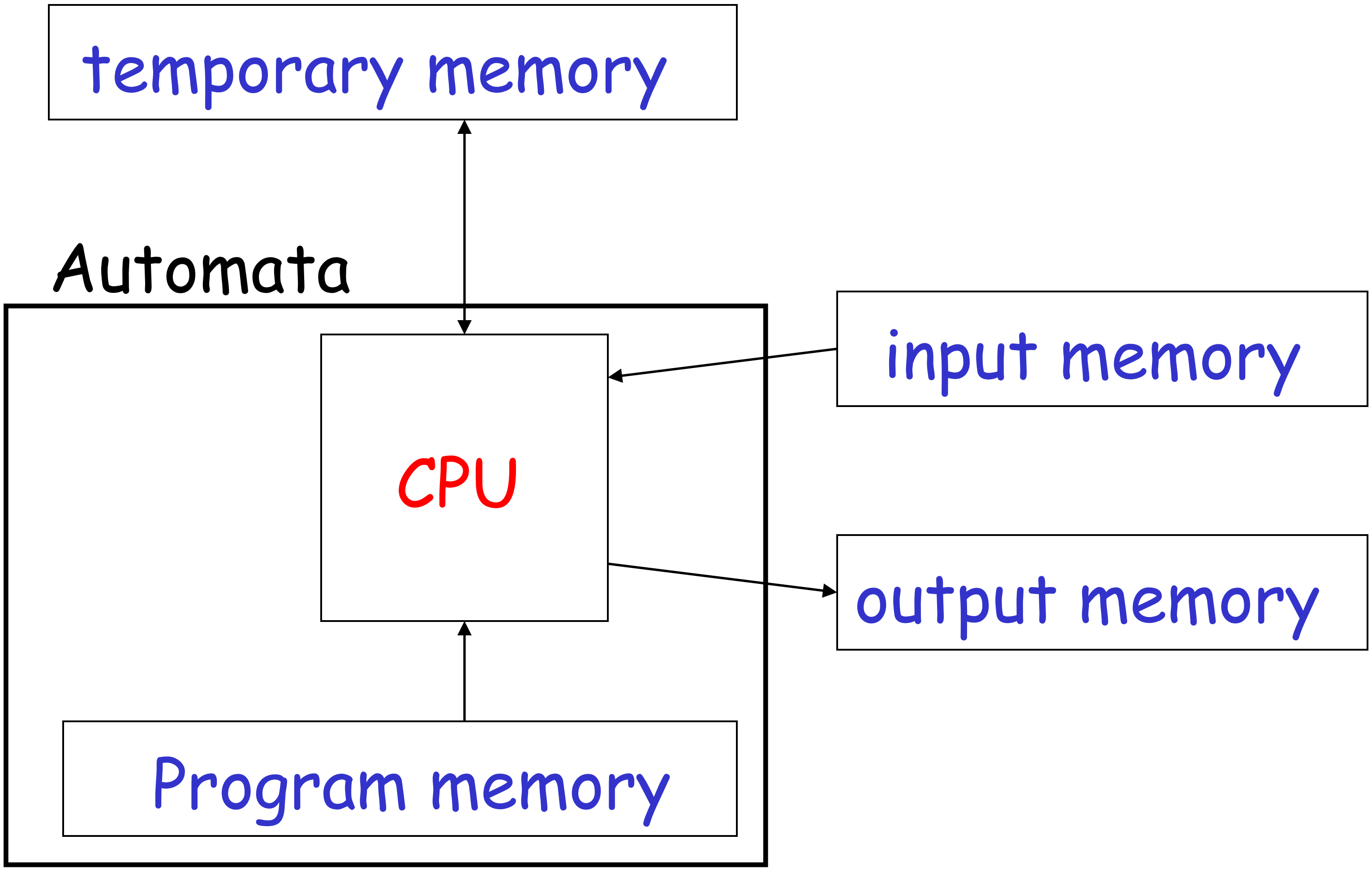
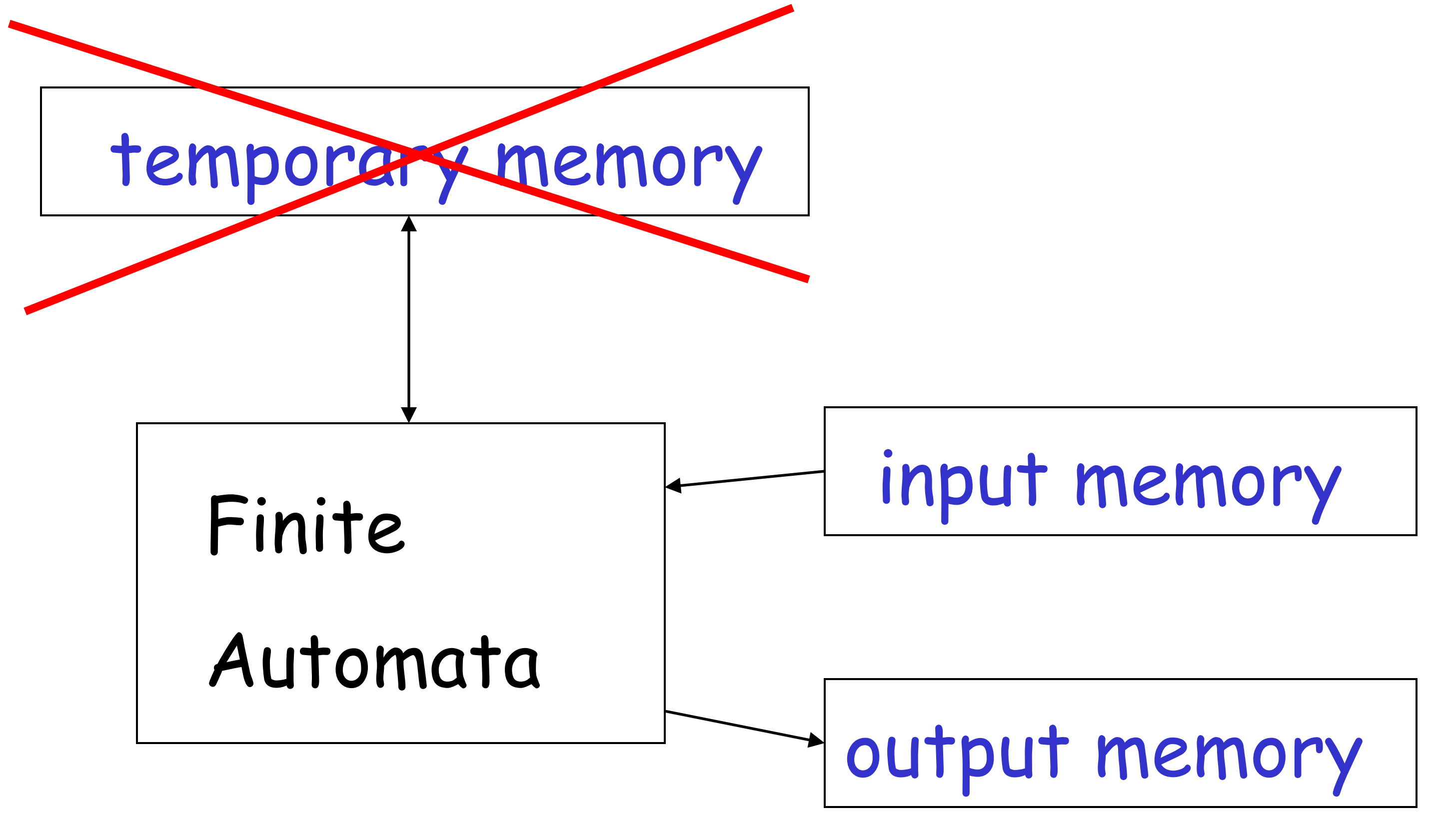
Finite Automata

Automata


temporary memory
Automata
CPU
input memory
output memory
Program memory

Finite Automata

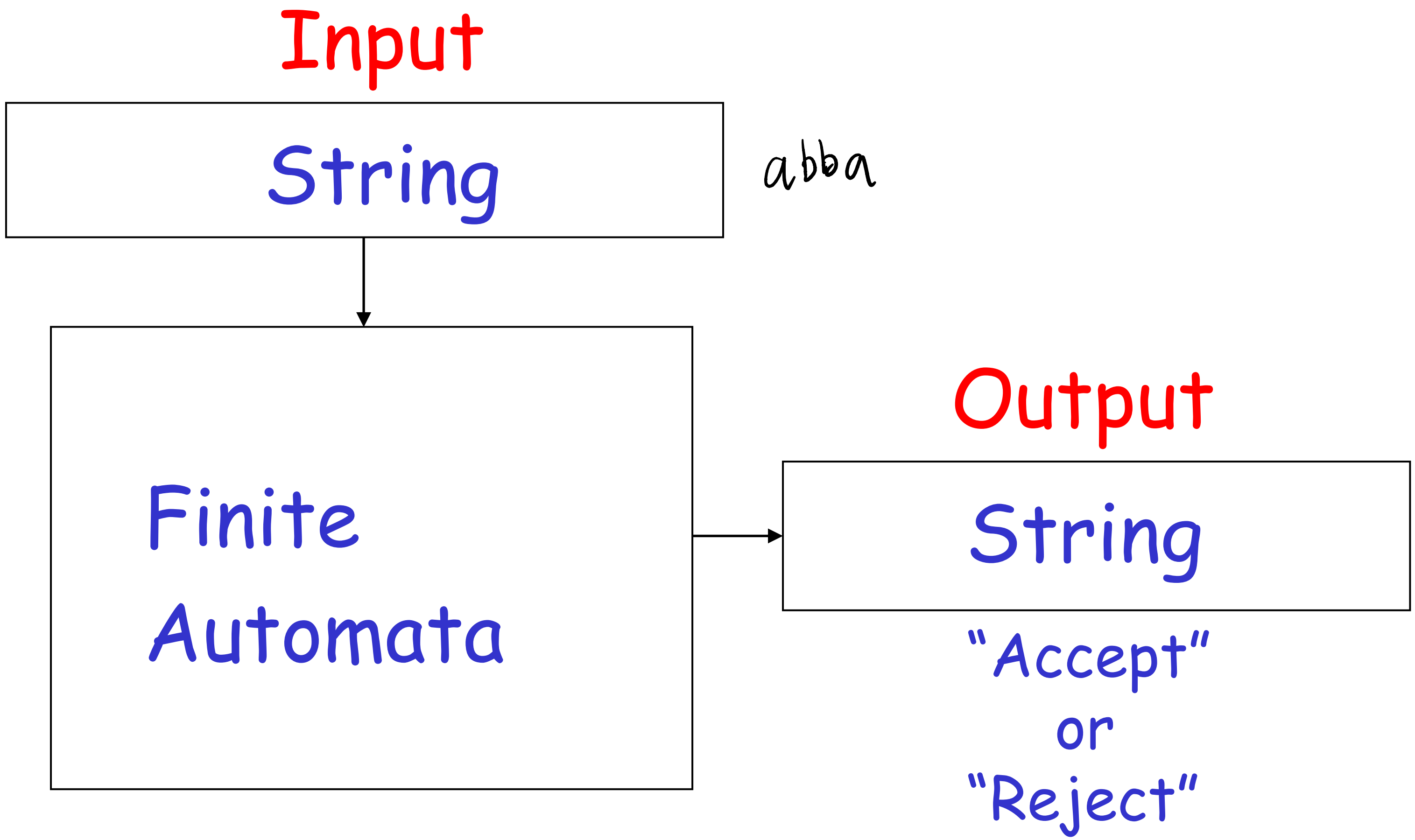


Example: Vending Machines

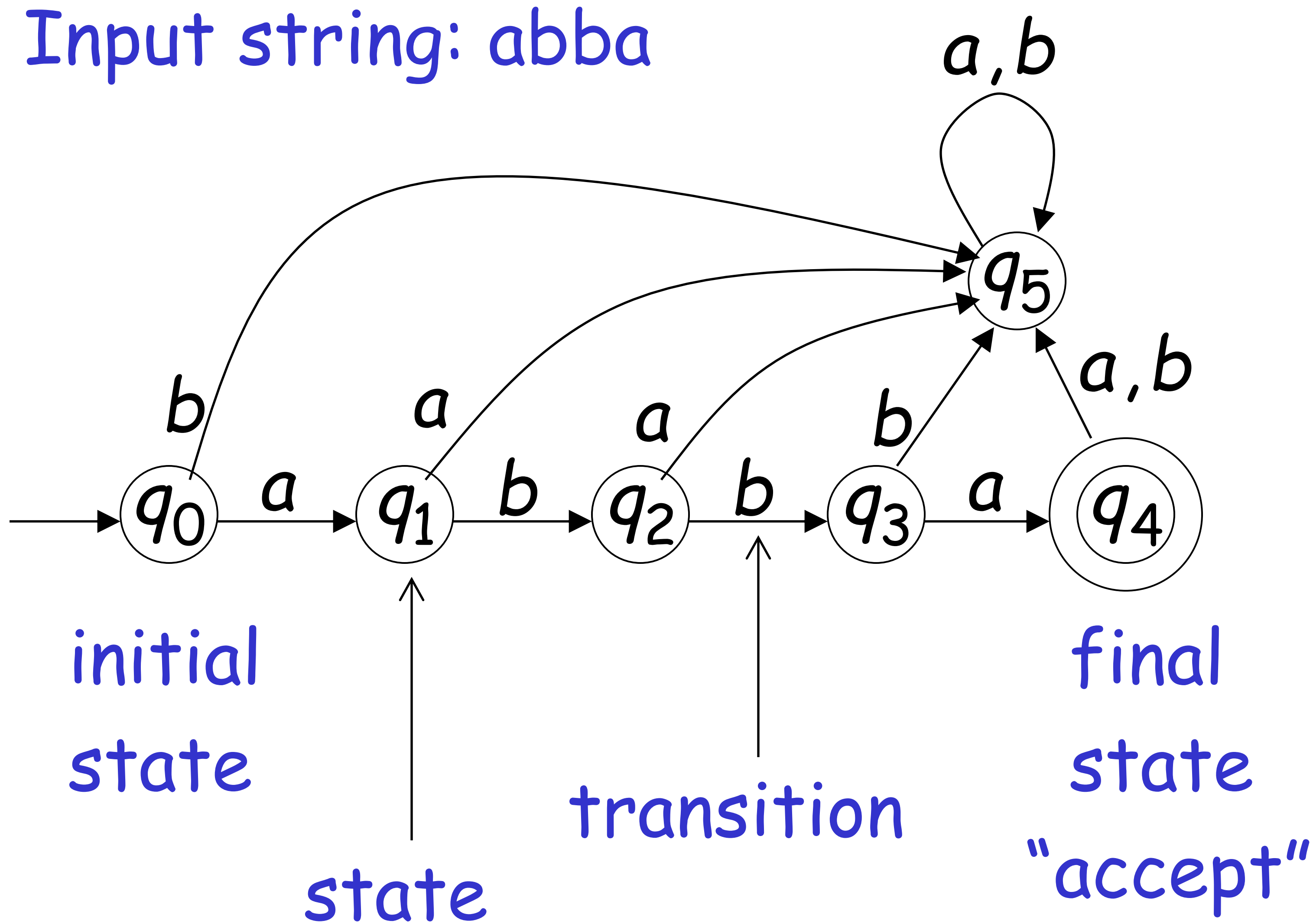
(small computing power)

Finite Automata

The simplest form of automata.



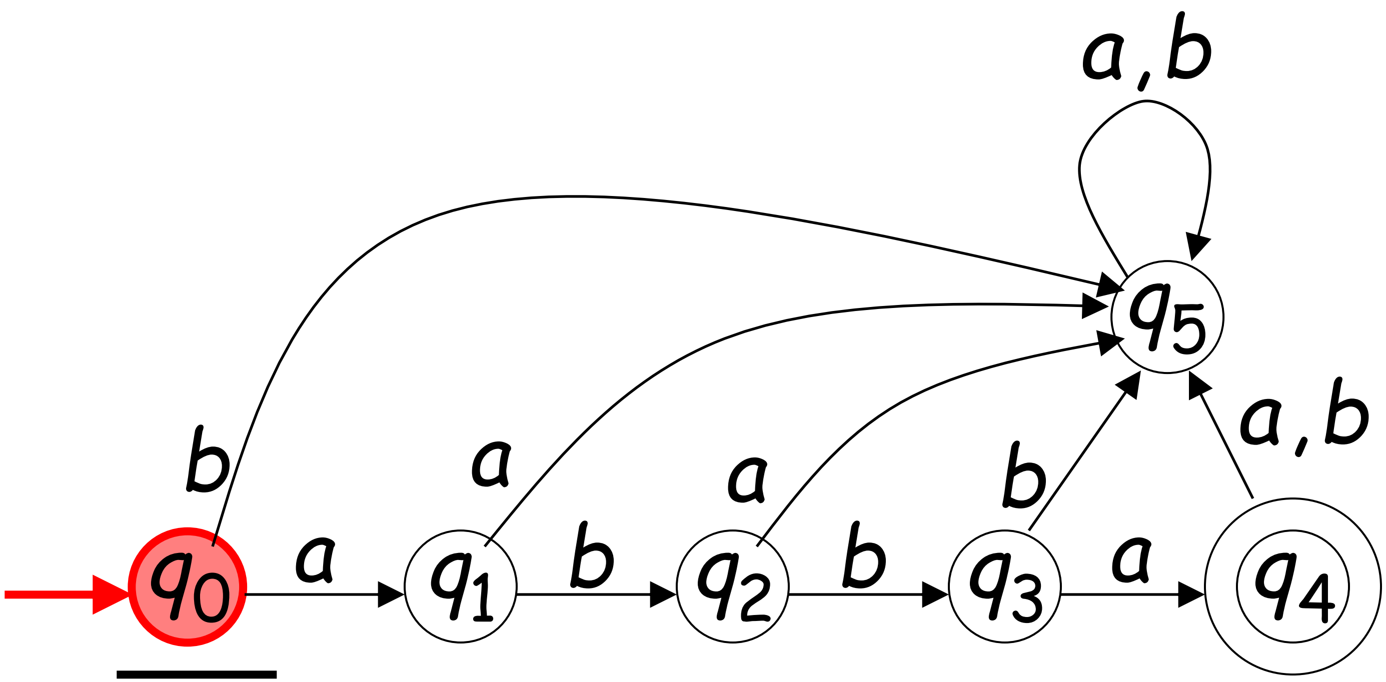
Transition Graph



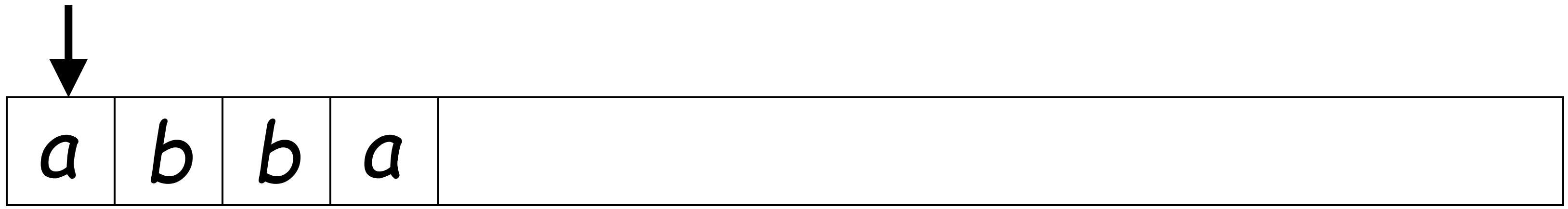
Initial Configuration

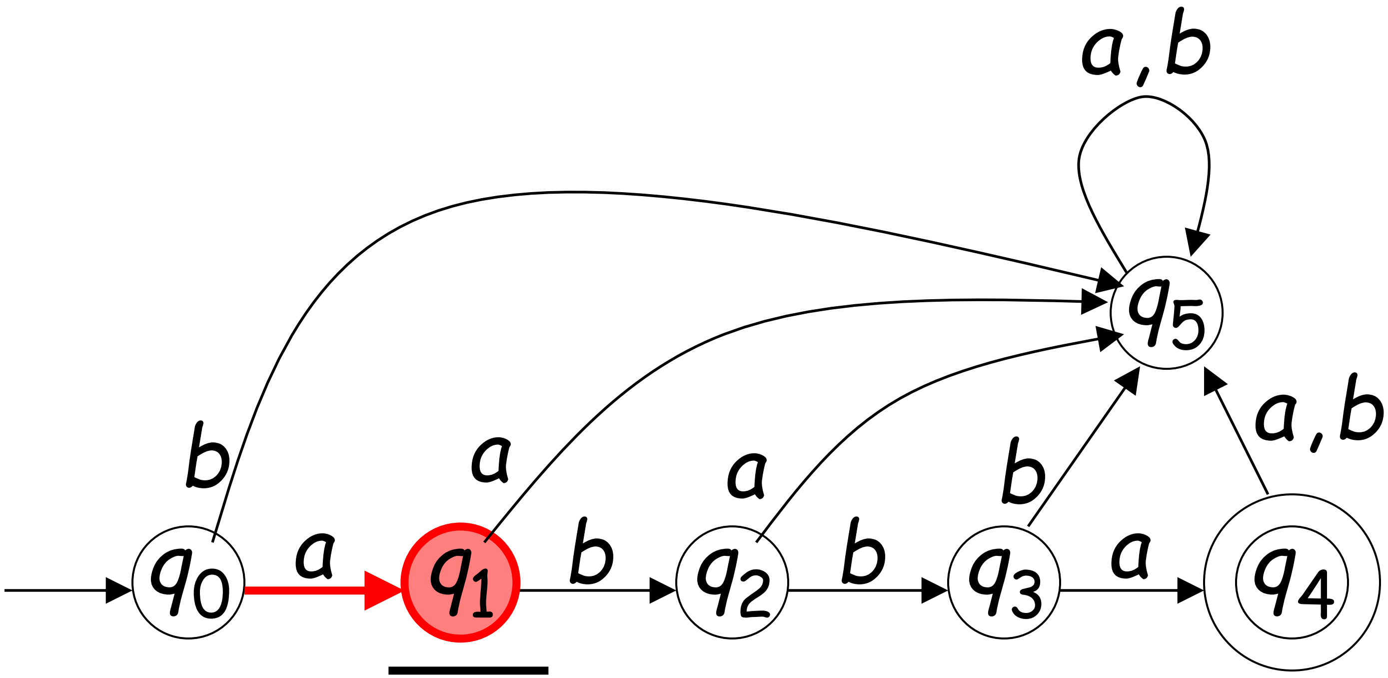
Input String

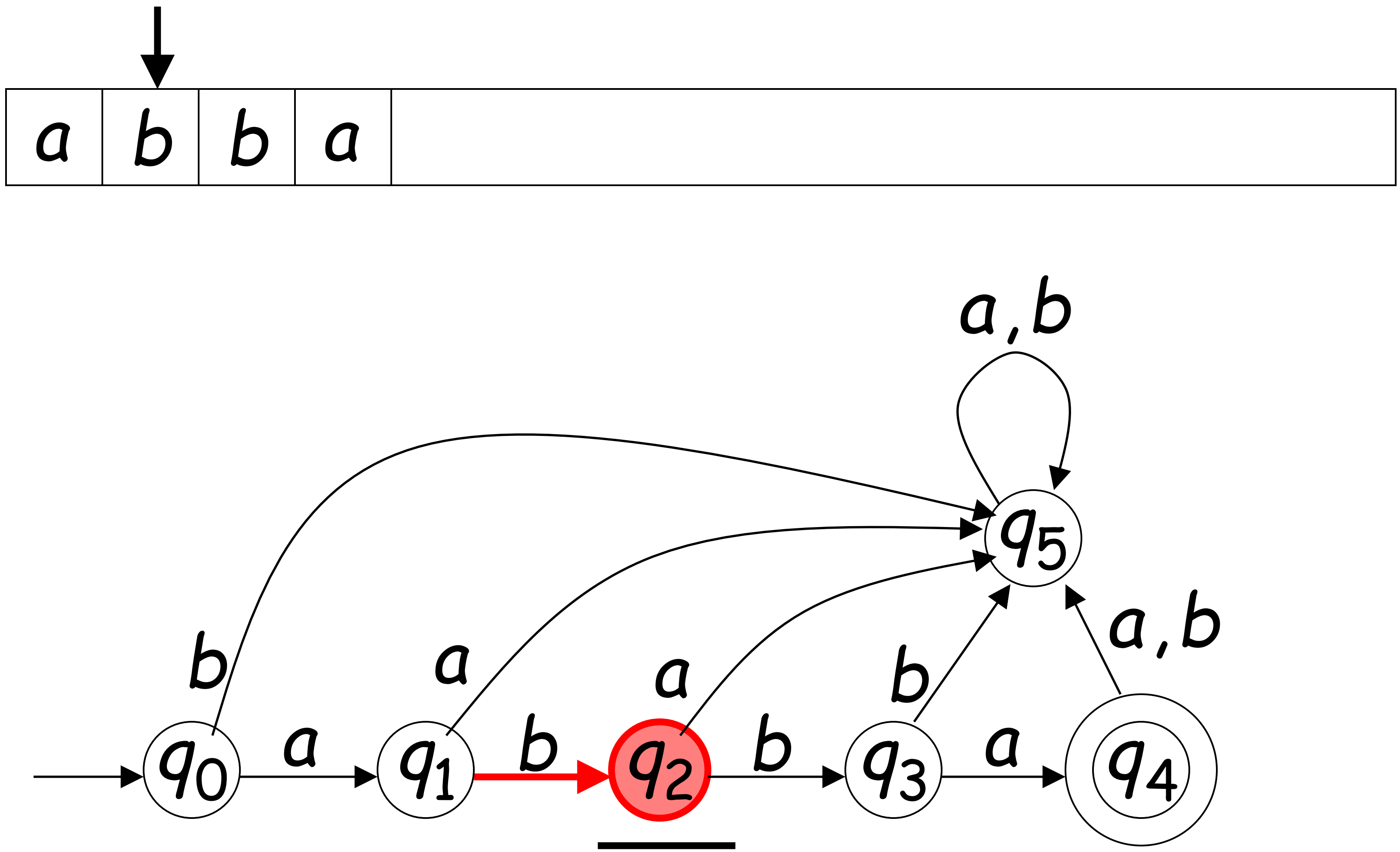
a	b	b	a	

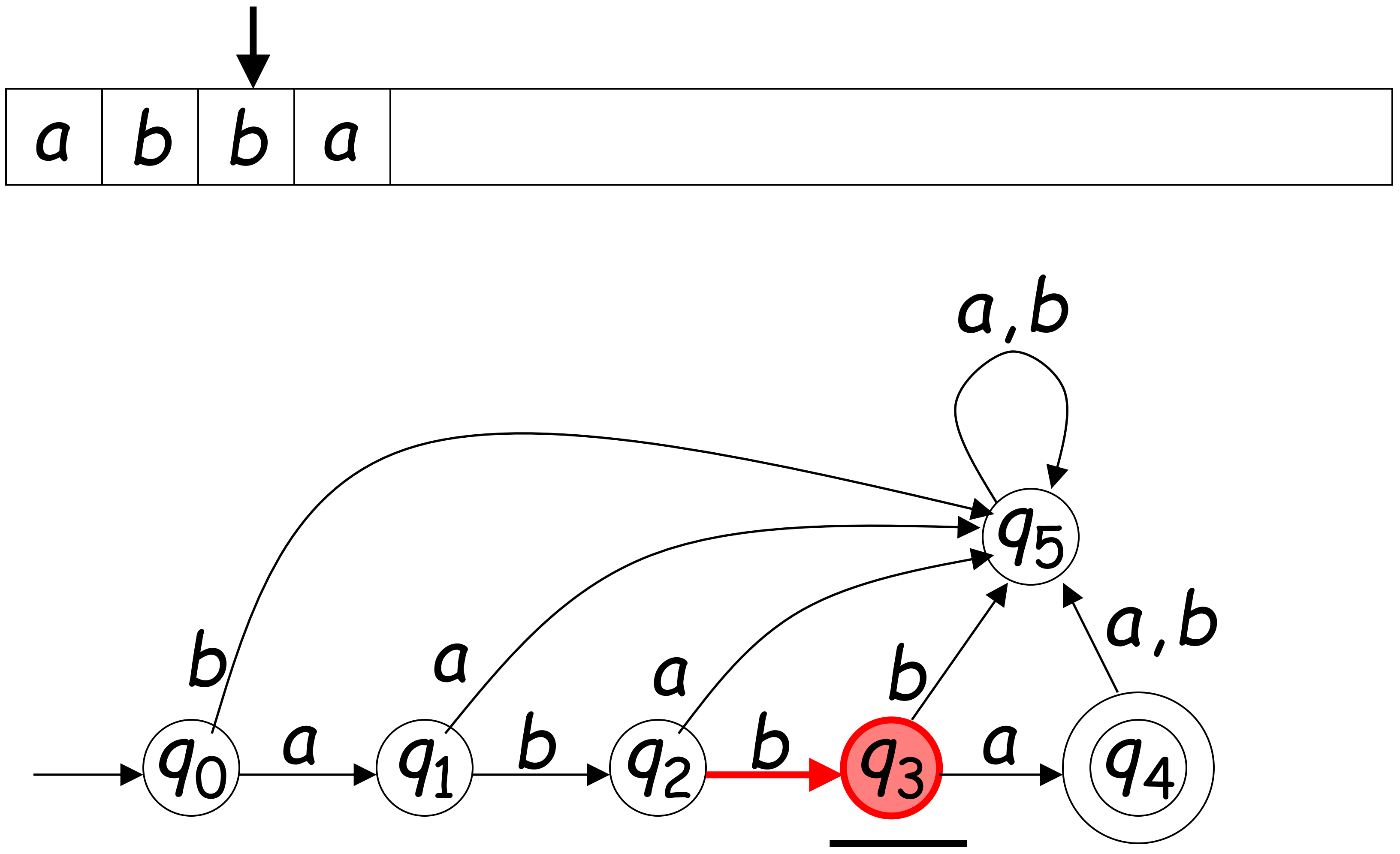


Reading the Input

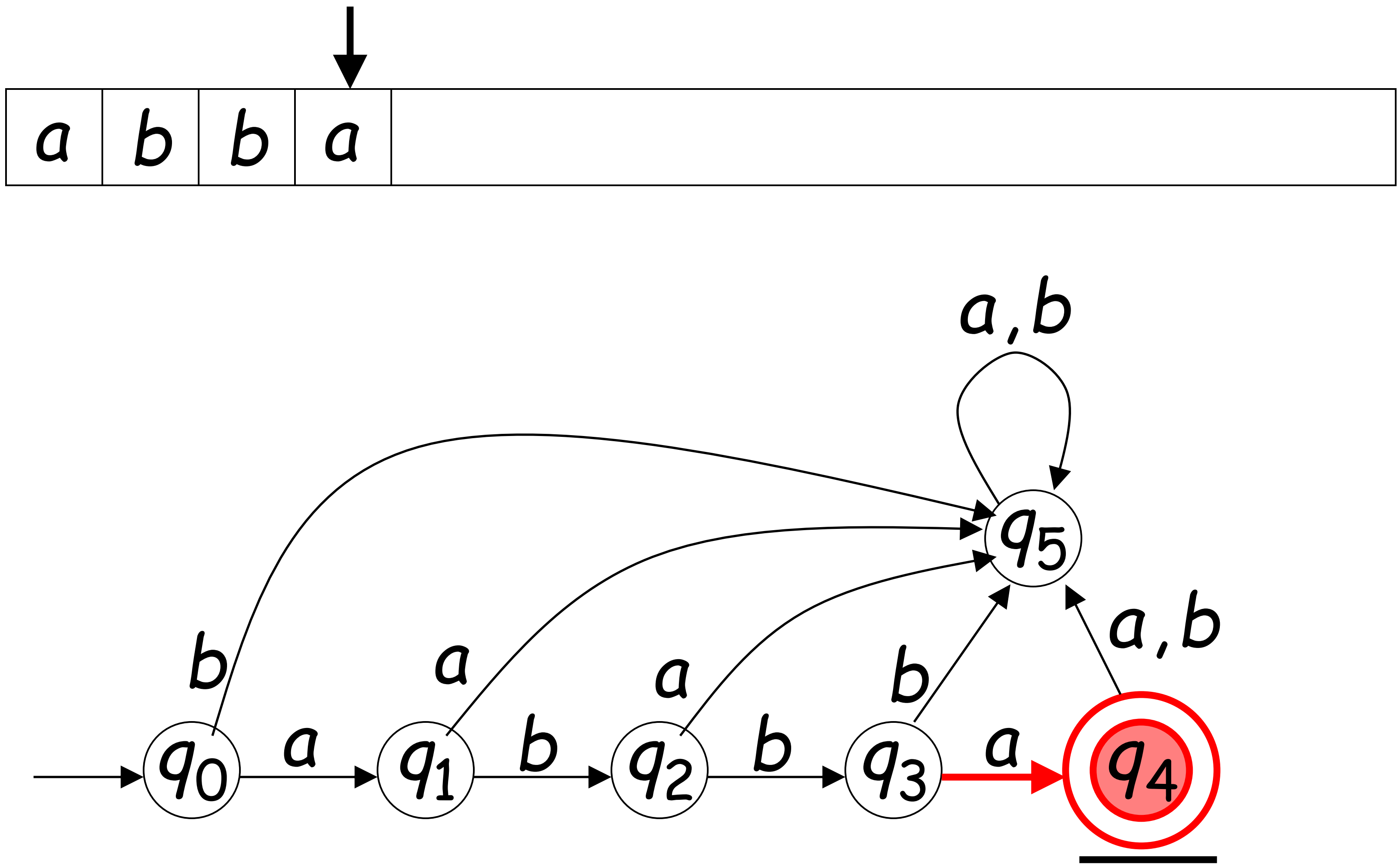




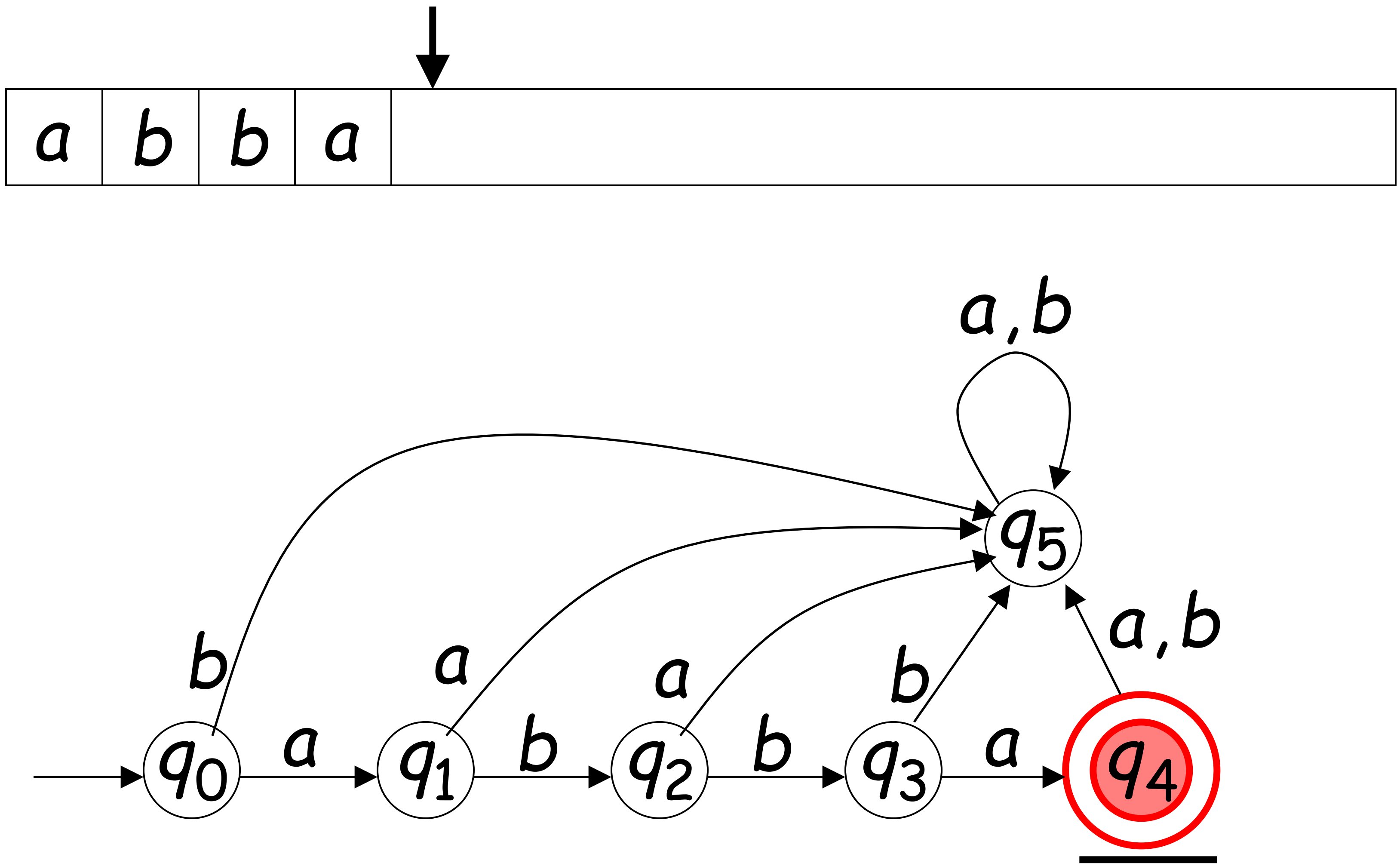

a
b
b
a
a, b
q_5
b
a
a
b
a, b
q_0
a
q_1
b
q_2
b
q_3
a
q_4



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

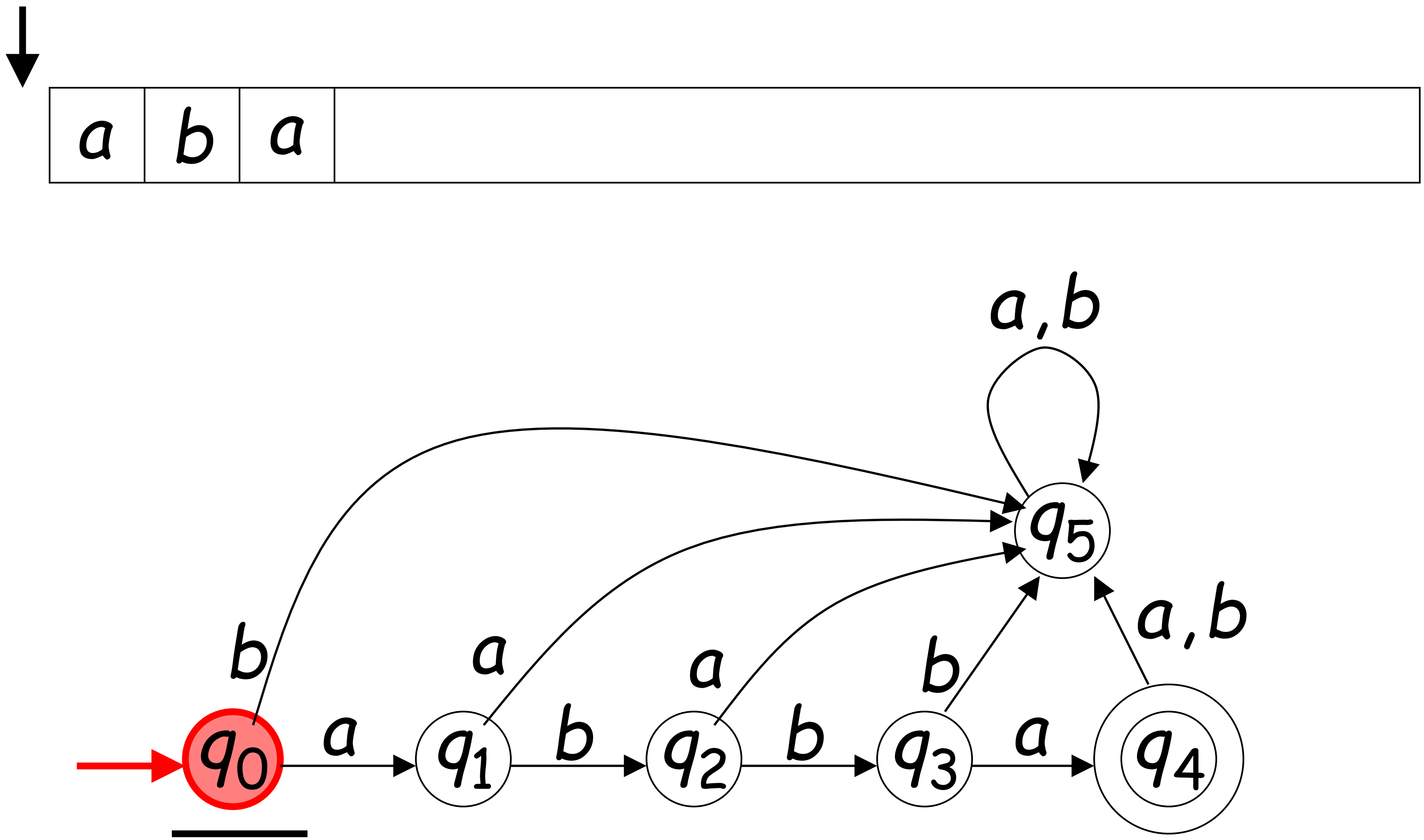

a b b a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4

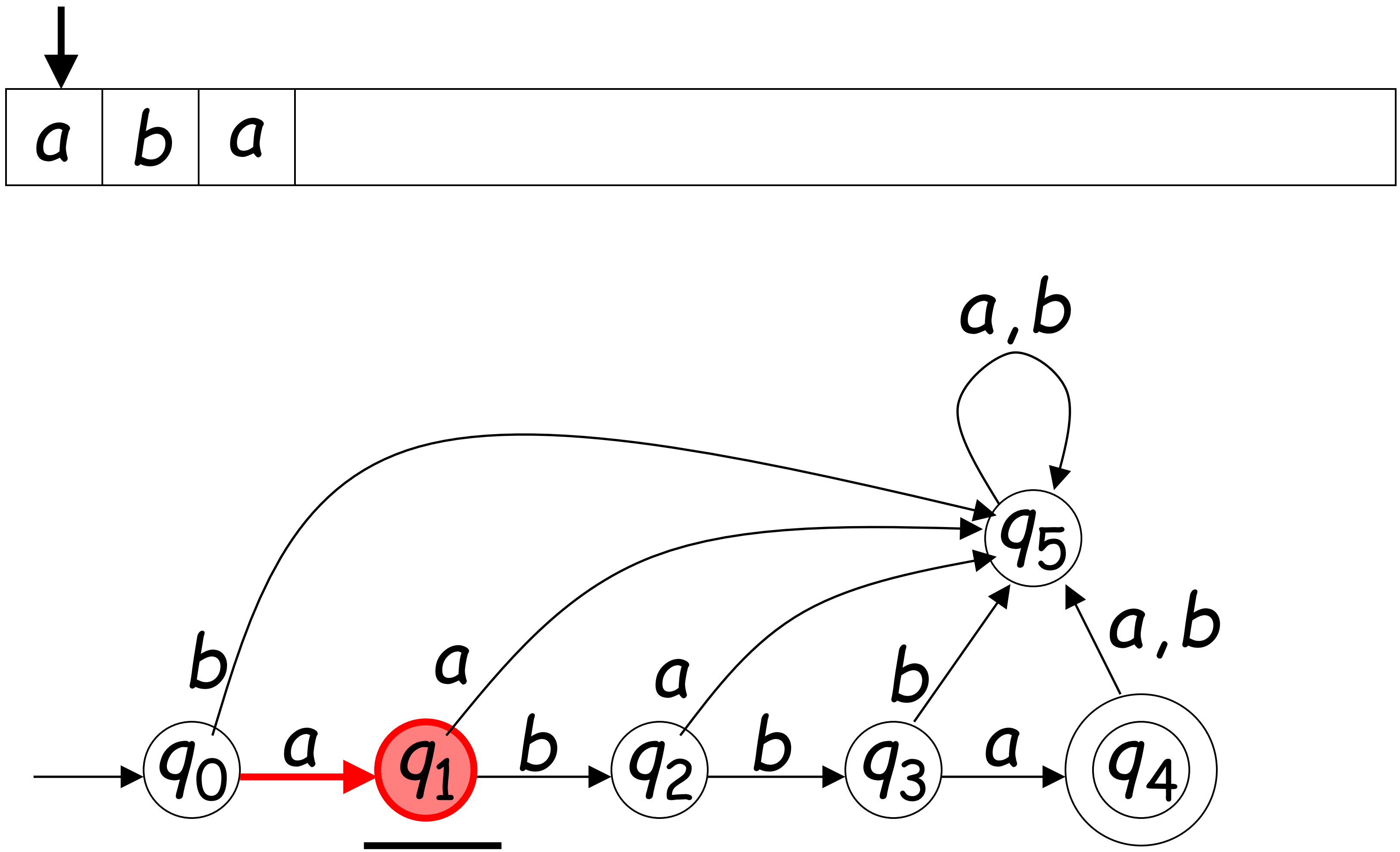
Input finished

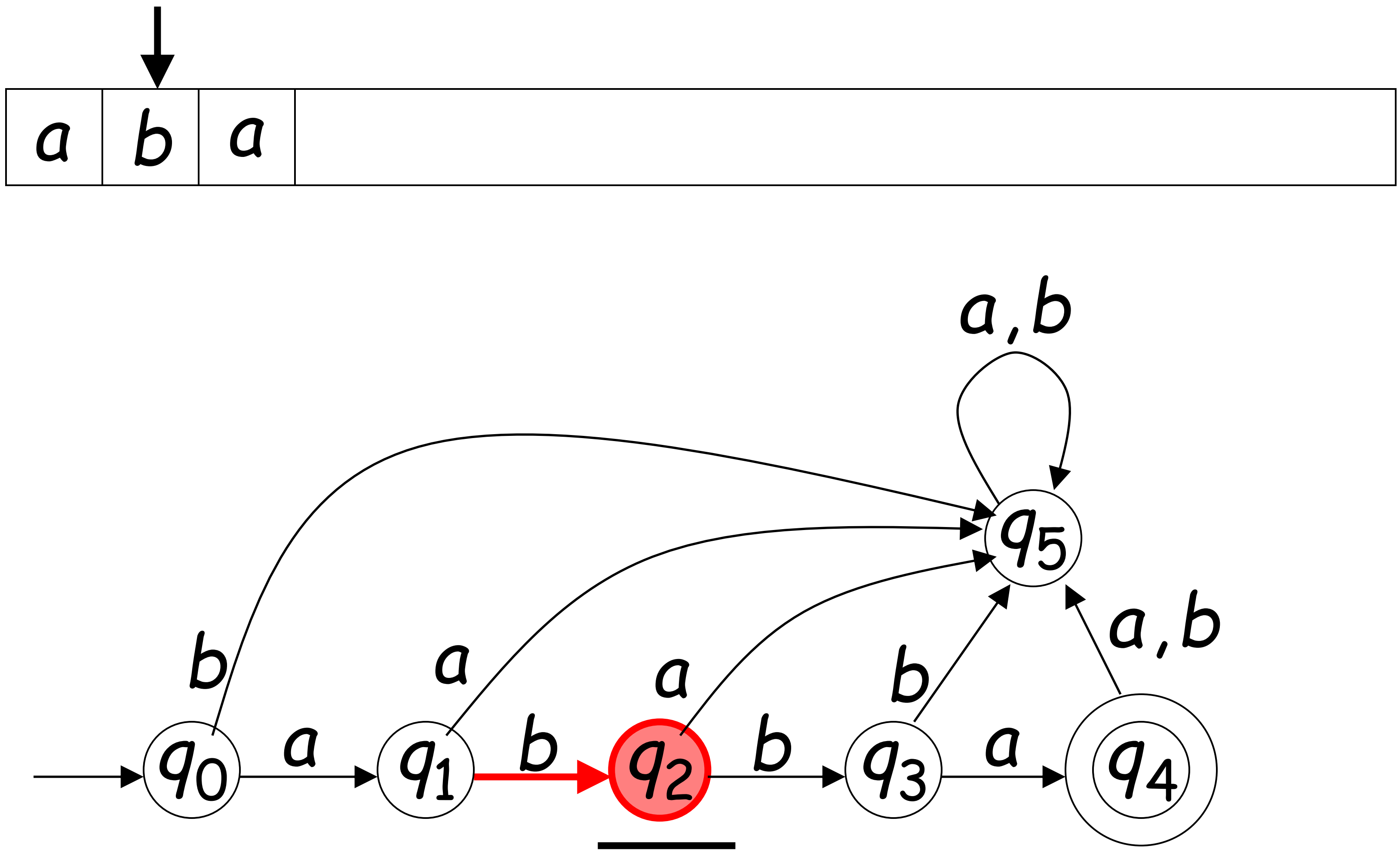


Output: "accept"

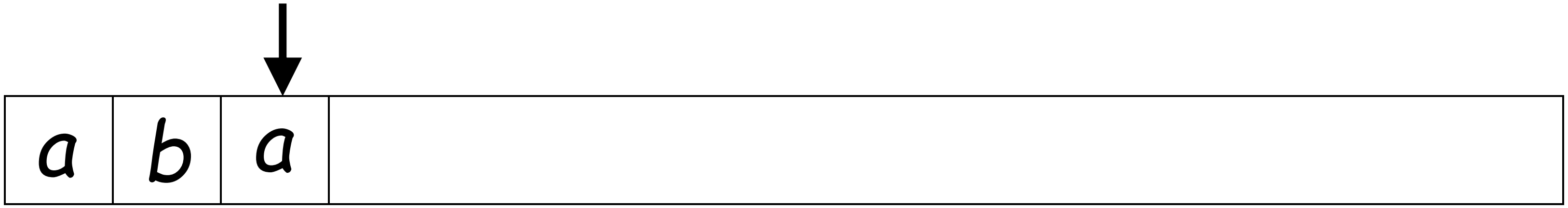
Rejection

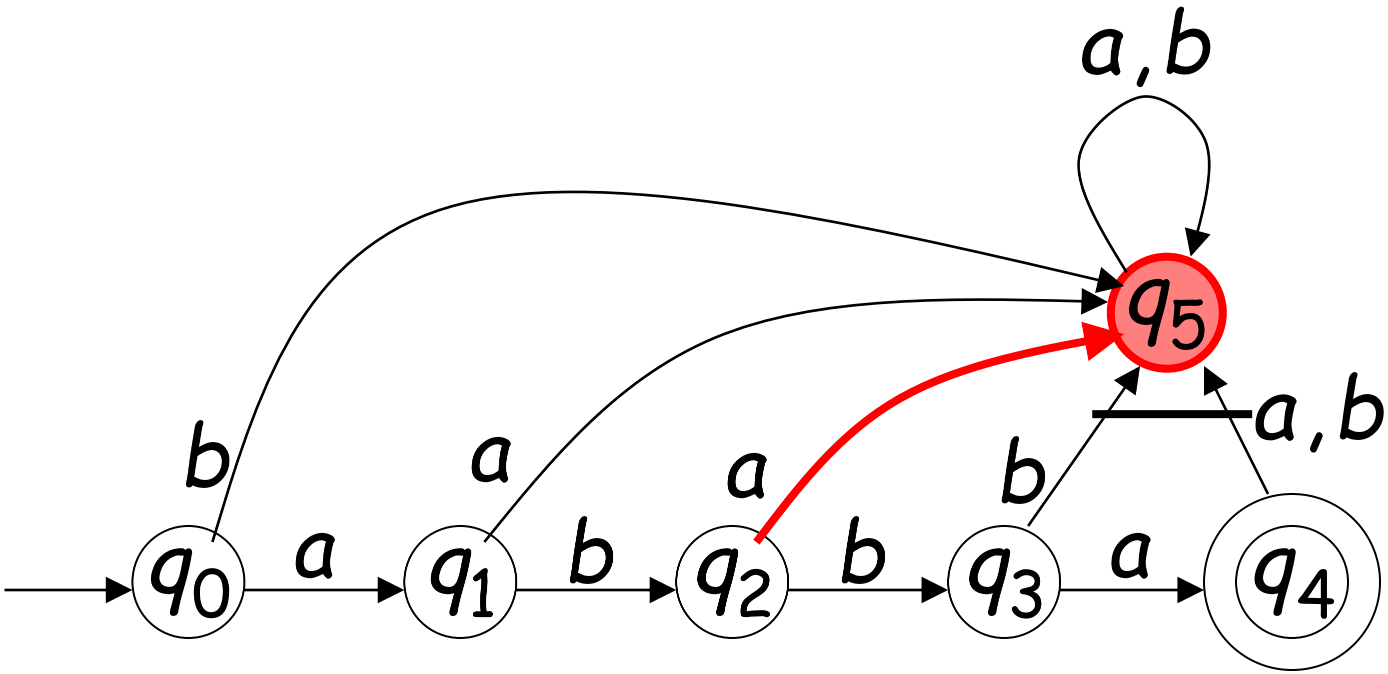



a
b
a
a,b
a
b
a
b
a
a,b
b
a
b
a
q_0
q_1
q_2
q_3
q_4
q_5

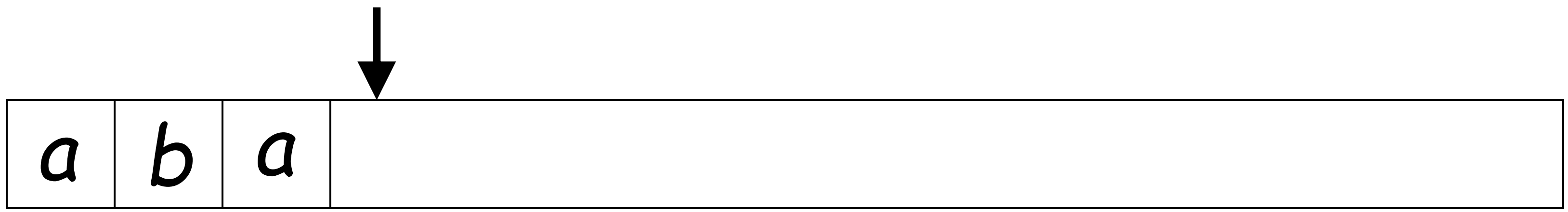


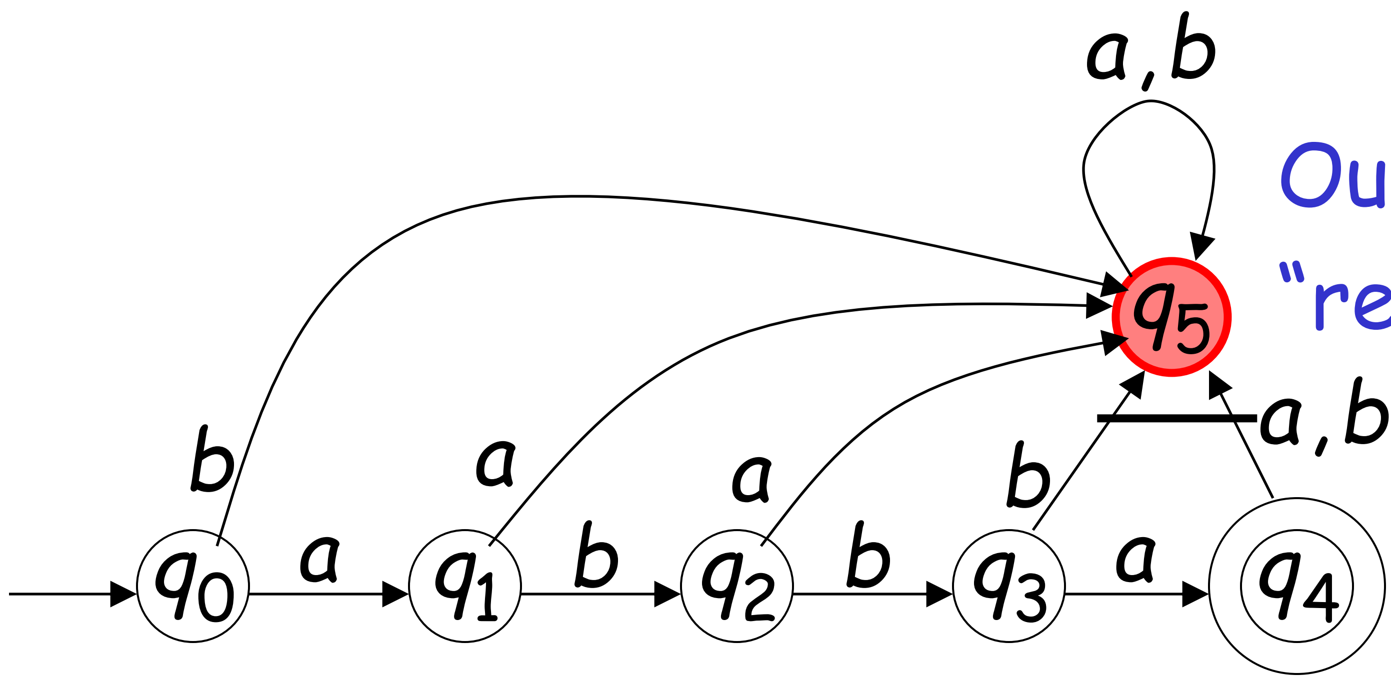
a
b
a
a,b
q5
a,b
b
a
a
b
q0
a
q1
b
q2
b
q3
a
q4


a
b
a


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

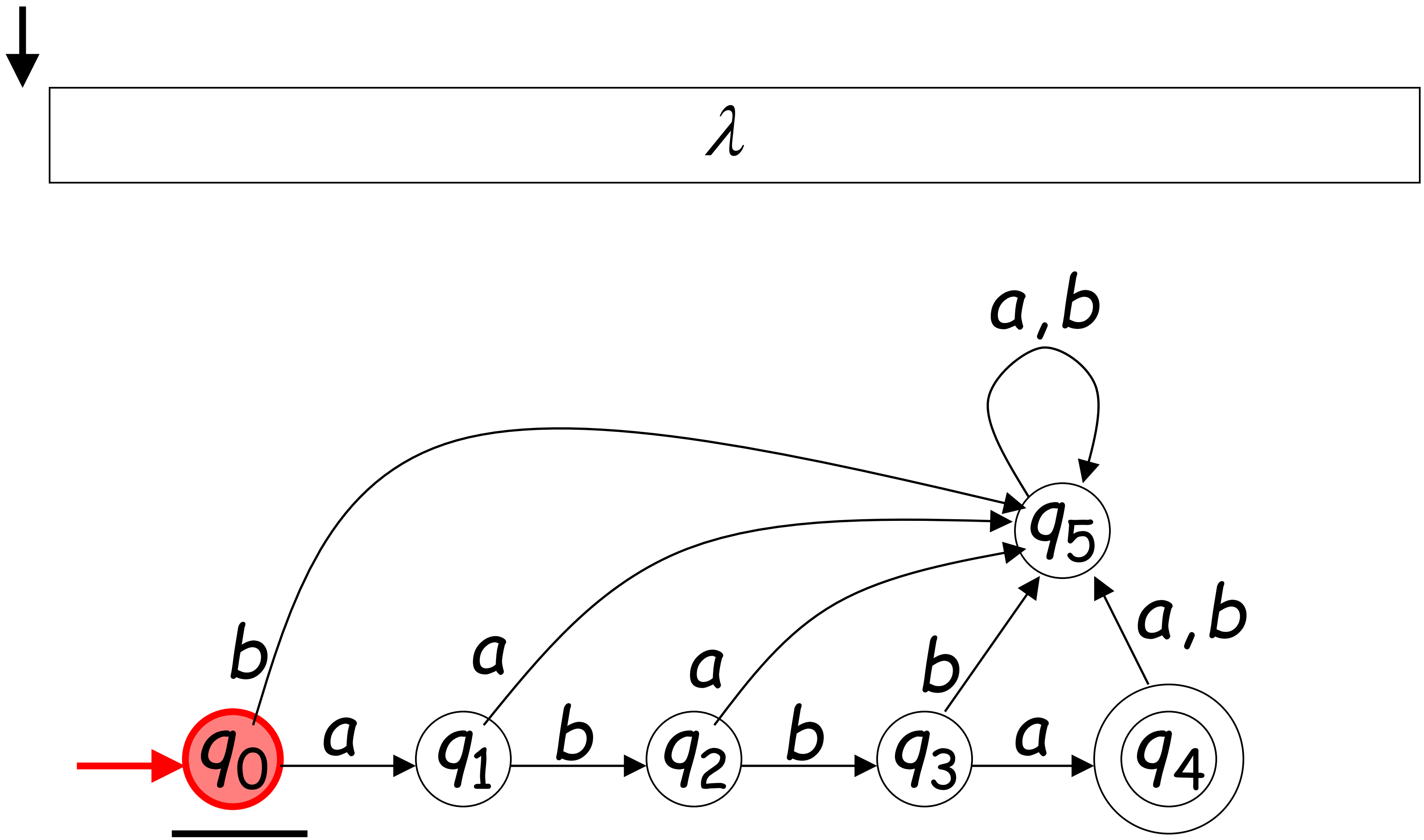
Input finished

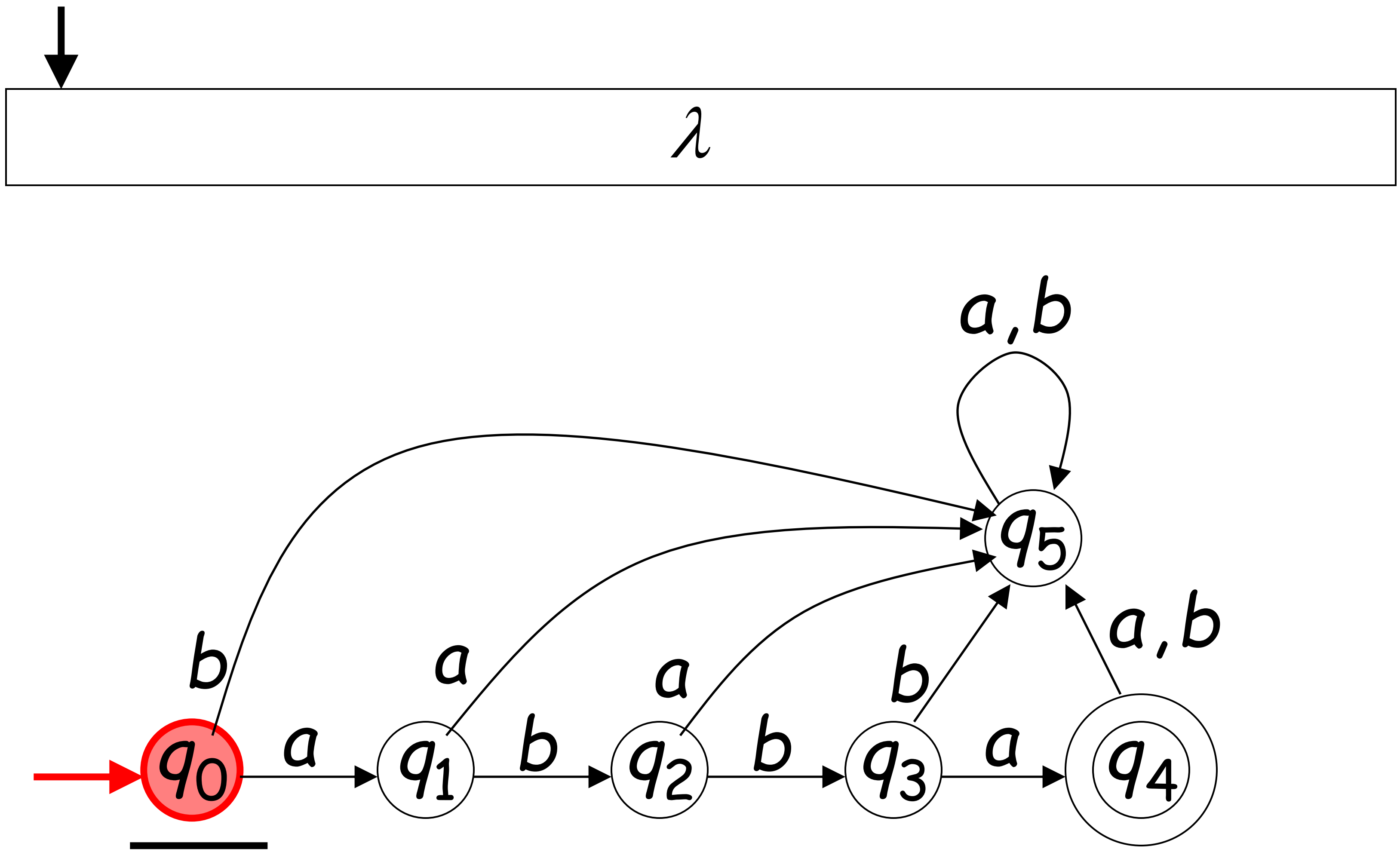




Output:
"reject"

Another Rejection

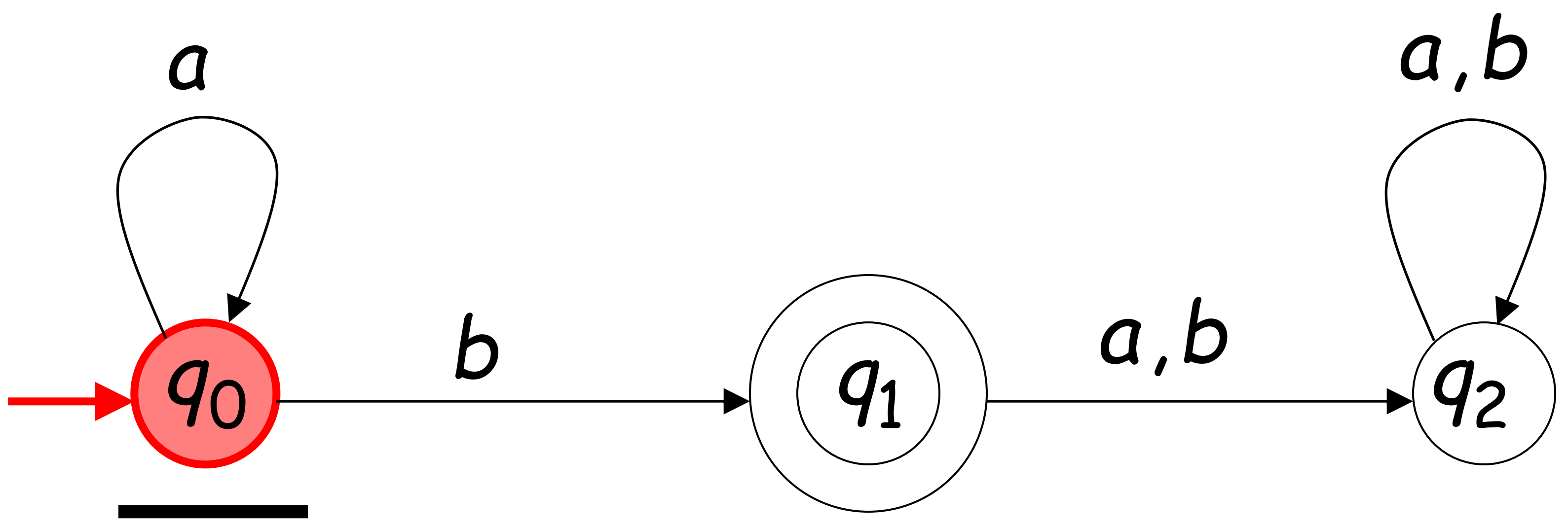


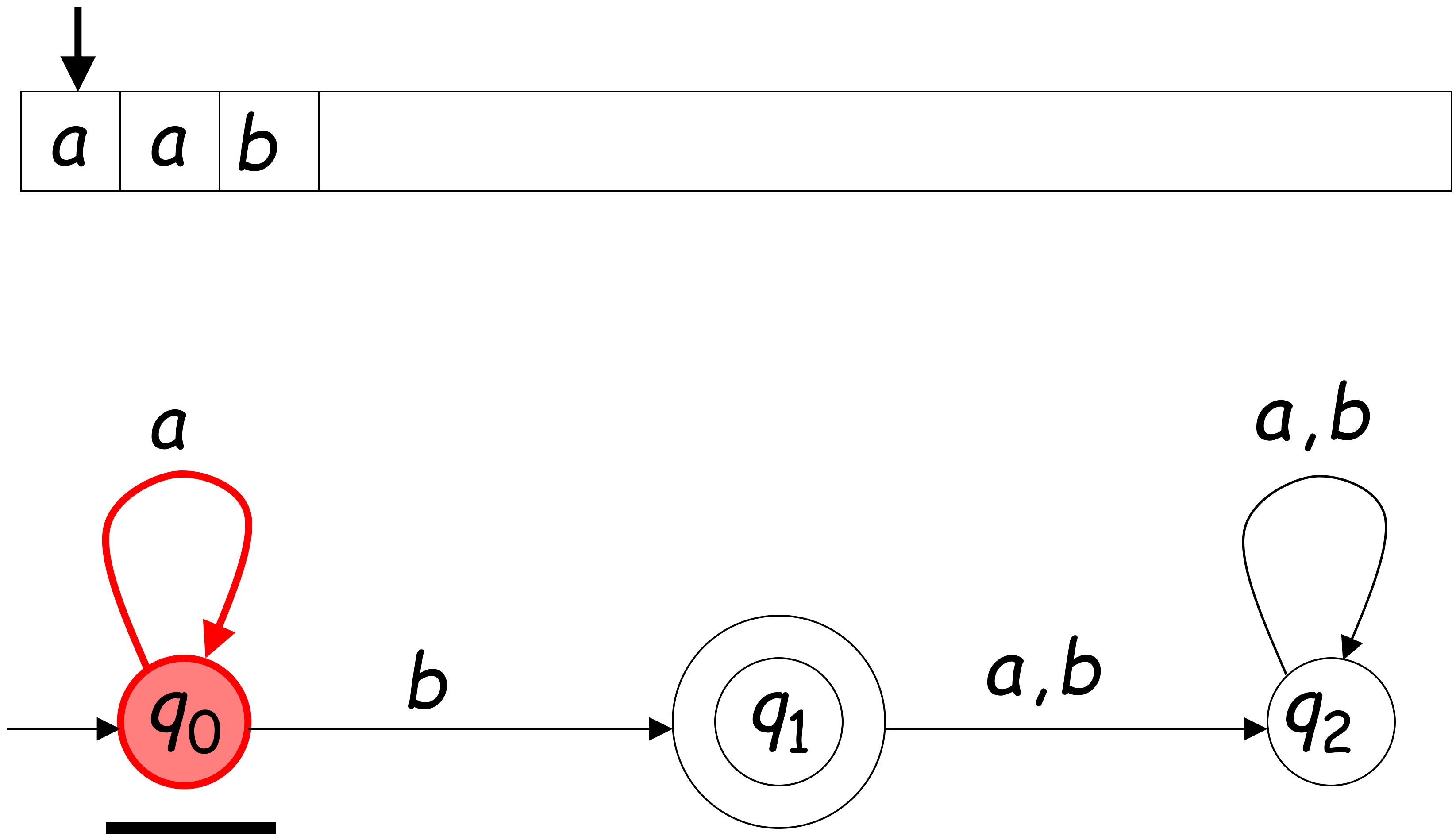


Output:
"reject"

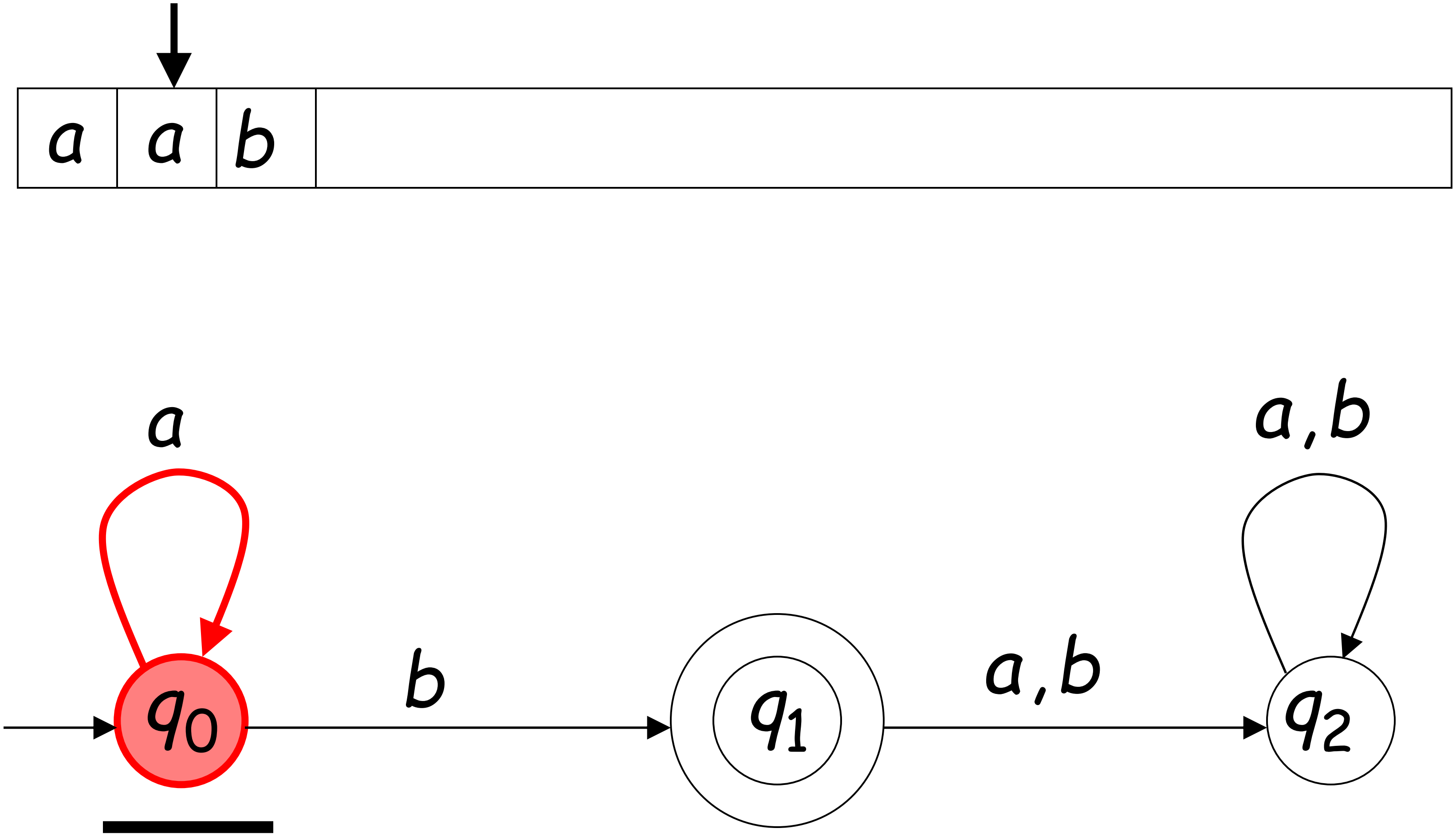
Another Example

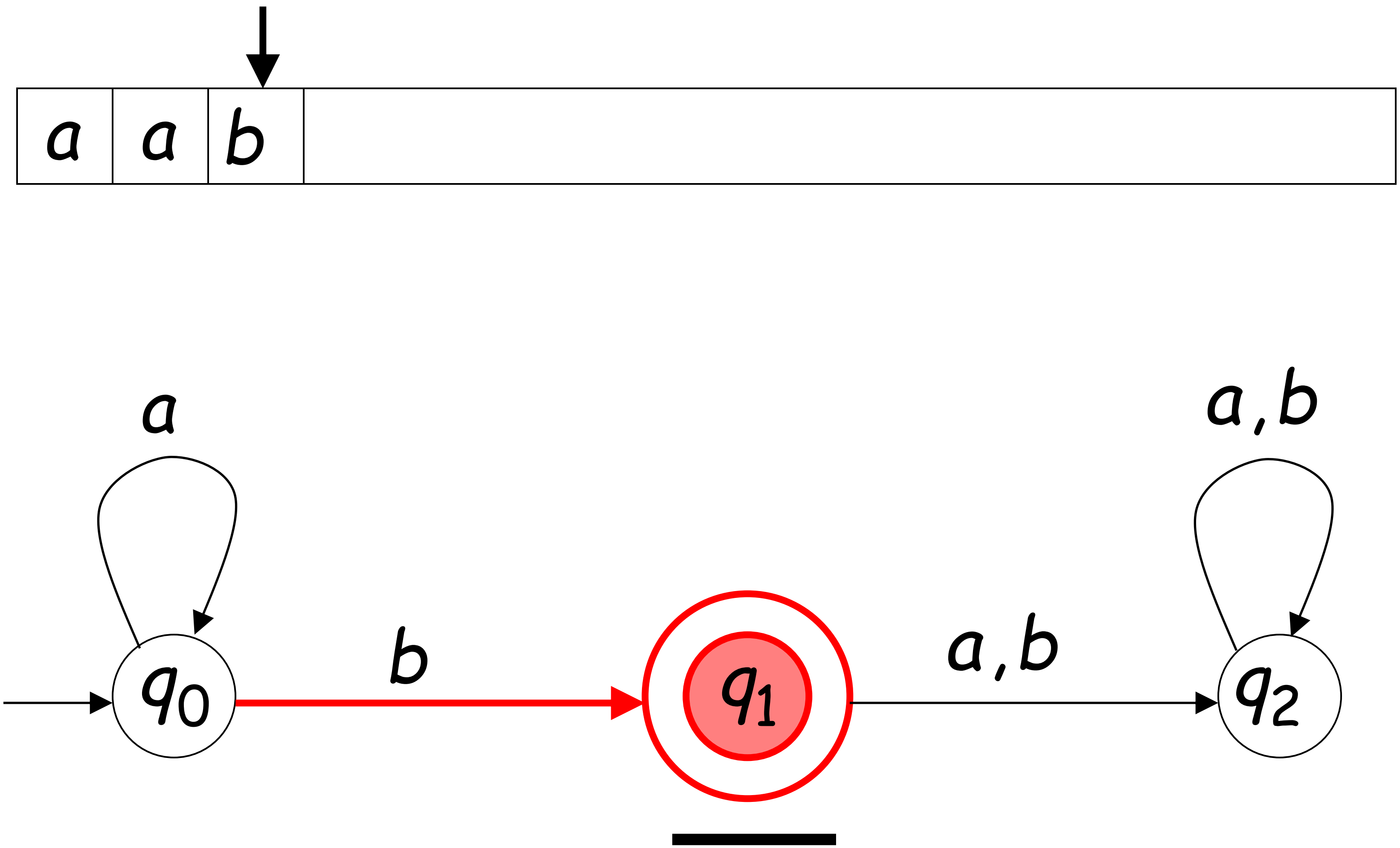
a	a	b



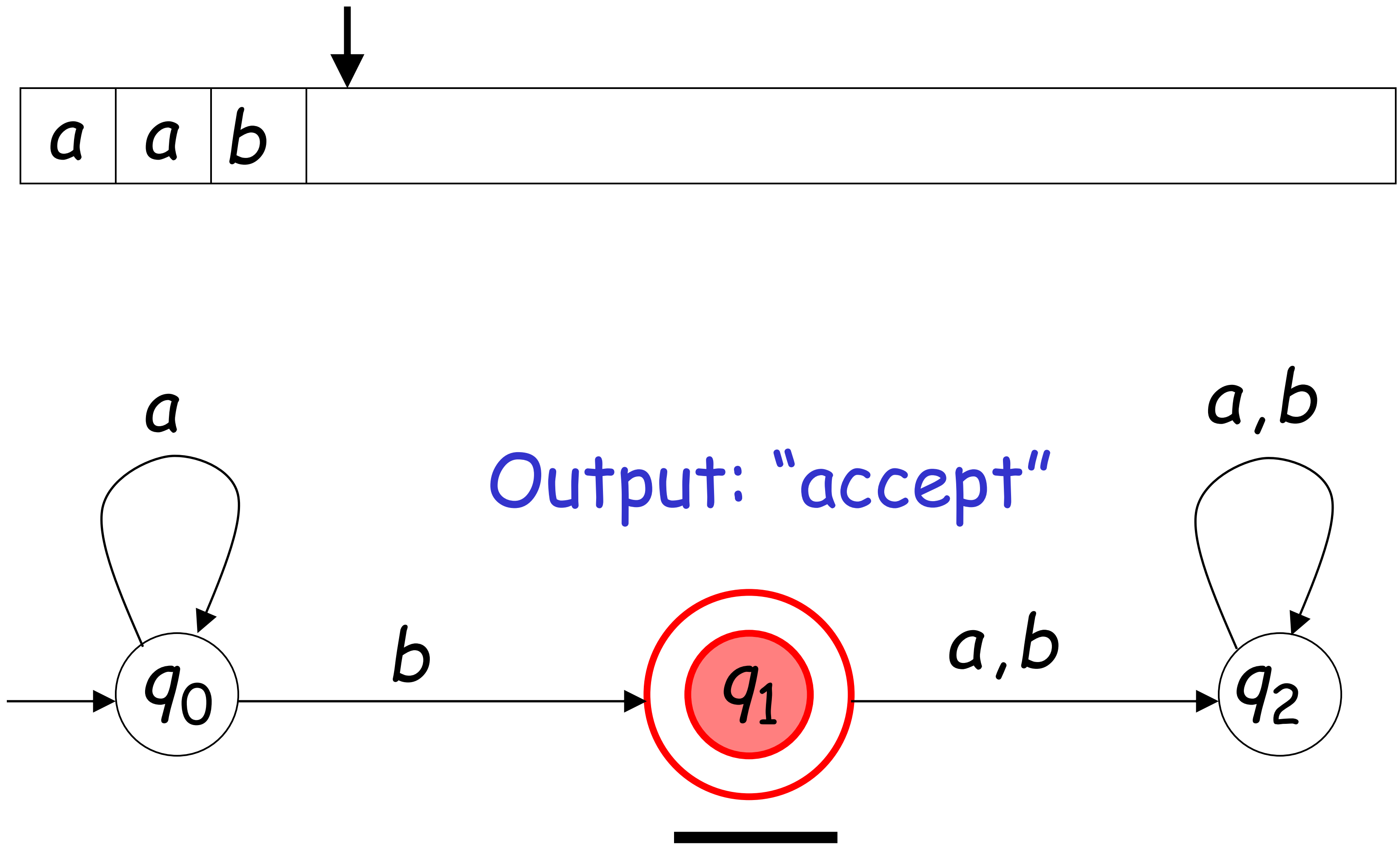


a
a
b
a
b
a,b
a,b
q_0
q_1
q_2

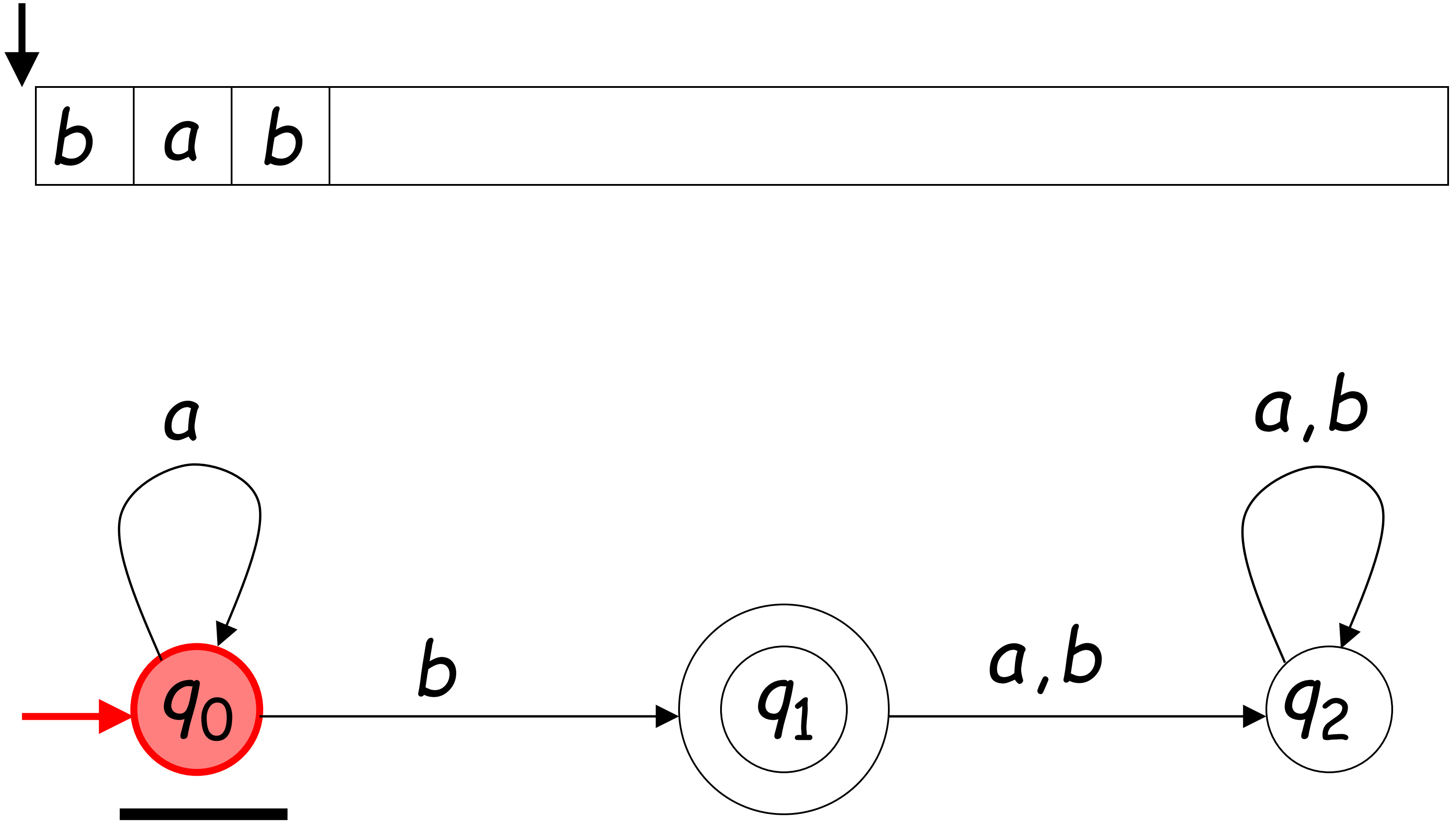

a
a
b
a
b
a,b
a,b
q_0
q_1
q_2

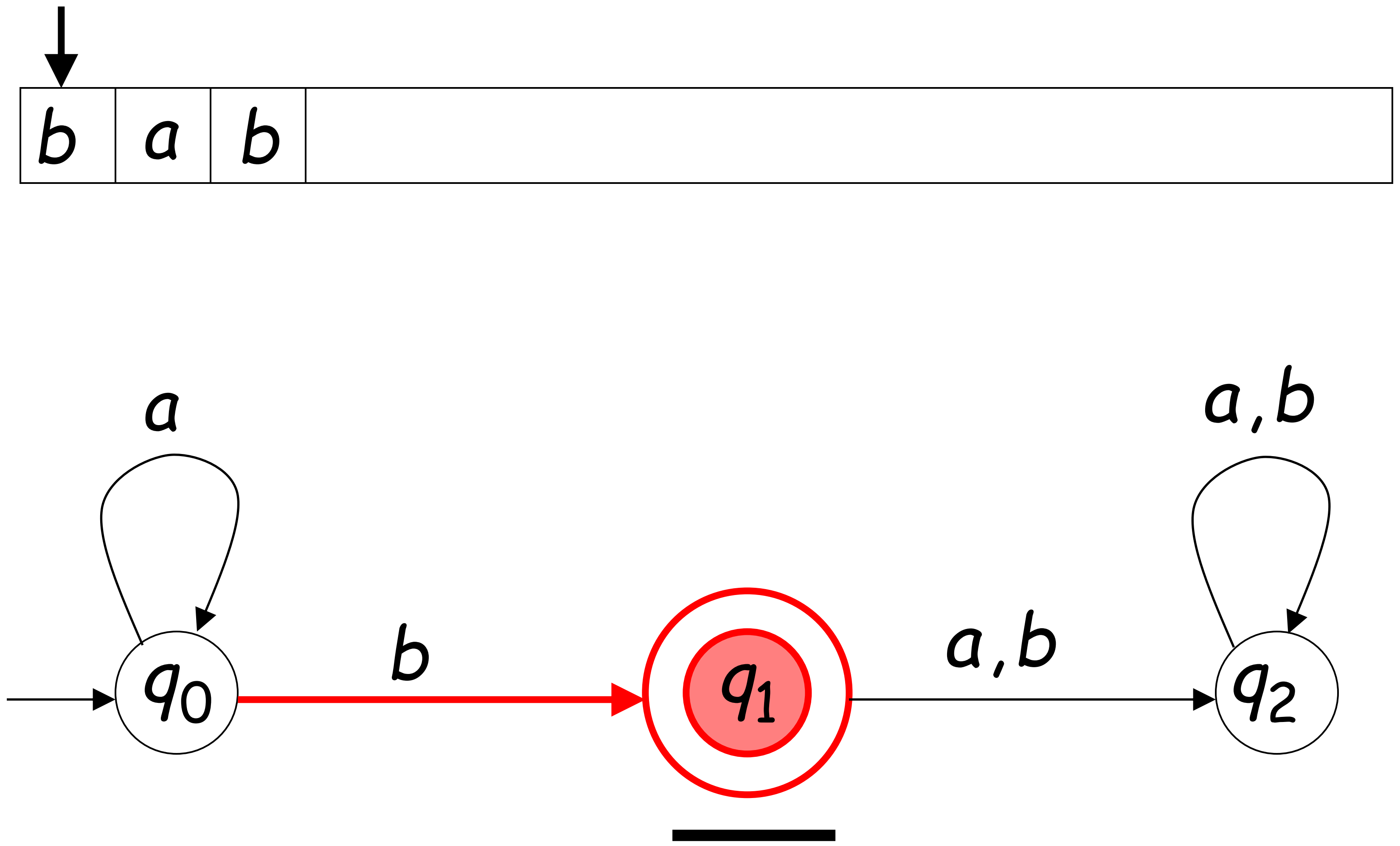

a a b
a
a,b
b
a,b
q_0
q_1
q_2

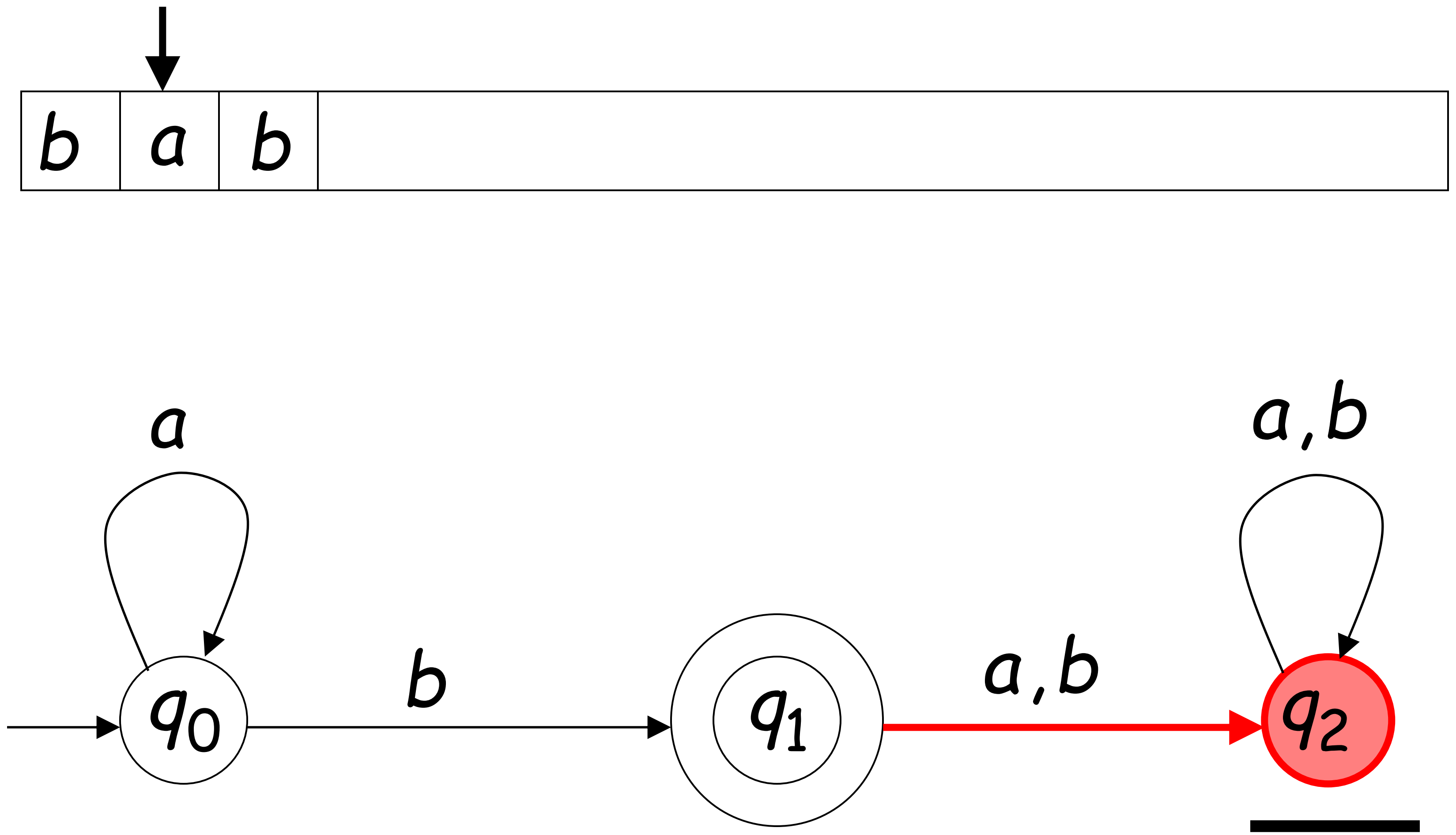
Input finished

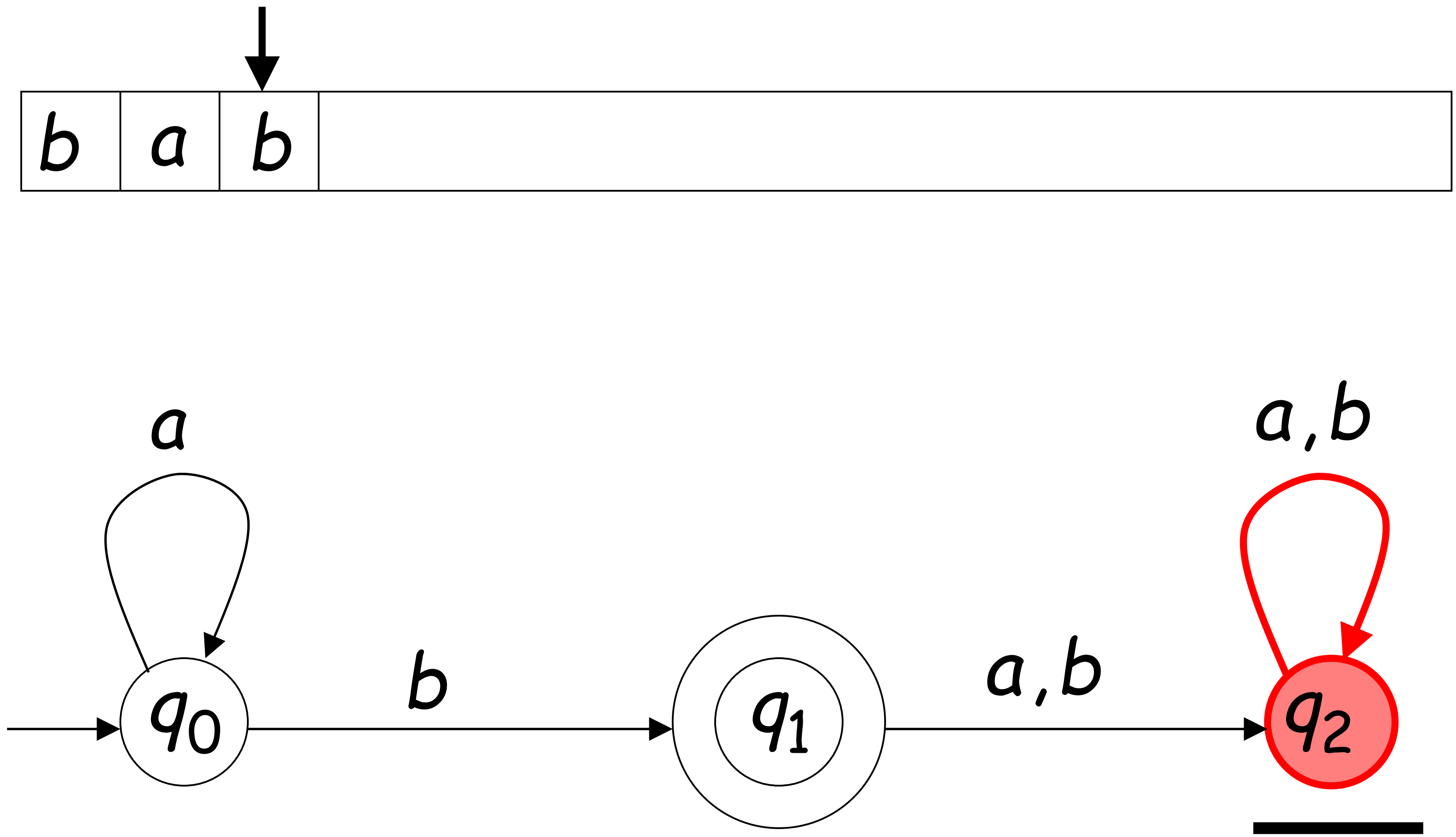


Rejection

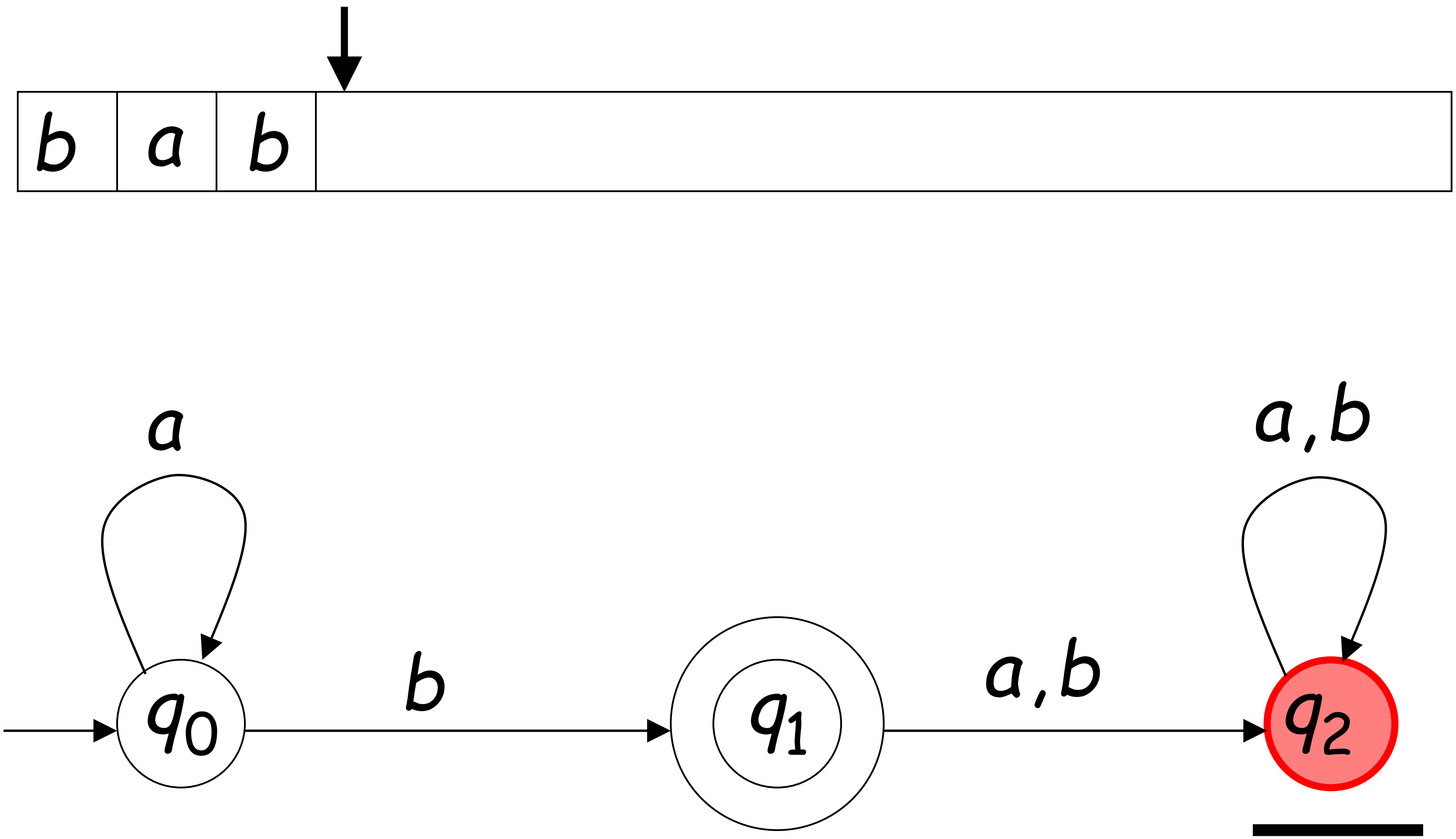



b
a
b
a
a,b
b
a,b
q_0
q_1
q_2


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


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

Input finished



Output: "reject"

Formalities

คาดเดาได้ M

Deterministic Finite Accepter (DFA)

machine — set state — alphabet — transition func — state เริ่ม — set final state

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

FA
DFA (Determinist...)
NFA (non . Deter, ...)

Q : set of states $Q = \{ q_0 \cdots \}$

Σ : input alphabet

Deter $q \rightarrow q_i$

δ : transition function

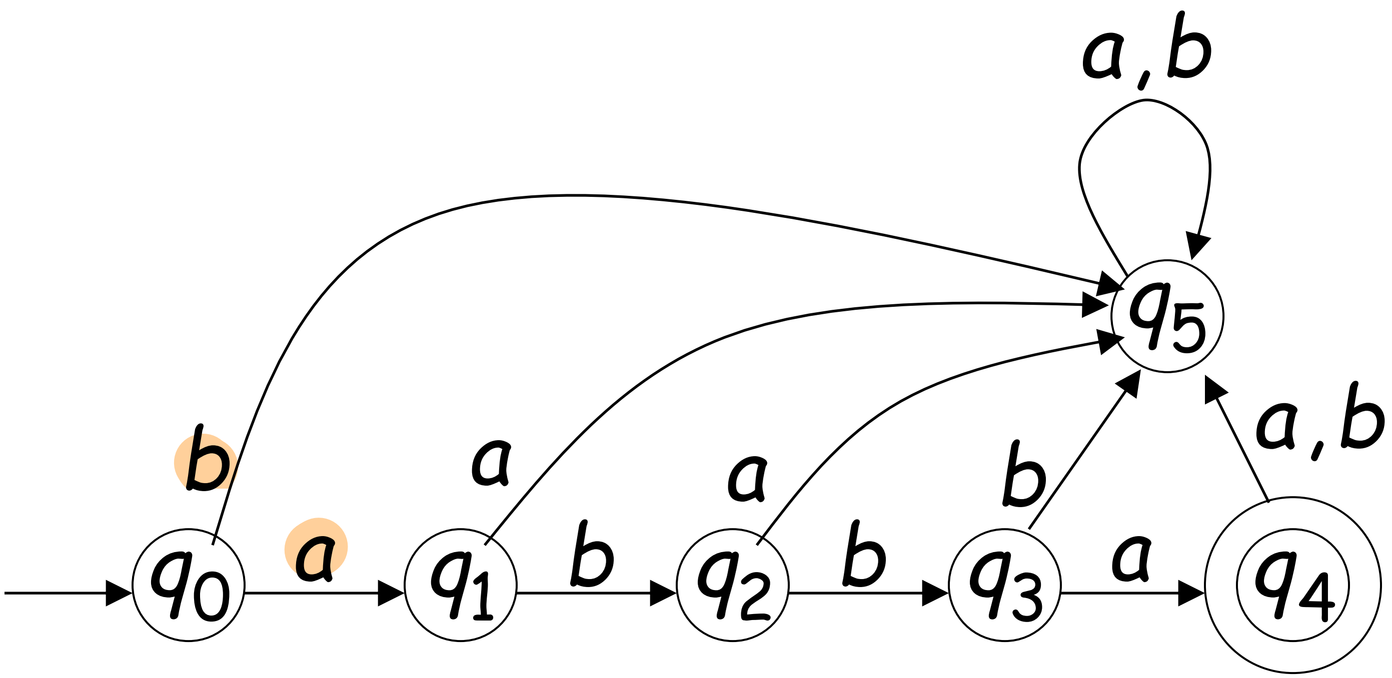
non Deter $q \xrightarrow{a} q_i$, $q \rightarrow q_j$

q_0 : initial state

F : set of final states มีได้มากกว่า 1 state

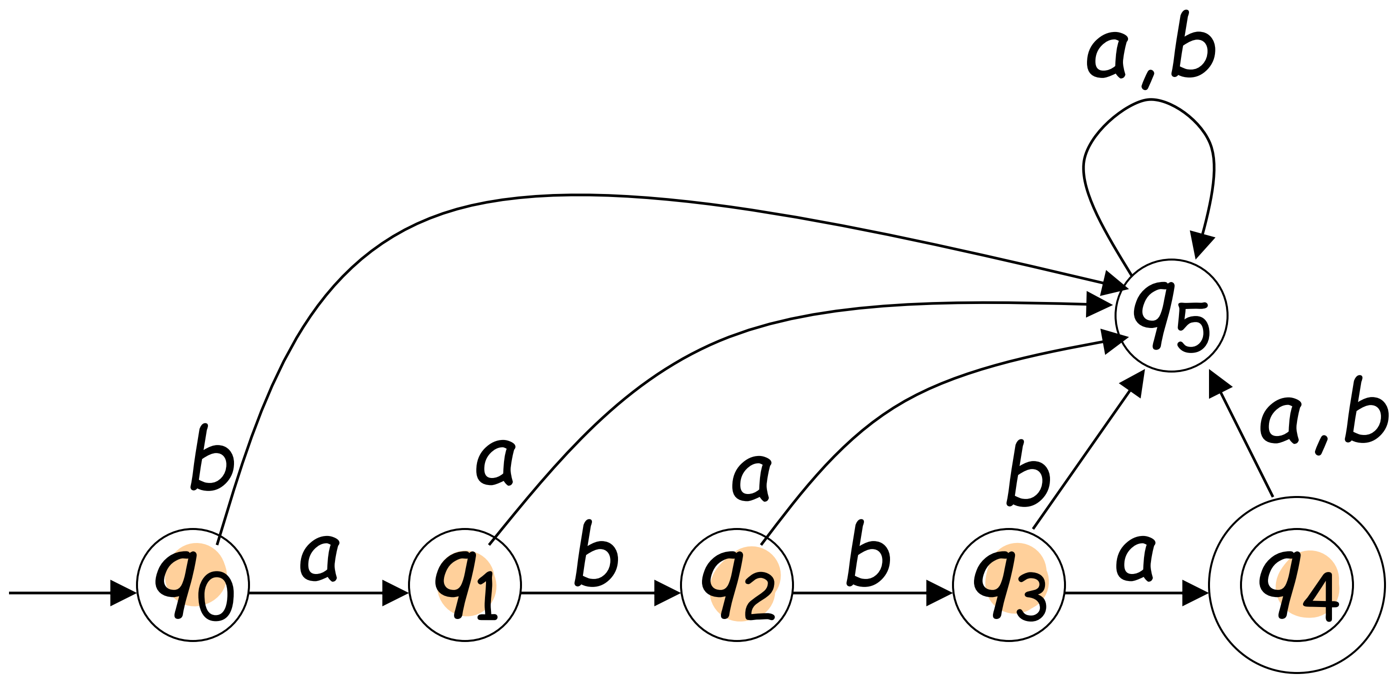
Input Alphabet Σ

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

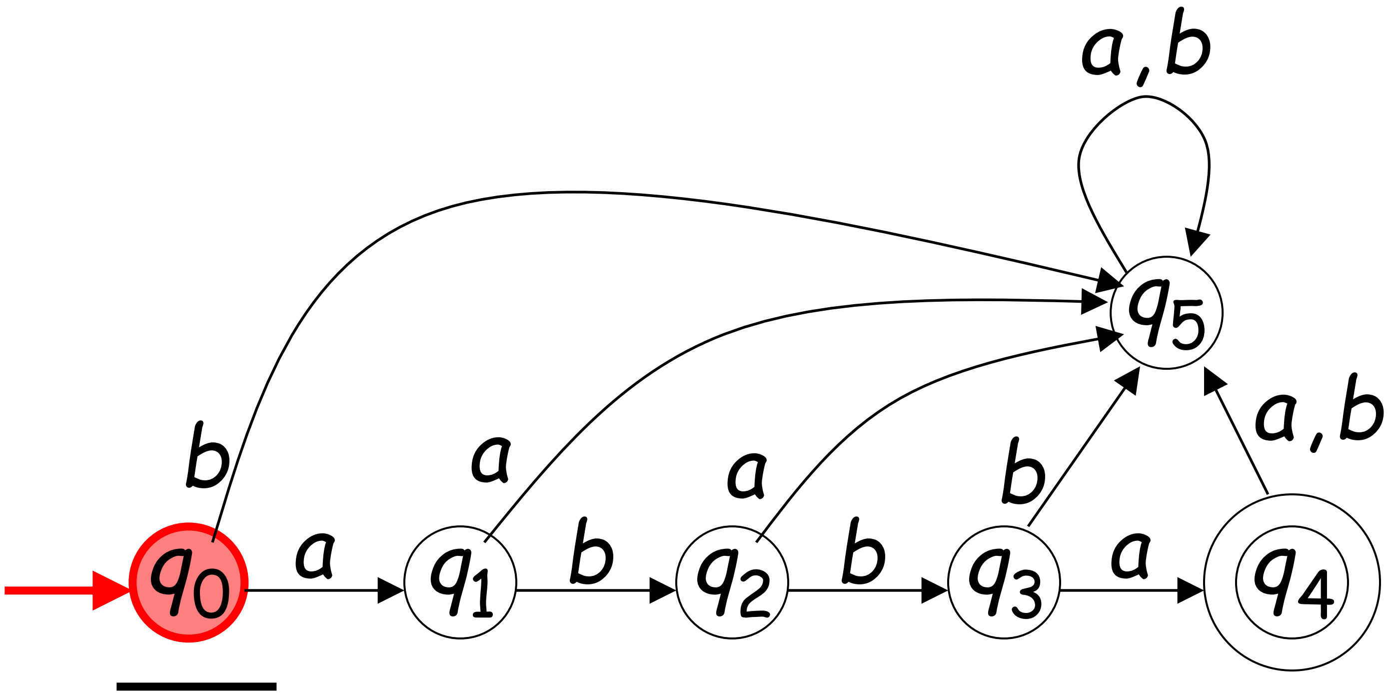


Set of States Q

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

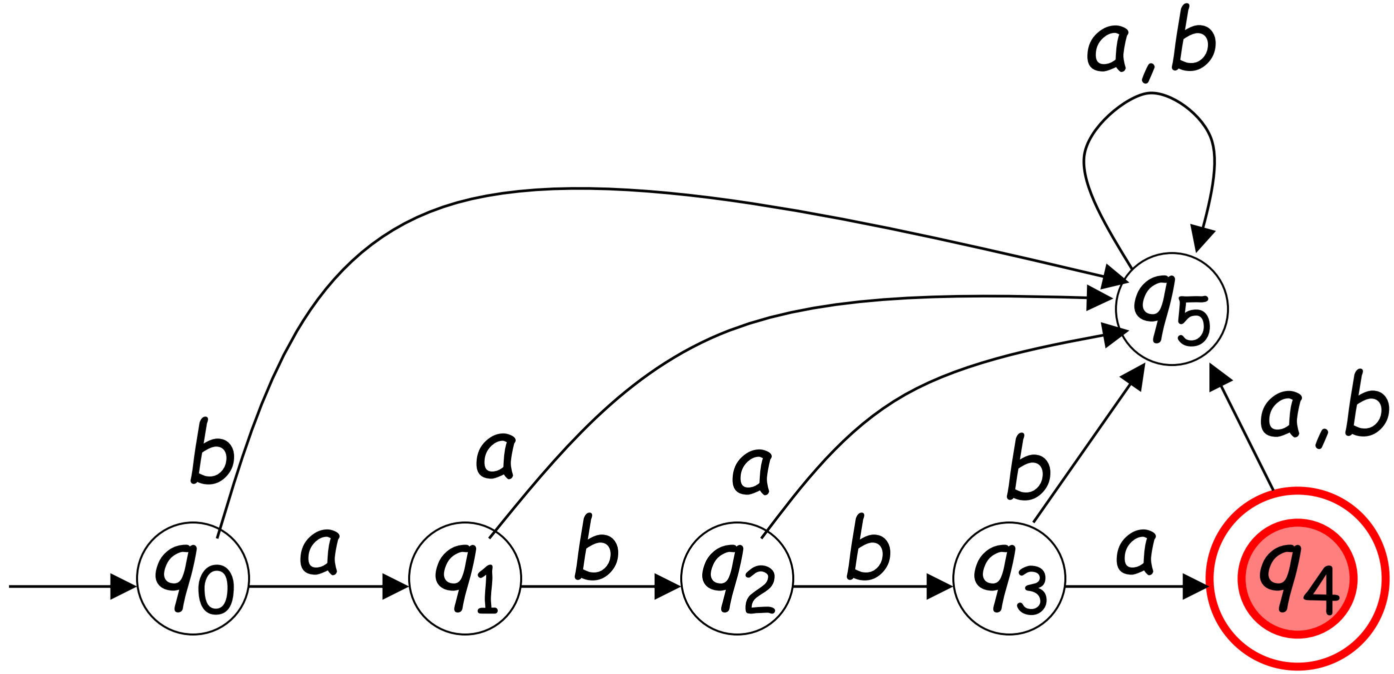


Initial State q_0



Set of Final States F

$$F = \{q_4\}$$

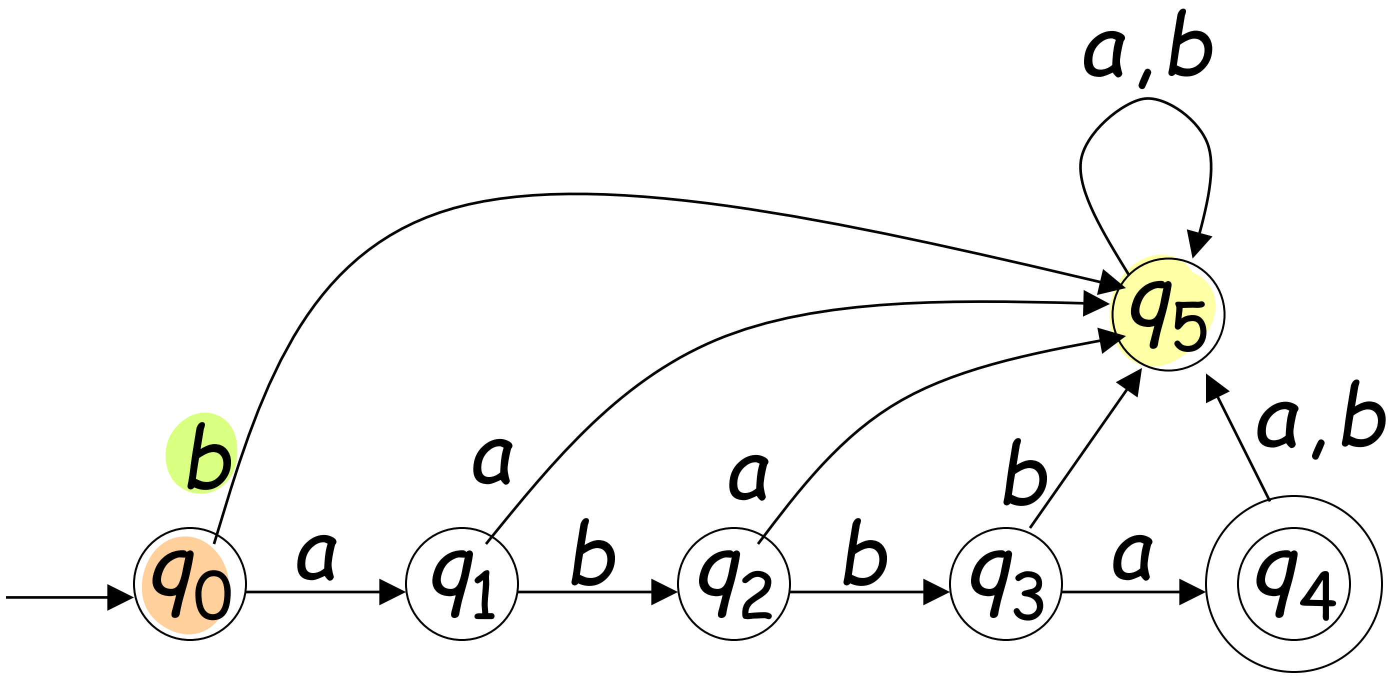


Transition Function δ

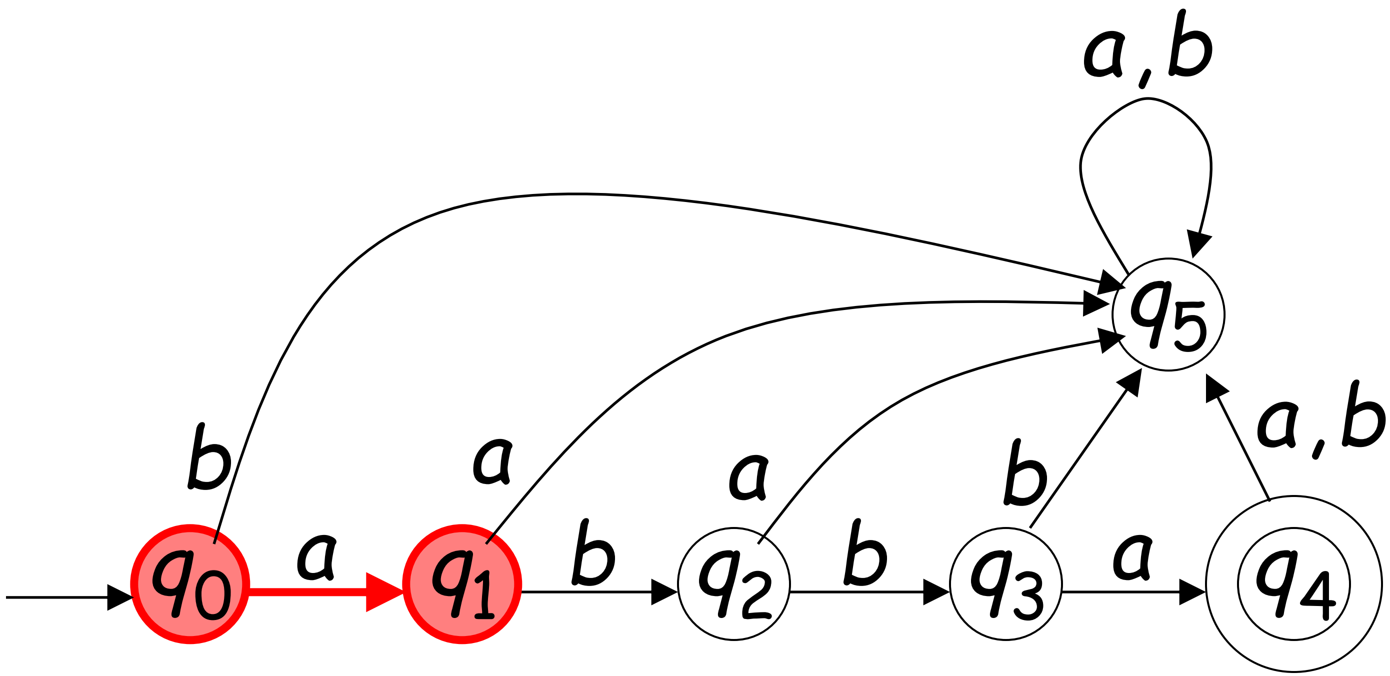
เดลตา

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

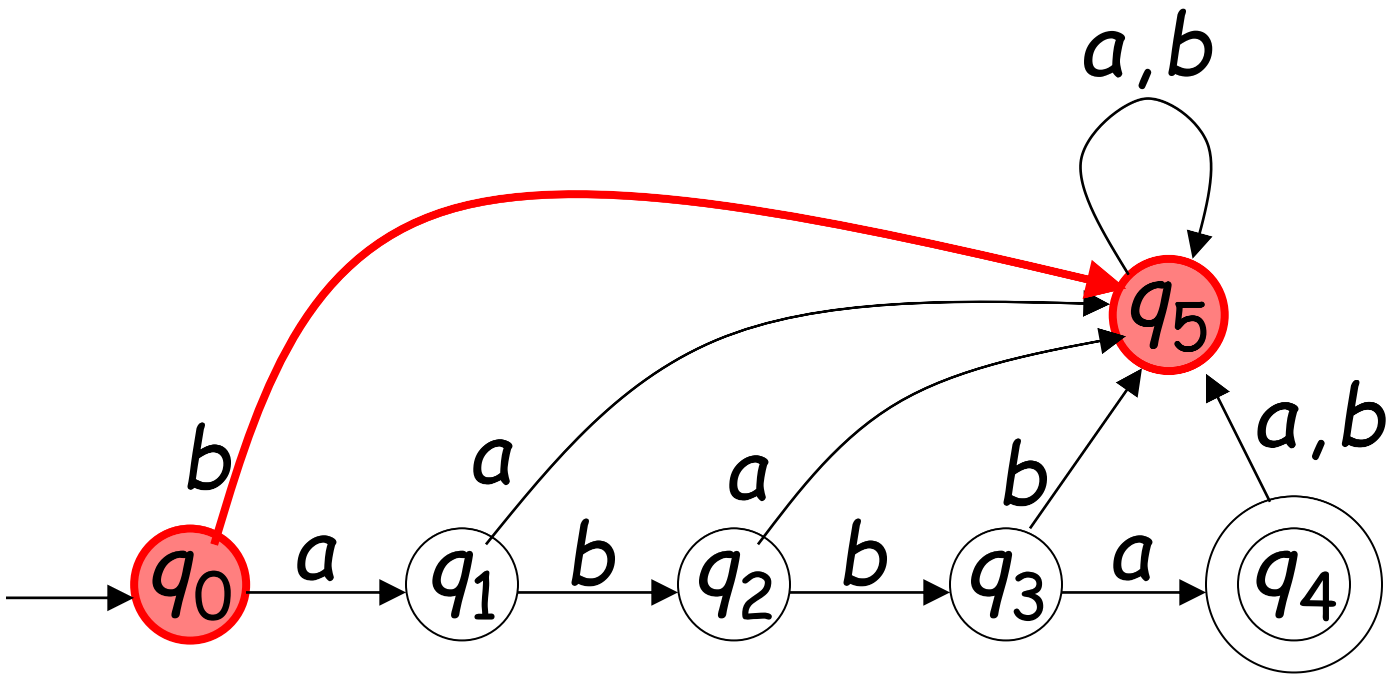
set state Input alphabet



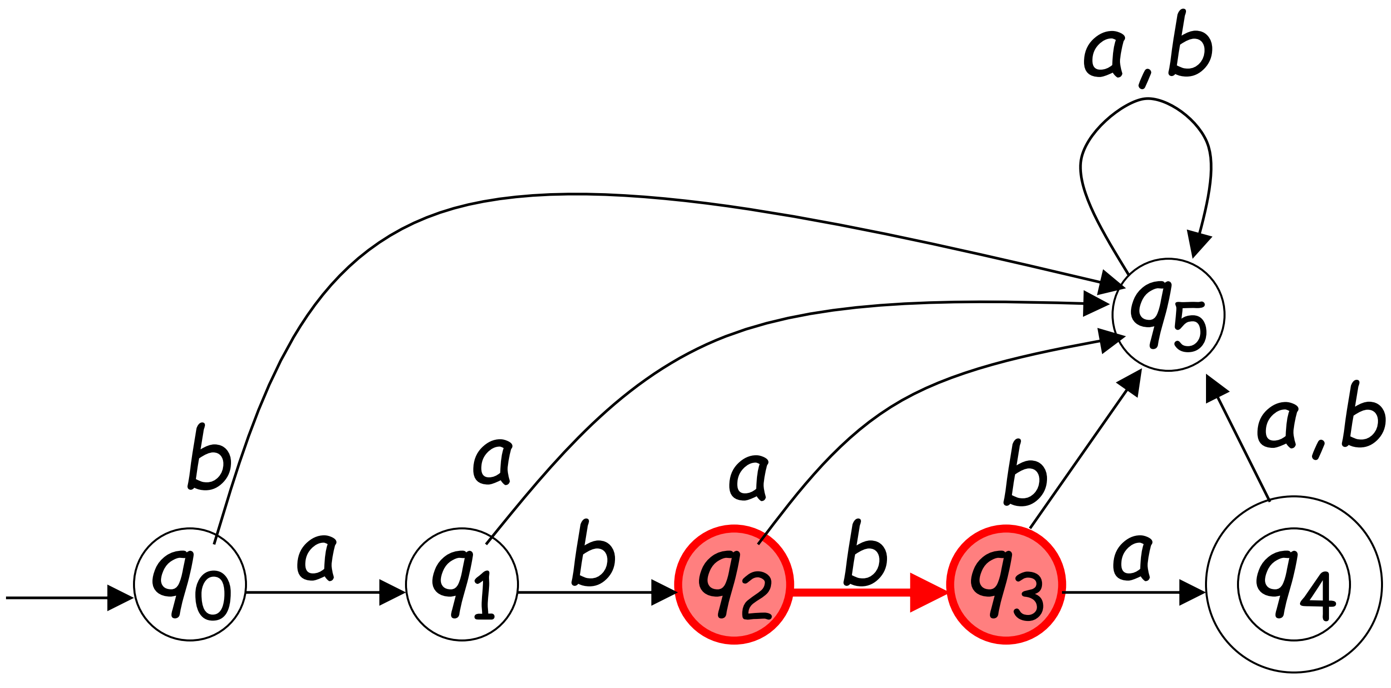
$$\delta(q_0, a) = q_1$$



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



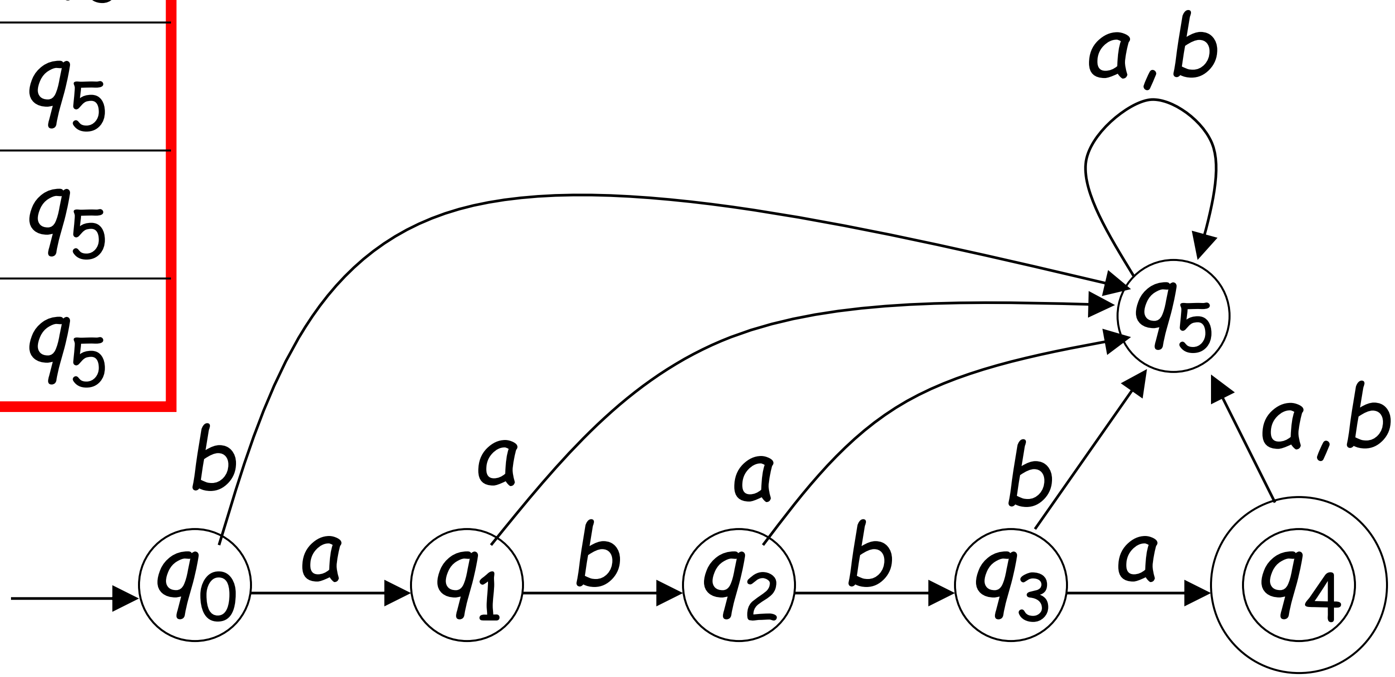
$$\delta(q_2, b) = q_3$$



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

$\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_5$

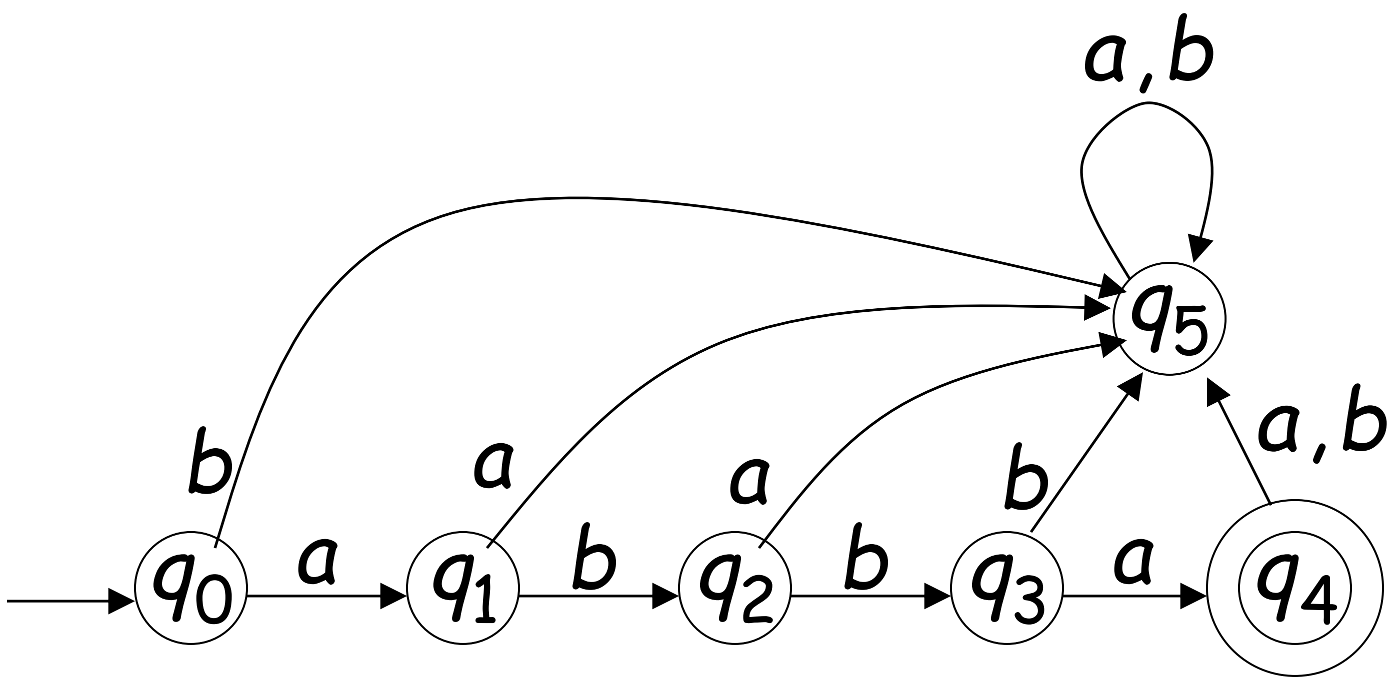


Extended Transition Function $\delta *$

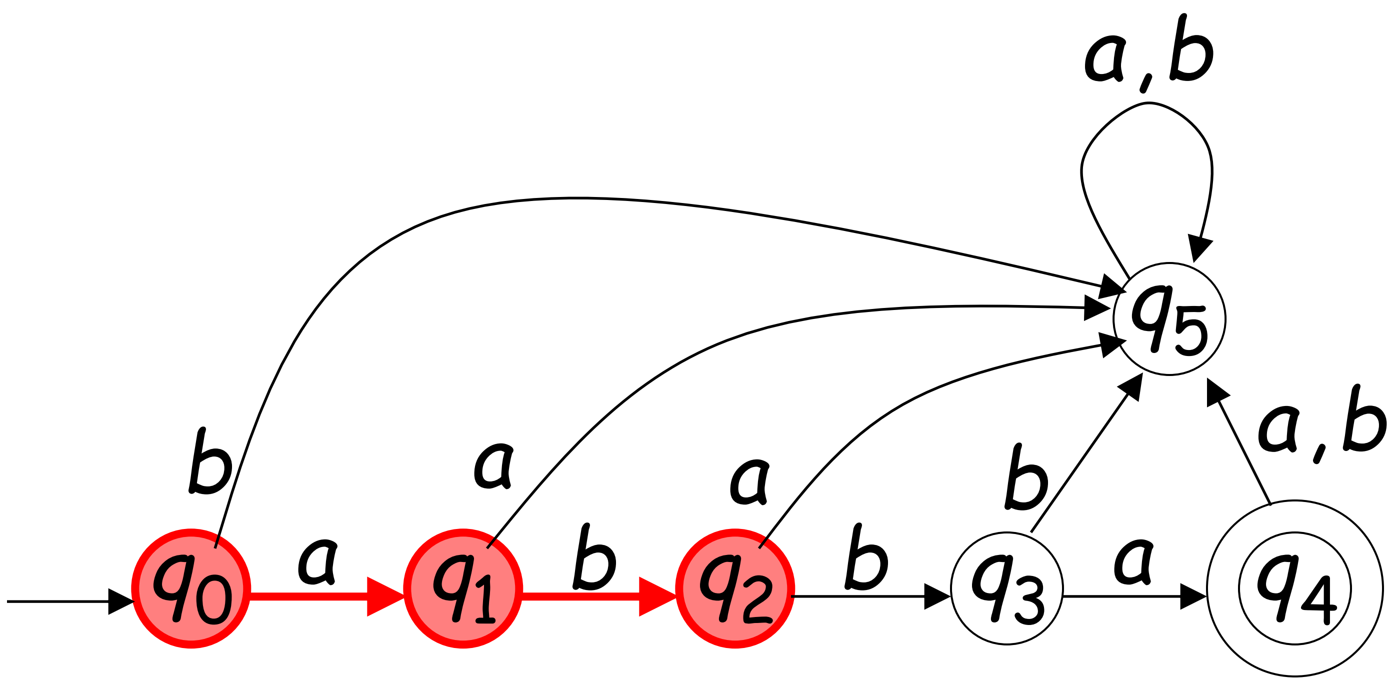
string

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

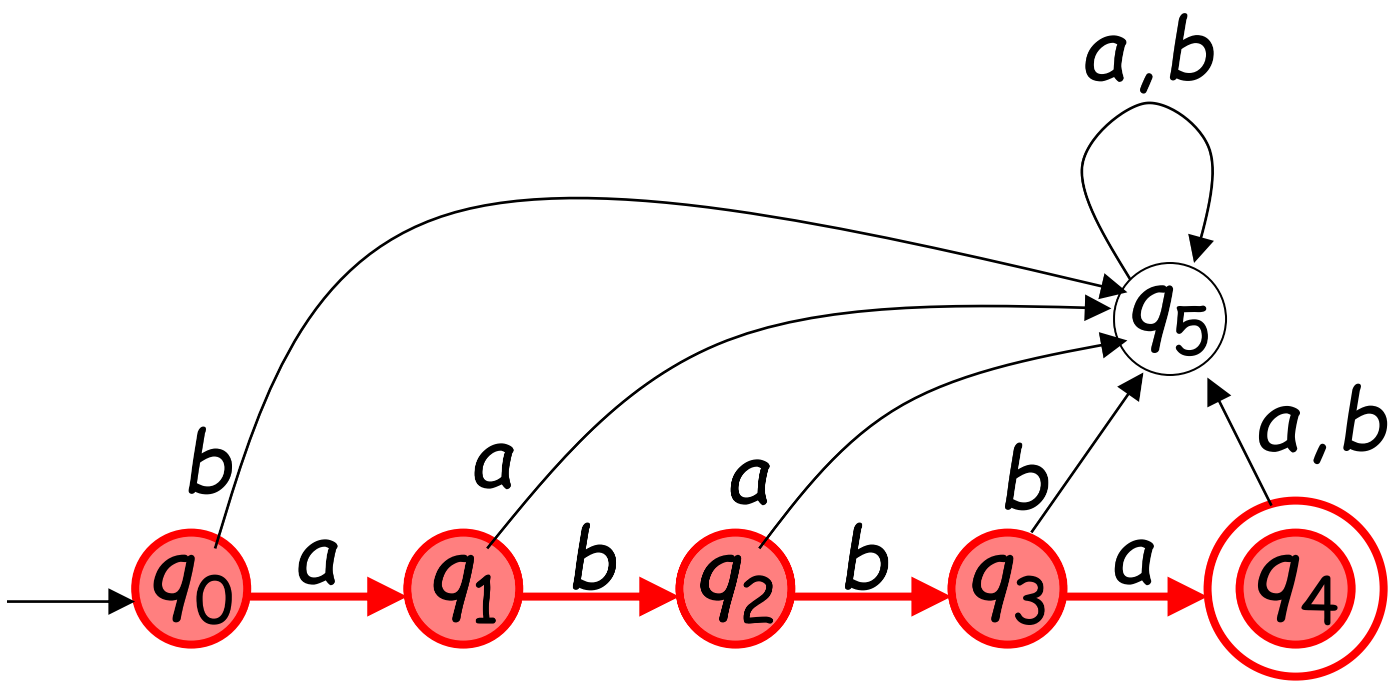
$$\delta : Q \times \Sigma \longrightarrow Q$$



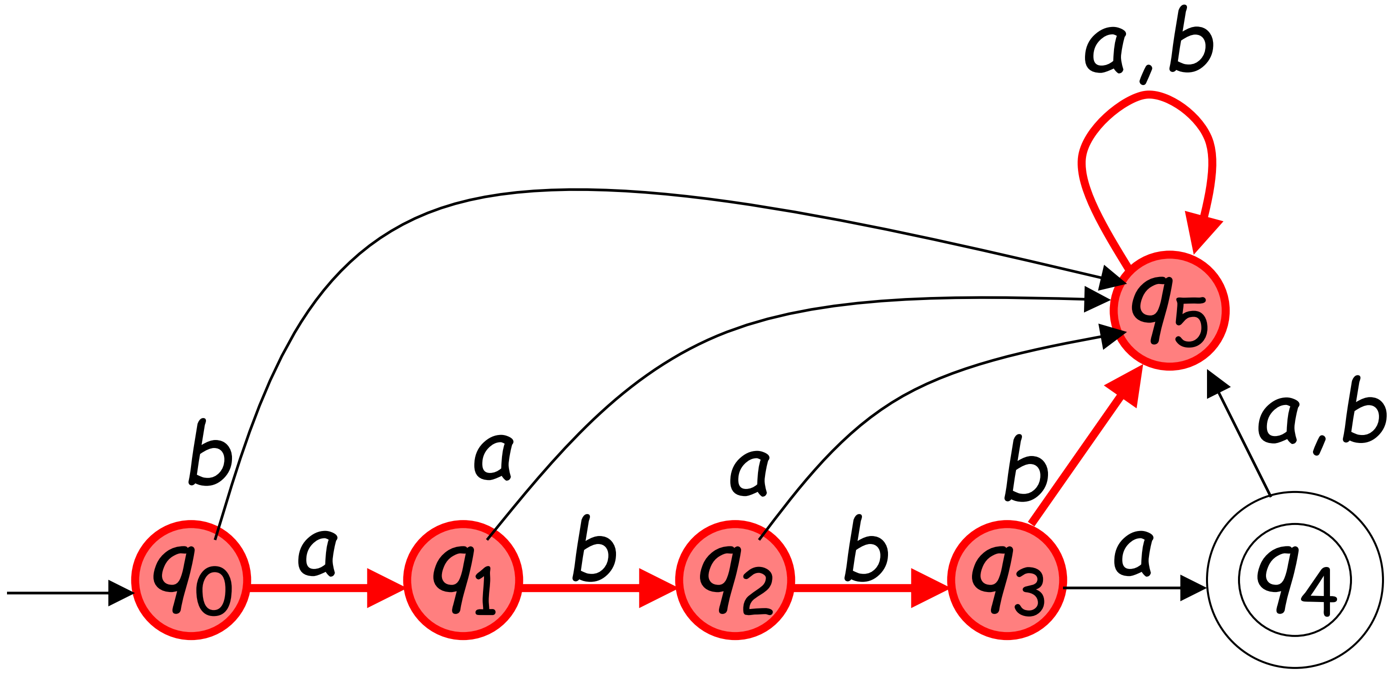
$$\delta^*(q_0, ab) = q_2$$



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



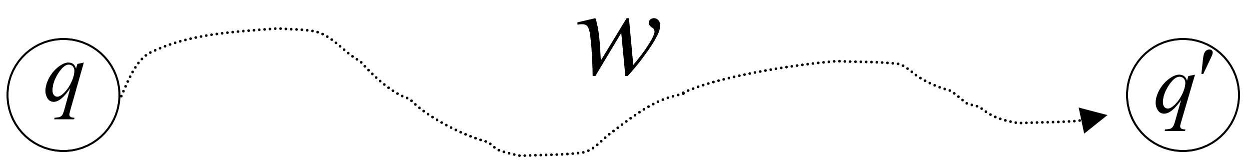
$$\delta^*(q_0, abbbaa) = q_5$$



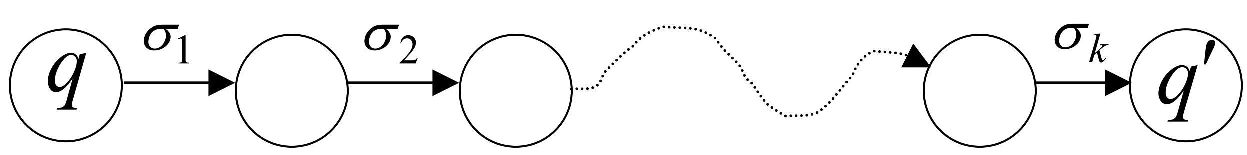
Observation: There is a walk from q to q' with label w

$\delta^*(q, \Sigma^*)$

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

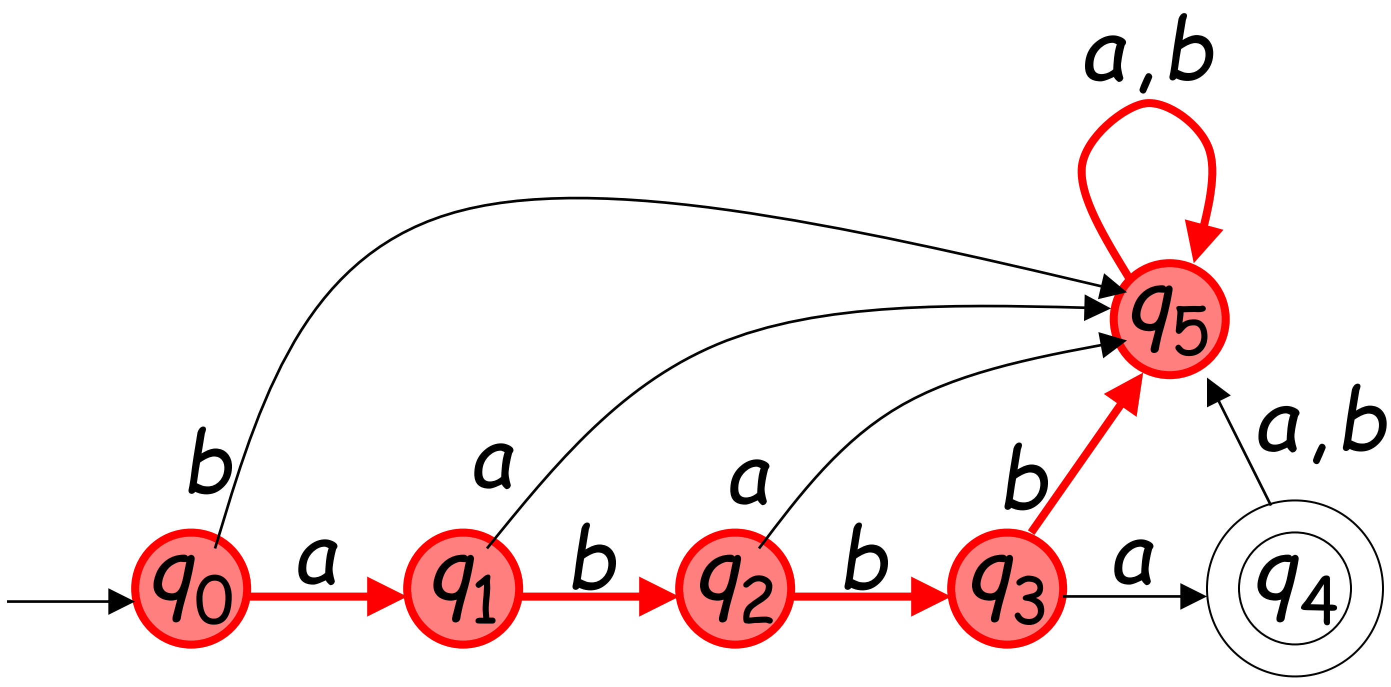


$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



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

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



Languages Accepted by DFAs

Take DFA M

Definition:

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

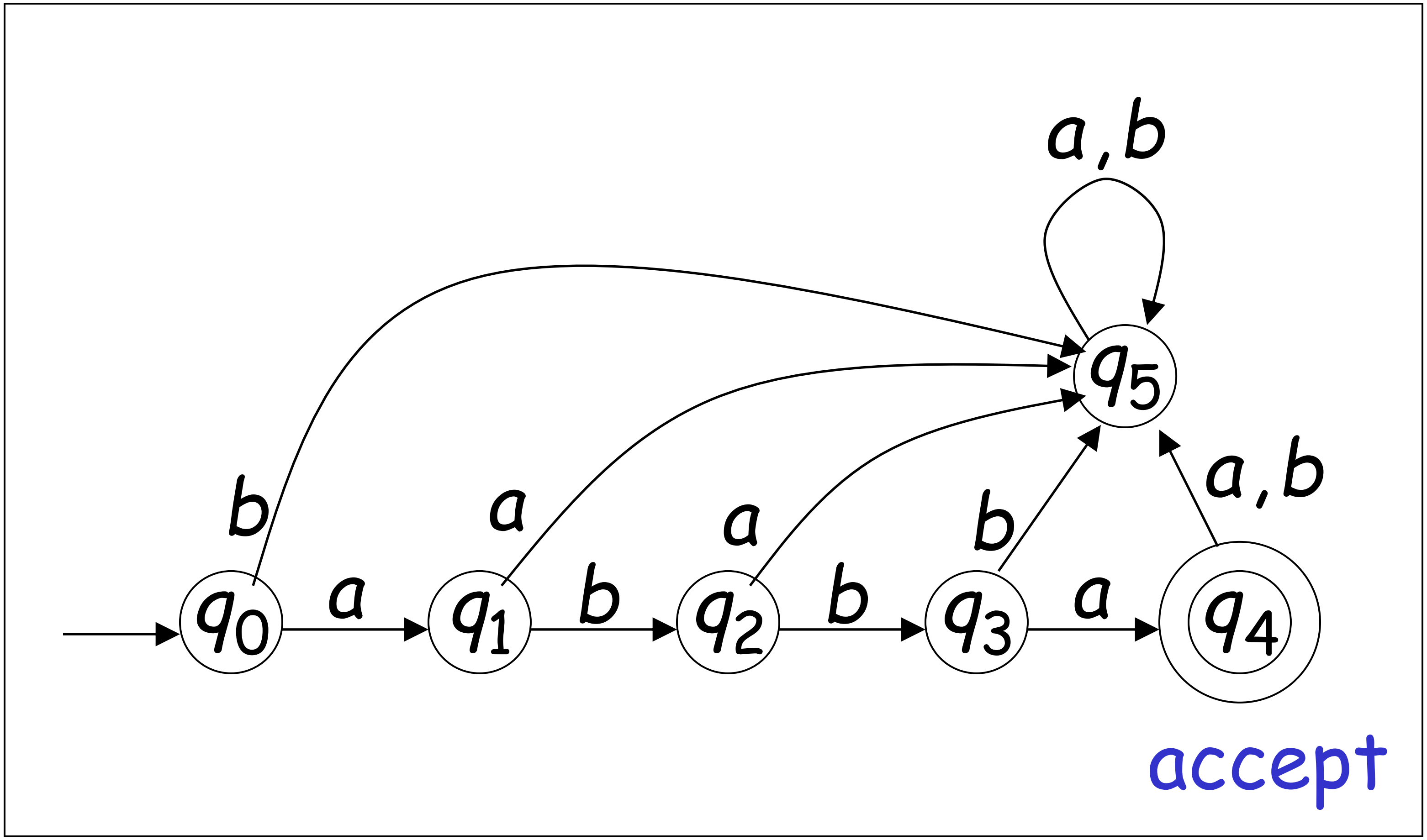
เซตของ string ที่ขับเคลื่อน M ให้เข้าสู่ final state ได้

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

Example

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

M

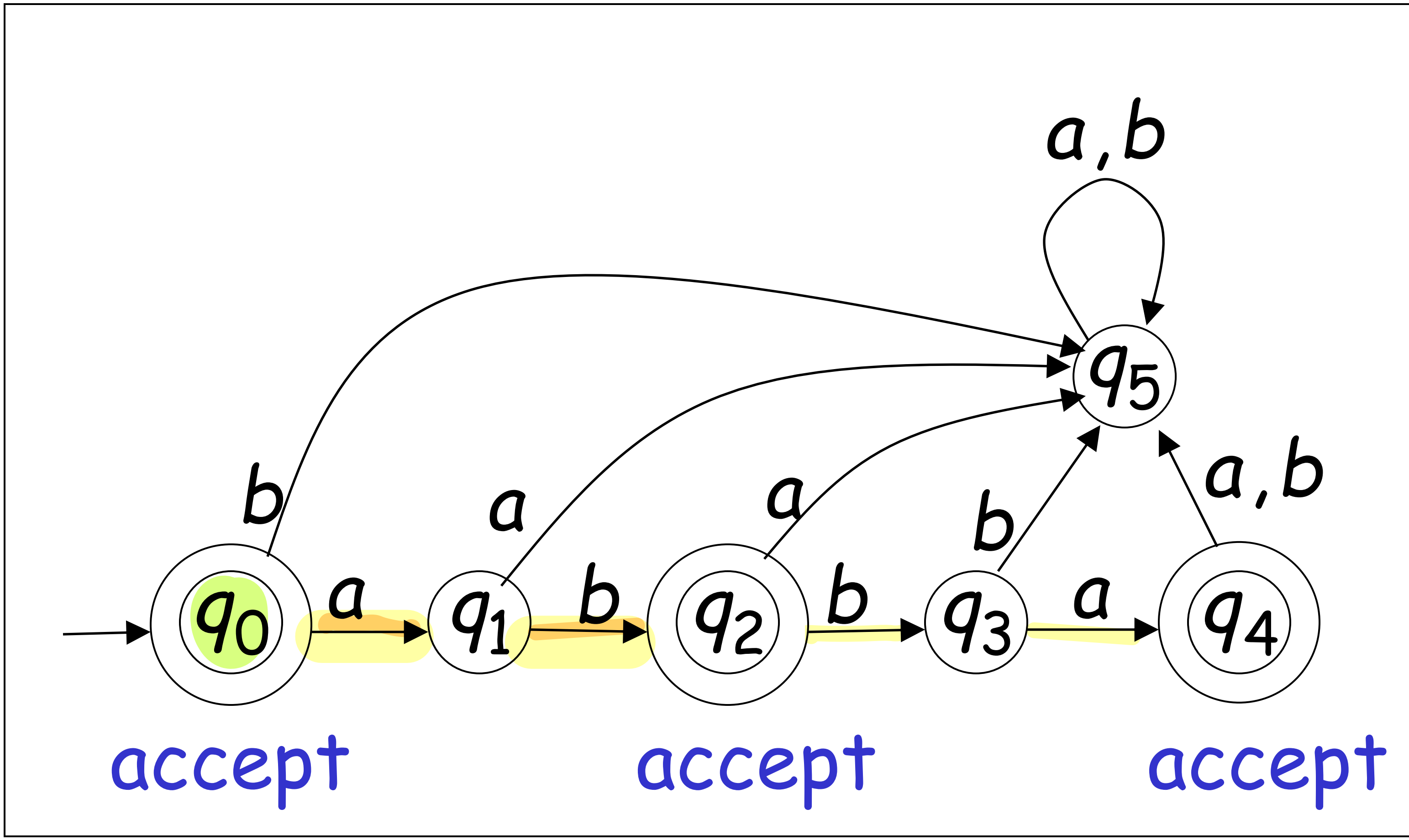


Another Example

มี string 3 string

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

M



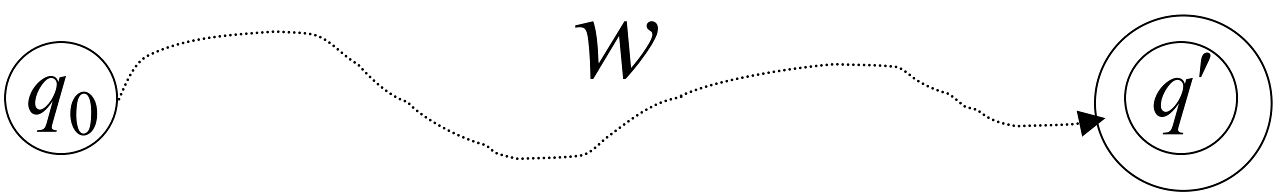
Formally

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

Language accepted by M :

ได้ state หนึ่ง ∈ ใน Final state

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



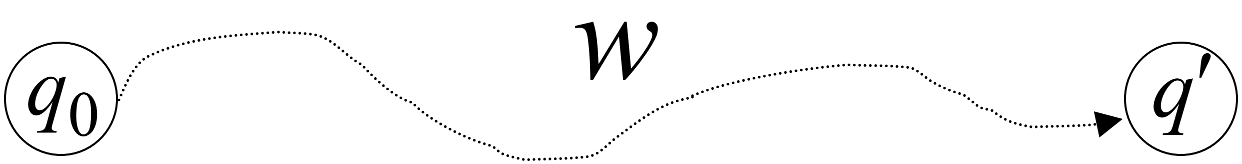
$q' \in F$

Observation

Language rejected by M :

ไม่เป็นสมาชิกใน Final state

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

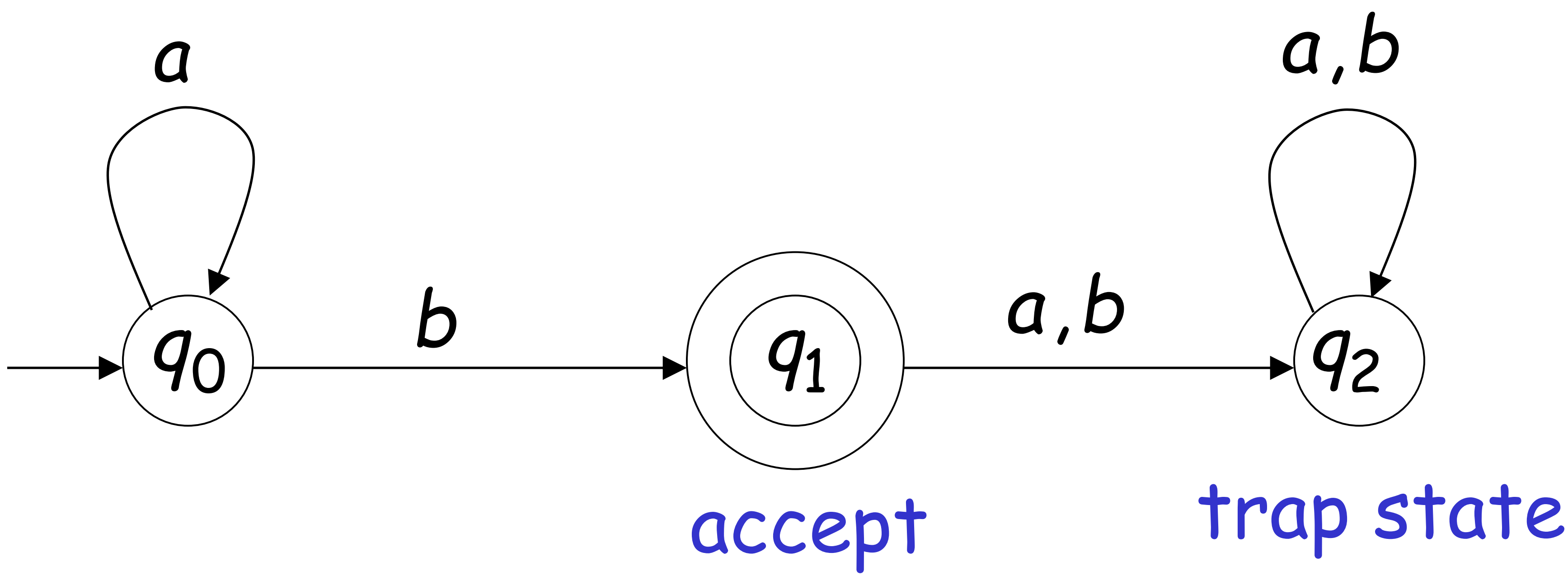


$q' \notin F$

More Examples

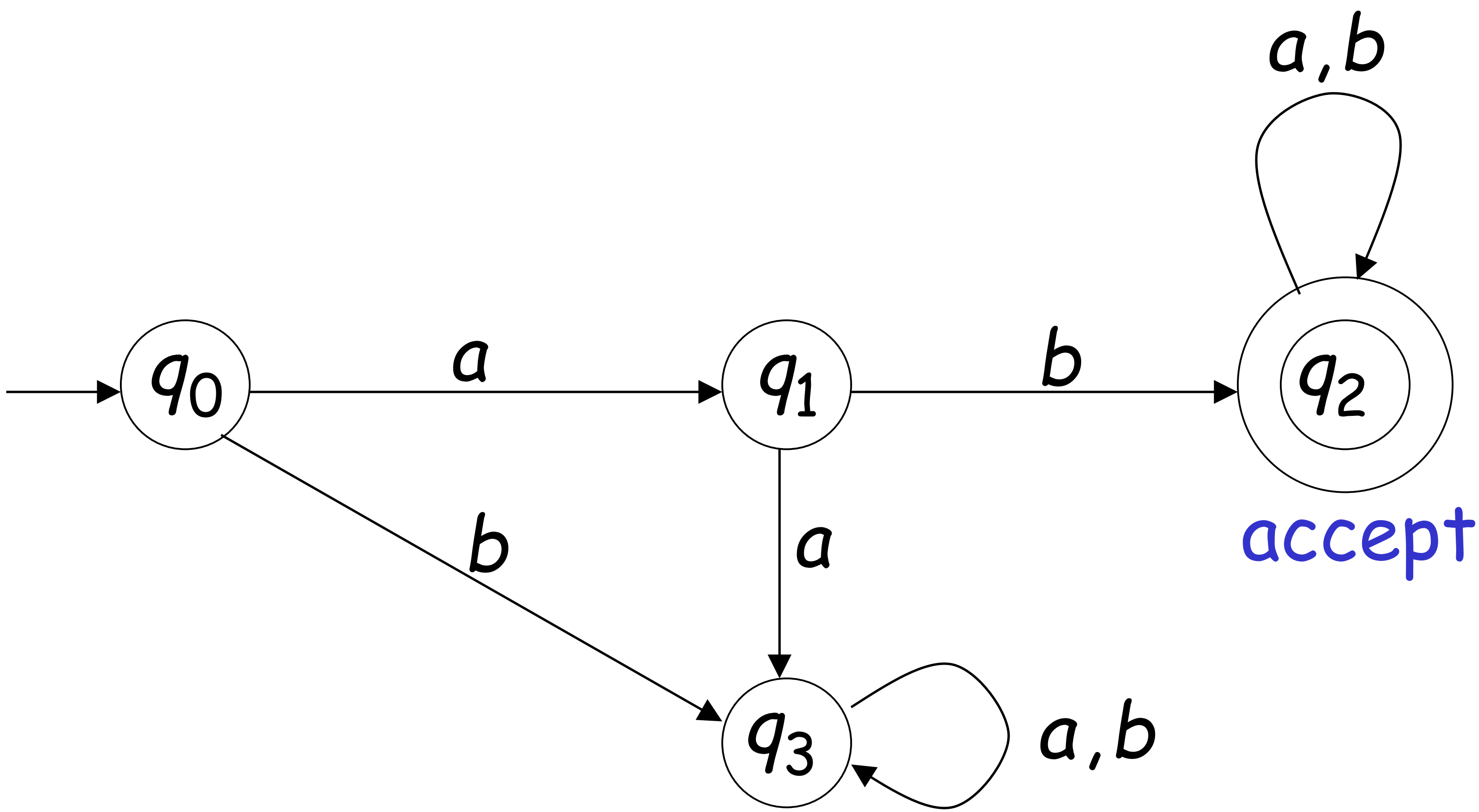
$$L(M) = \{a^n b : n \geq 0\}$$

b n = 0
ab n = 1
aab n = 2
aaab n = 3



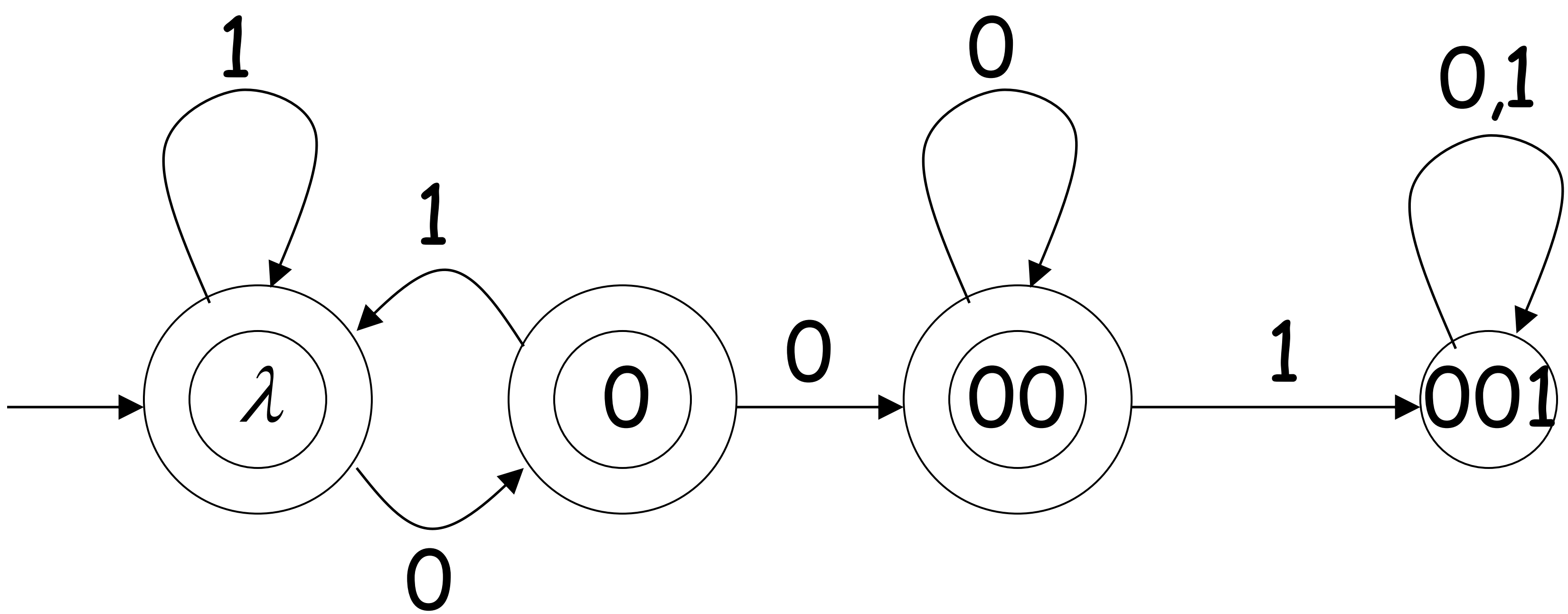
ข้างหน้า

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



ไม่มี substring 001

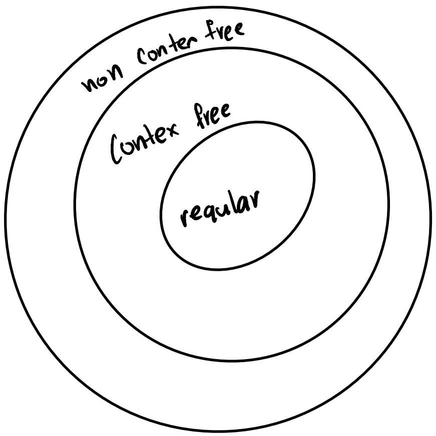
$L(M)$ = { all strings without substring 001 }



Regular Languages

ถ้าสร้าง DFA รองรับ L ได้ มันคือ L คือภาษา regular

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

0, 1, 2, 3, 4

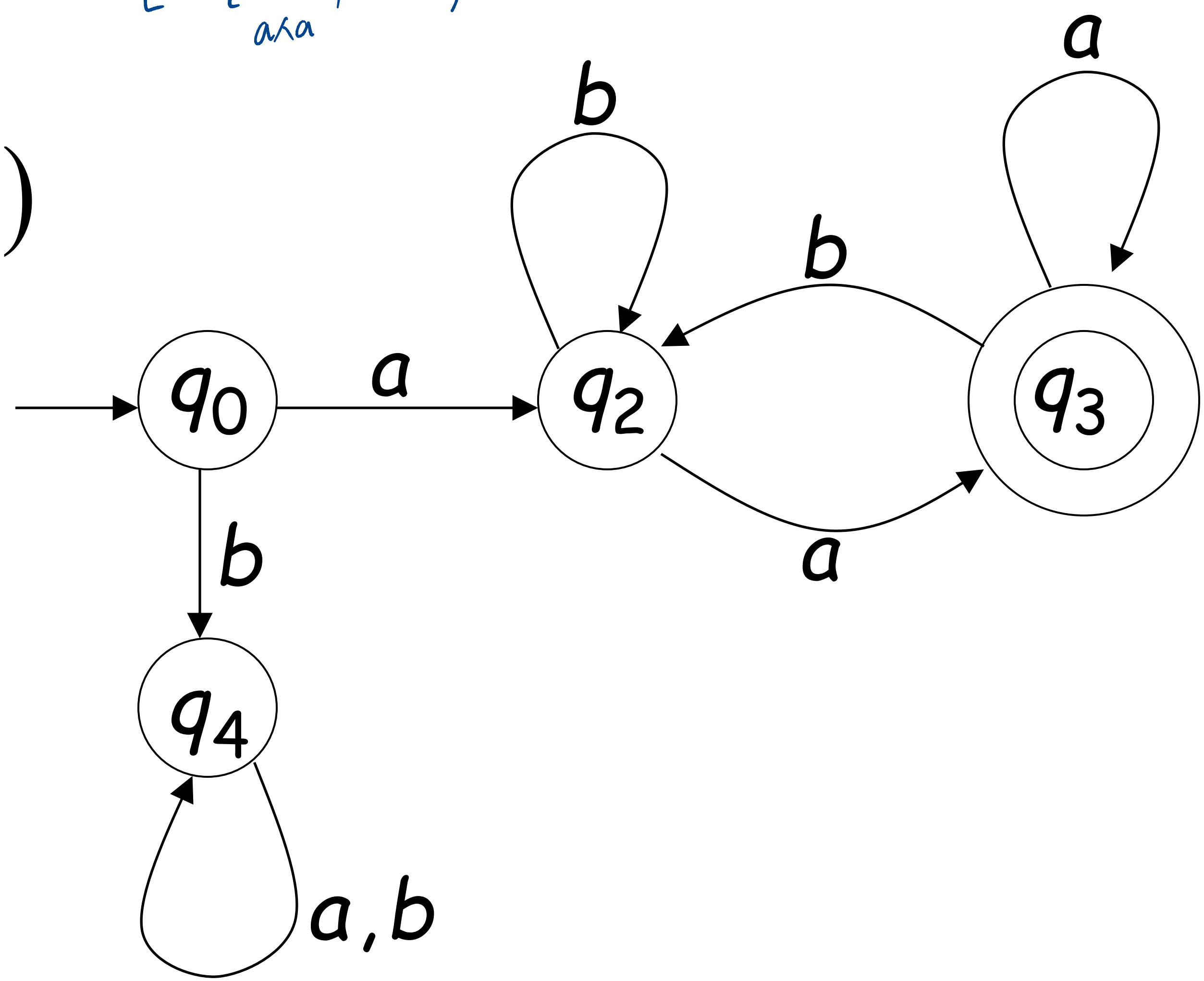
The language $L = \{awa : w \in \{a,b\}^*\}$ is regular:

$\{a,b\}^0 = \lambda$, $\{a,b\}^1 = \{a,b\}$

$L = \{ aa, aaa, aba$

$a\lambda a$

$$L = L(M)$$



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

Example: $$\underbrace{L = \{a^n b^n : n \ge 0\}}_{\text{สร้าง DFA ไม่ได้ แสดงว่า ไม่ใช่ ภาษา regular (ความจำมีจำกัด)}}$$

There is no DFA that accepts such a language

(we will prove this later in the class)