Deterministic Finite Automata

Finite Automaton

Definition: An **automaton** is an abstract model of a digital computer.

Remark:

Computer : any computing device

Finite memory

Finite Automaton

Example: Vending Machines

Finite Automaton

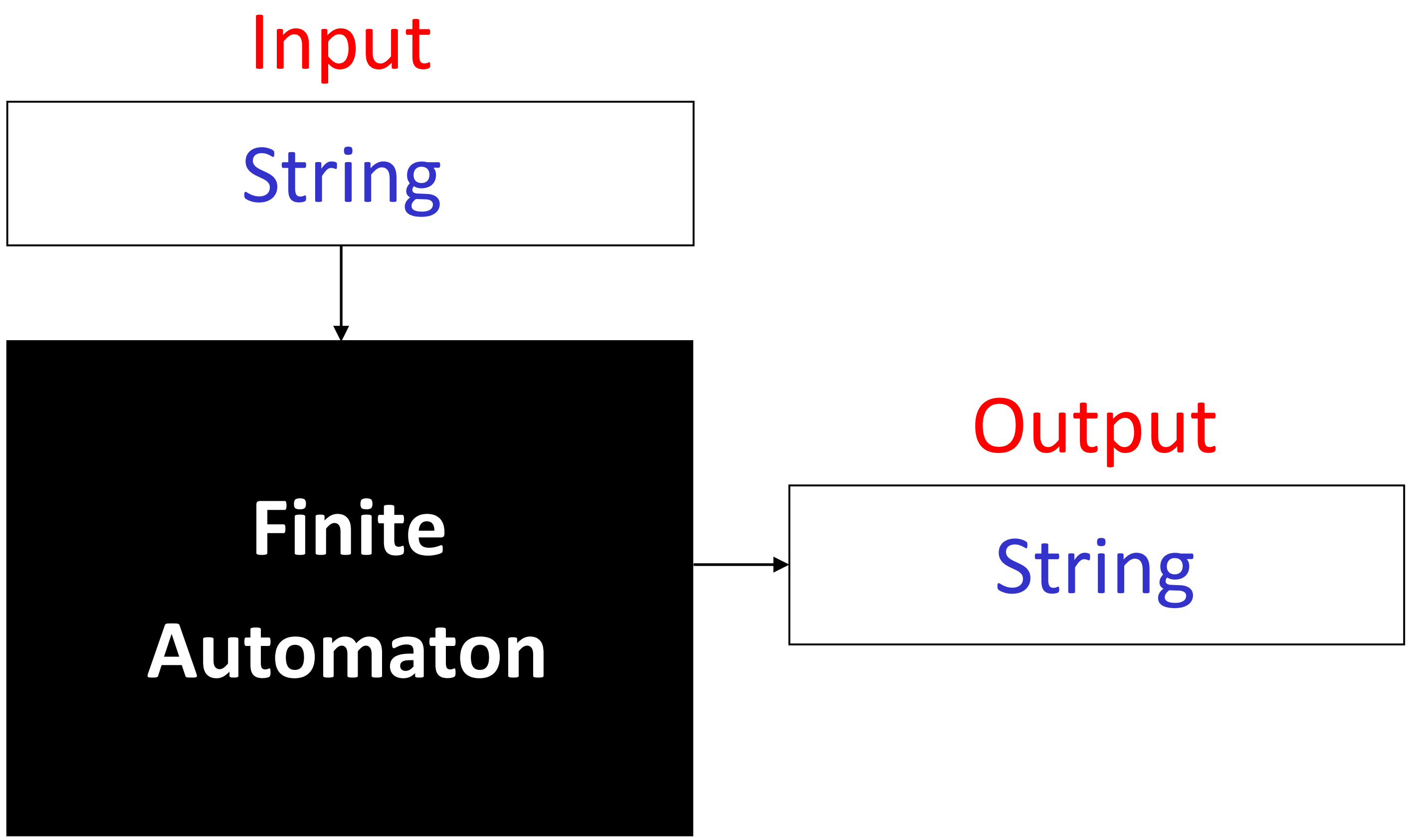
Example: Card terminals

Finite Automaton

Example: Auto barriers

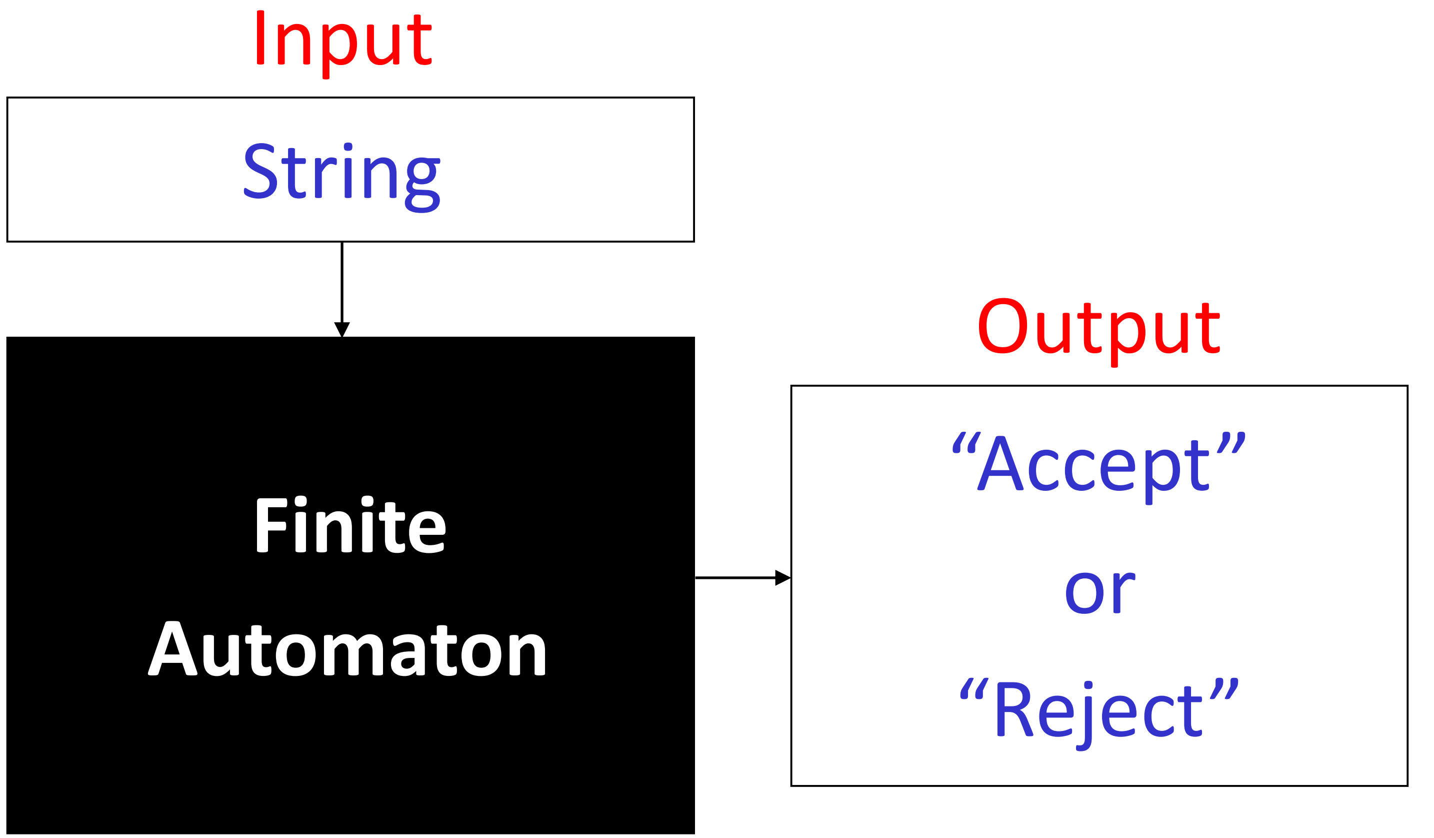
Finite Automaton

- An **automaton** is an abstract model of a digital computer / computing devices.

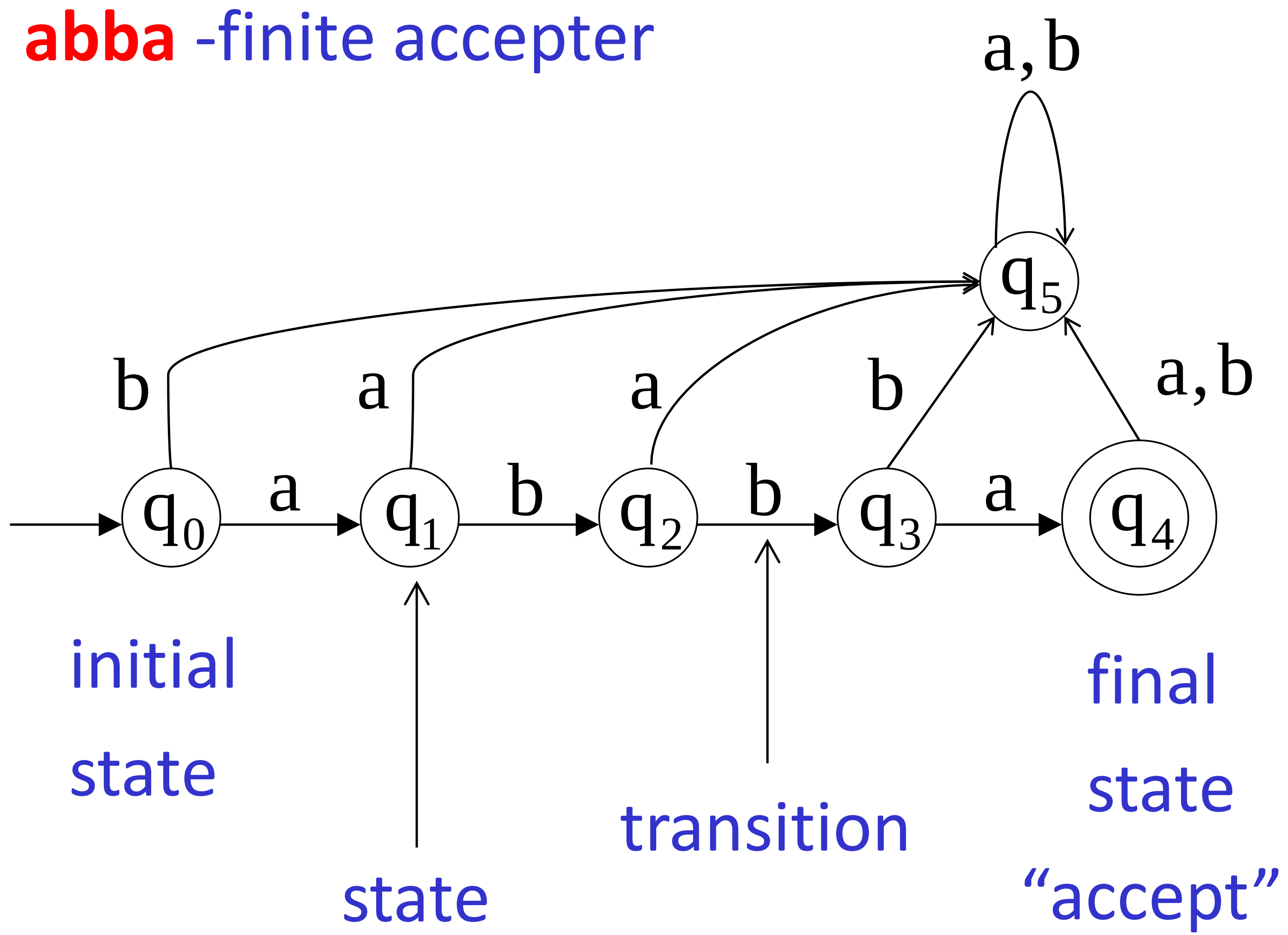


Finite Accepter

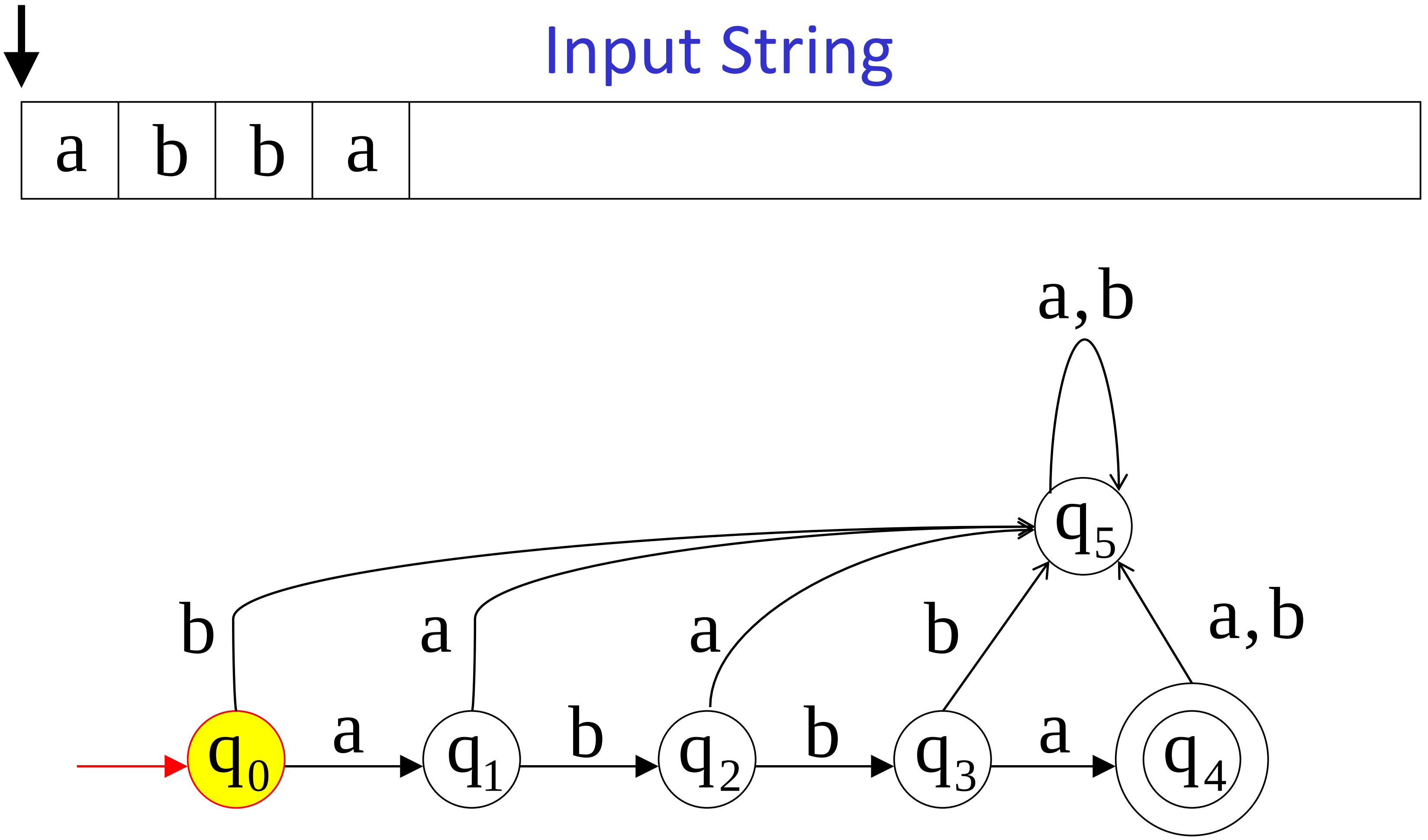
- An **automaton** is an abstract model of a digital computer / computing devices.



Transition Graph



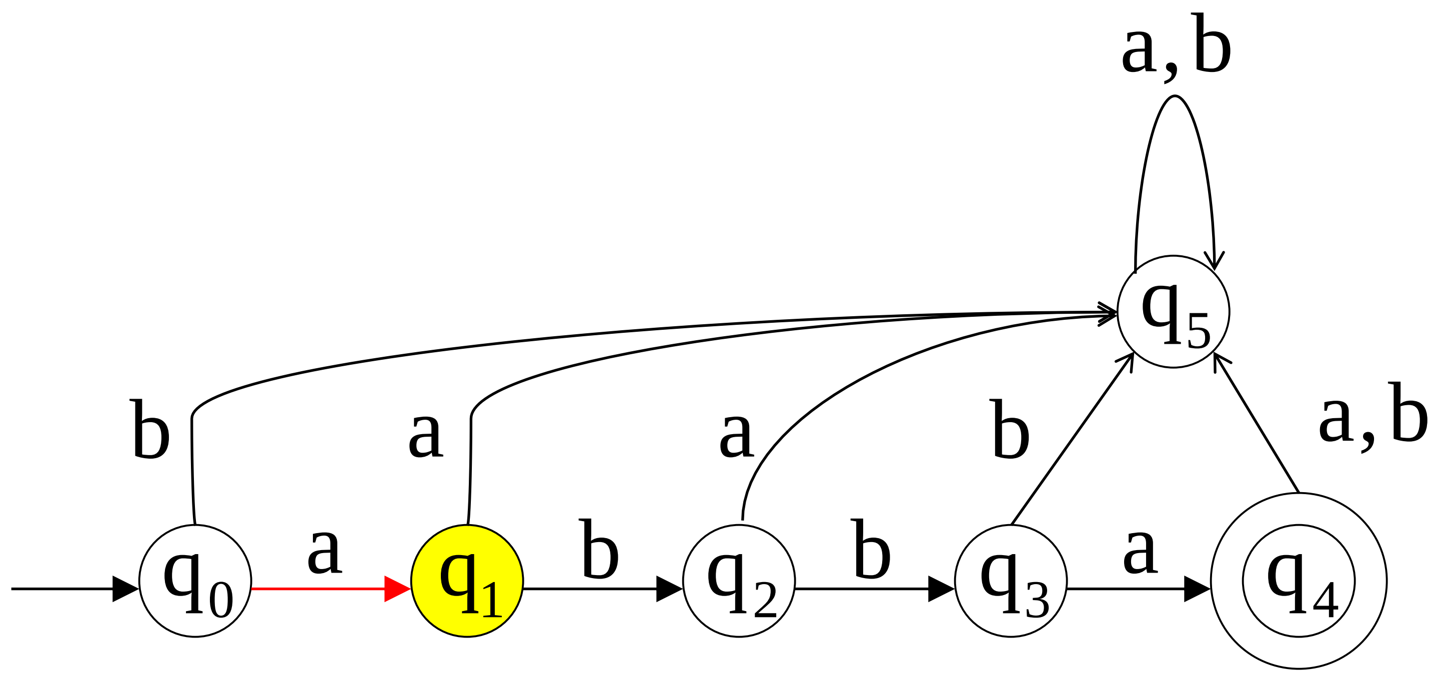
Initial Configuration



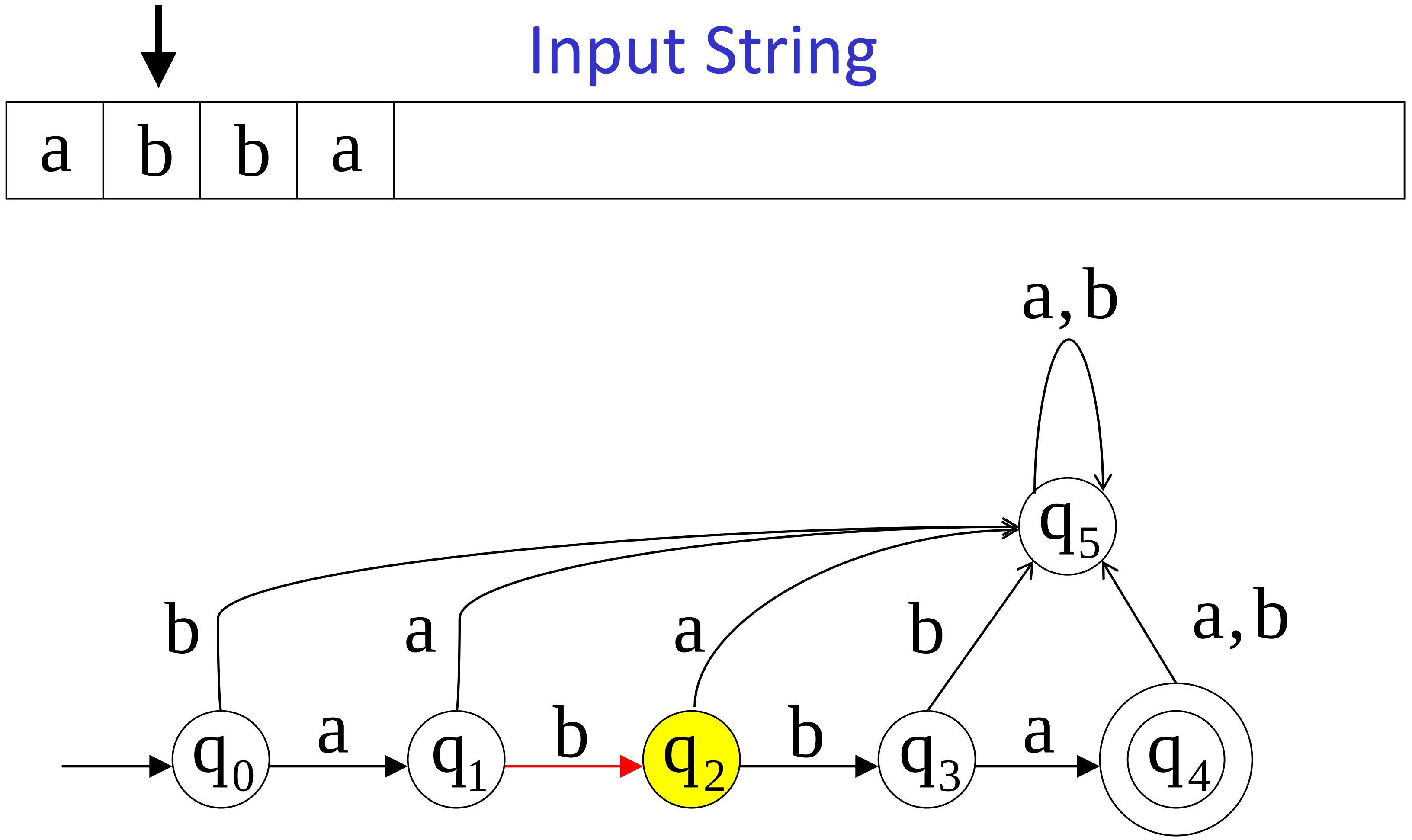
Reading the Input

Input String

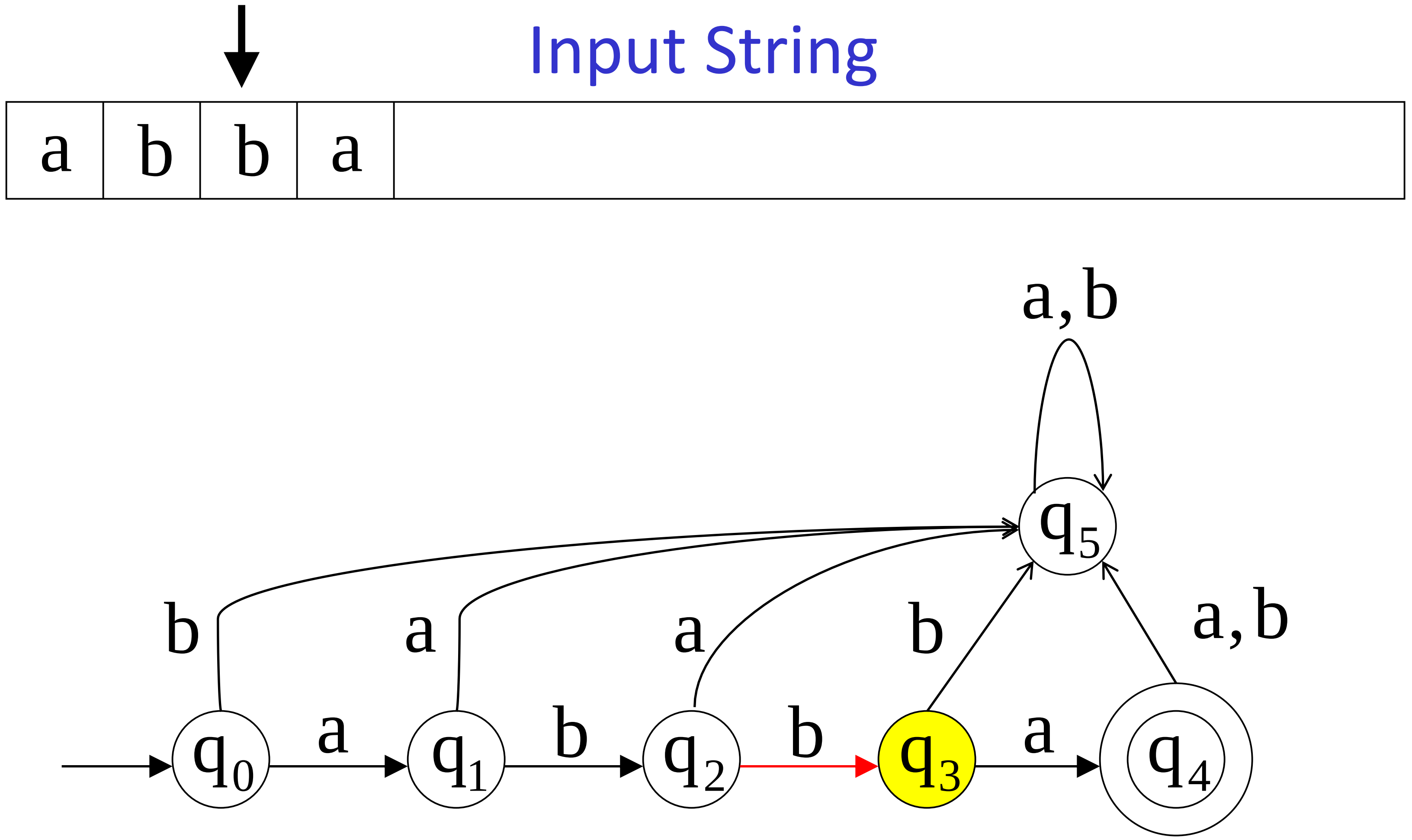
a	b	b	a	



Reading the Input



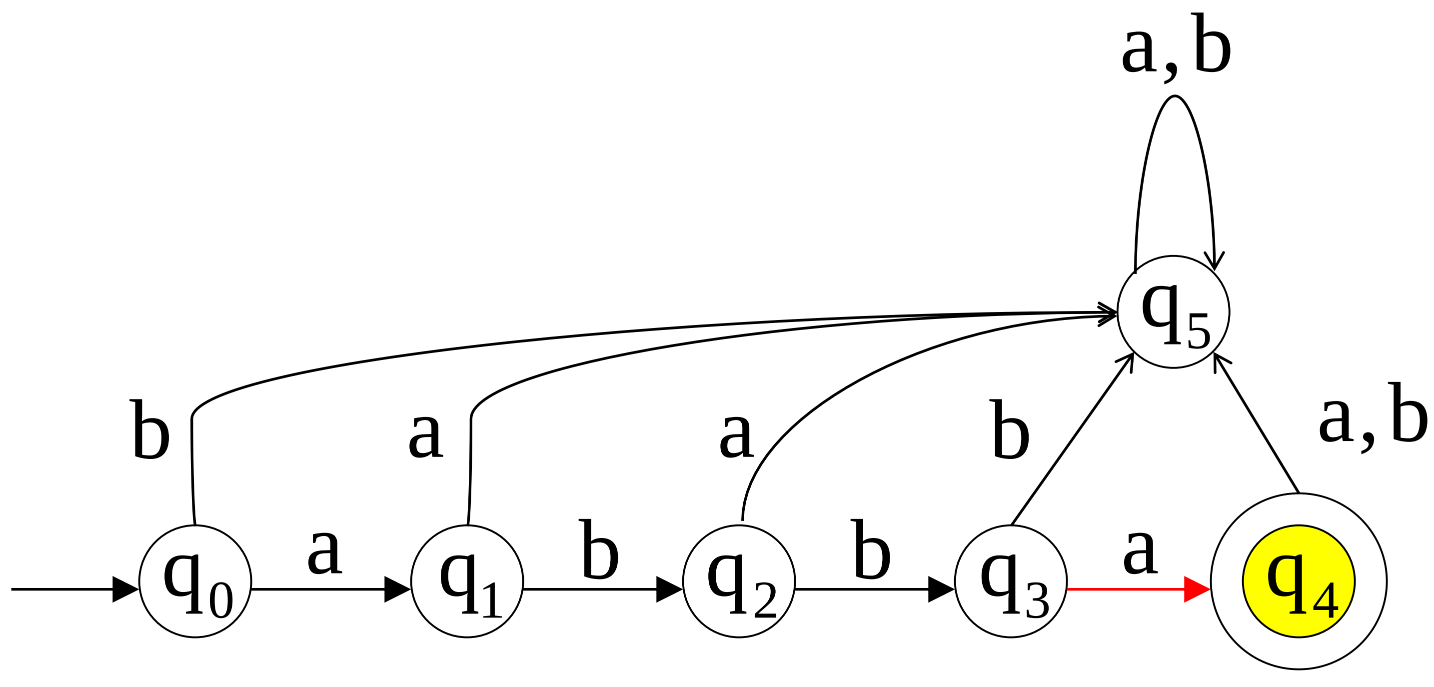
Reading the Input



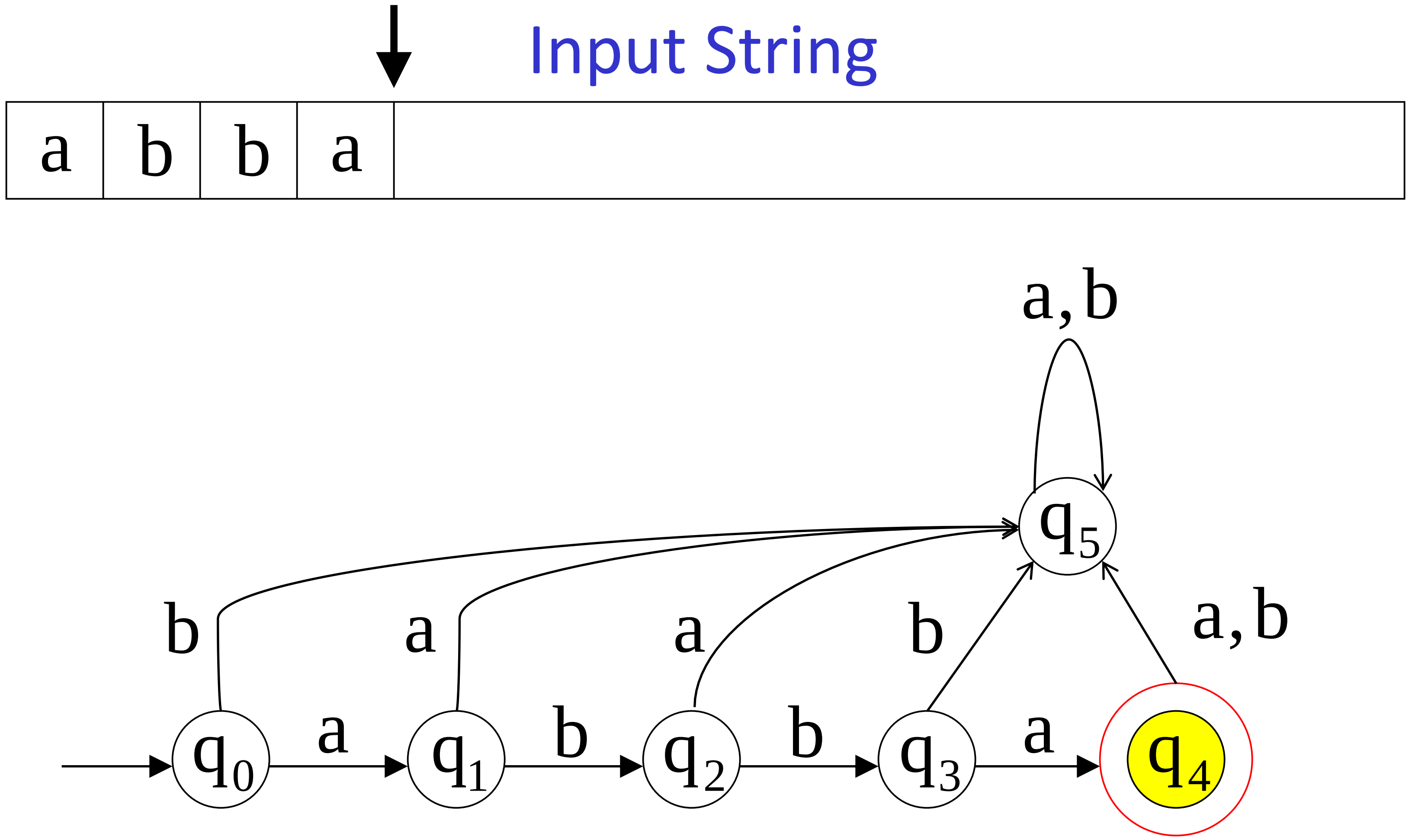
Reading the Input

Input String

a	b	b	a	

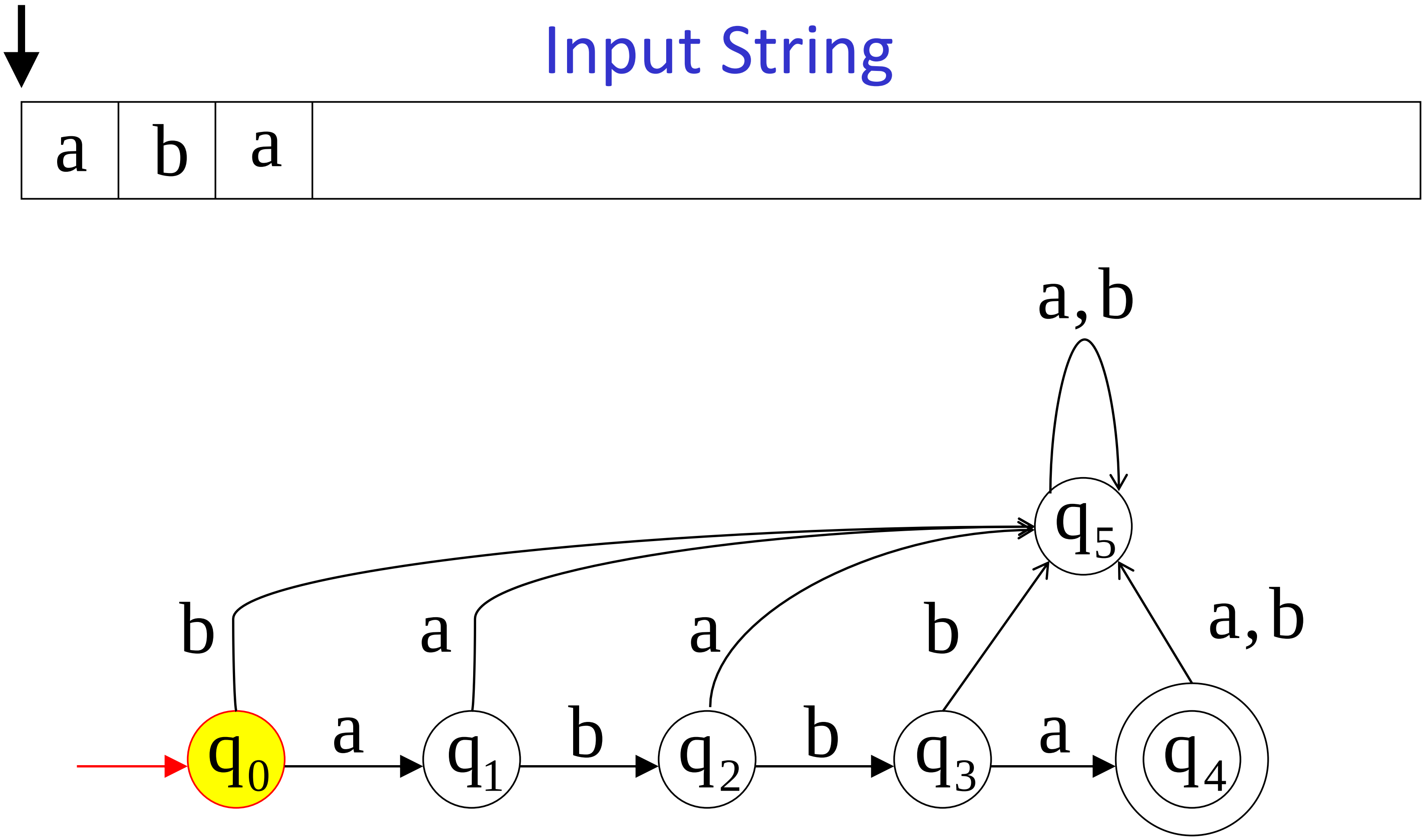


Reading the Input

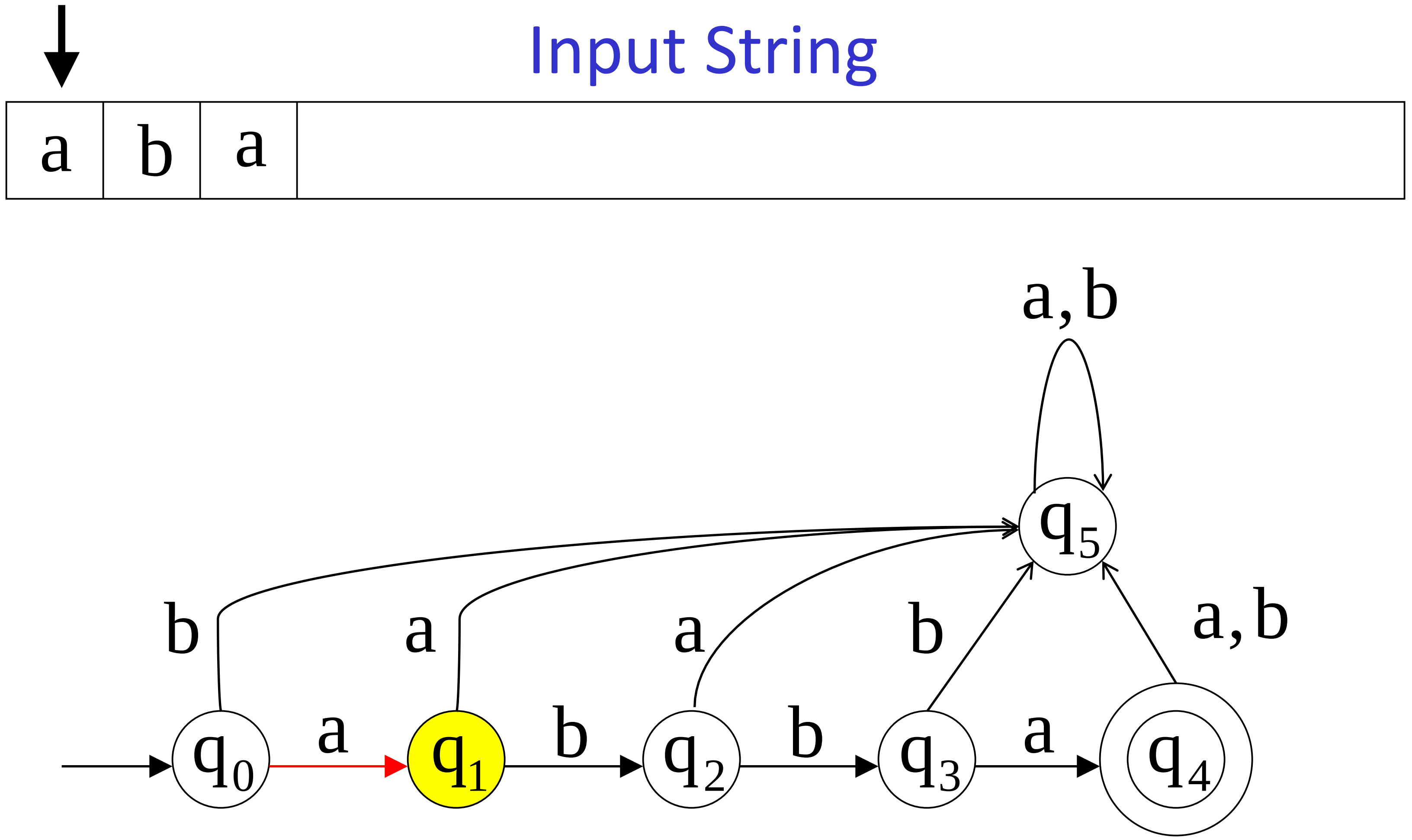


Output: "accept"

Rejection



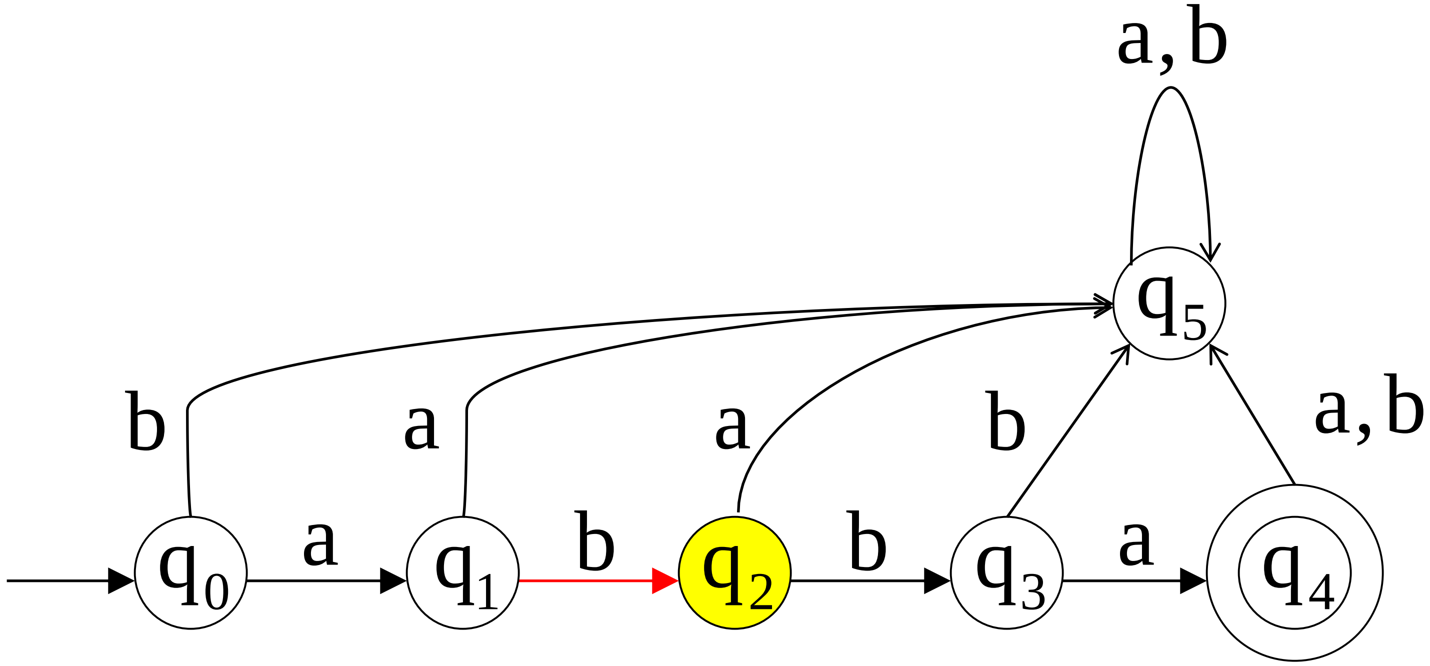
Rejection



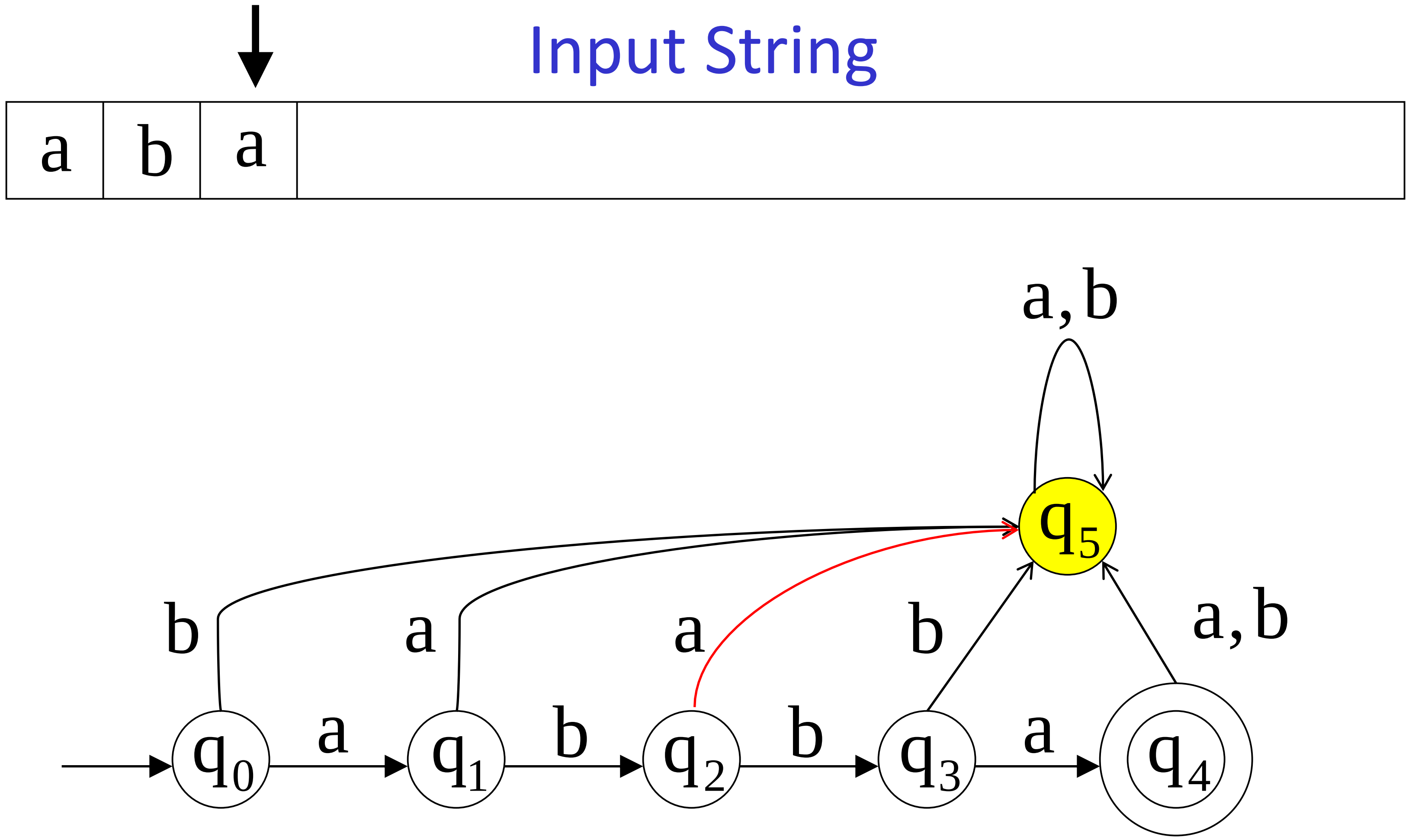
Rejection

Input String

a	b	a	

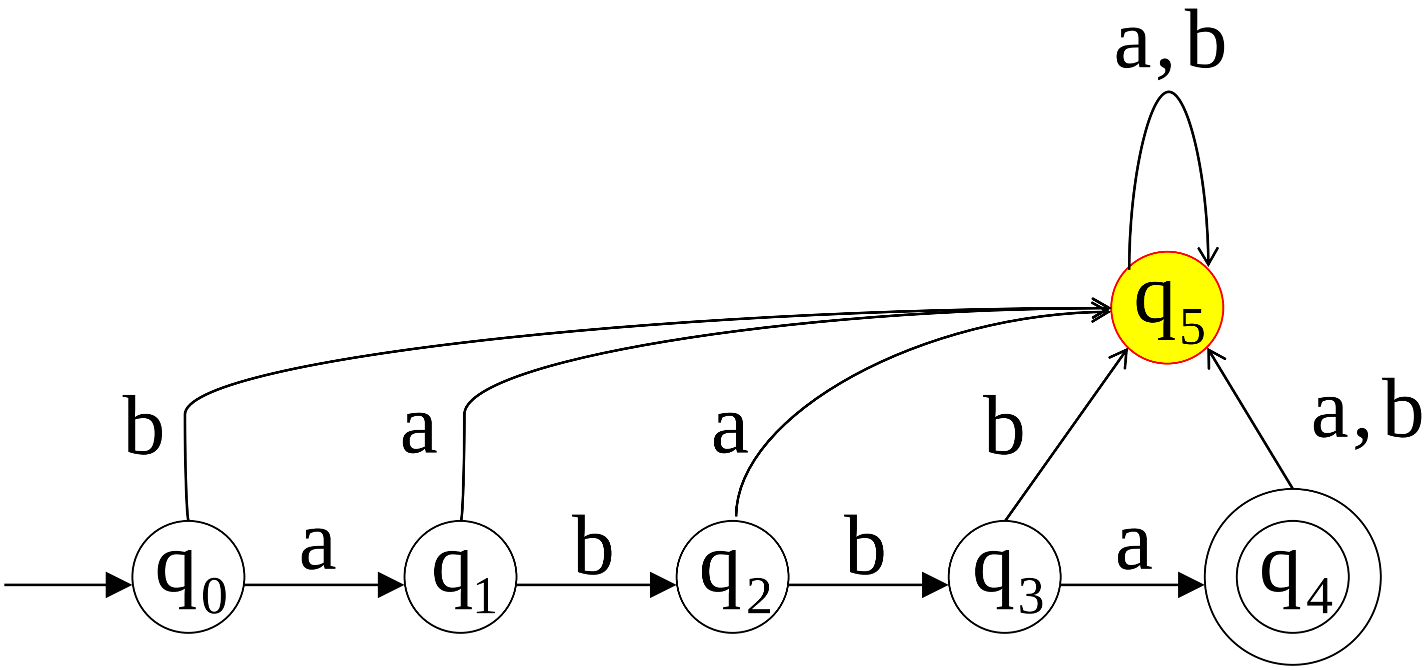


Rejection



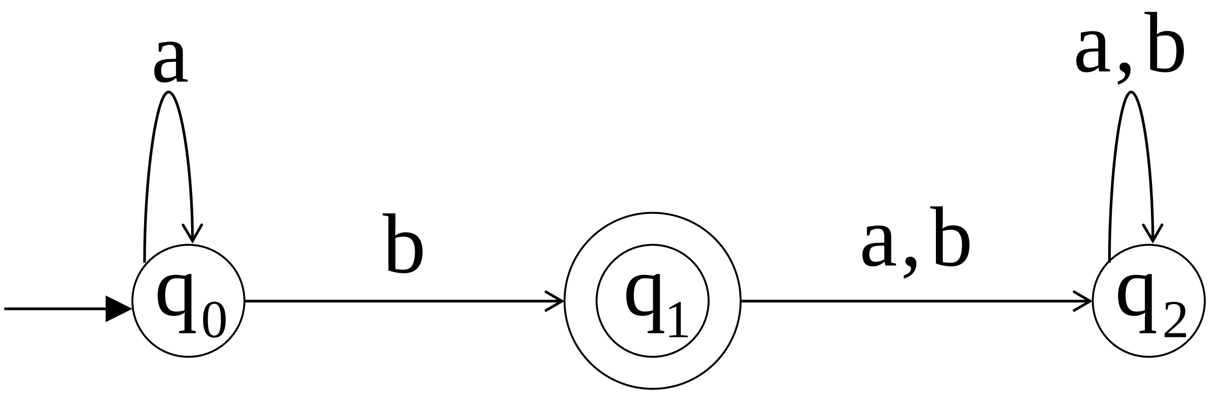
Rejection

Input String

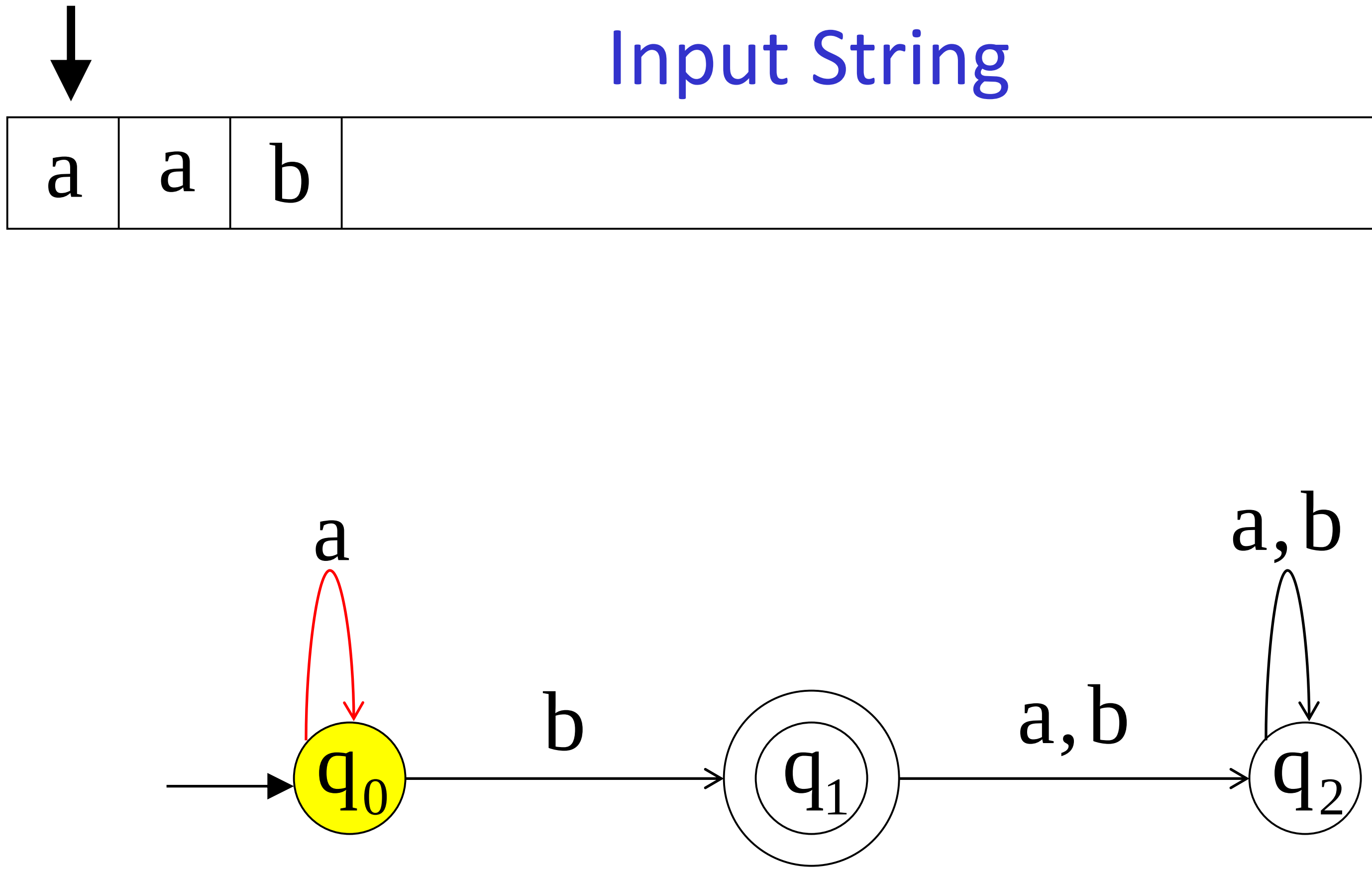


Output: "reject"

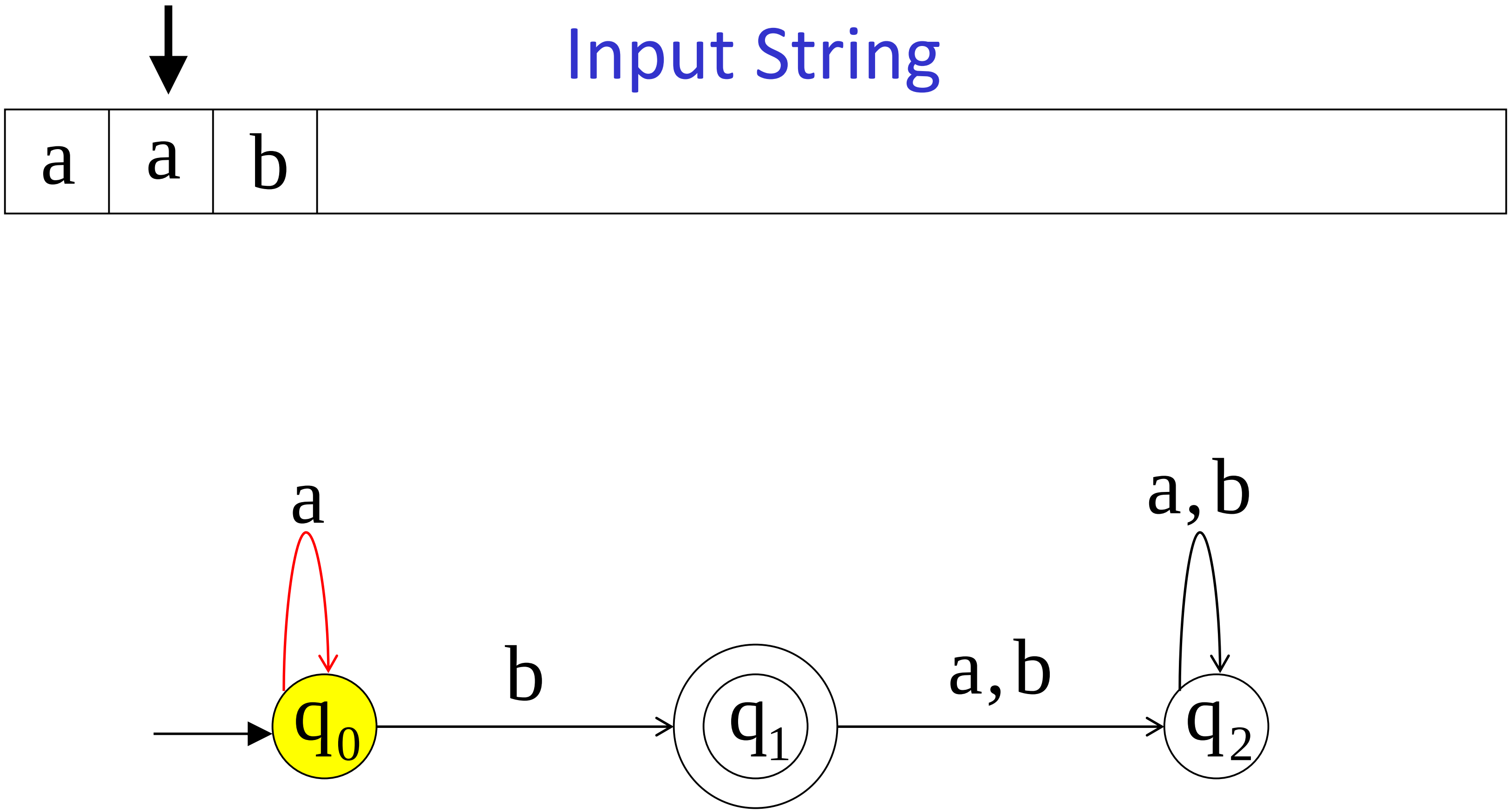
Another Example



Another Example

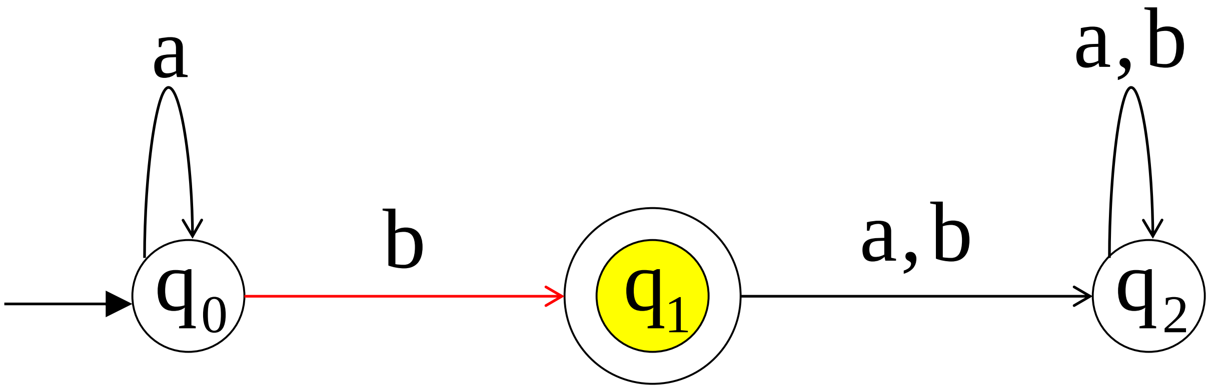


Another Example



Another Example

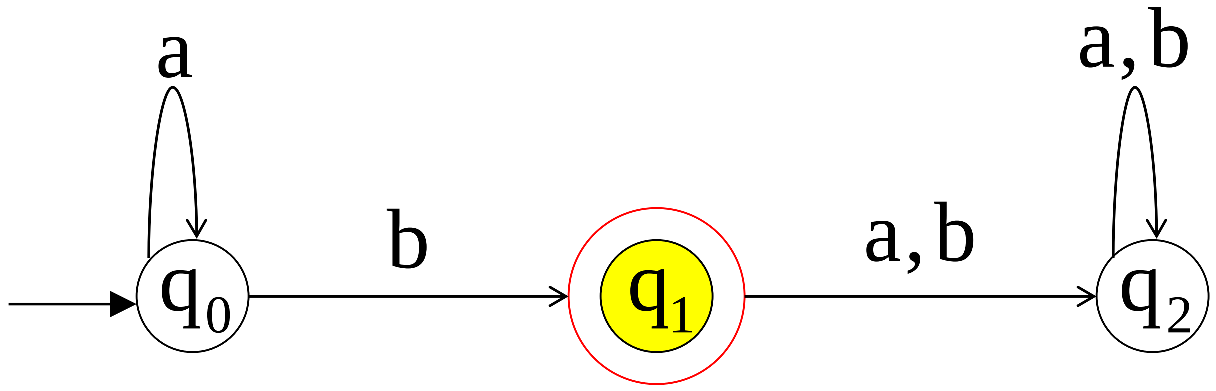
Input String



Another Example

Input String

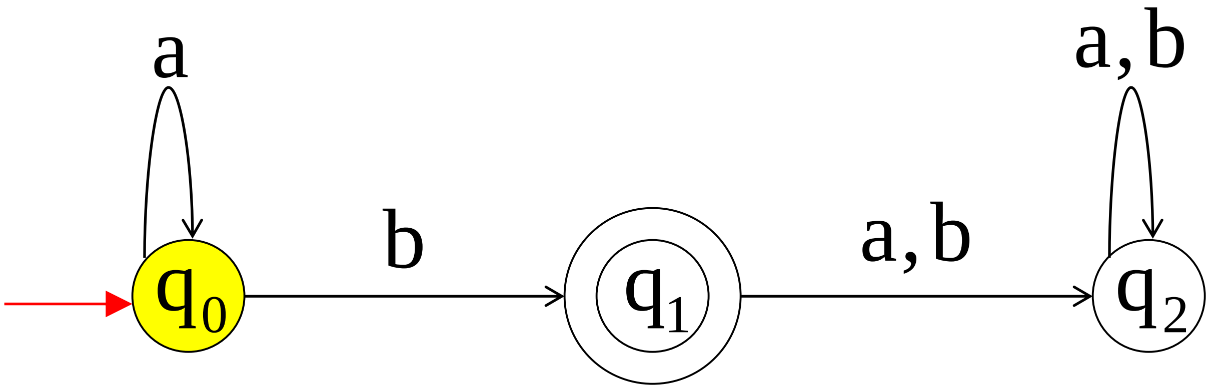
a	a	b	



Output: "accept"

Rejection

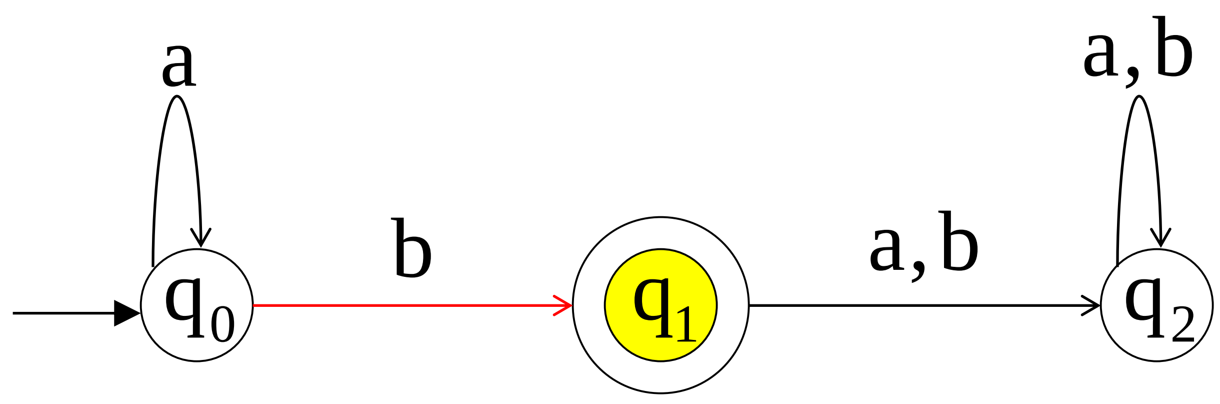
Input String



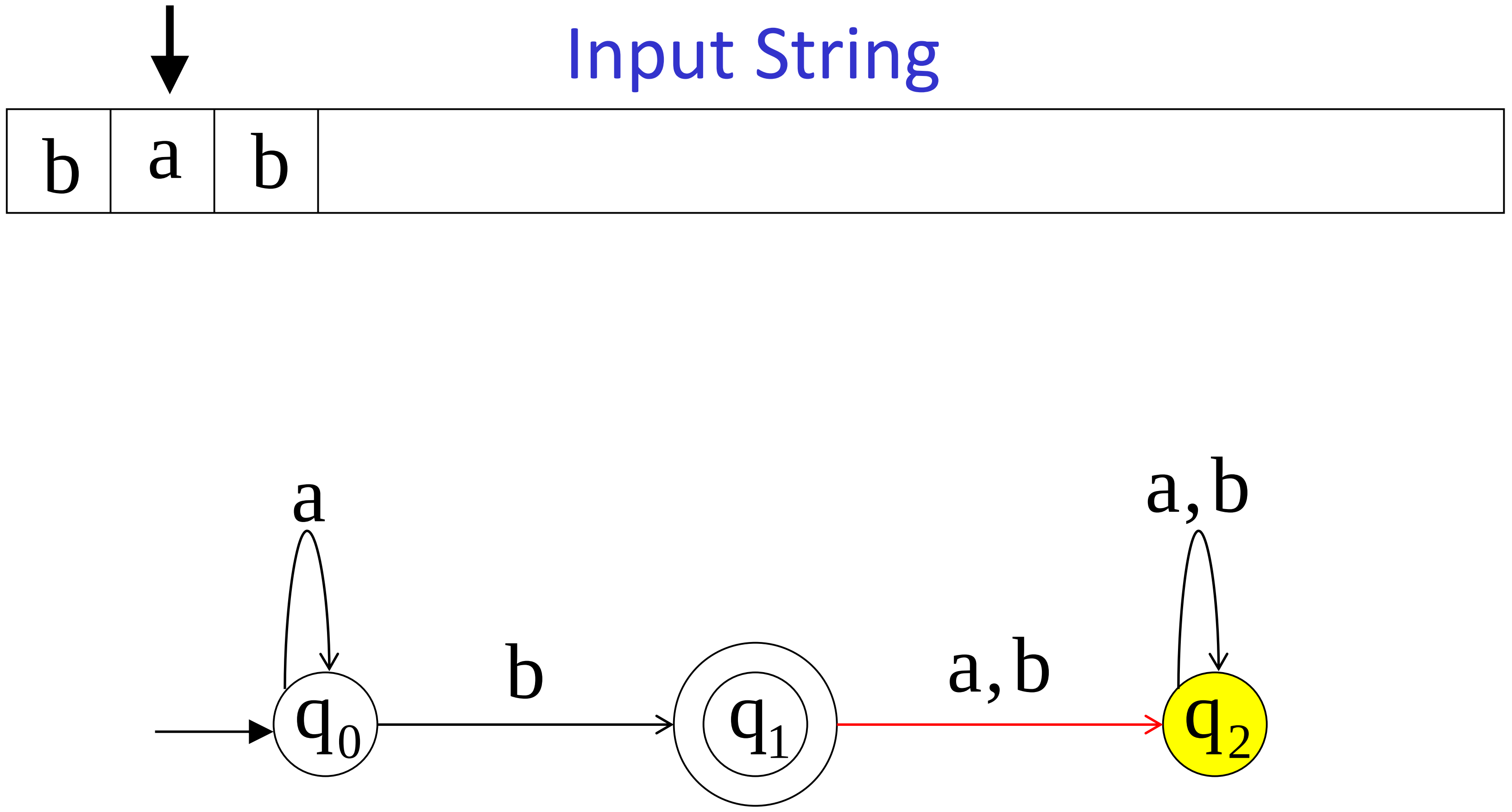
Rejection

Input String

b	a	b	



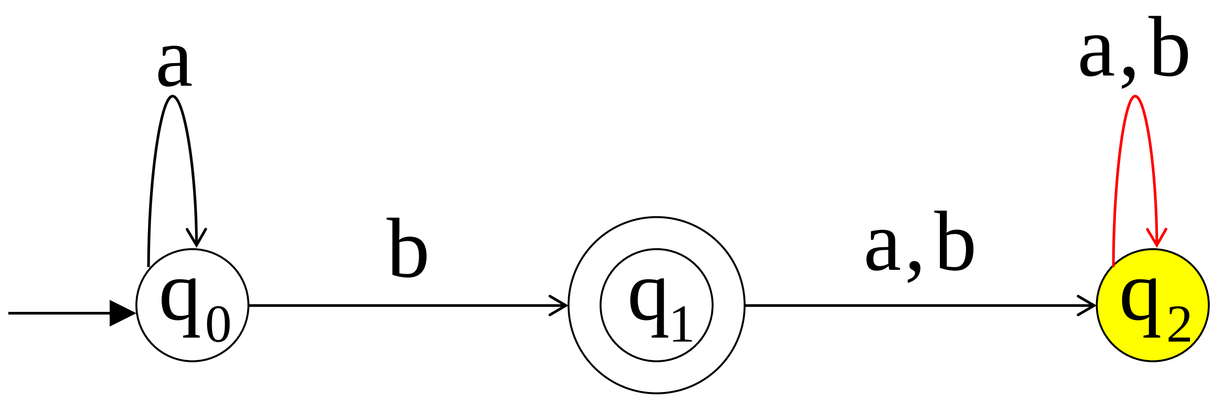
Rejection



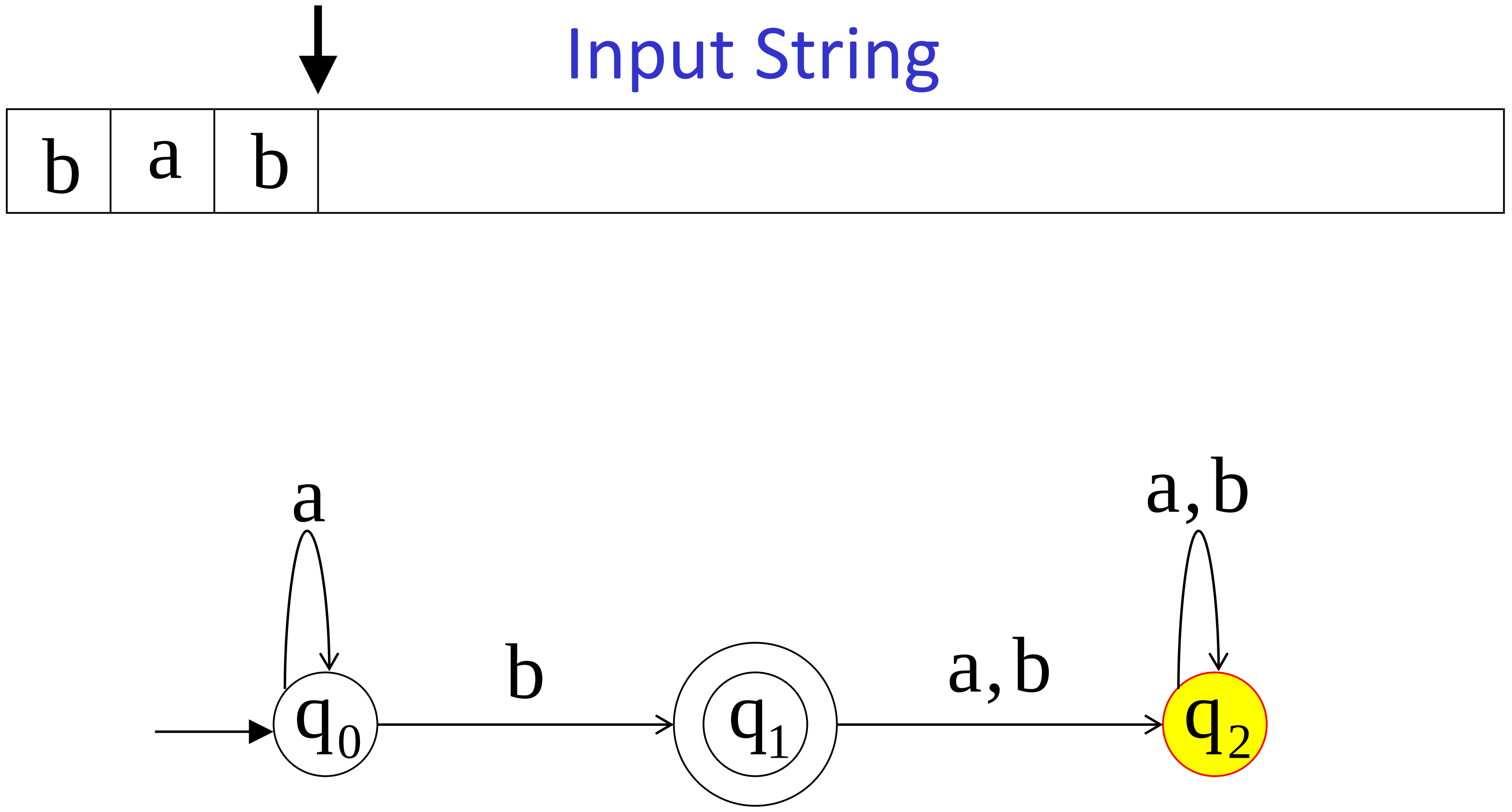
Rejection

Input String

b	a	b	



Rejection



Output: “reject”

Regular Languages

- A language L is **regular** if there is a DFA M such that

$$L = L(M).$$

- All regular languages form the **family of regular languages**.

Exercises

- For $\sum = \{0,1\}$, construct DFAs that accept the languages consisting of

1) all strings with **exactly one** 0.
2) all strings with **at least one** 0.
3) all strings with **no more than two** 0's.
4) all strings **starting with** 00.
5) all strings **ending with** 00.

Formalities

- **Deterministic Finite Accepter (DFA)**

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

Q : set of states

Σ : input alphabet

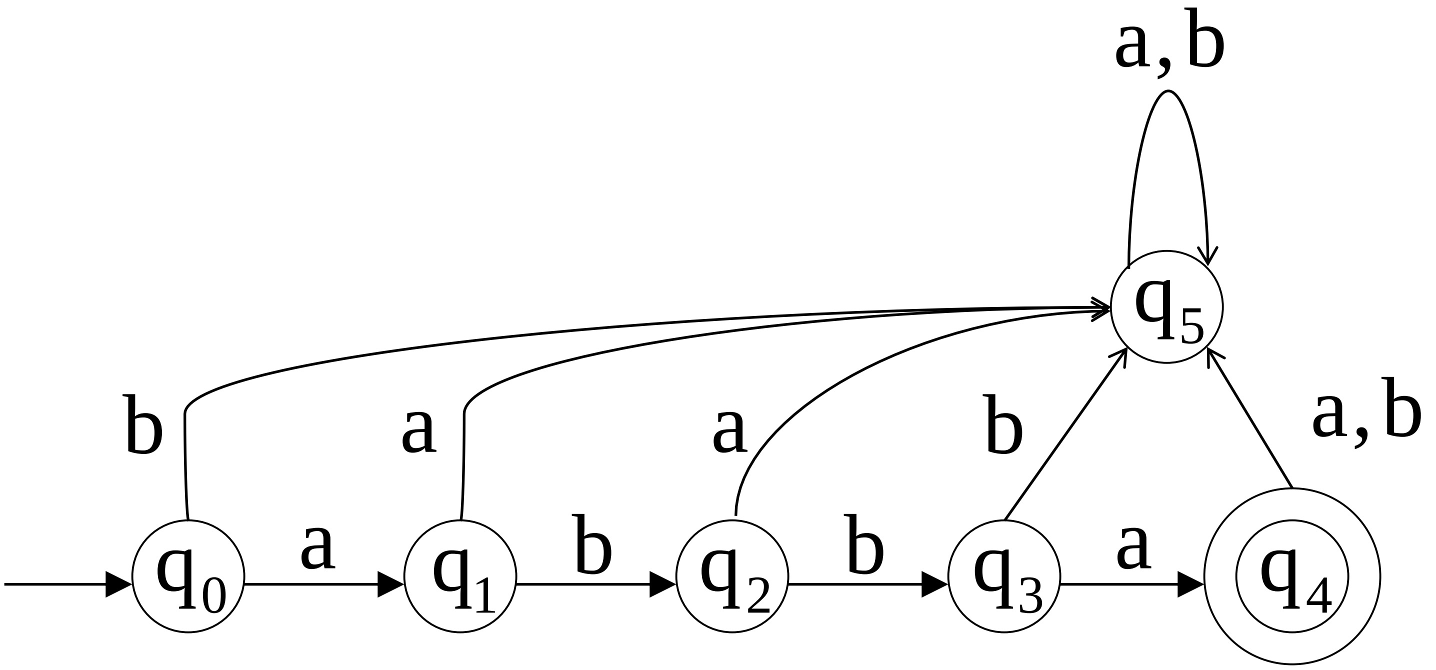
δ : transition function

q_0 : initial state

F : set of final states

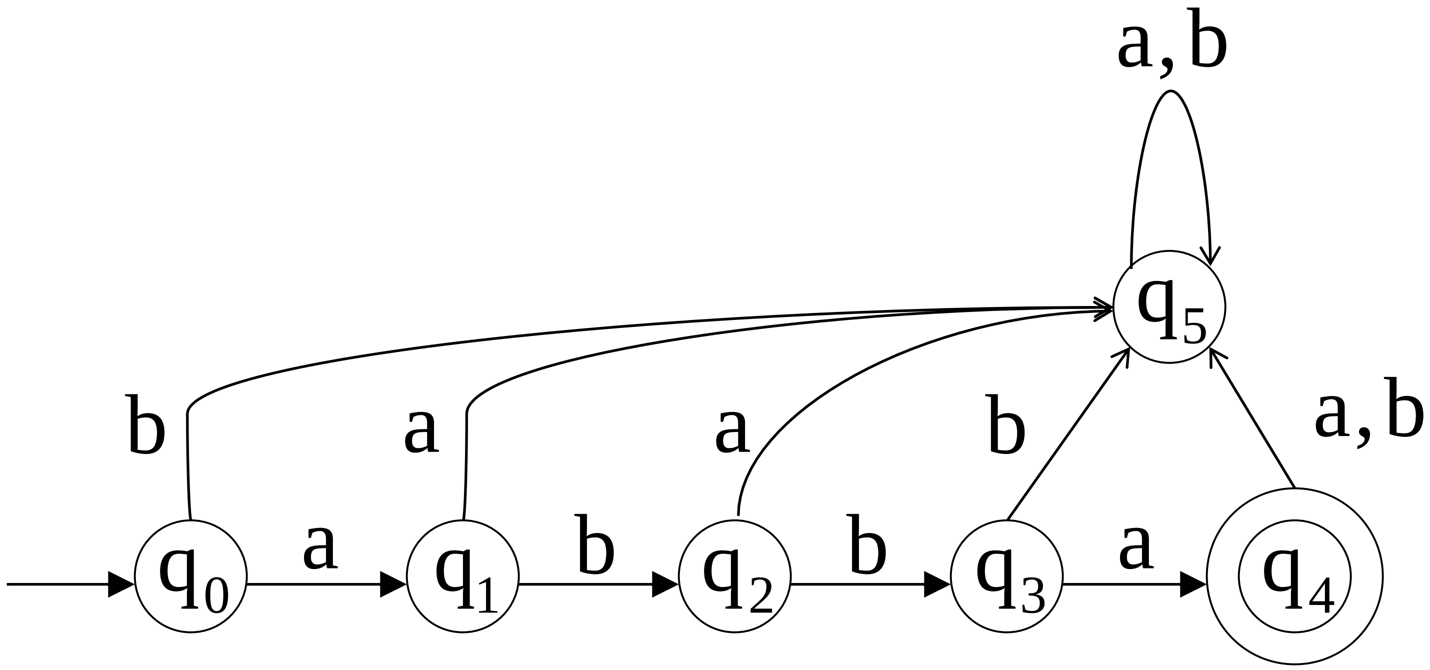
Input Alphabet Σ

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

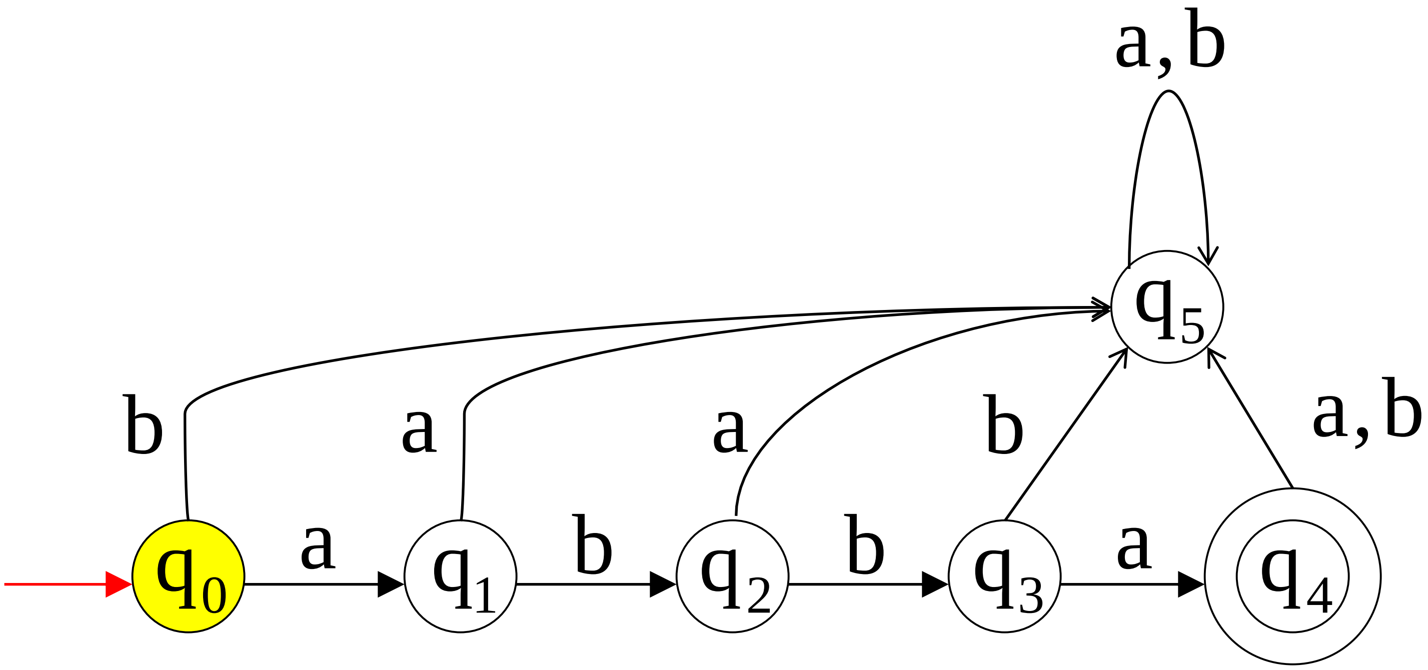


Set of States Q

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

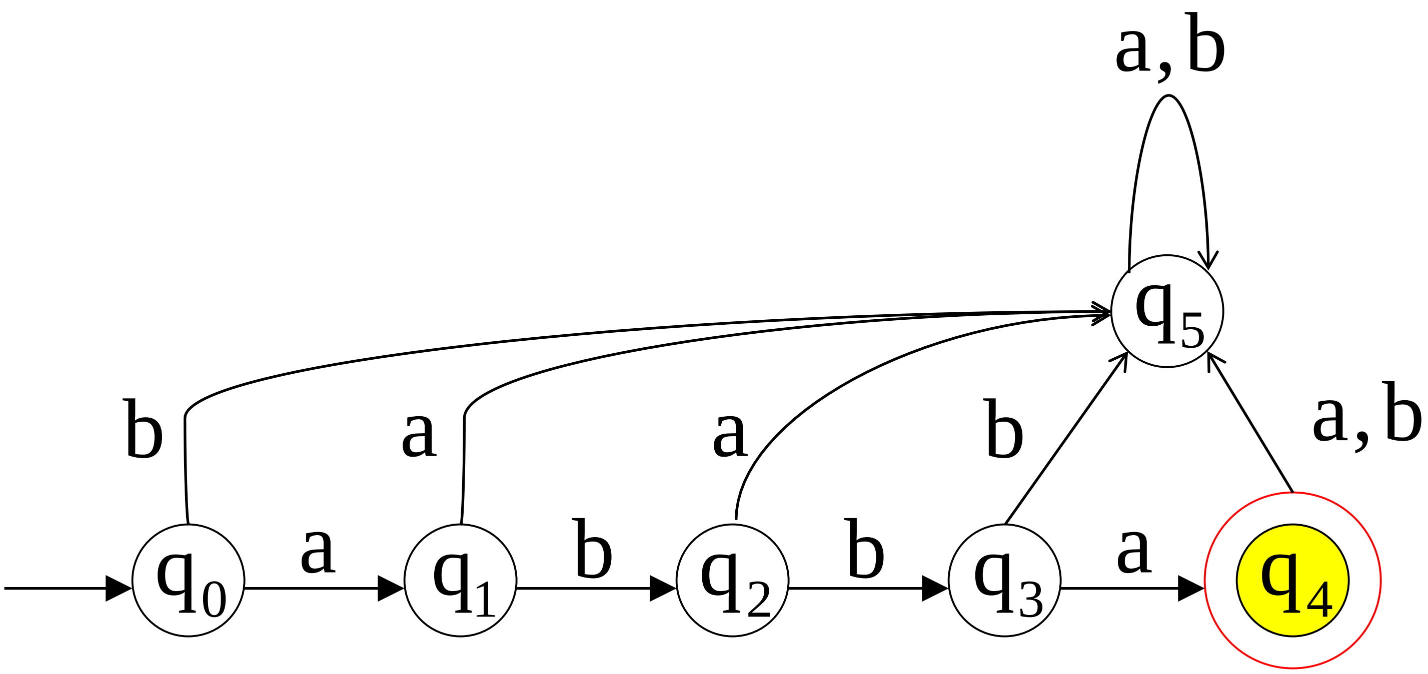


Initial State q_0



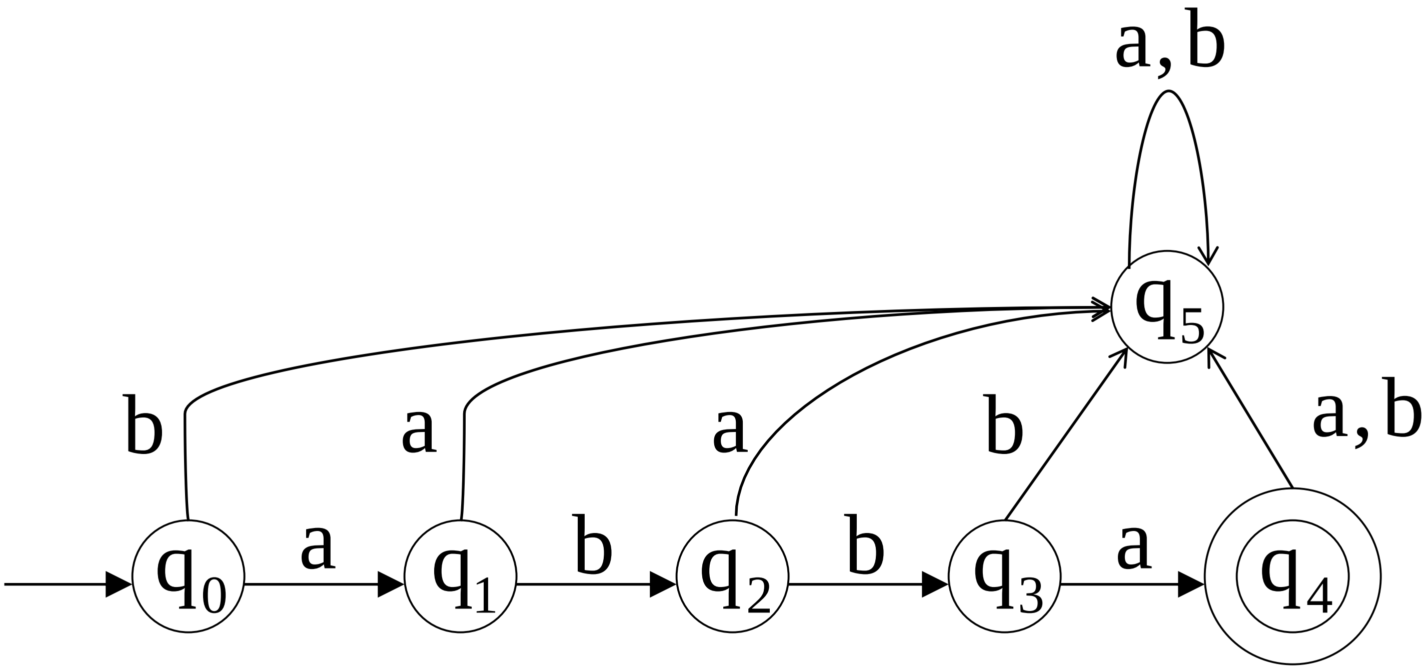
Set of Final States F

$$F = \{q_4\}$$



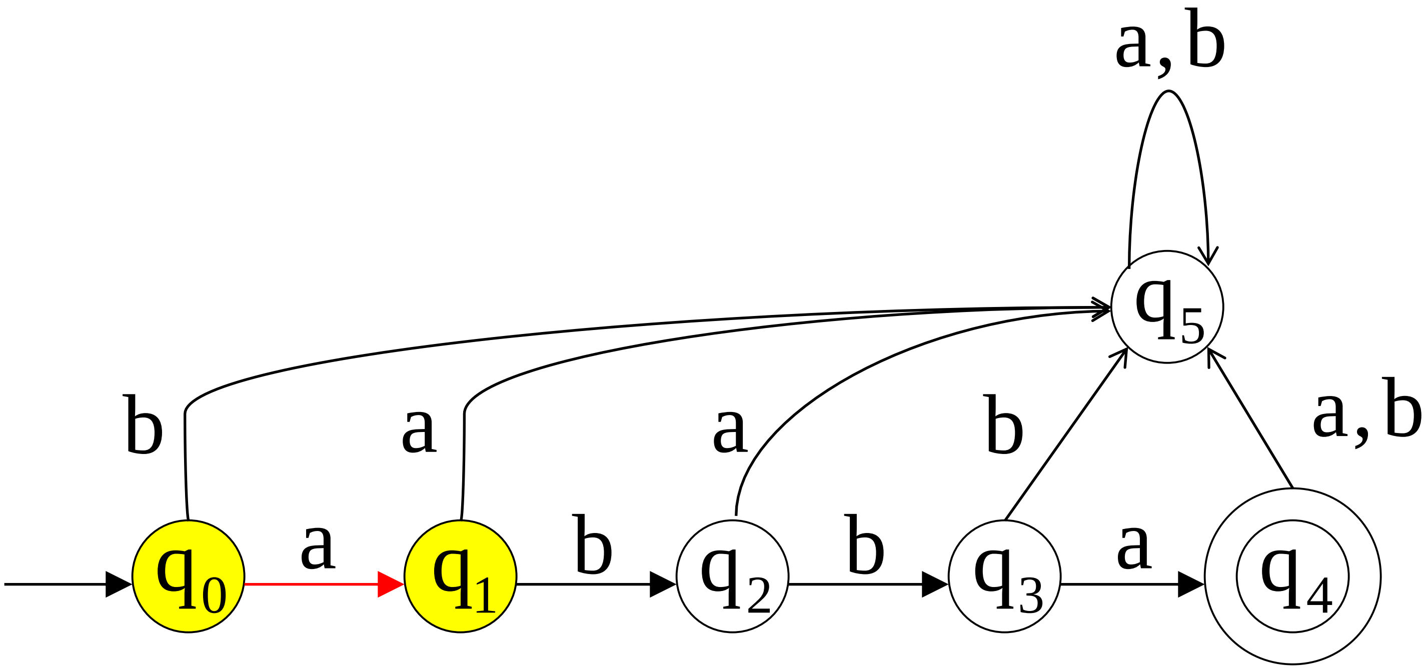
Transition Function δ

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



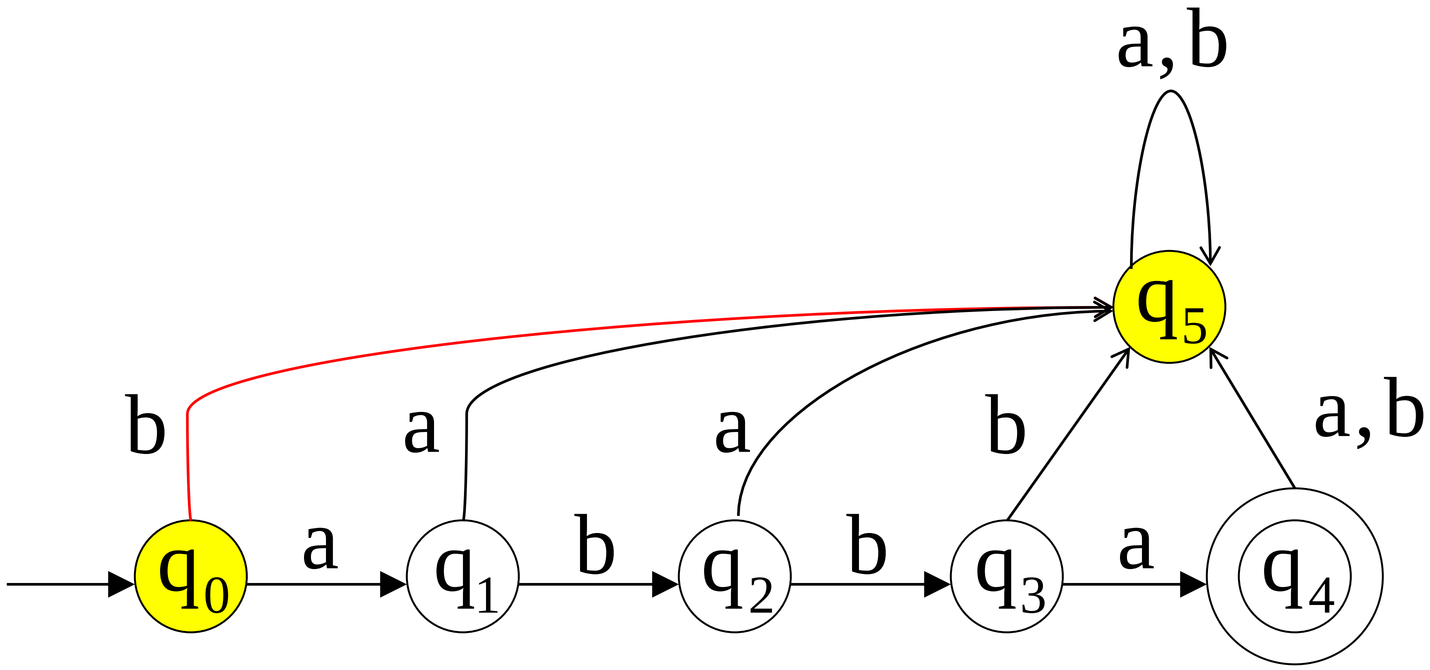
Transition Function δ

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



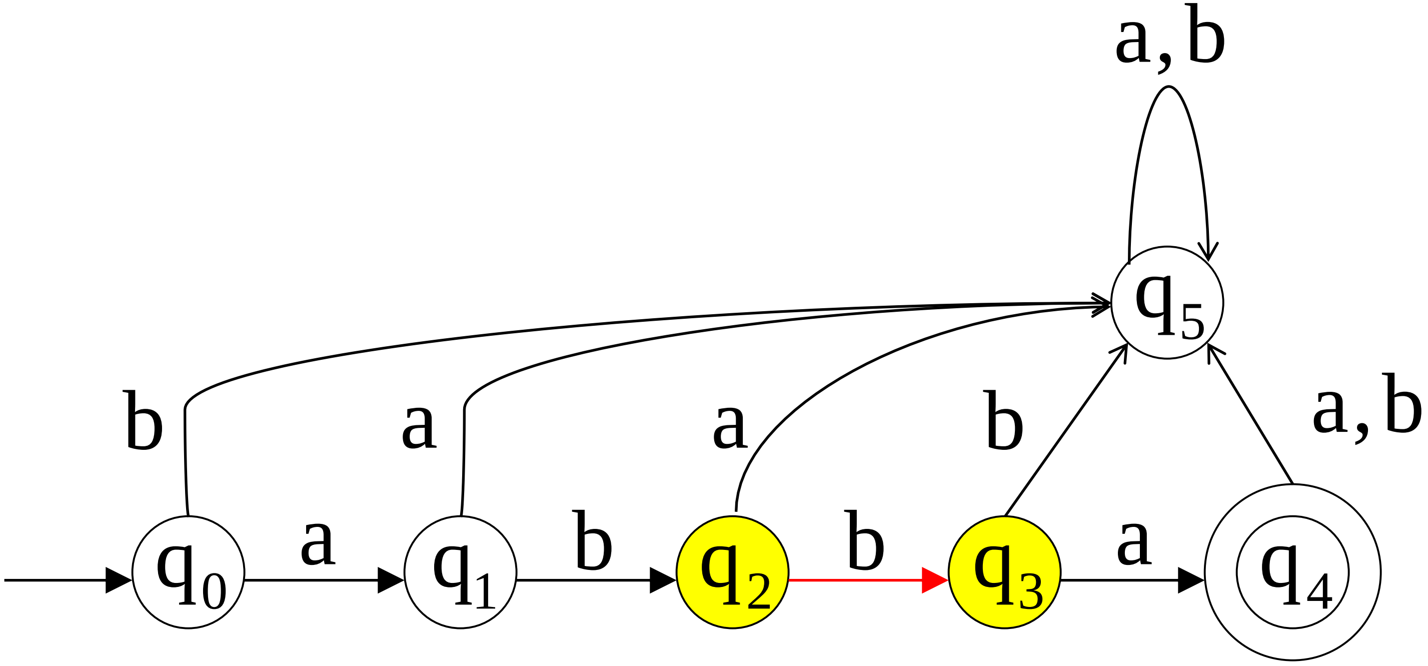
Transition Function δ

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



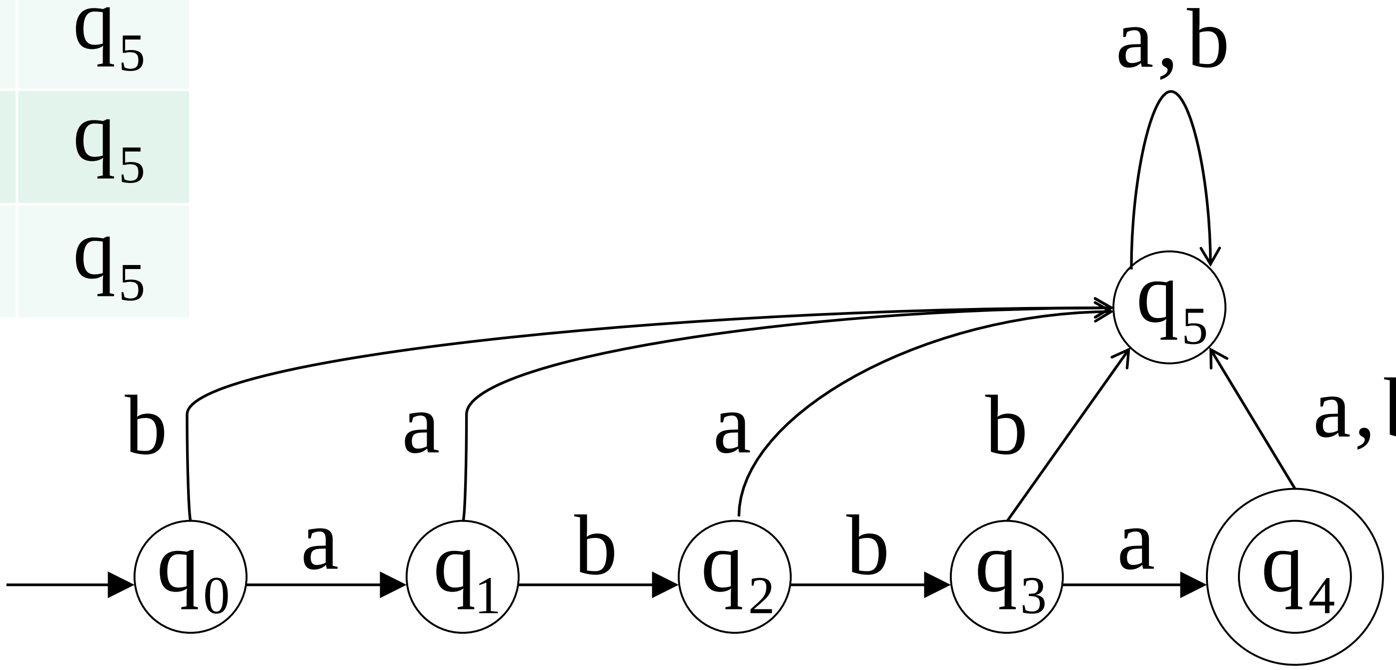
Transition Function δ

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



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



Languages Accepted by DFAs

- Take DFA M

- **Definition:**

  The language $L(M)$ contains

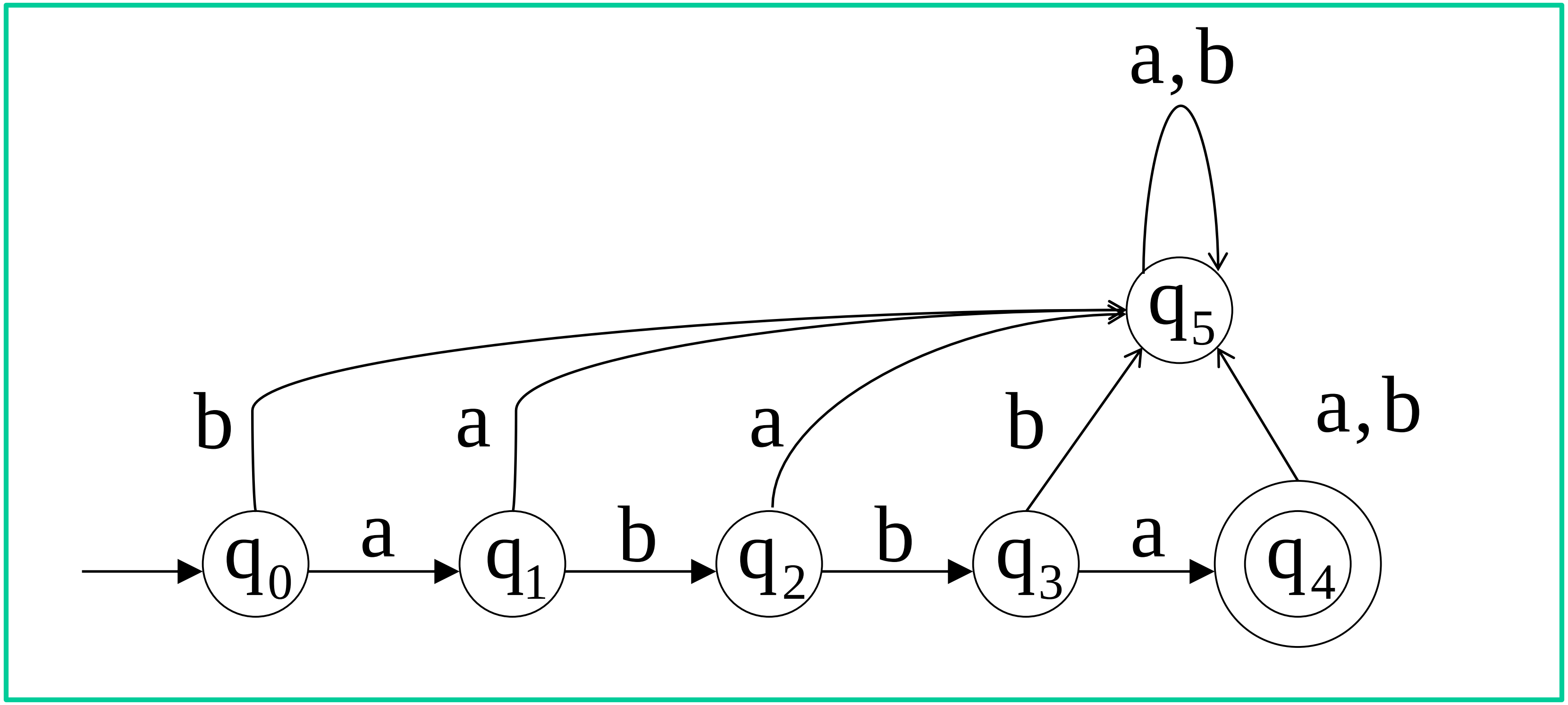
  all input strings accepted by M

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

Example

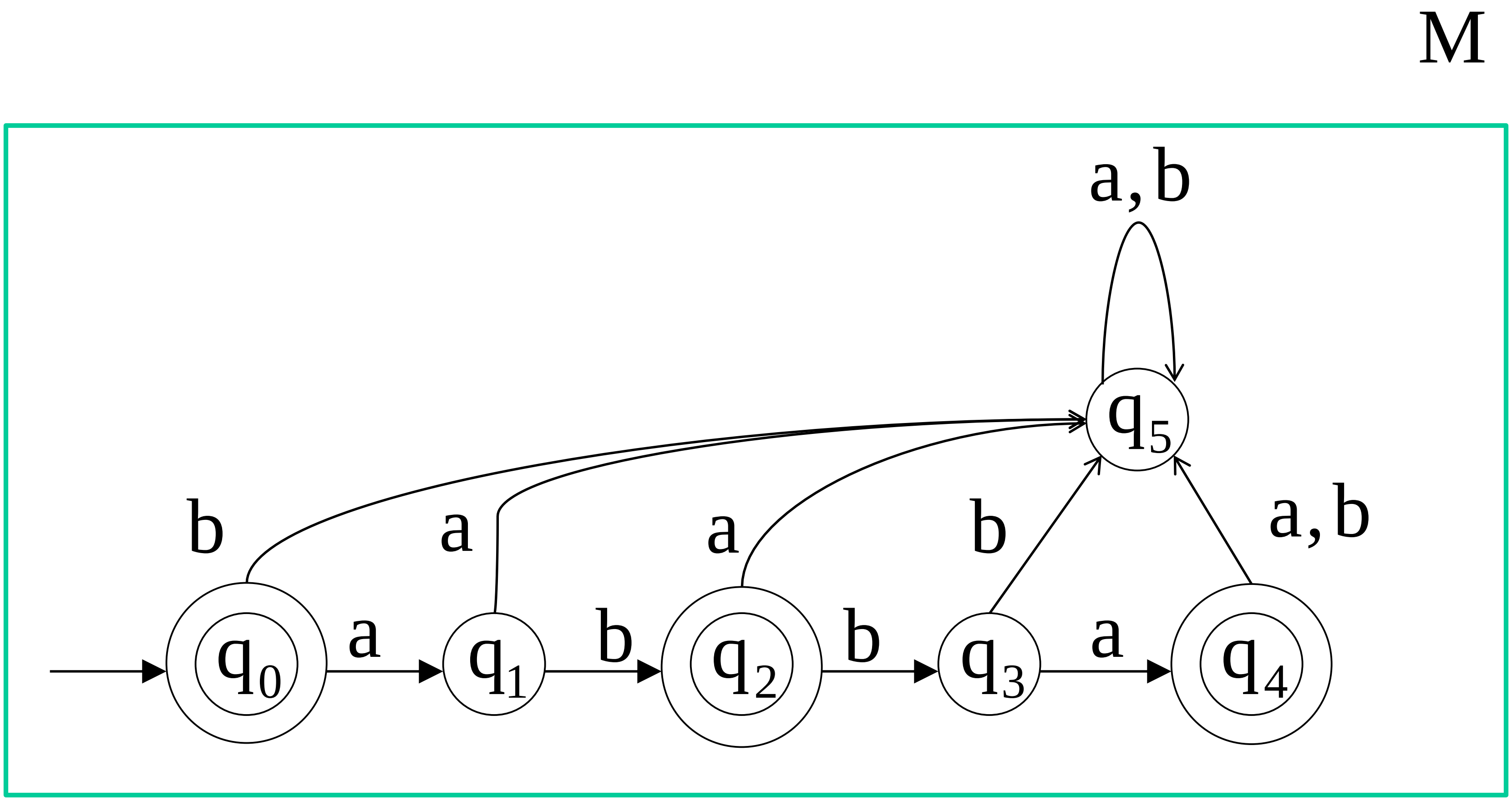
$$L(M) = \{abba\}$$

M



accept

Example

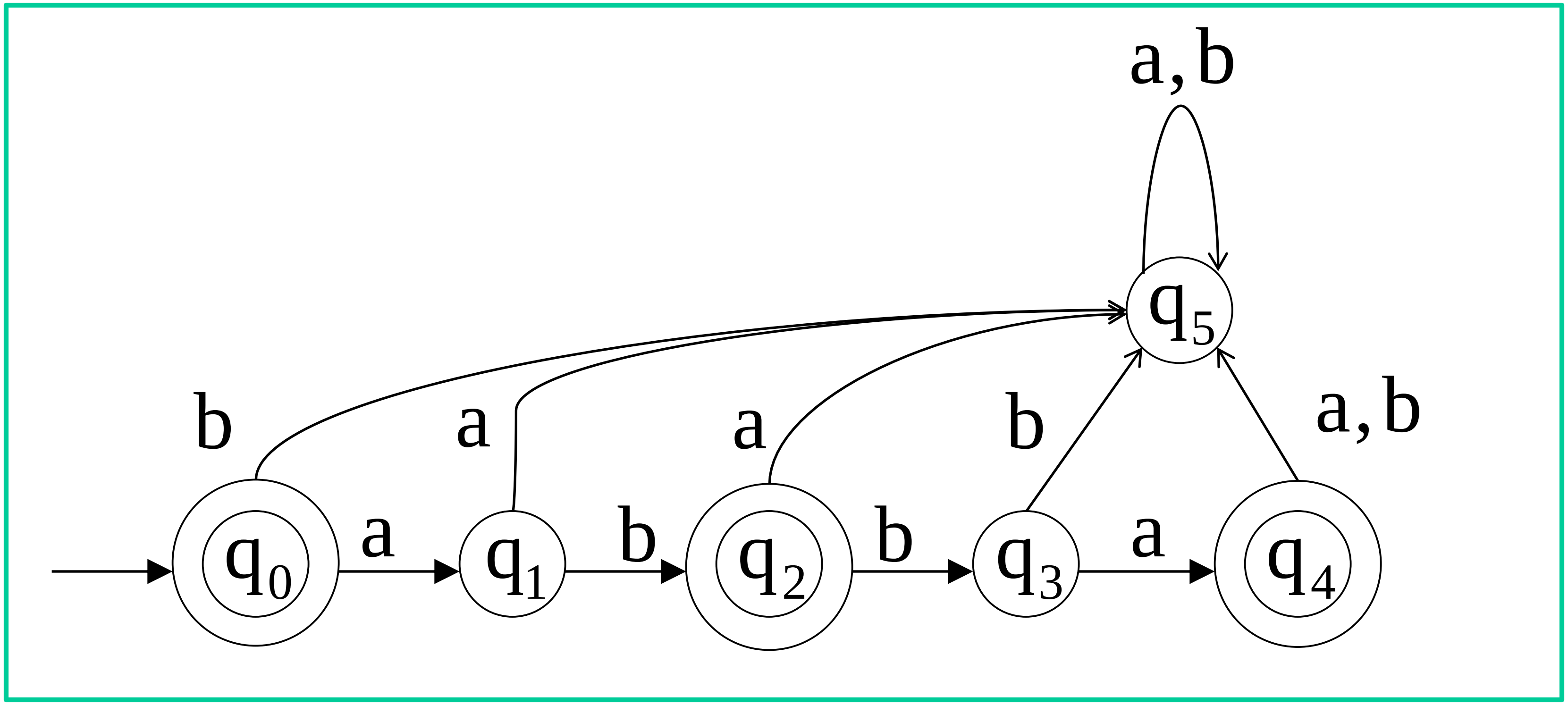


accept accept accept

Example

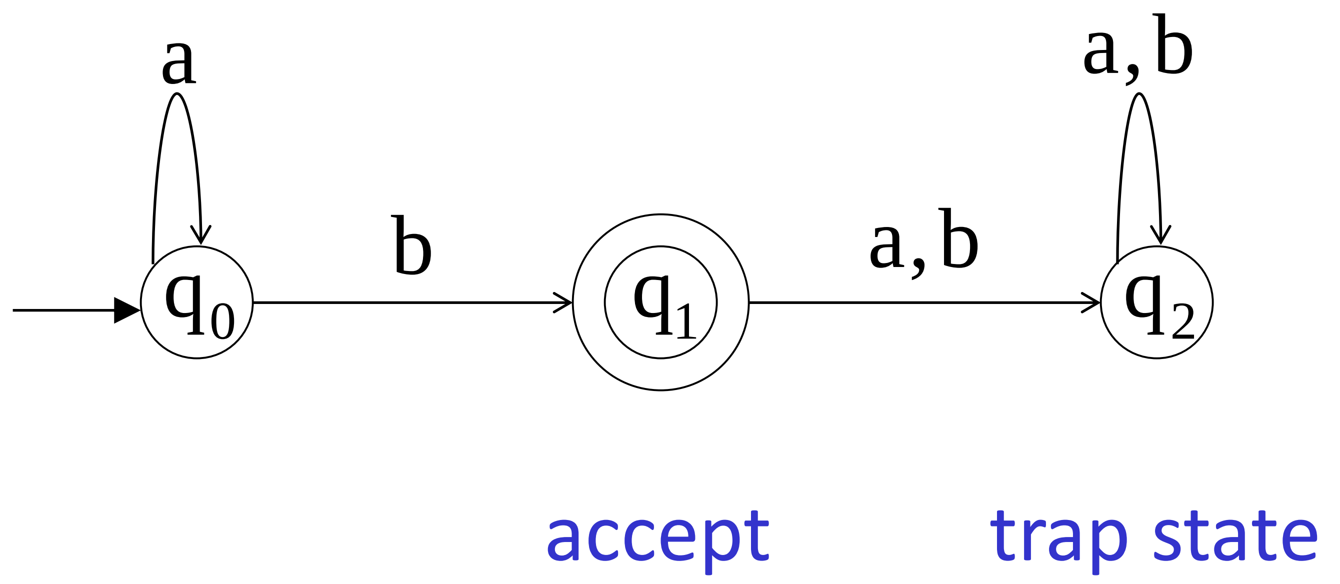
$$L(M) = \{\lambda, ab, abba\}$$

M



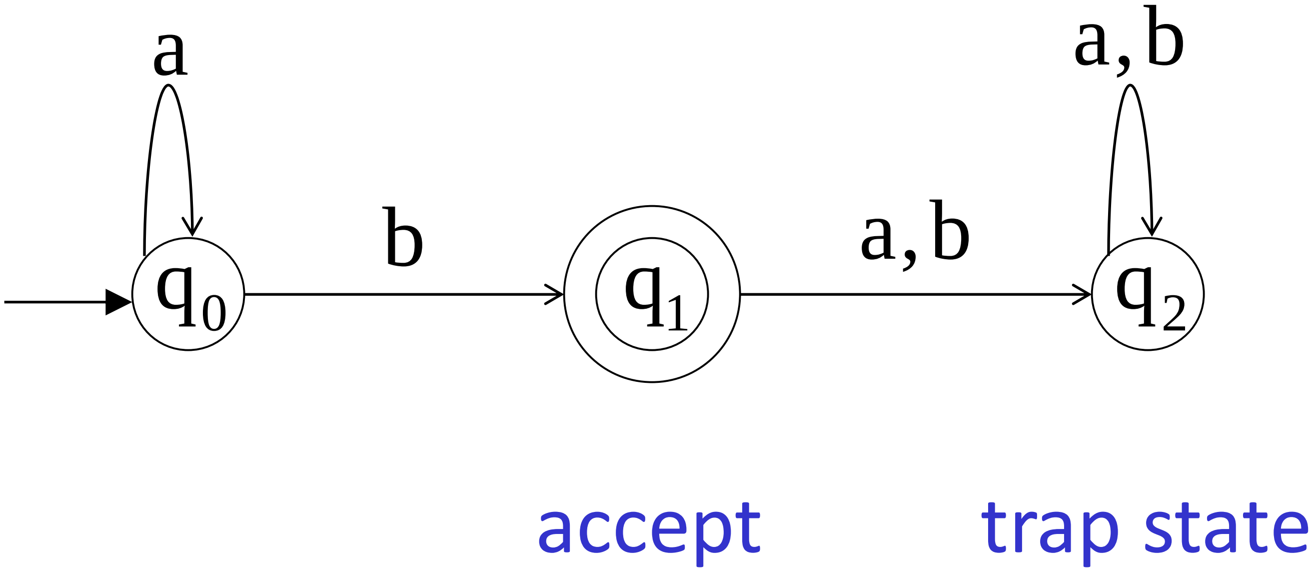
accept accept accept

More Examples

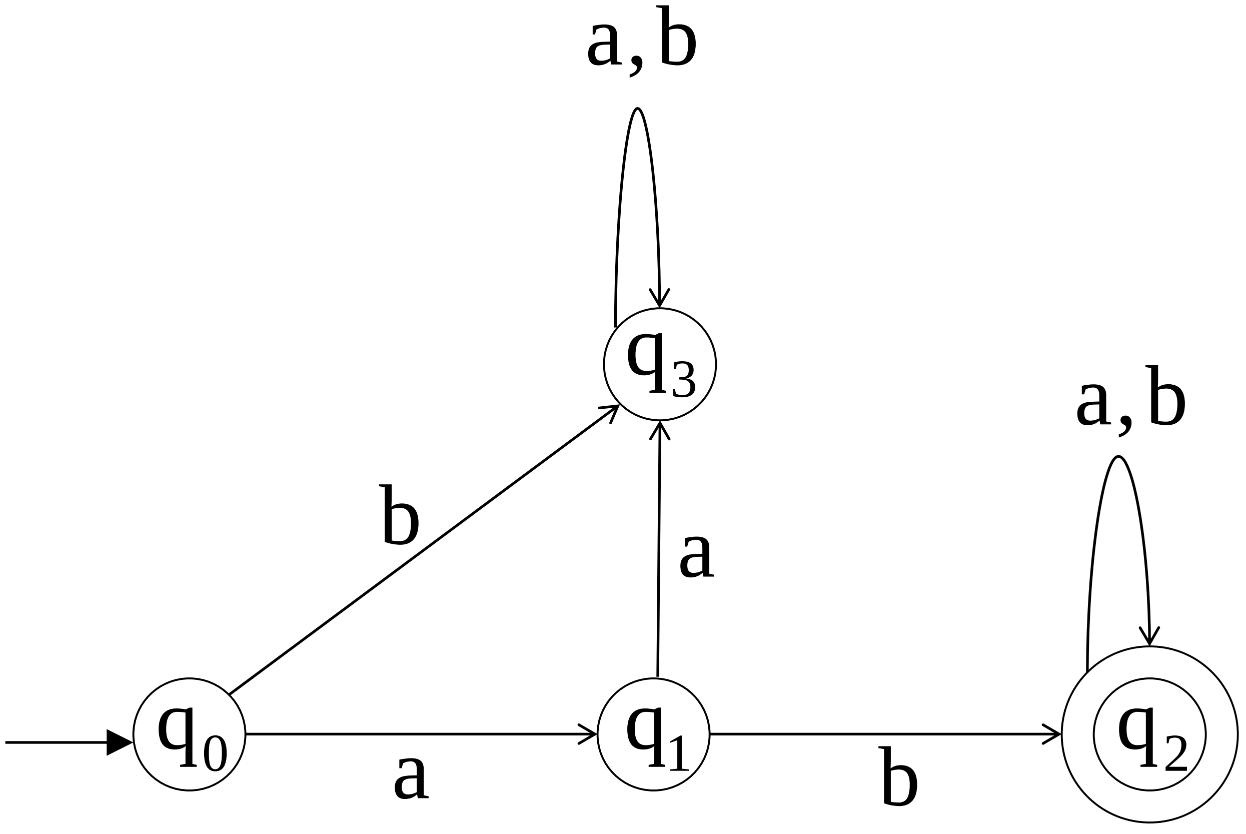


More Examples

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

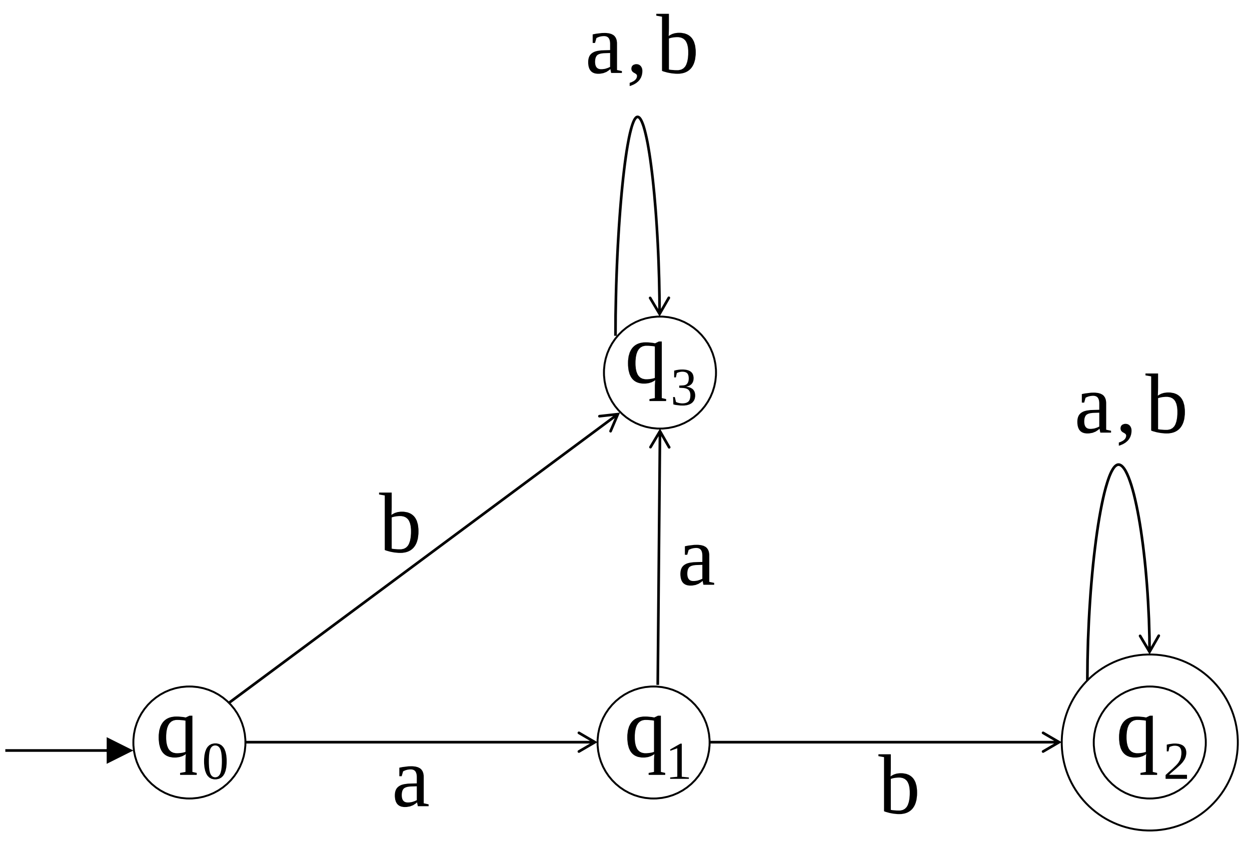


More Examples



More Examples

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



Regular Languages

- A language L is **regular** if there is a DFA M such that

$$L = L(M).$$

- All regular languages form the **family of regular languages**.

Exercises

- For $\Sigma = \{0,1\}$, construct DFAs that accept the languages consisting of

1) all strings with **exactly one** 0.
2) all strings with **at least one** 0.
3) all strings with **no more than two** 0's.
4) all strings **starting with** 00.
5) all strings **ending with** 00.