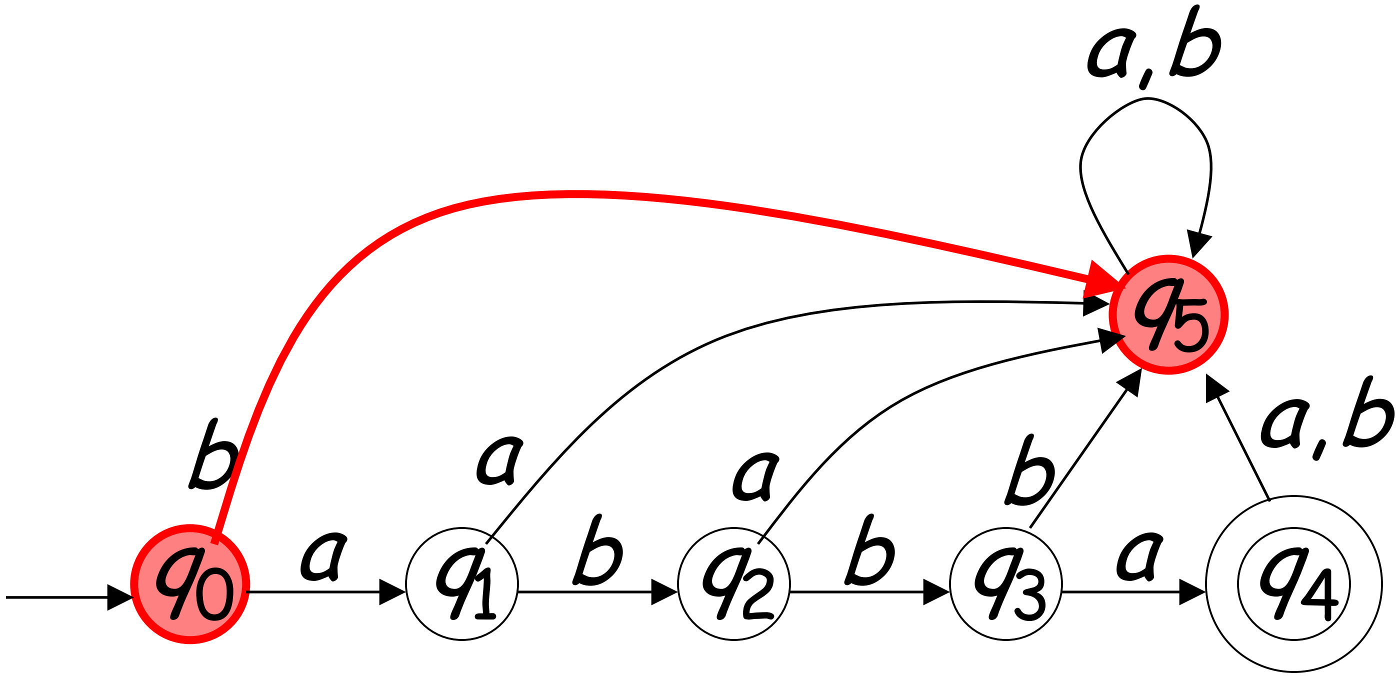
$$\delta : Q \times \Sigma \to Q$$



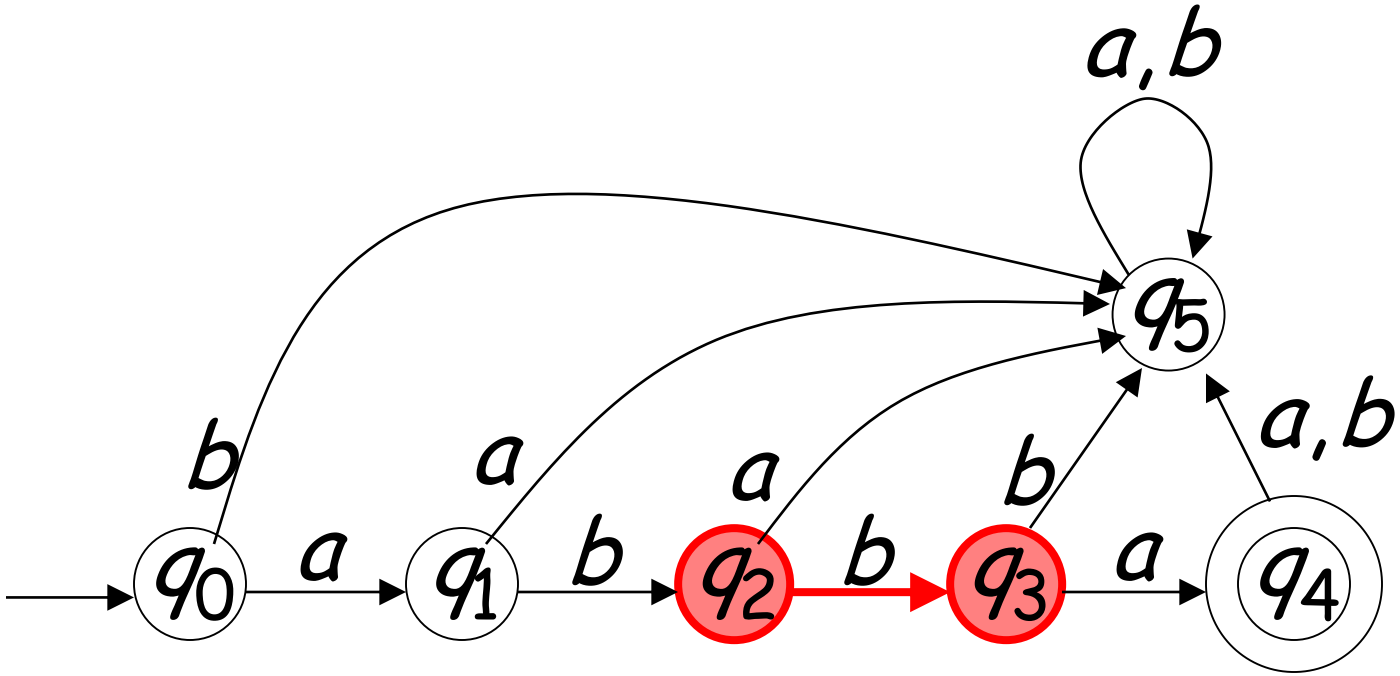
$$\delta(q_0, a) = q_1$$



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

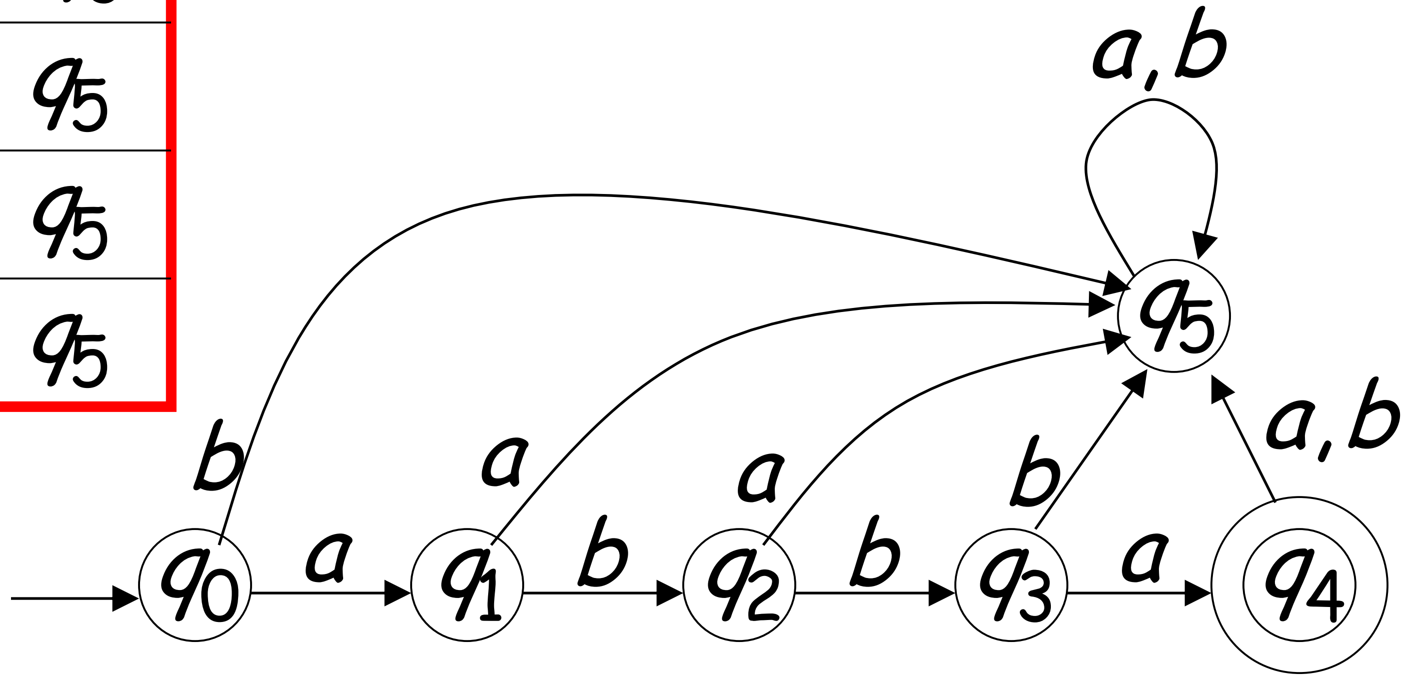


$$\delta(q_2, b) = q_3$$



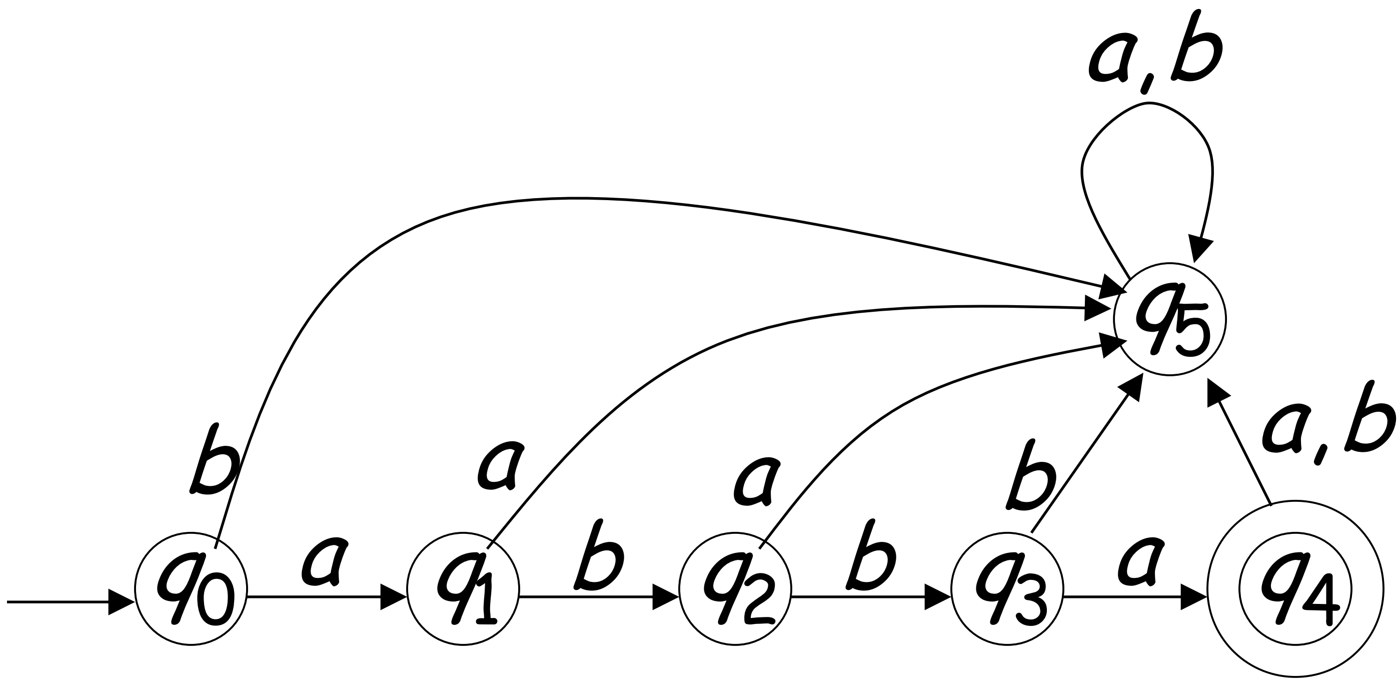
Transition Function δ

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

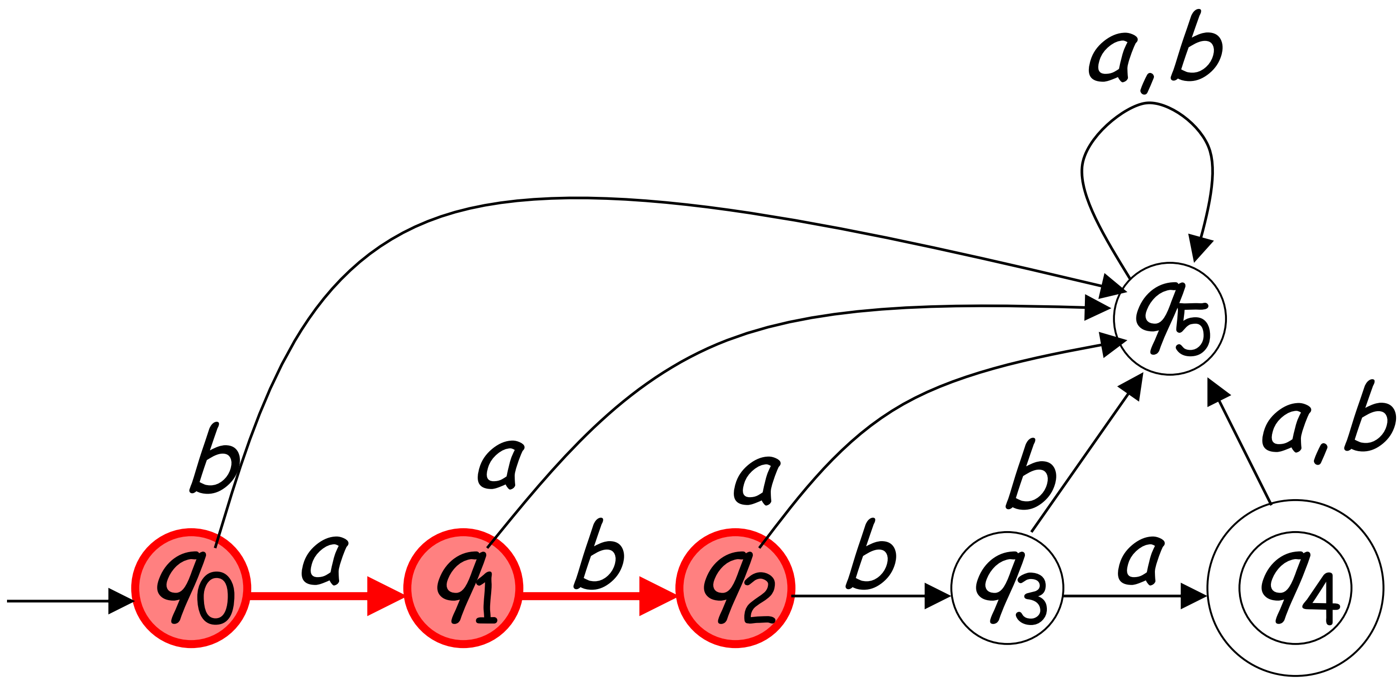


Extended Transition Function δ^*

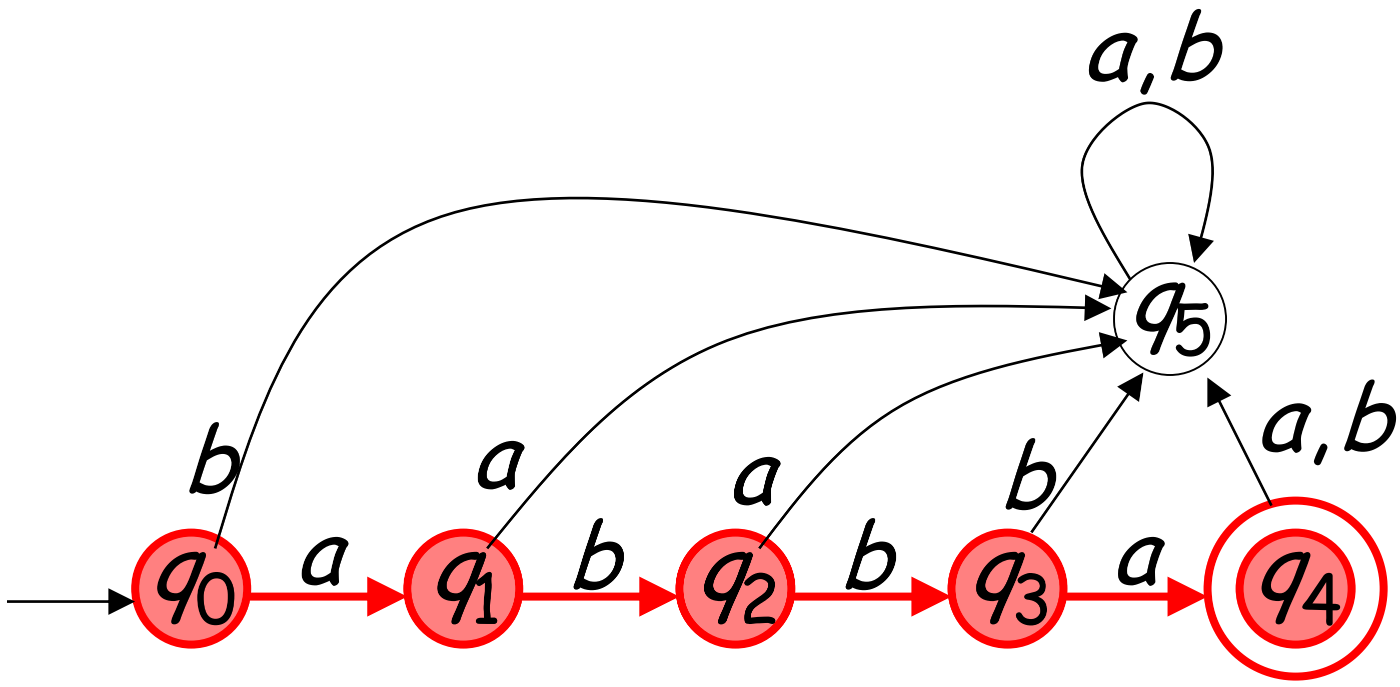
$$\delta^* : Q \times \Sigma^* \to Q$$



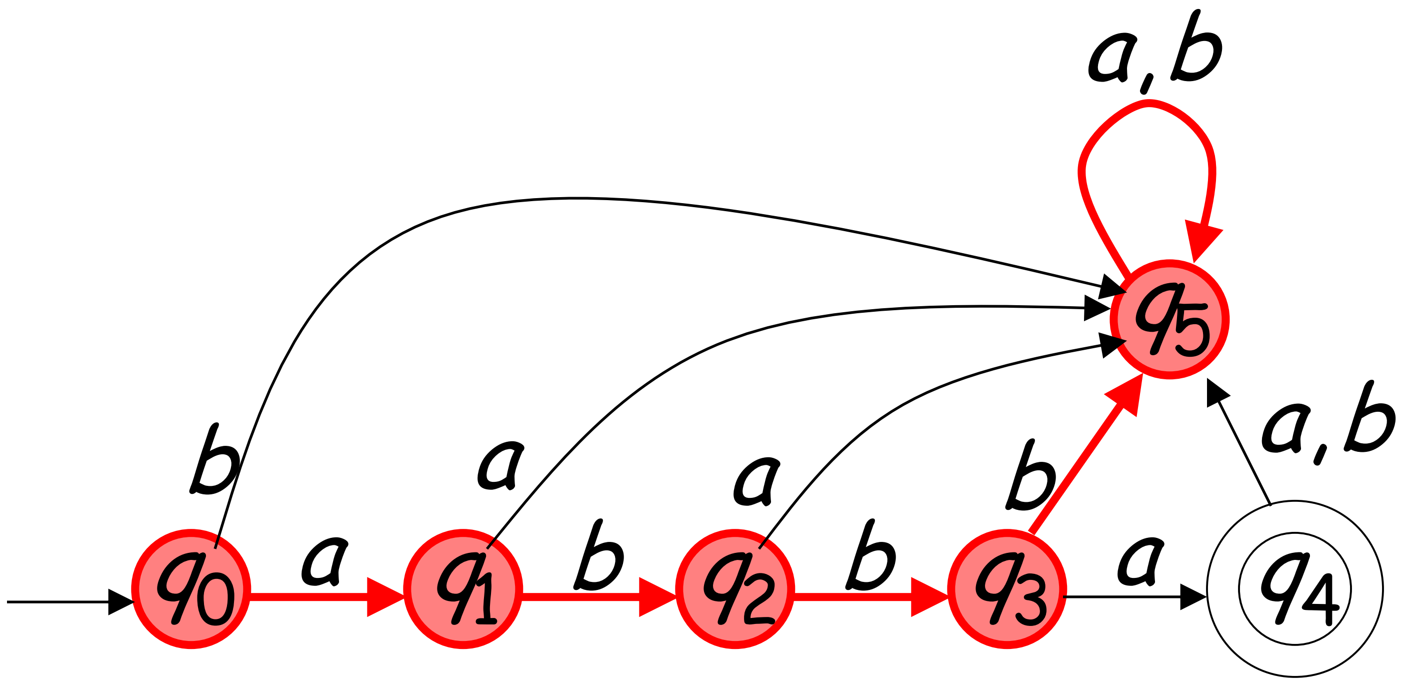
$$\delta^*(q_0, ab) = q_2$$



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

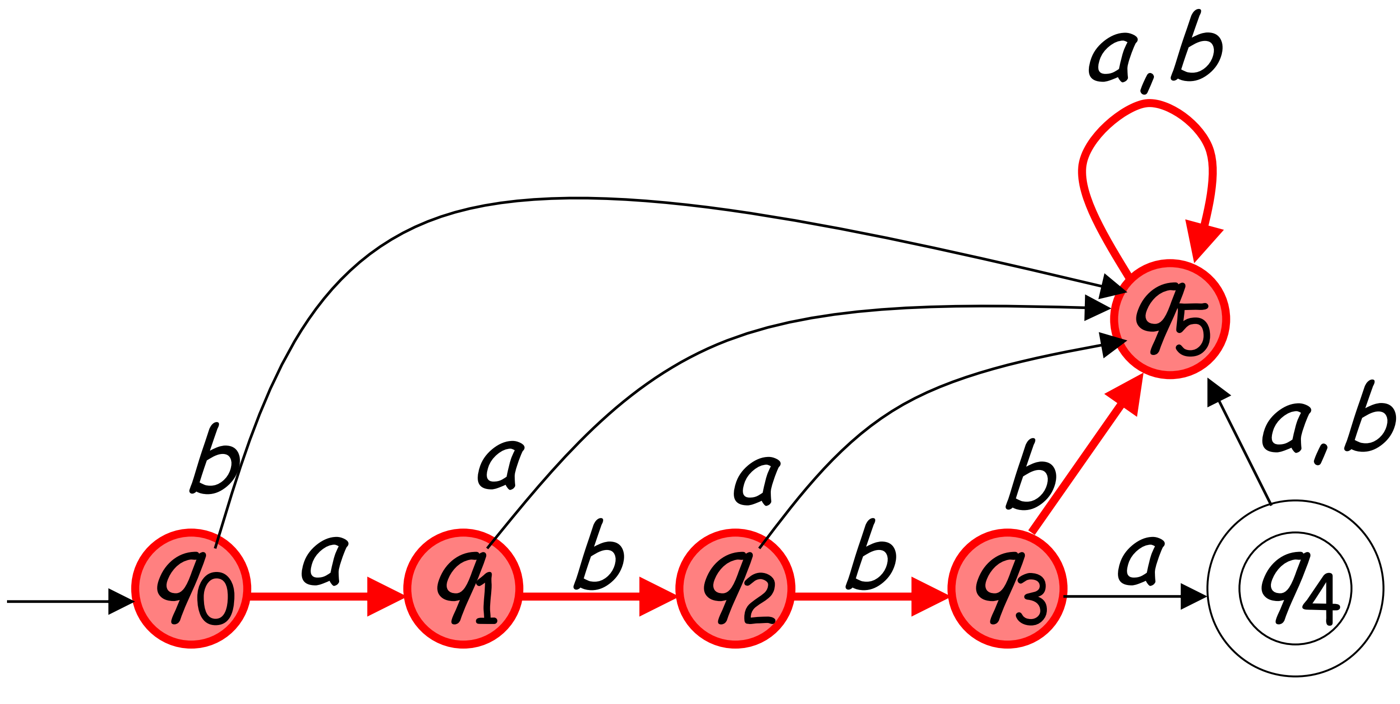


$$\delta^*(q_0, abbbaa) = q_5$$



Observation: There is a walk from q_0 to q_1 with label $abbbaa$

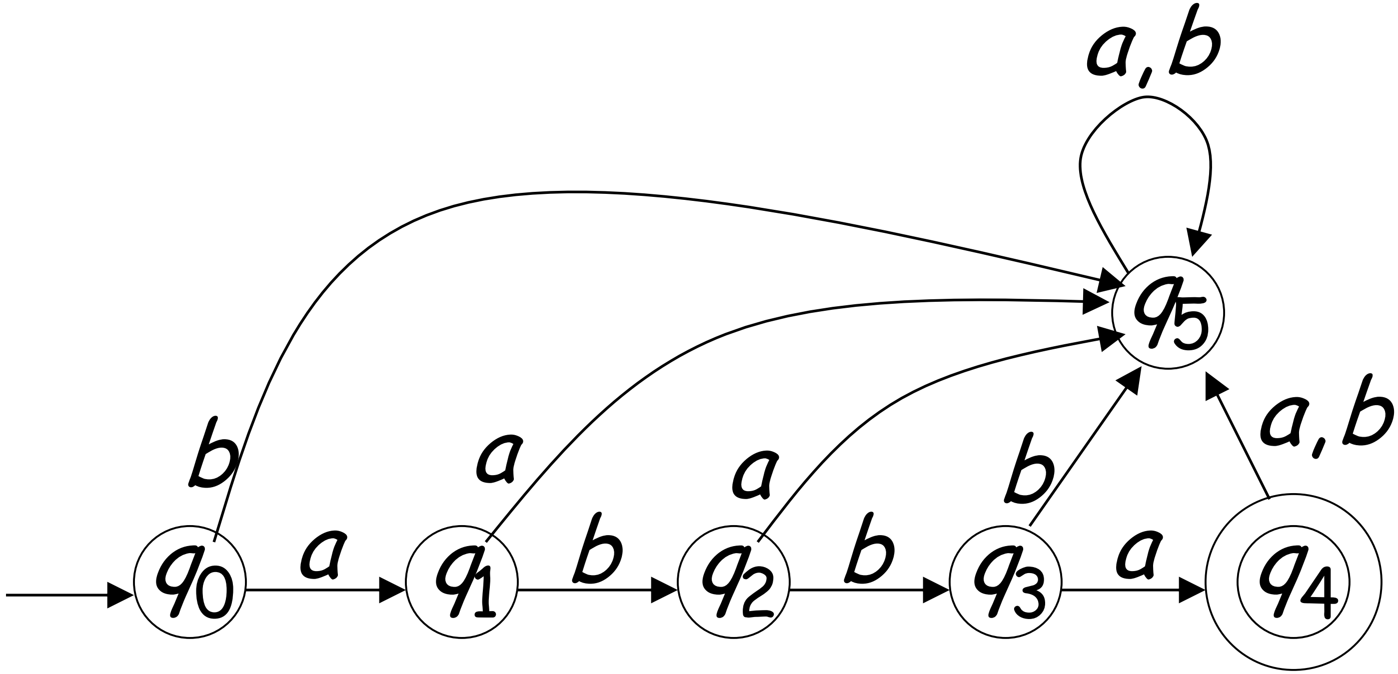
$$\delta^*(q_0, abbbaa) = q_5$$



Recursive Definition

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

$$\delta^*(q, wa) = \delta(\delta^*(q, w), a)$$



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

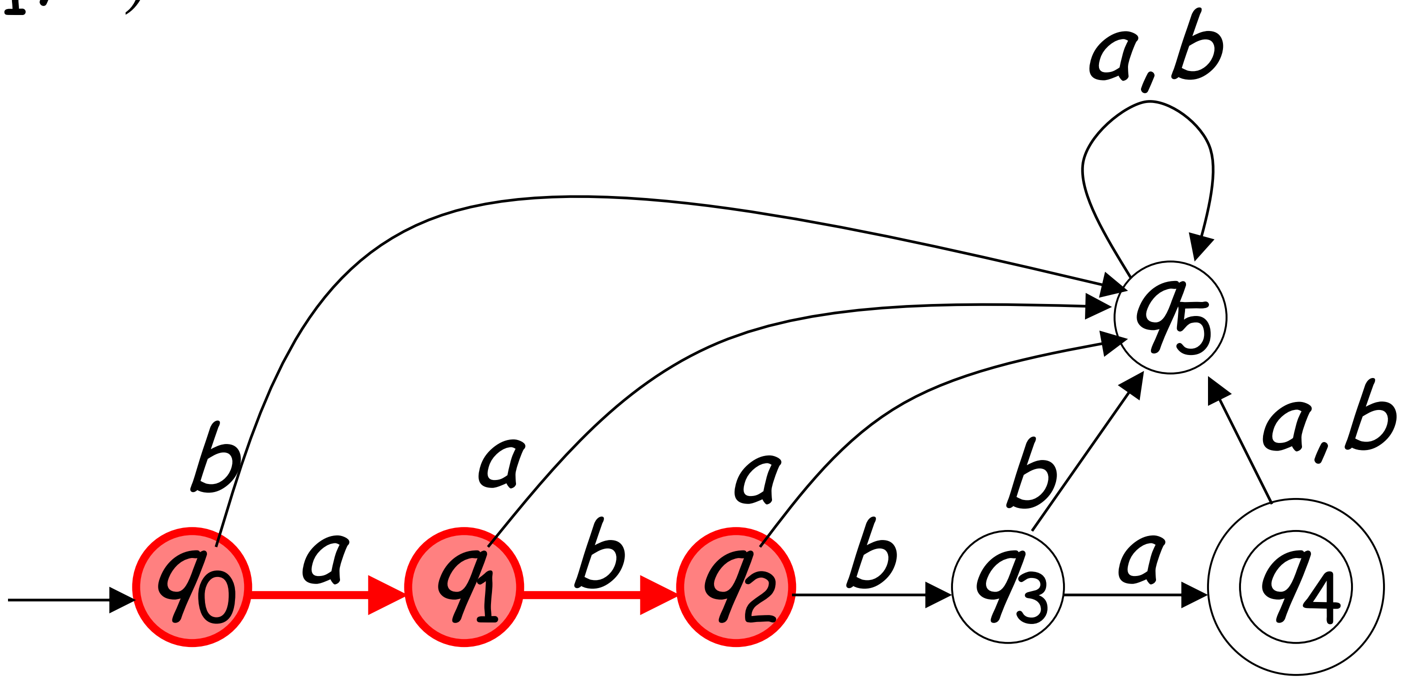
$$\delta(\delta^*(q_0, a), b) =$$

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

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

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

$$q_2$$



Languages Accepted by DFAs

Take DFA M

Definition:

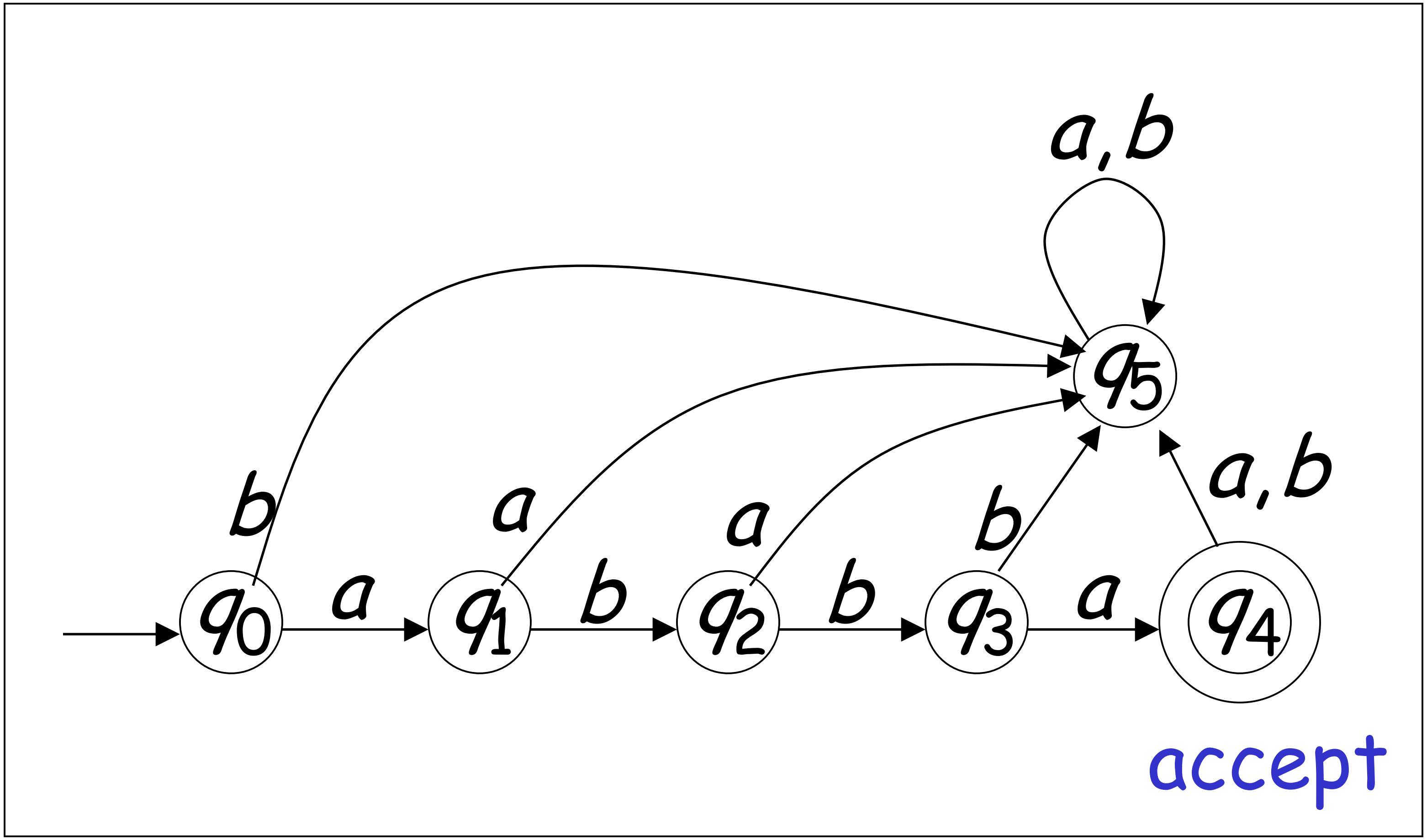
The language $L(M)$ contains all input strings accepted by M

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

Example

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

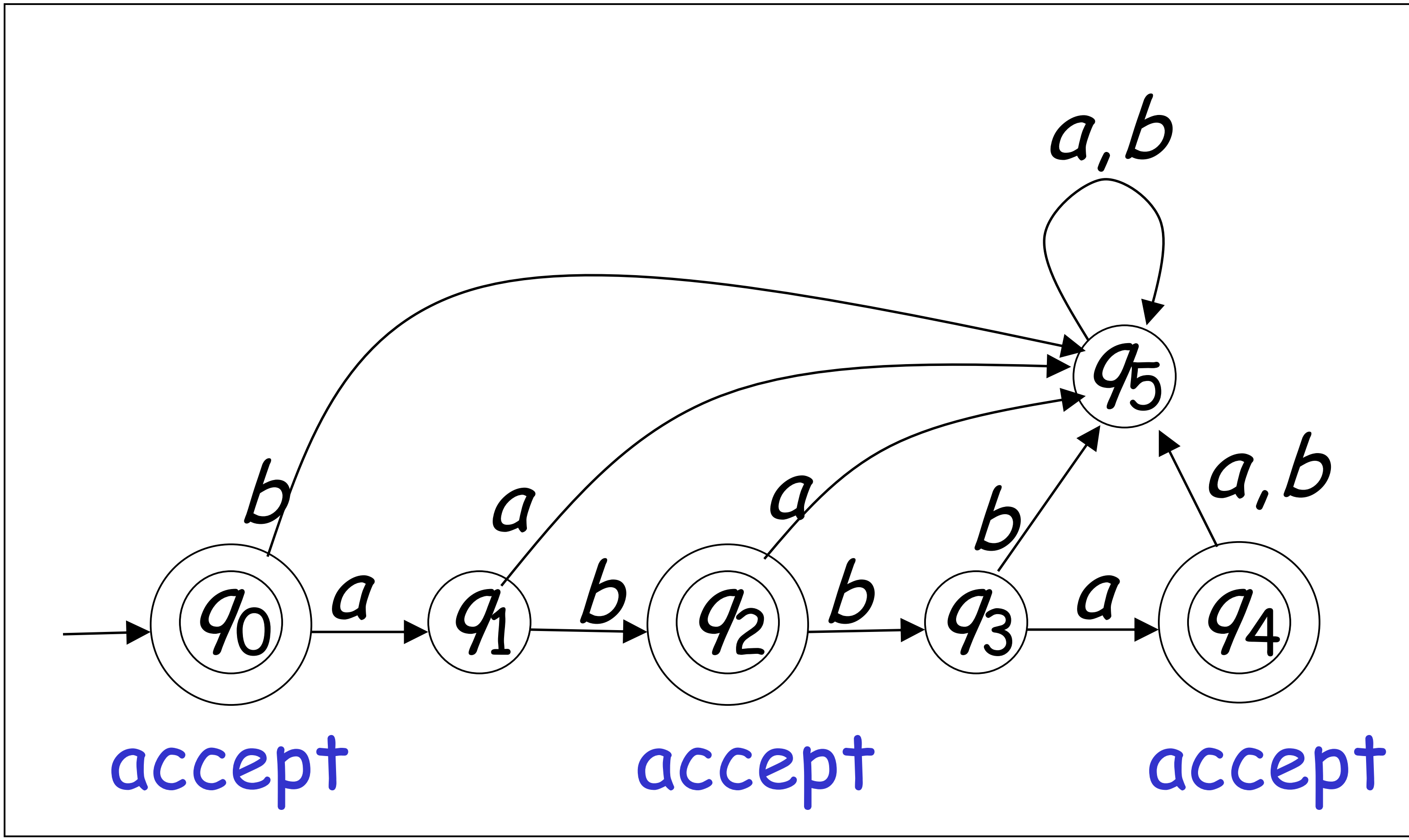
M



Another Example

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

M



Formally

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

Language accepted by M :

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

alphabet

transition function

initial state

final states

Observation

Language accepted by M:

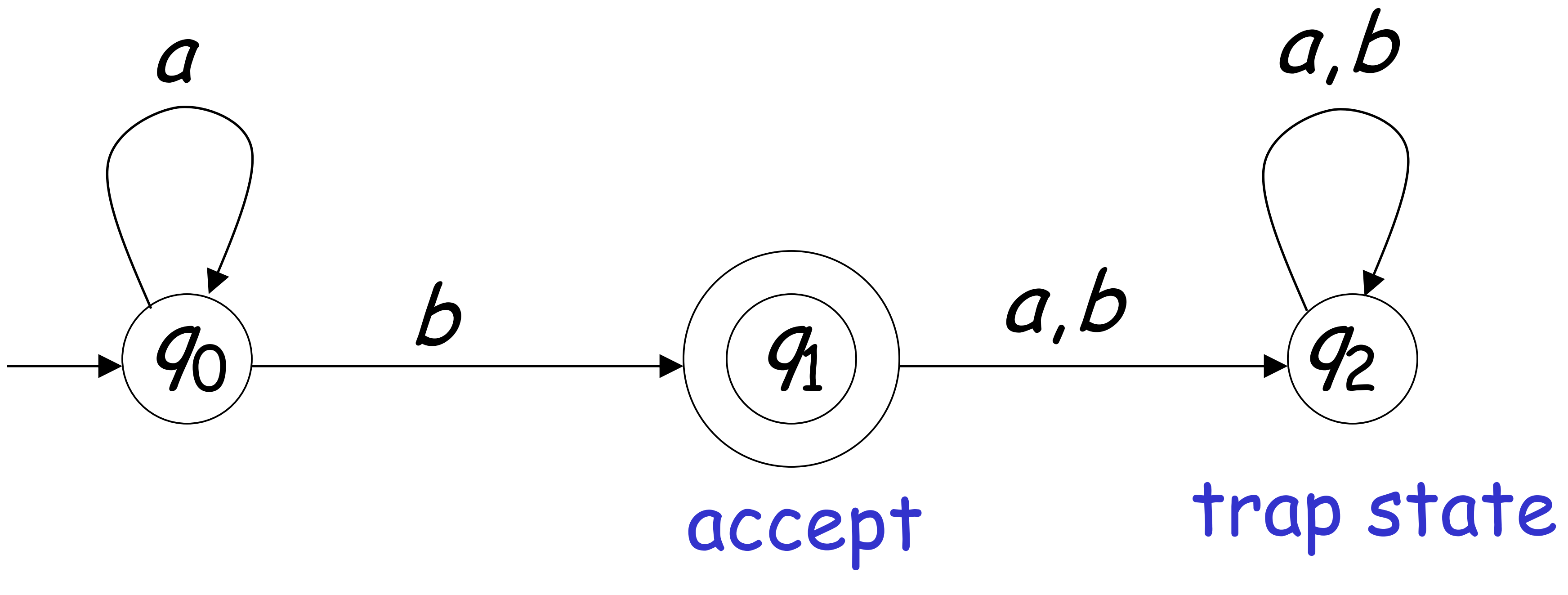
$$L(M) = \{ w \in \Sigma^* : \delta^*(q_0, w) \in F \}$$

Language rejected by M:

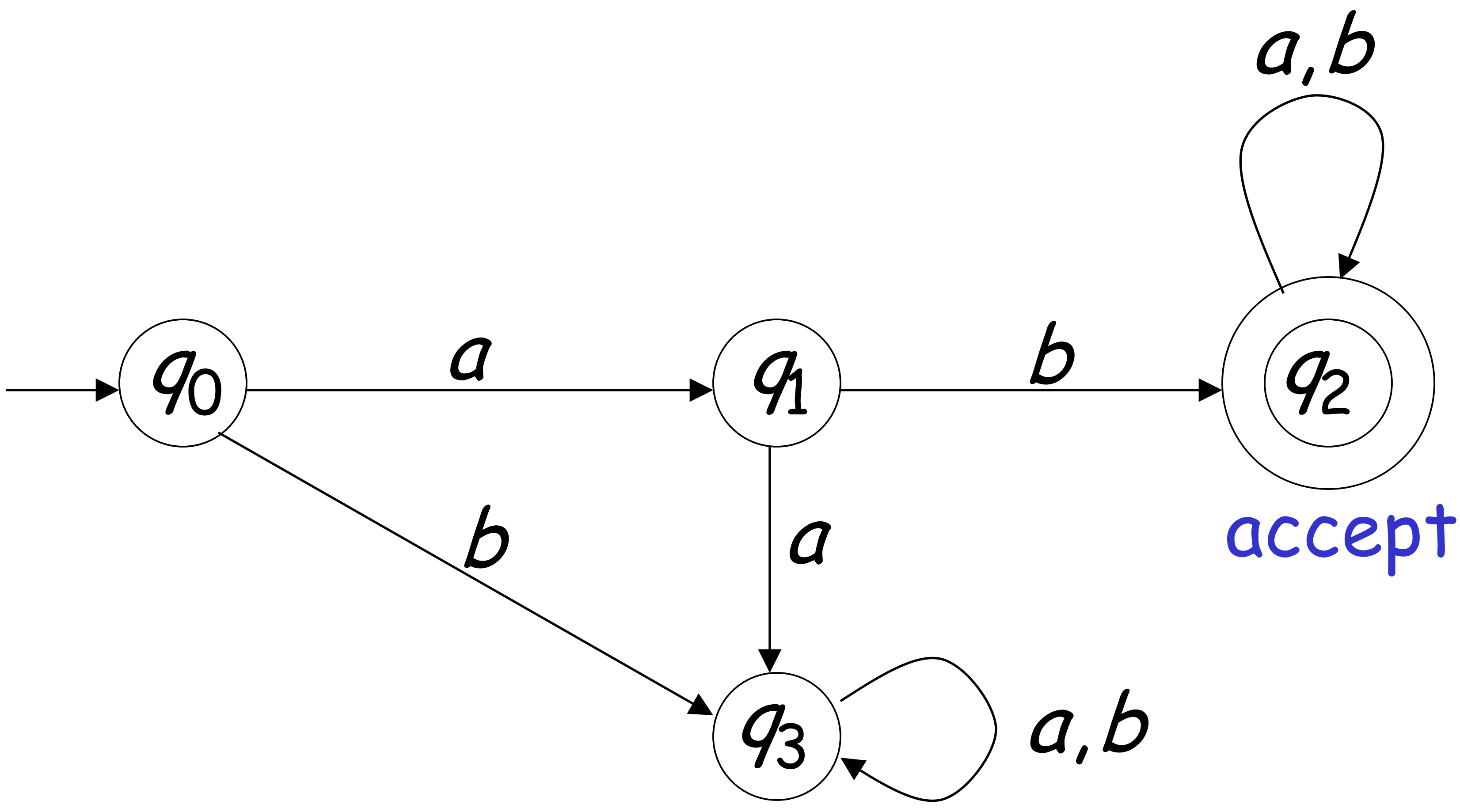
$$\overline{L(M)} = \{ w \in \Sigma^* : \delta^*(q_0, w) \notin F \}$$

More Examples

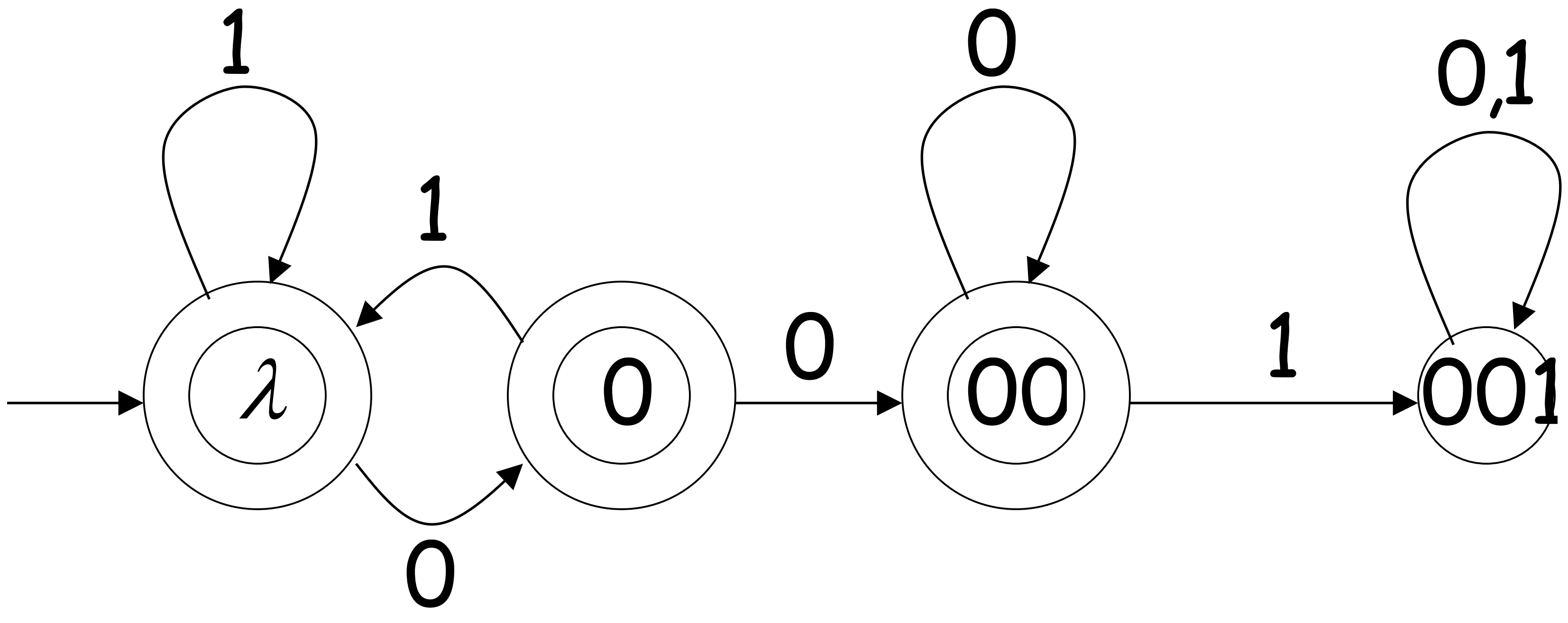
$$L(M) = \{a^n b : n \geq 0\}$$



$L(M)$= { all substrings 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

Example

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