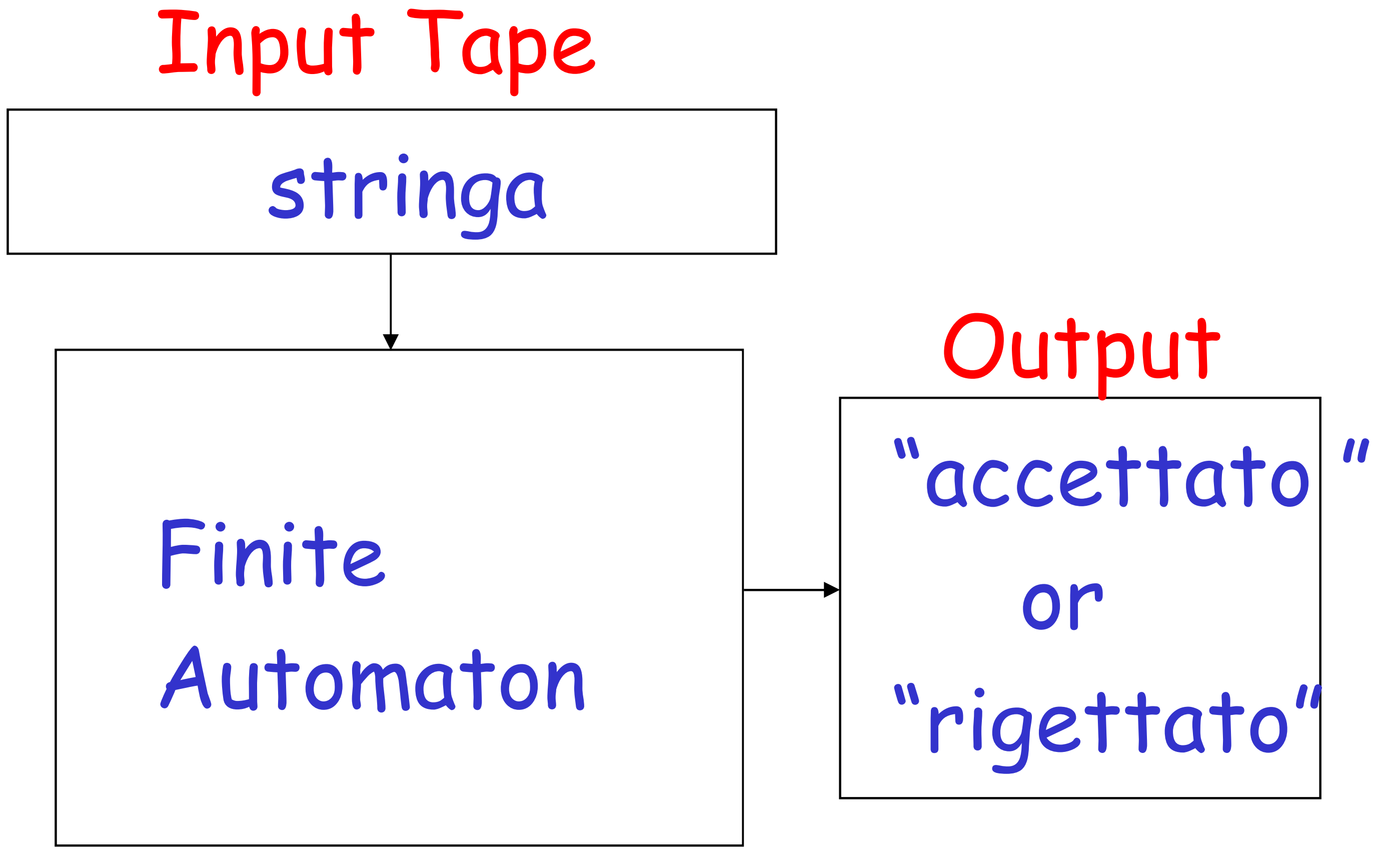
Deterministic

Finite Automata

E linguaggi regolari

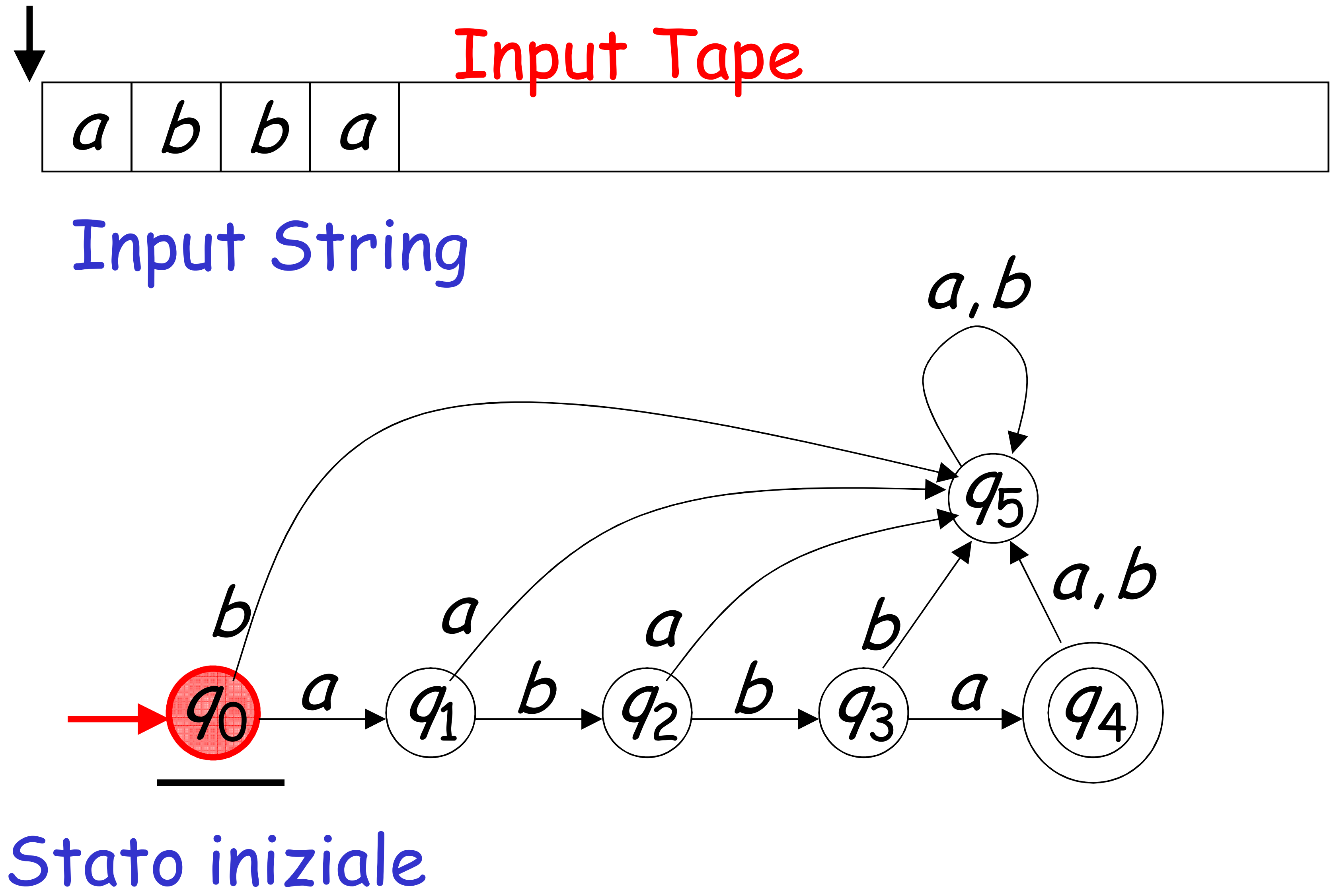
Simulatore http://www.jflap.org/

Deterministic Finite Automaton (DFA)

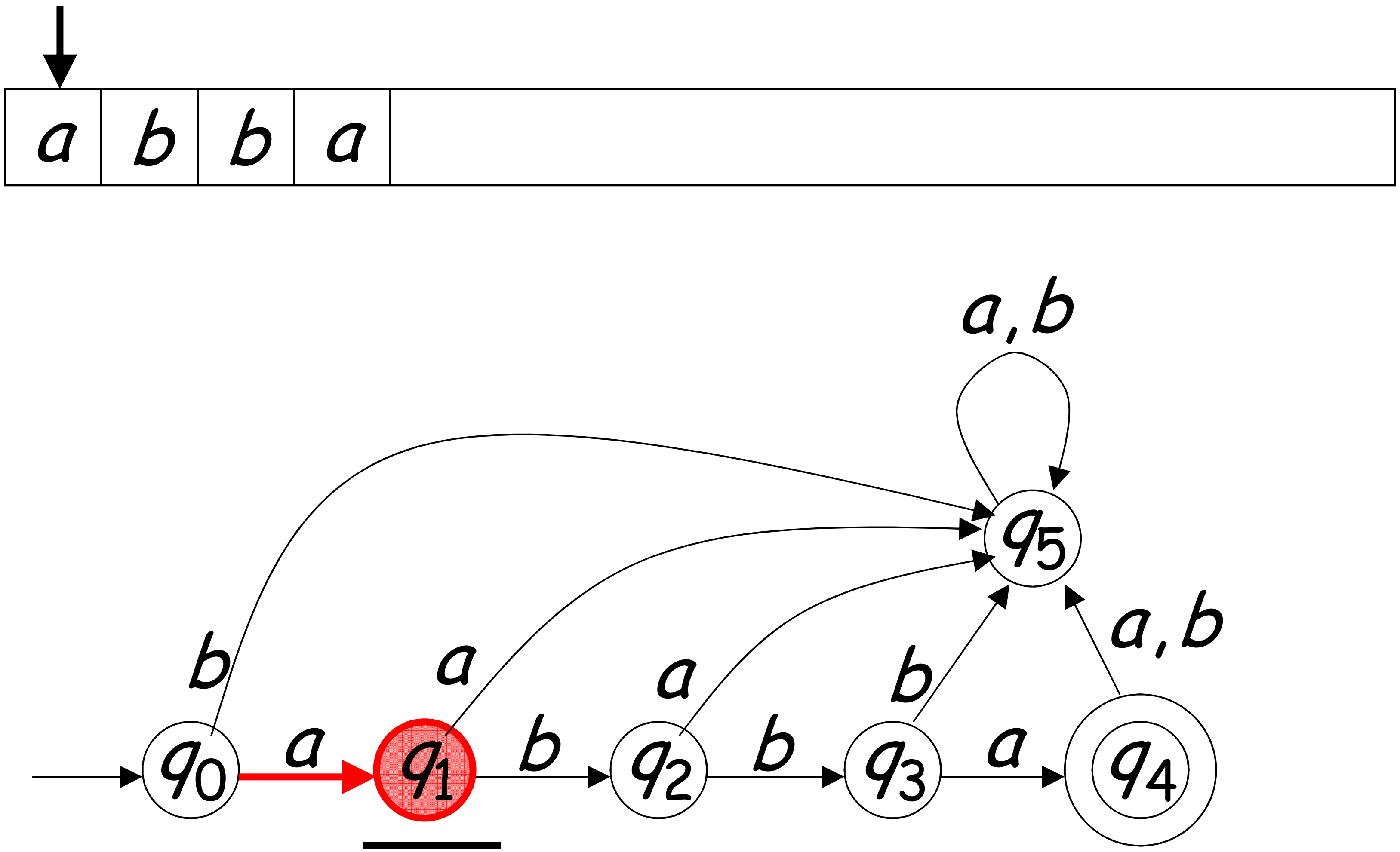


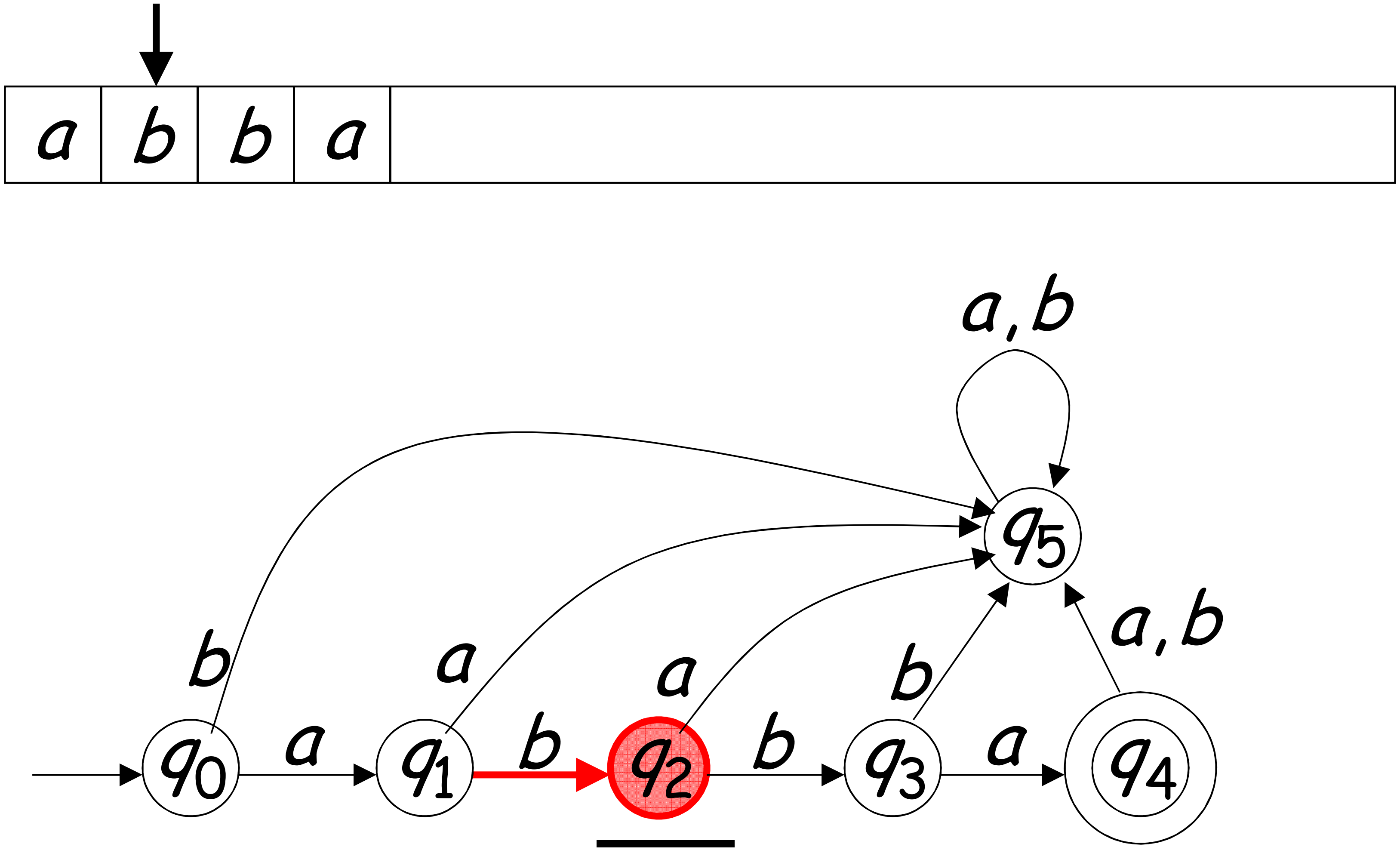
Configurazione iniziale

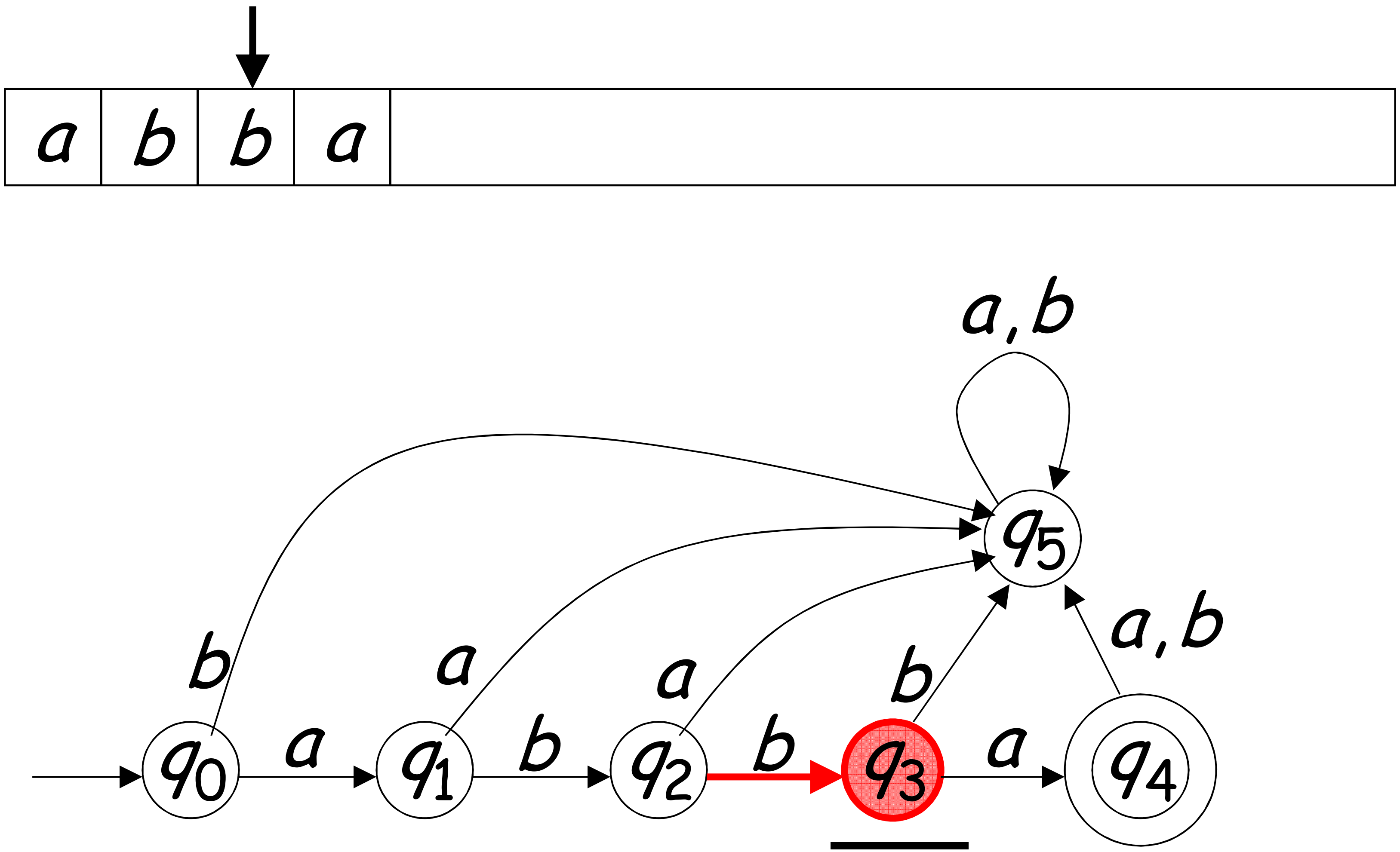
testa



Analizzare l'Input

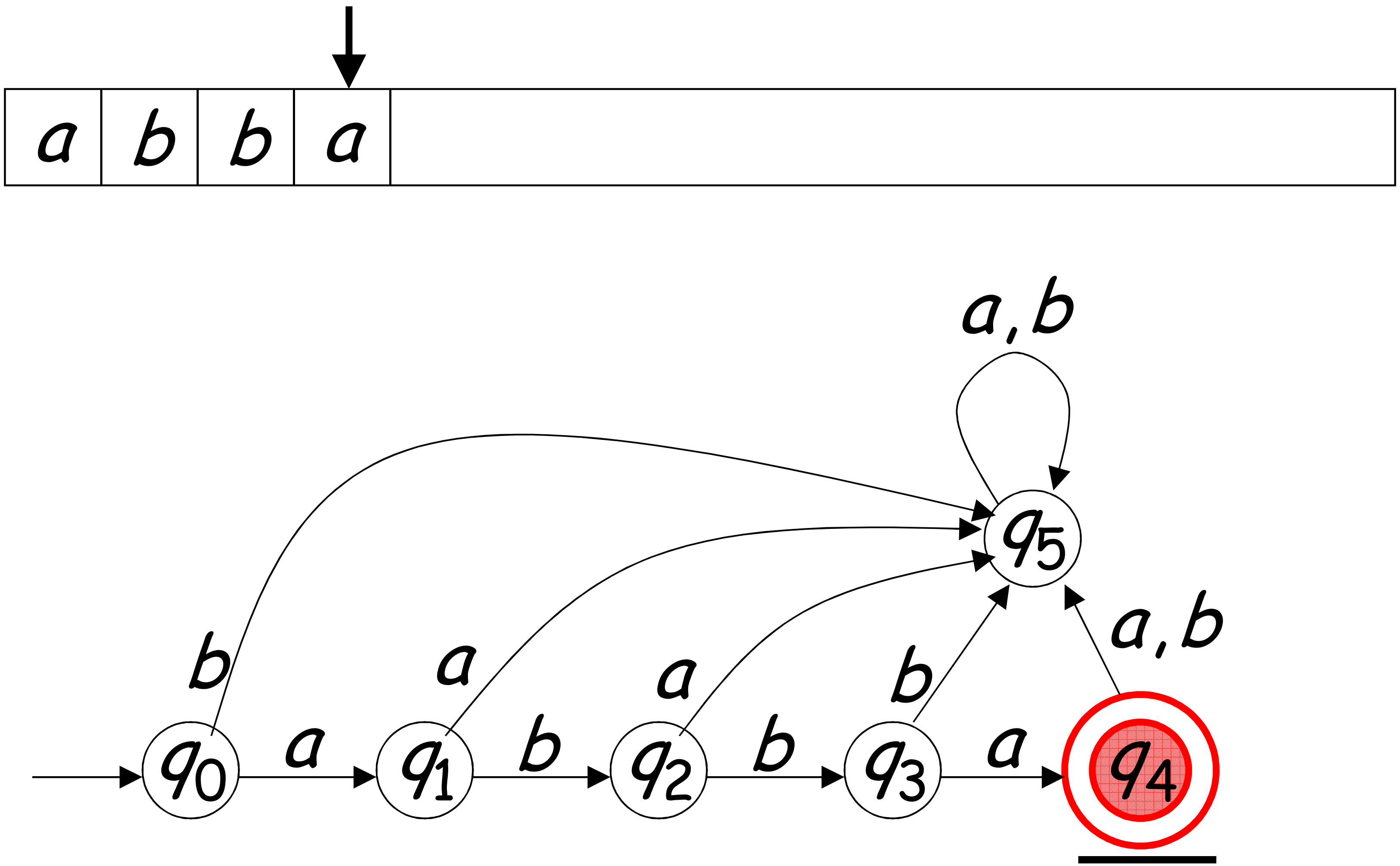



a
b
b
a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4



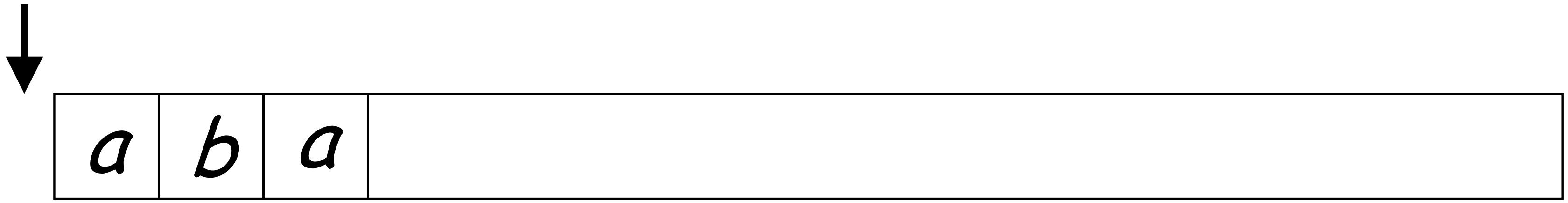
a
b
b
a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4

Input finito

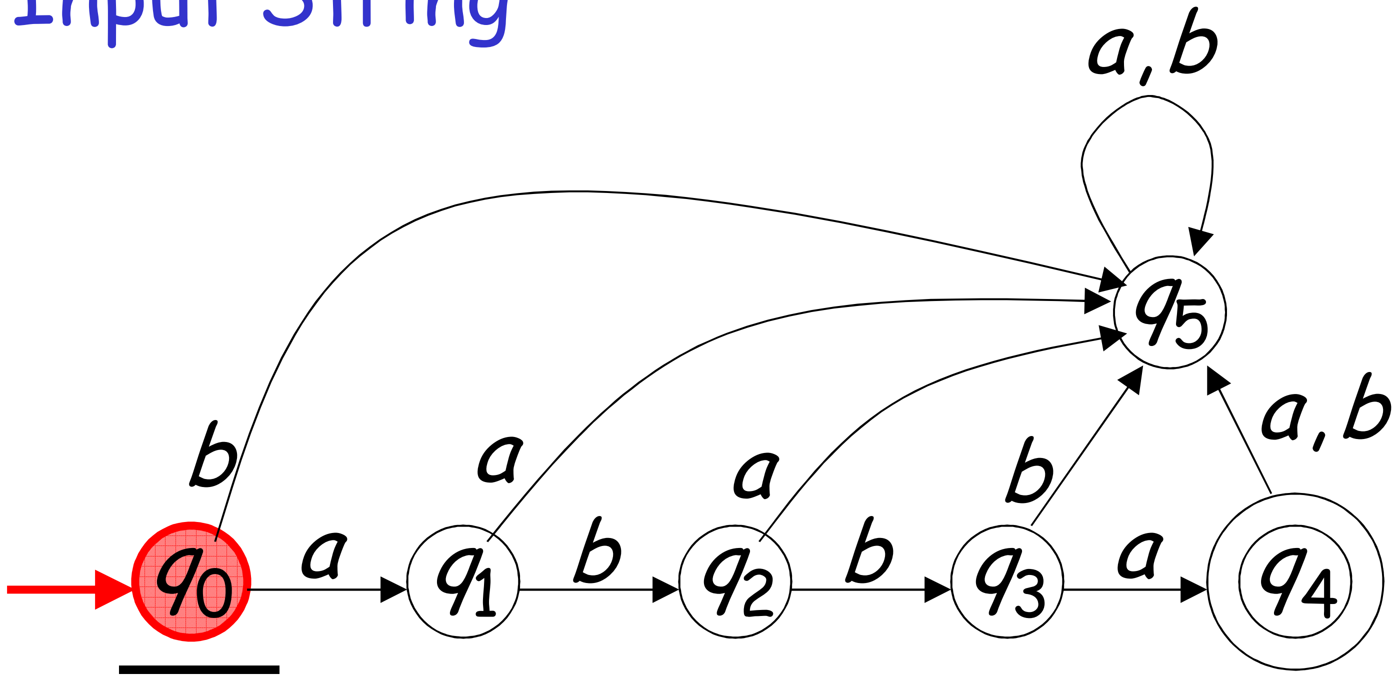


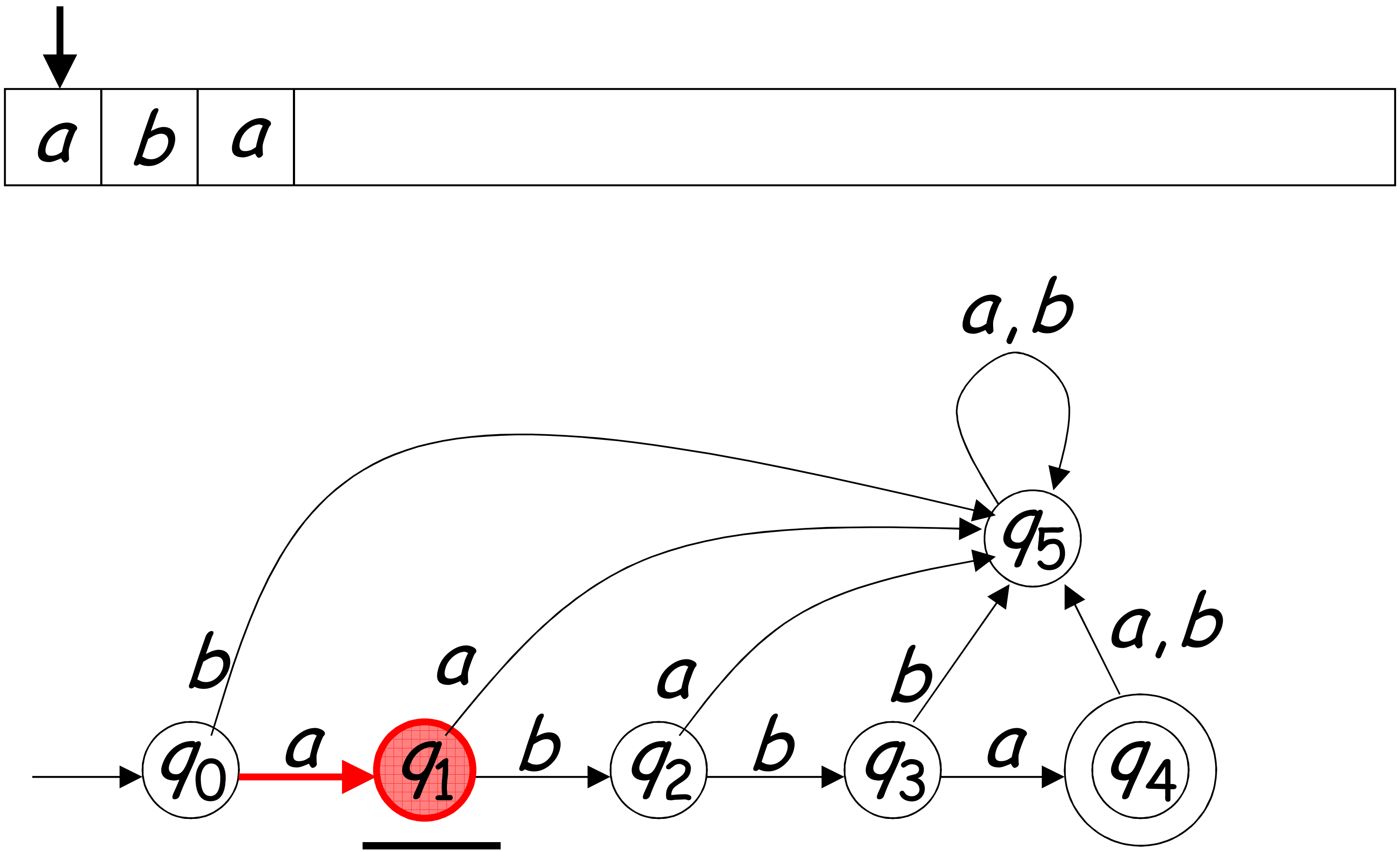
accettato

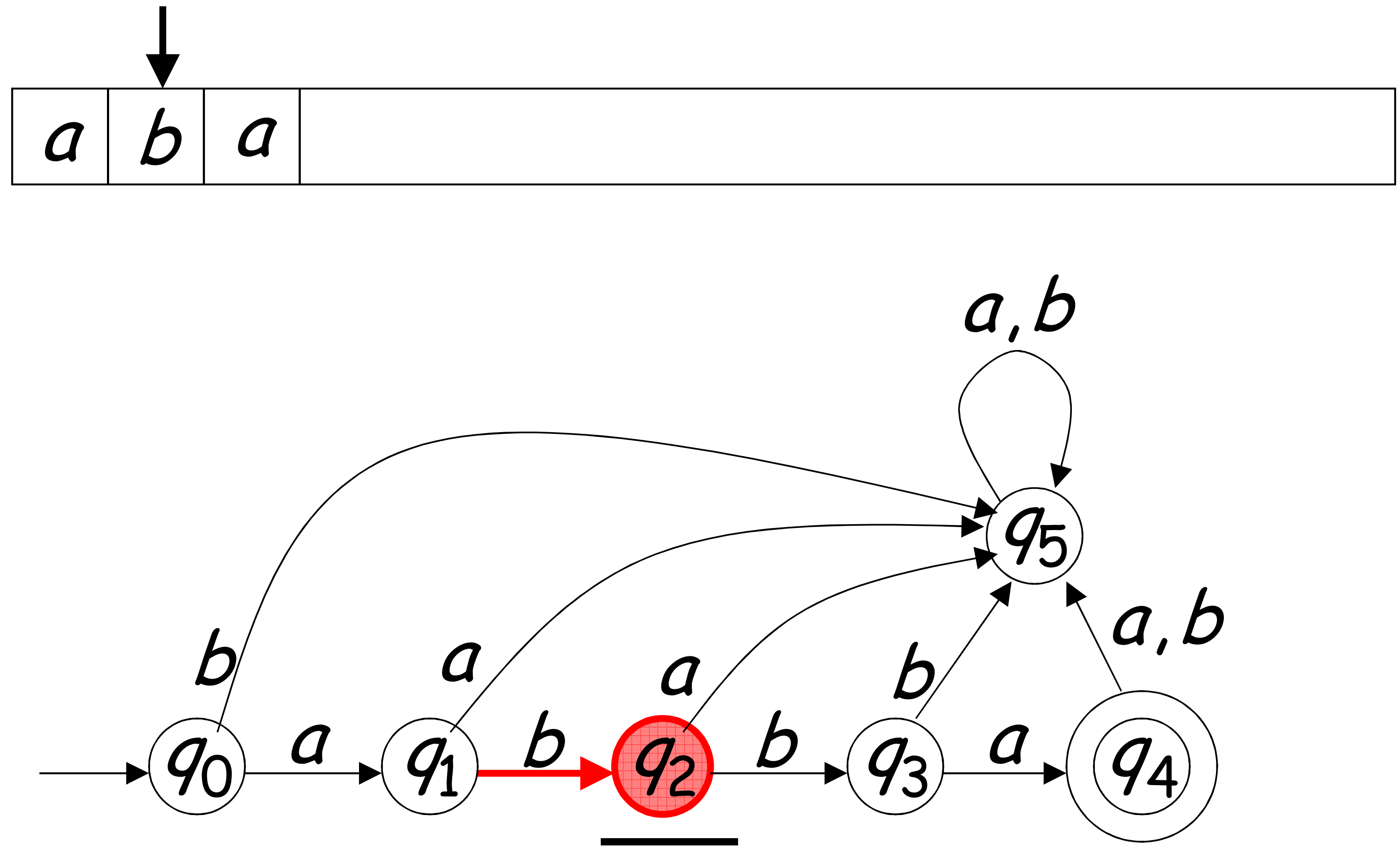
Un caso rigettato



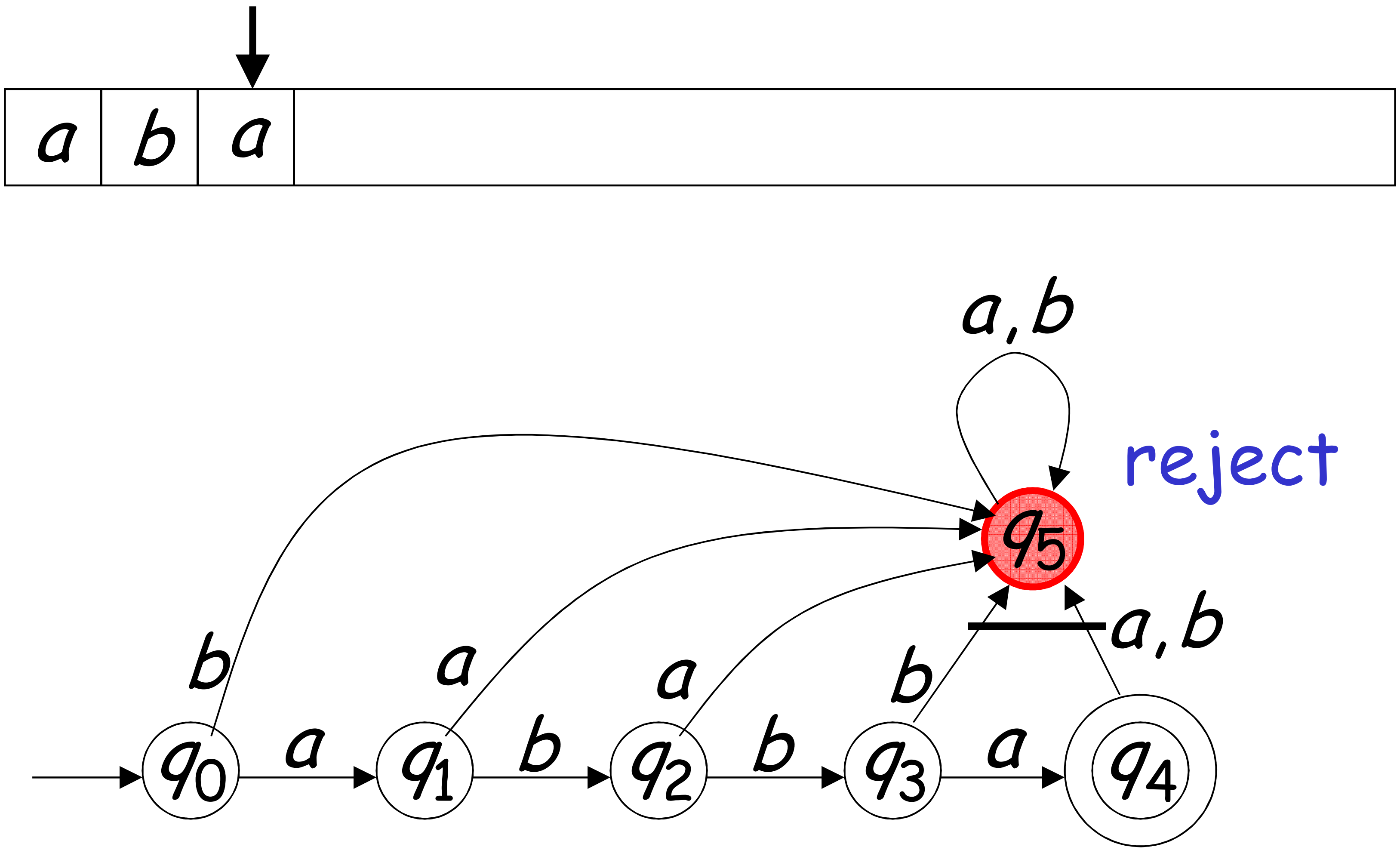
Input String




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

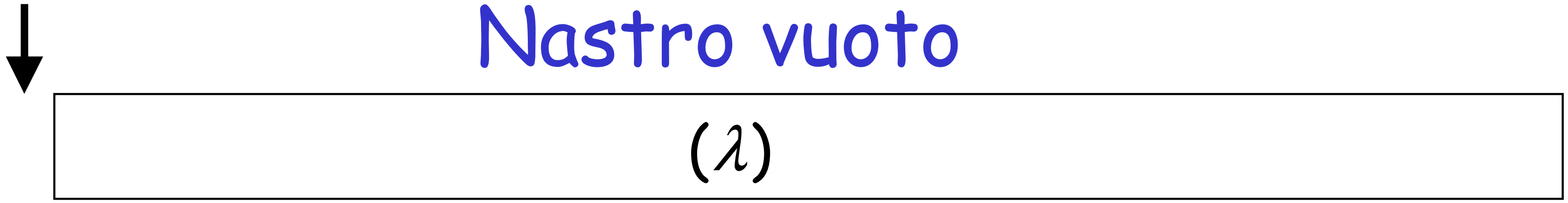

a
b
a
a,b
q_5
a,b
b
a
a
b
q_0
a
q_1
b
q_2
b
q_3
a
q_4

Input finito

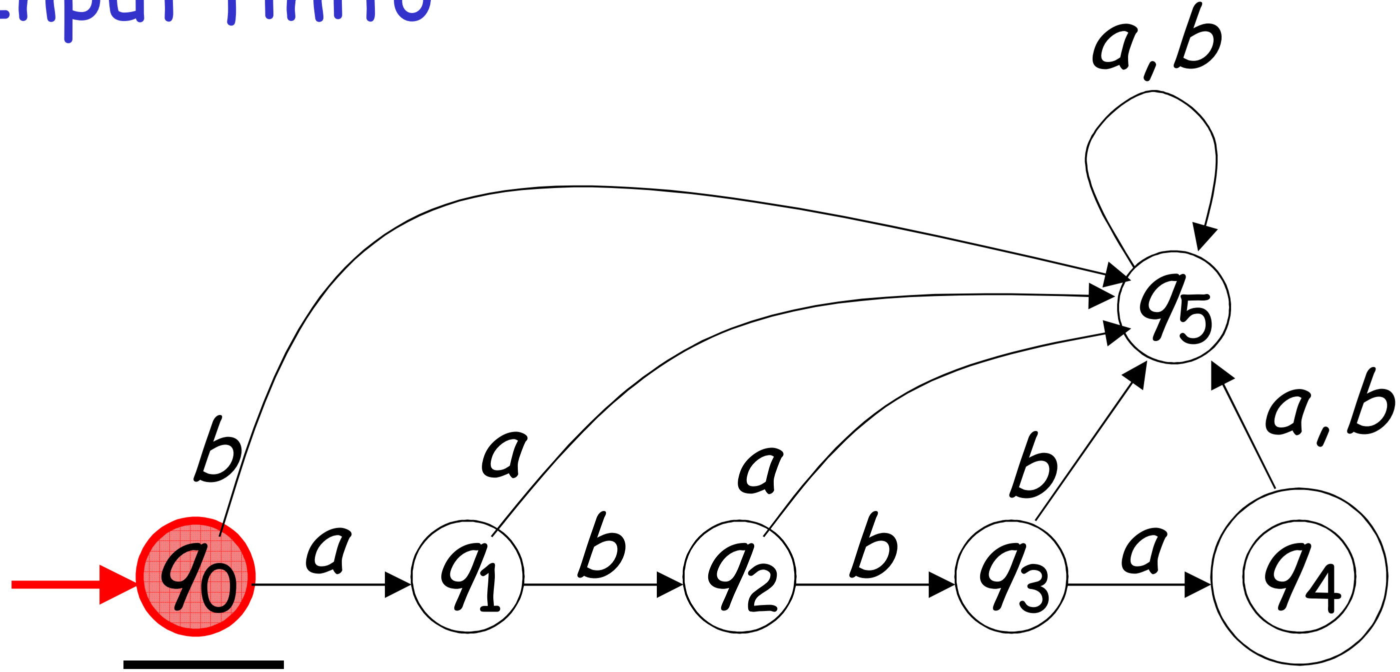


Un altro caso rigettato

Nastro vuoto

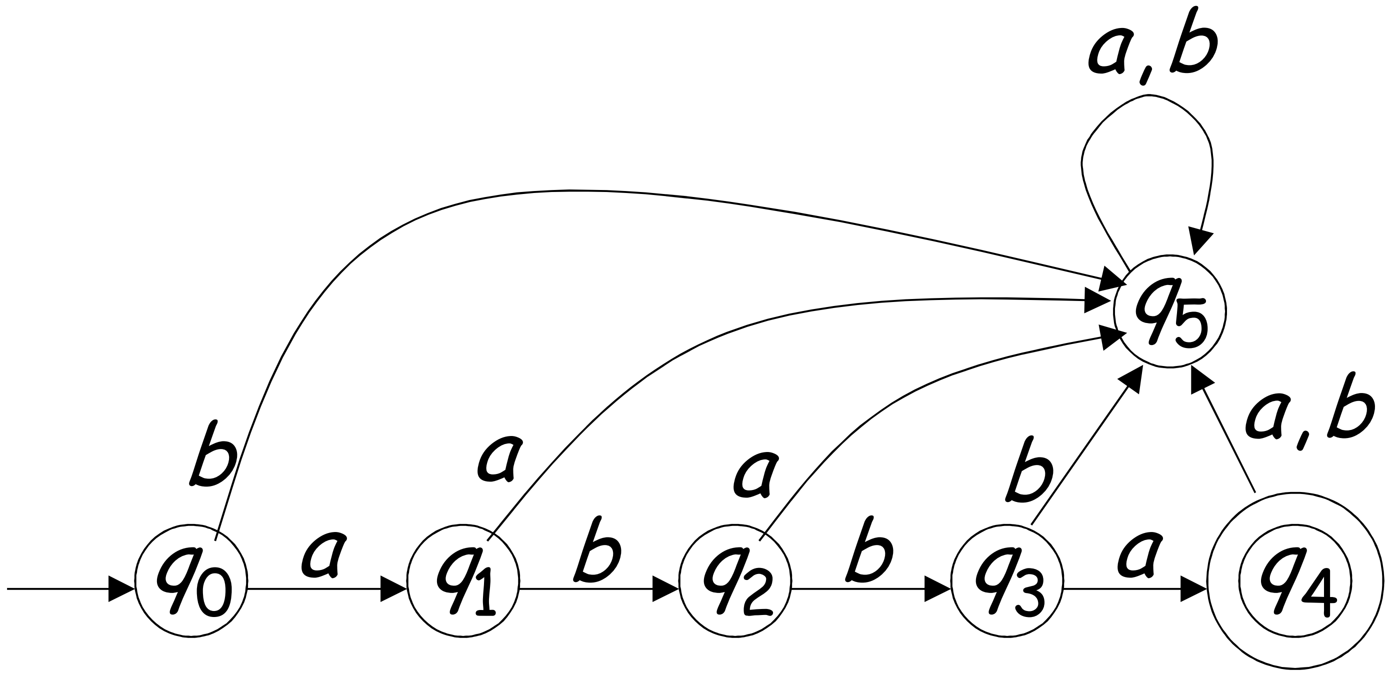


Input finito



rigettato

Linguaggio accettato: $L = \{abba\}$



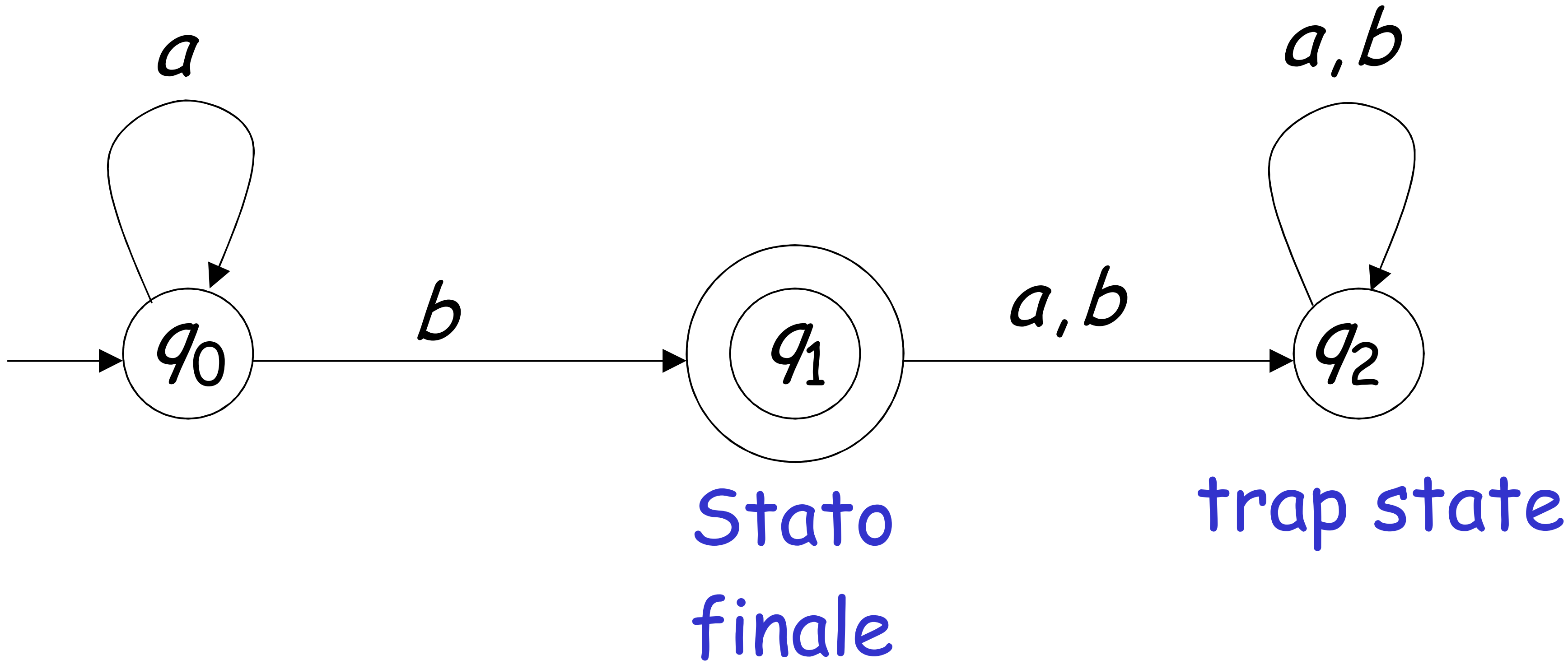
Per accettare una stringa :

Devono essere esaminati tutti i caratteri di
Input e l'ultimo stato è uno stato finale

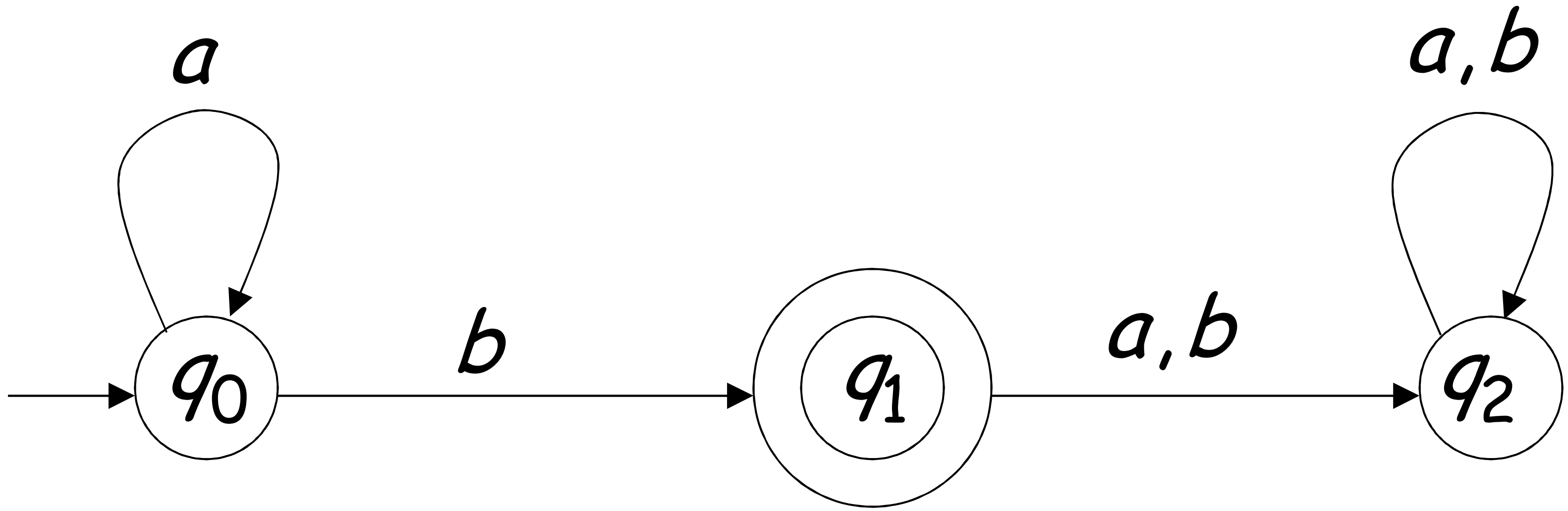
Per rigettare una stringa :

Tutti i caratteri di input sono stati esaminati
E non si è raggiunto uno stato finale

Un altro esempio

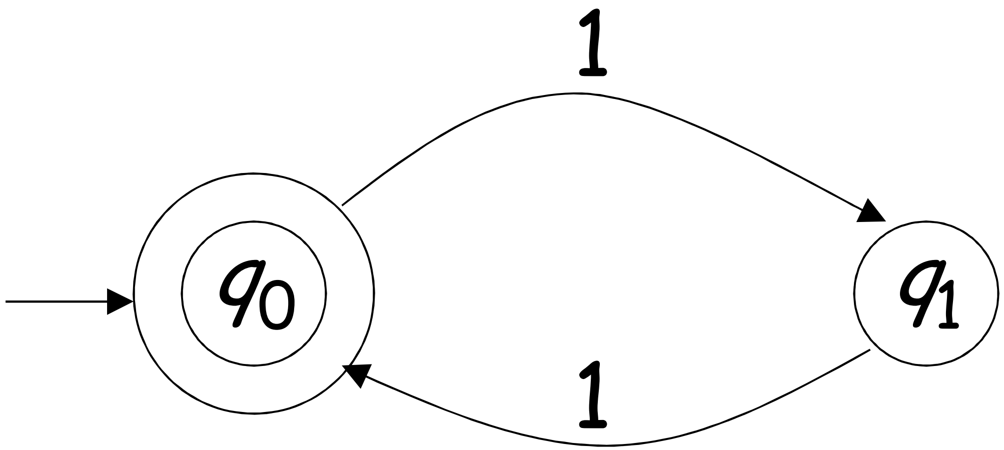


Language Accepted: $L = \{a^n b : n \geq 0\}$



Un altro esempio

Alfabeto: $\Sigma = \{1\}$



Linguaggio accettato:

$$EVEN = \{x : x \in \Sigma^* \text{ and } x \text{ is even}\}$$
$$= \{\lambda, 11, 1111, 111111, \ldots\}$$

Definizione formale

un automa deterministico formale(DFA)

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

Q : insieme degli stati

Σ : alfabeto di input $\lambda \notin \Sigma$

δ : funzione di transizione

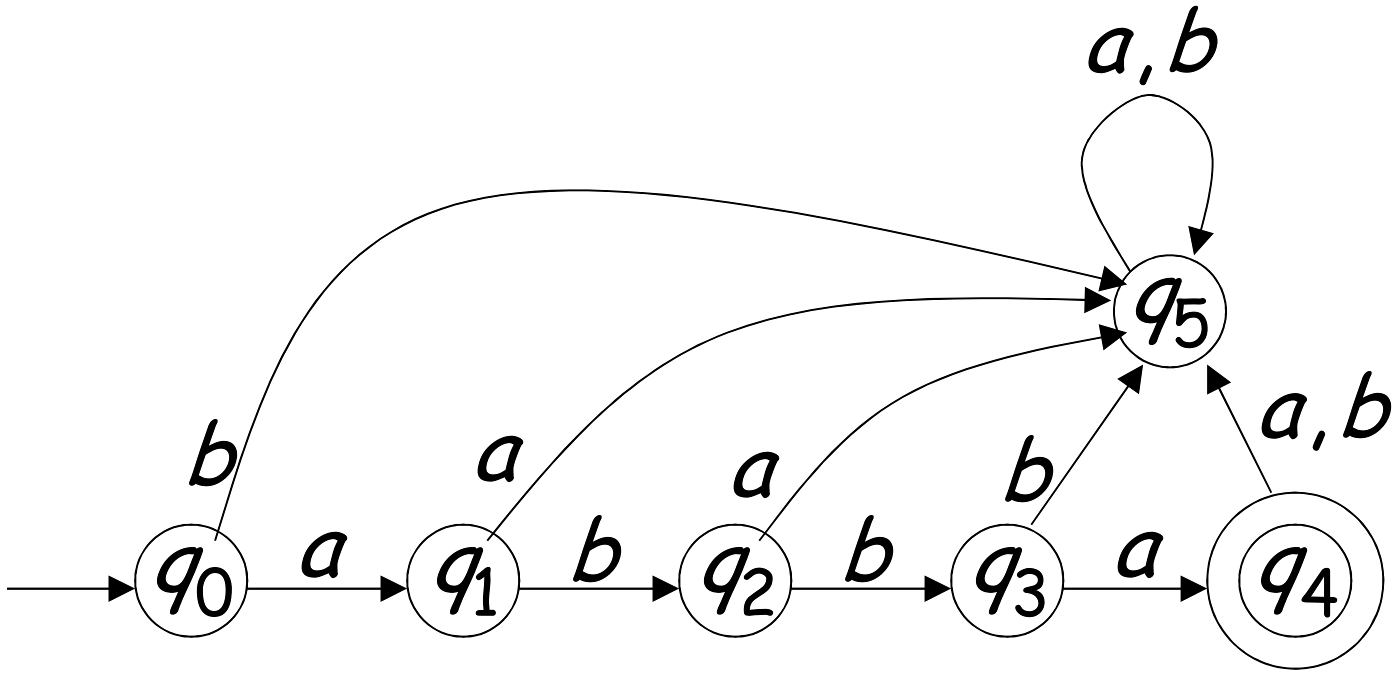
q_0 : stato iniziale

F : insieme degli stati di accettazione (finale)

Insieme degli stati Q

esempio

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

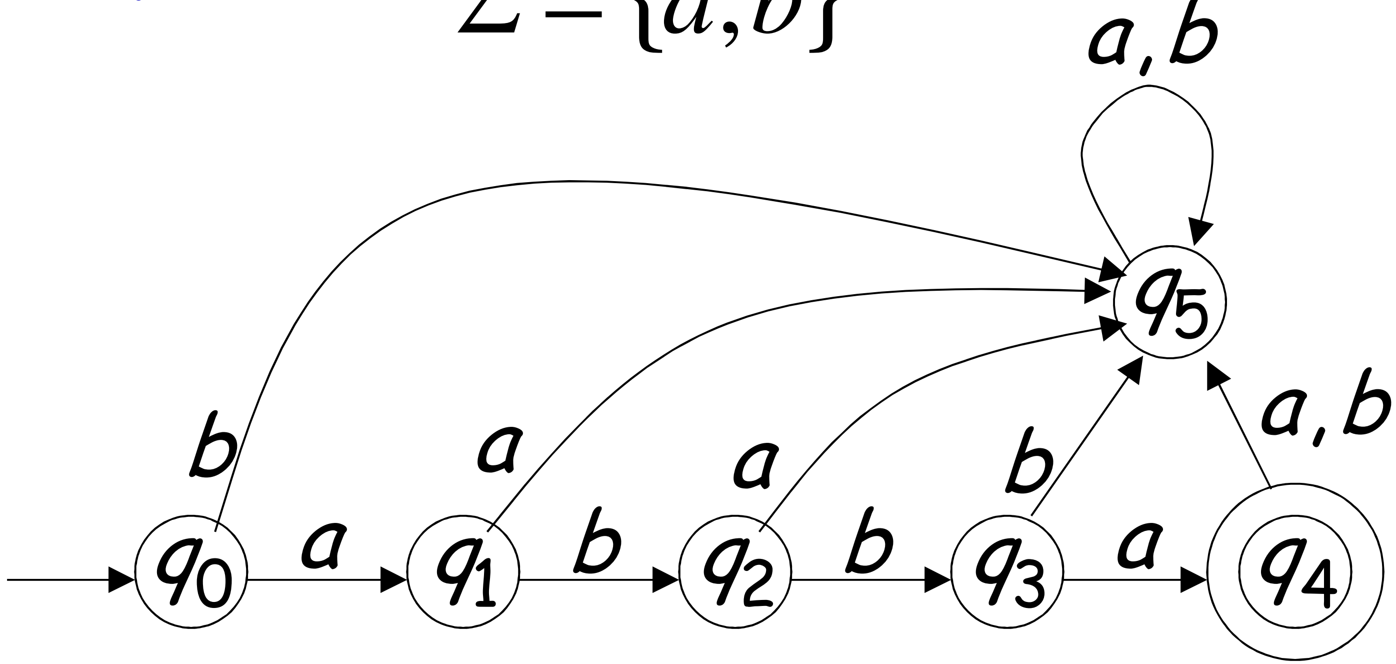


Alfabeto di input Σ

$\lambda \notin \Sigma$:l'alfabeto di input non contene λ

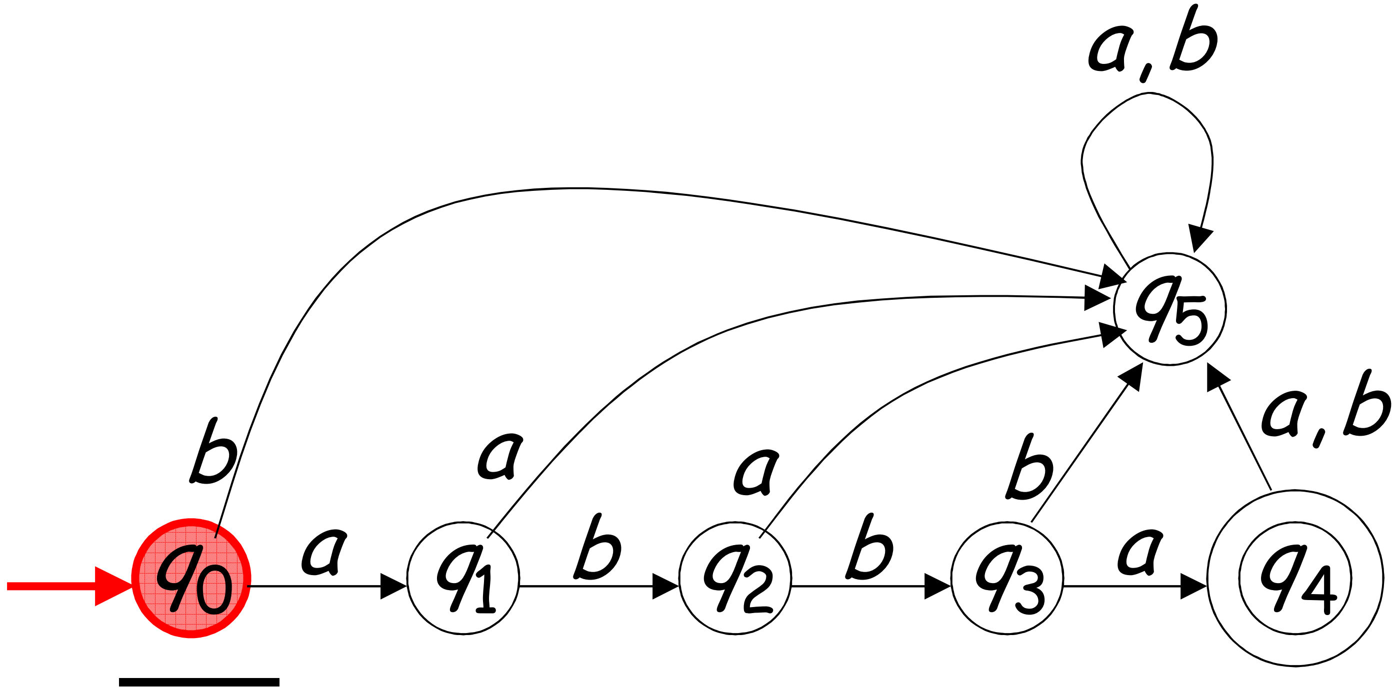
esempio

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



Stato iniziale q_0

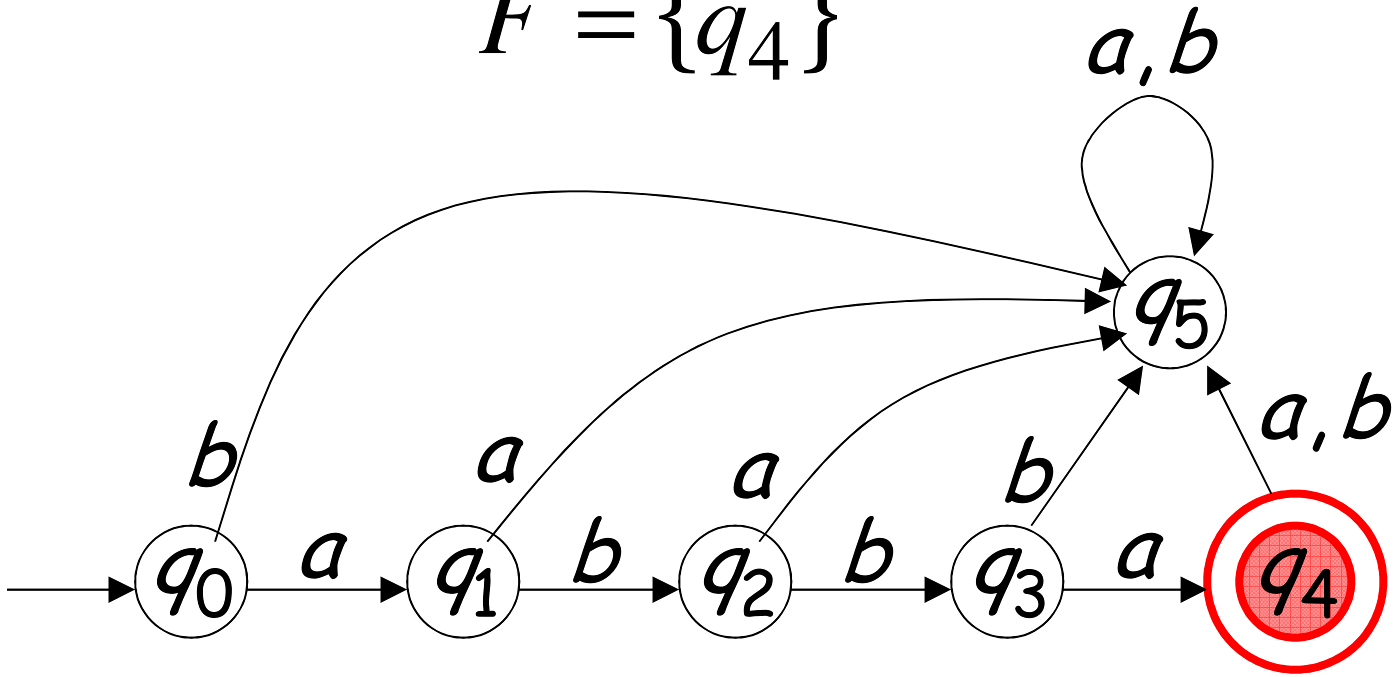
esempio



Insieme stati finali $F \subseteq Q$

esempio

$$F = \{q_4\}$$



Funzione di transizione

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

$$\delta(q, x) = q'$$

$$q \xrightarrow{x} q'$$

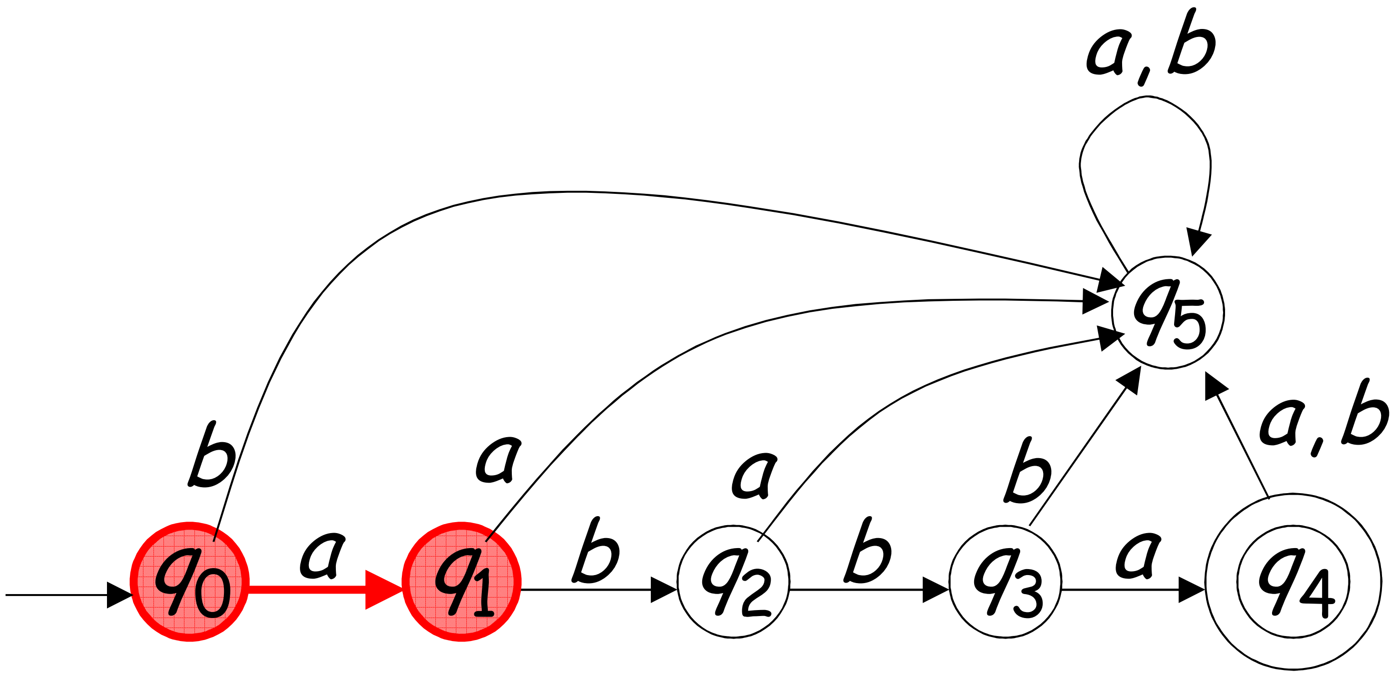
Descrive il risultato della

Transizione dallo stato q

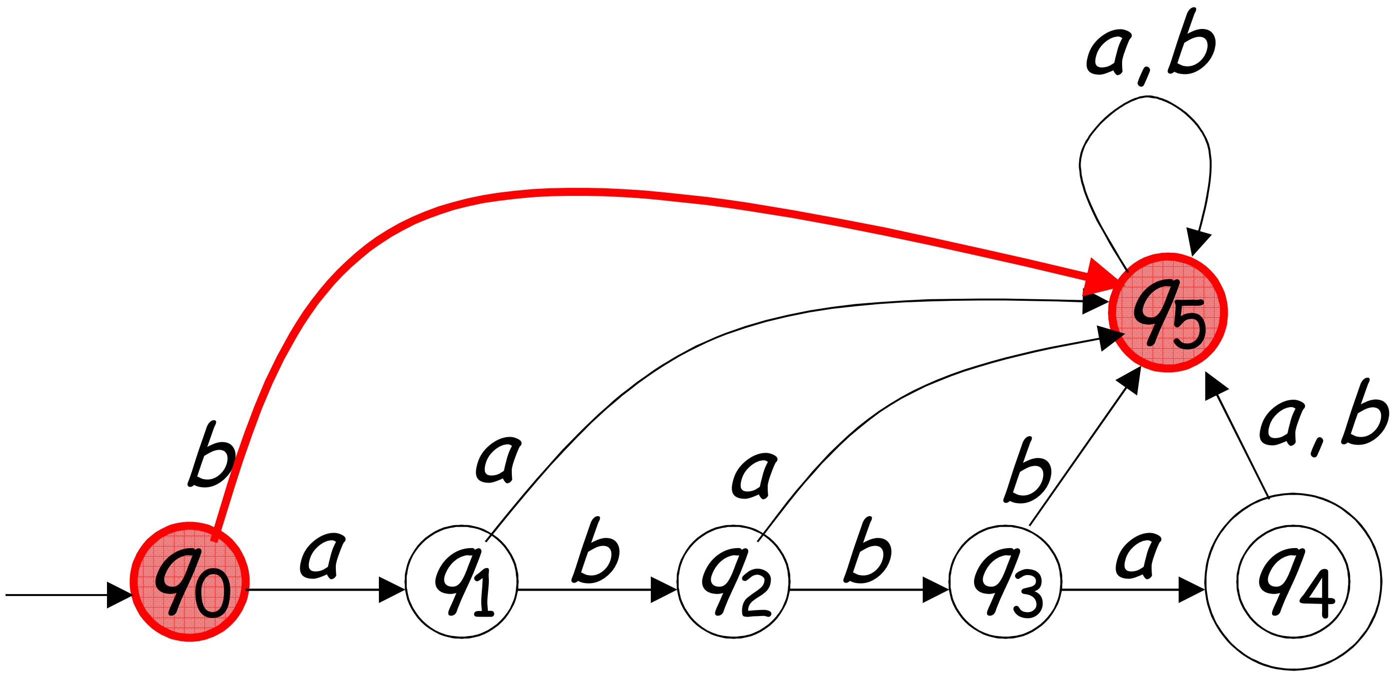
Con simbolo x

esempio:

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



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



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

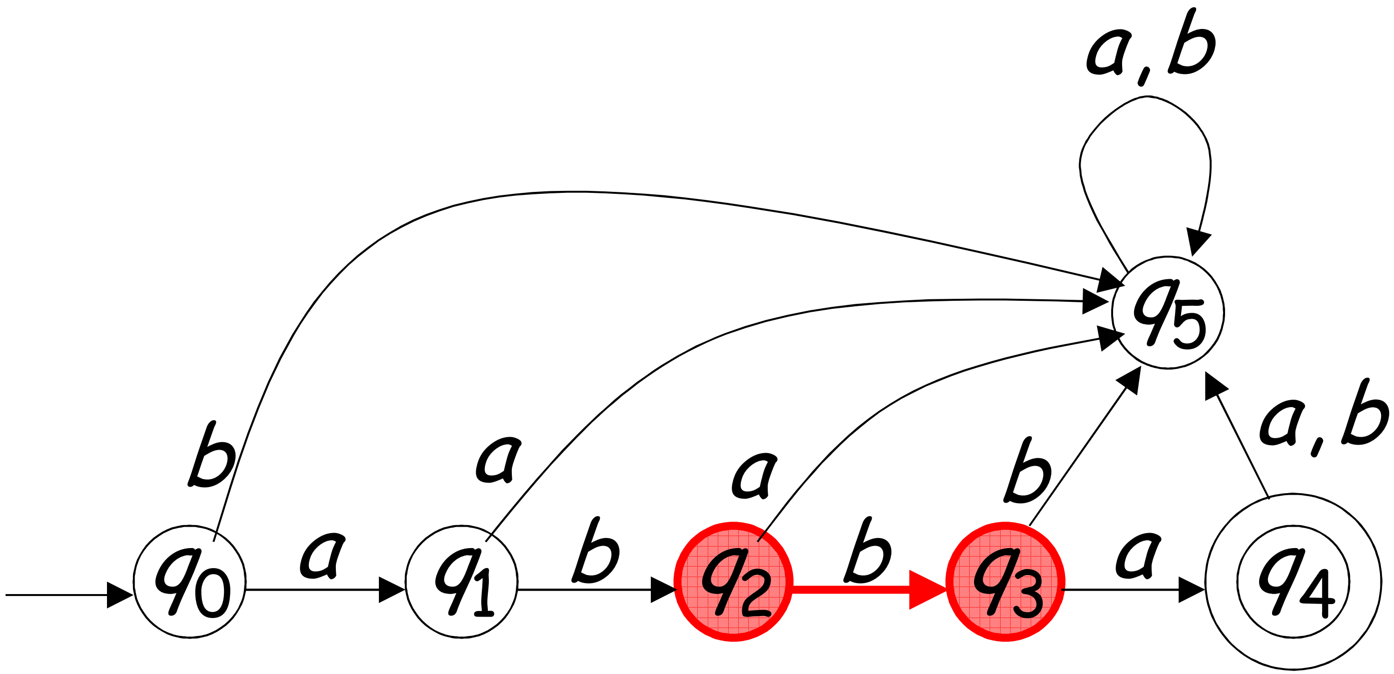
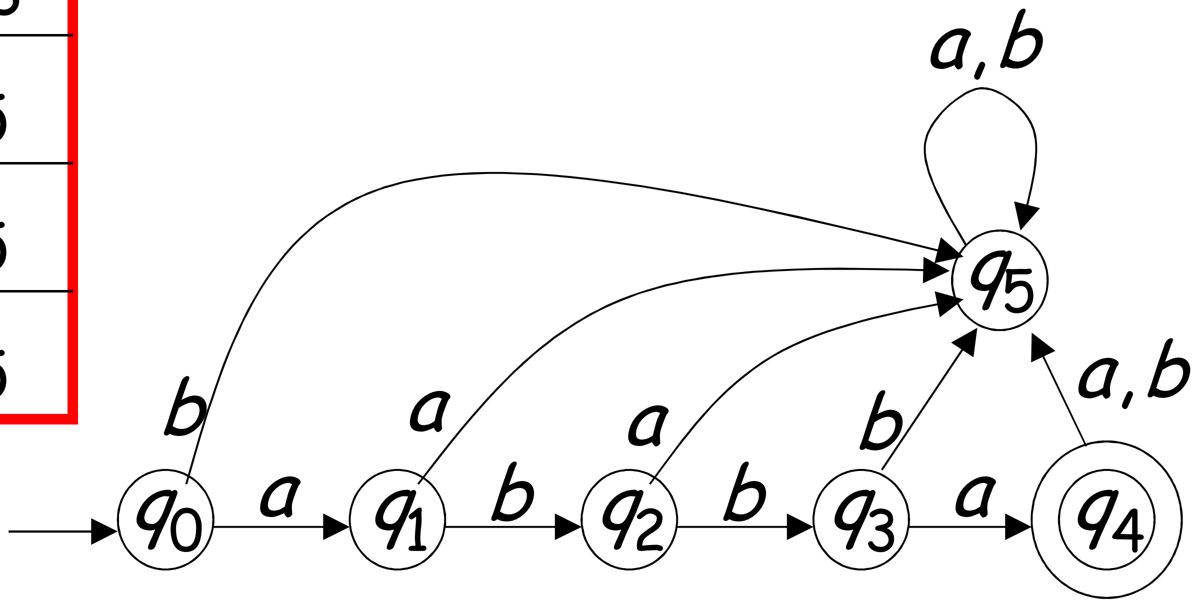


Tavola di transizione per δ

symbols

states

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



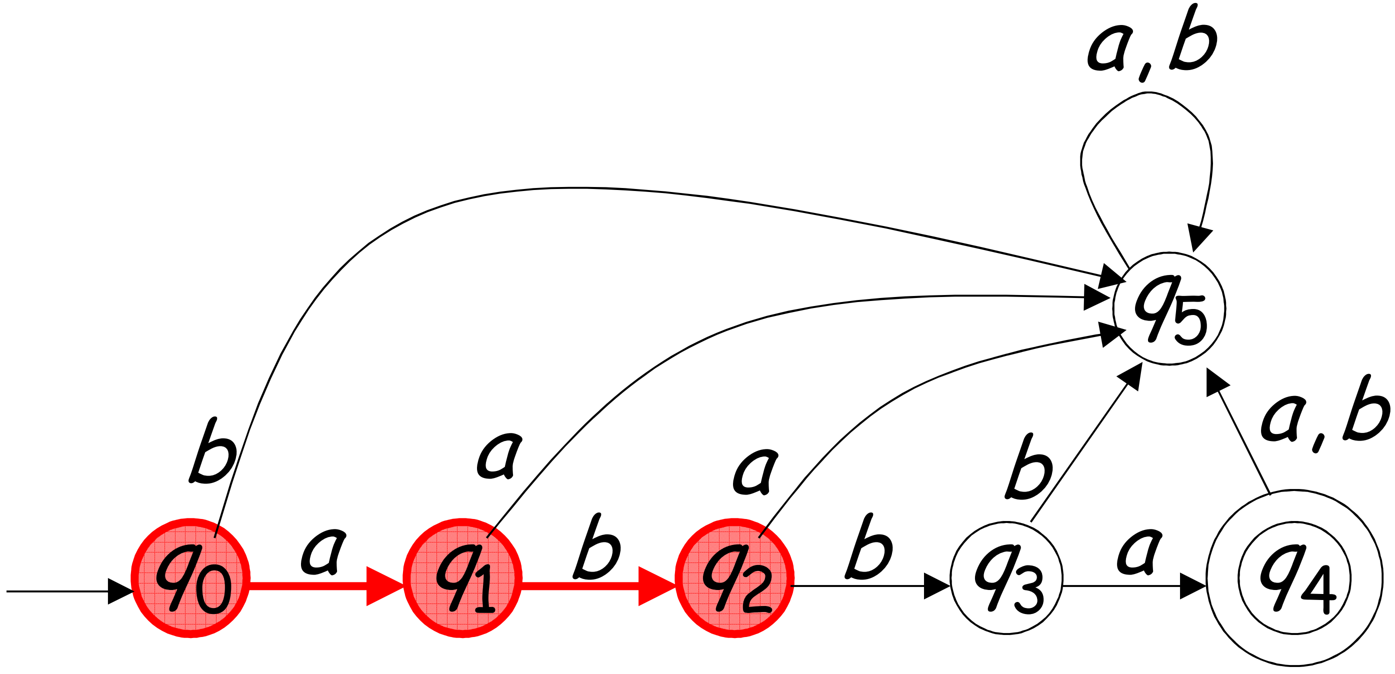
Funzione estesa di transizione

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

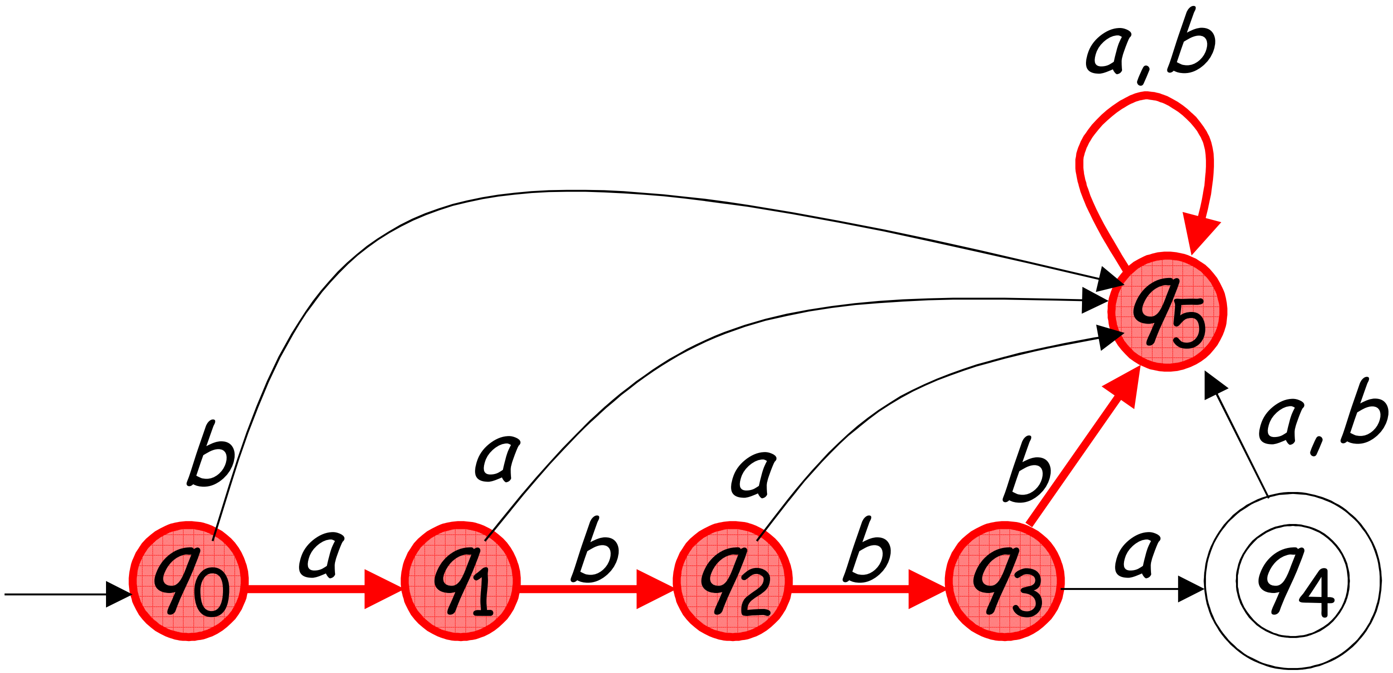
$$\delta^*(q, w) = q'$$

Descrive lo stato che risulta dopo aver
Esaminata la stringa w
a partire dallo stato q

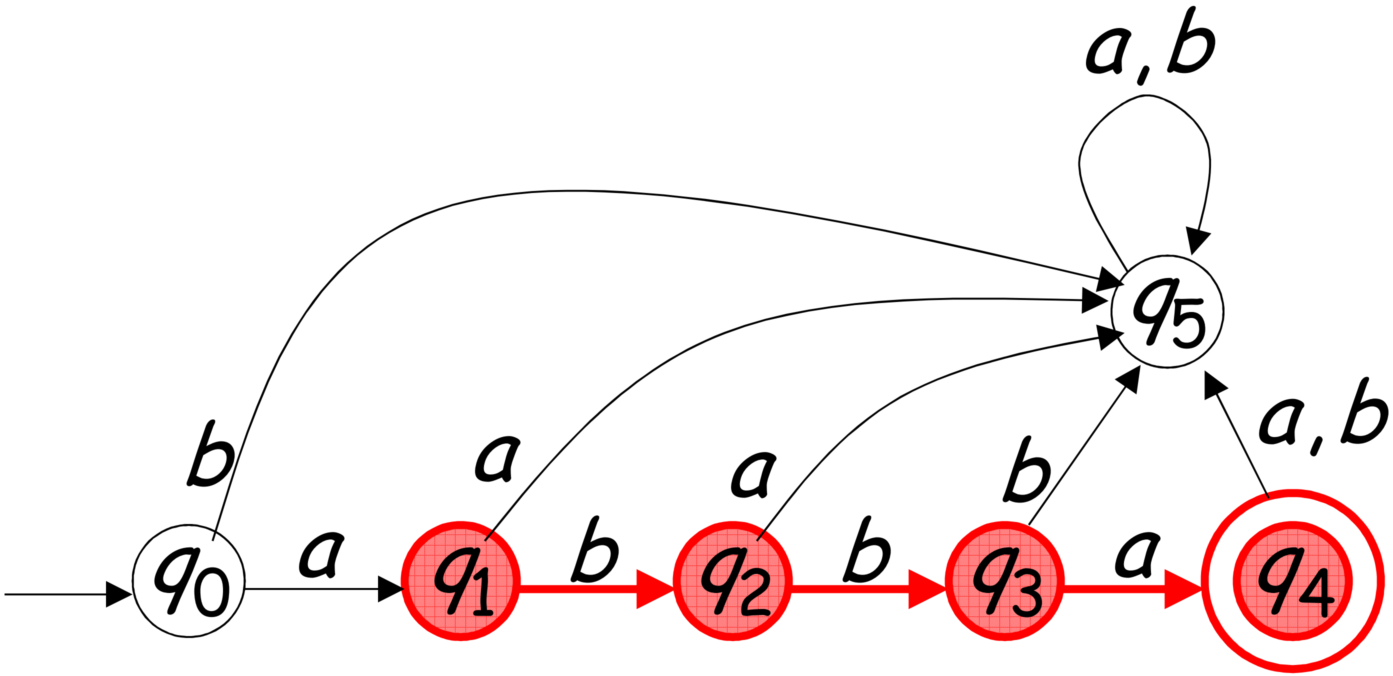
esempio: $\delta^*(q_0, ab) = q_2$



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



$$\delta^*(q_1, bba) = q_4$$



Caso speciale:

Per ogni stato q

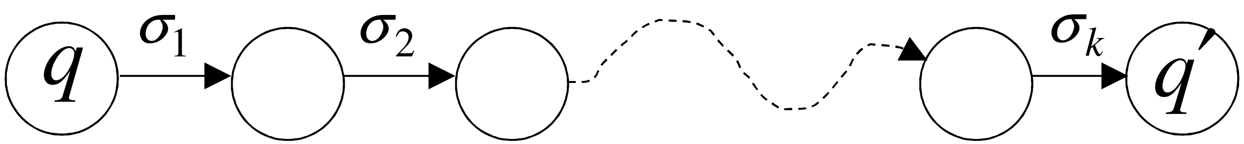
$$\delta^*(q, \lambda) = q$$

:

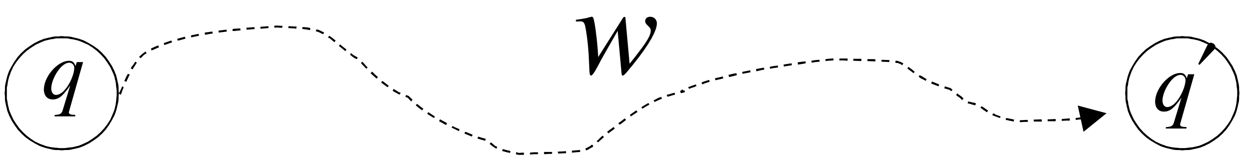
$$\delta^*(q,w) = q'$$

Implica che vi è un cammino di transizione

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



Alcuni stati possono essere ripeturi



Linguaggio accettato da un DFA

Linguaggio di un DFA: M

È denotato come $L(M)$

E contiene tutte le stringhe

Accettate da M

Un linguaggio L'

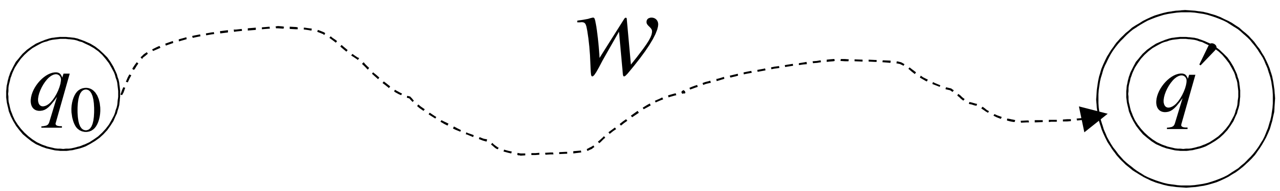
È accettato (o riconisciuto)

Da un DFA M se $L(M) = L'$

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

Il linguaggio accettato da M:

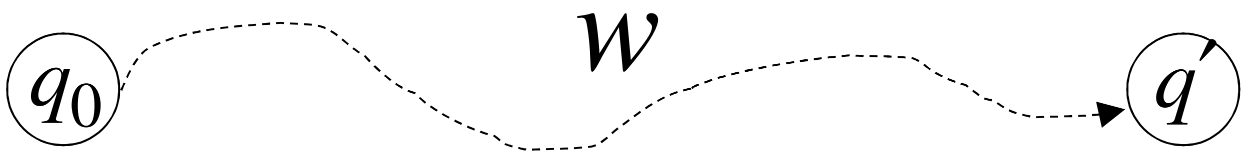
$$L(M) = \{ w \in \Sigma^* : \delta^*(q_0, w) \in F \}$$



$q' \in F$

Linguaggio rifiutato da M:

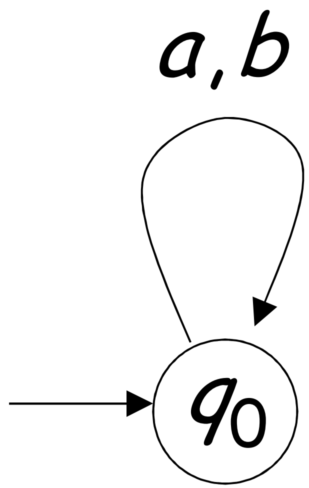
$$\overline{L(M)} = \{ w \in \Sigma^* : \delta^*(q_0, w) \notin F \}$$

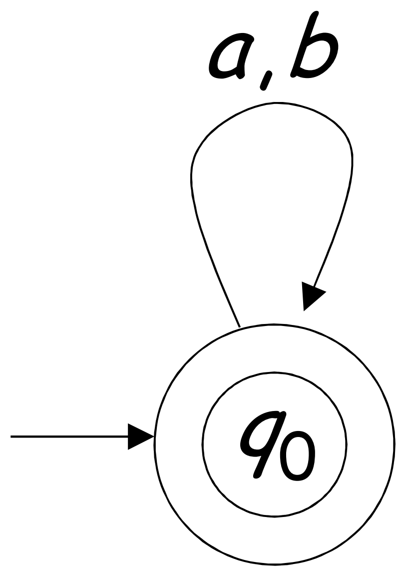


$q' \notin F$

DFA esempi

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





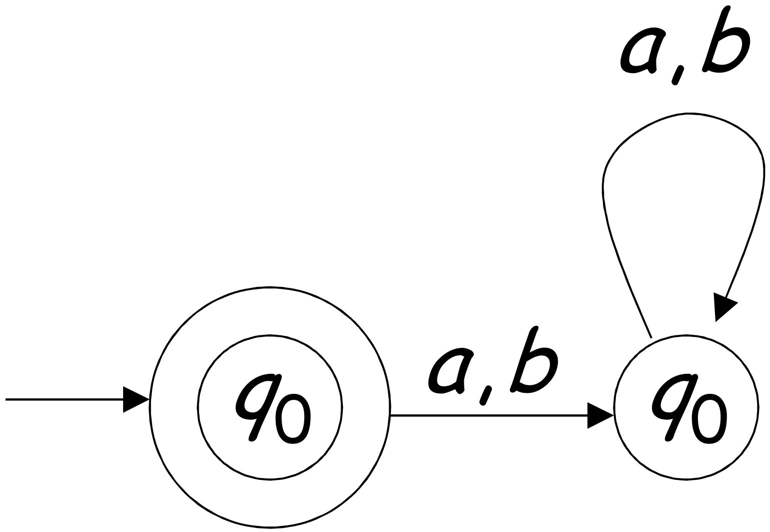
$$L(M) = \{\ \}$$

Linguaggio vuoto

$$L(M) = \Sigma^*$$

Tutte le stringhe

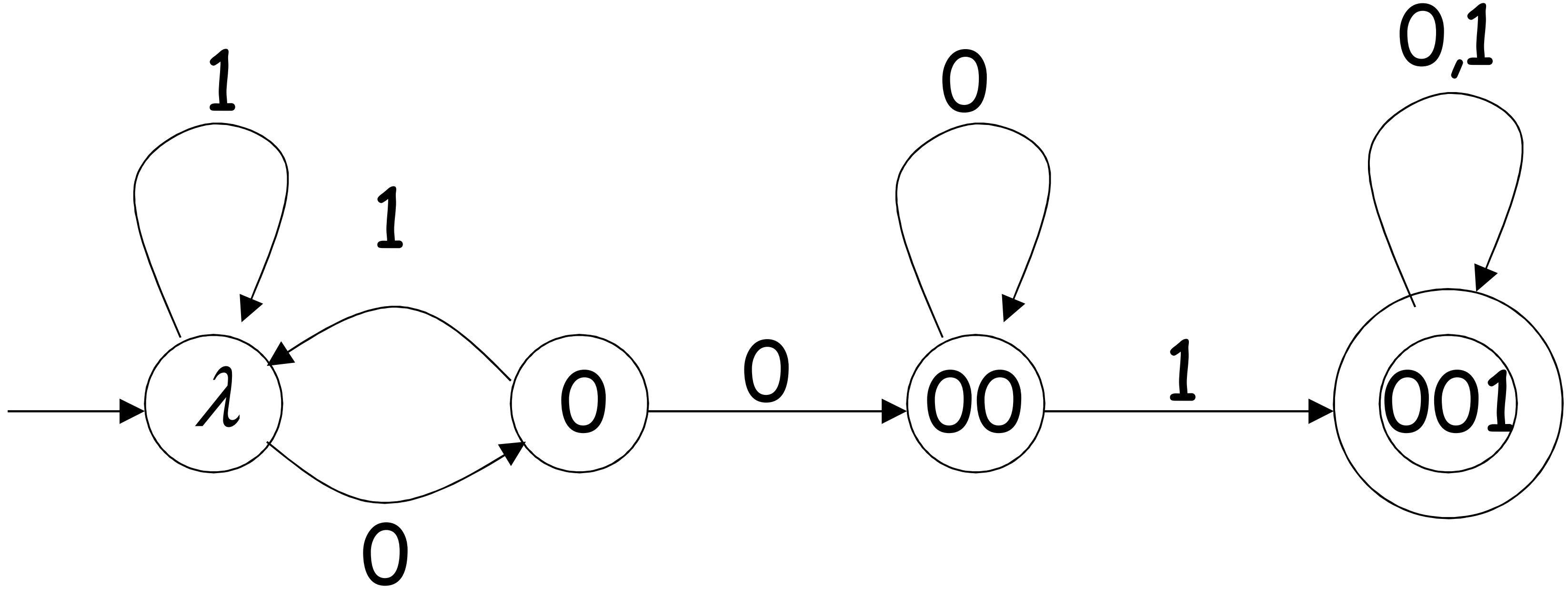
$$\Sigma = \{a, b\}$$



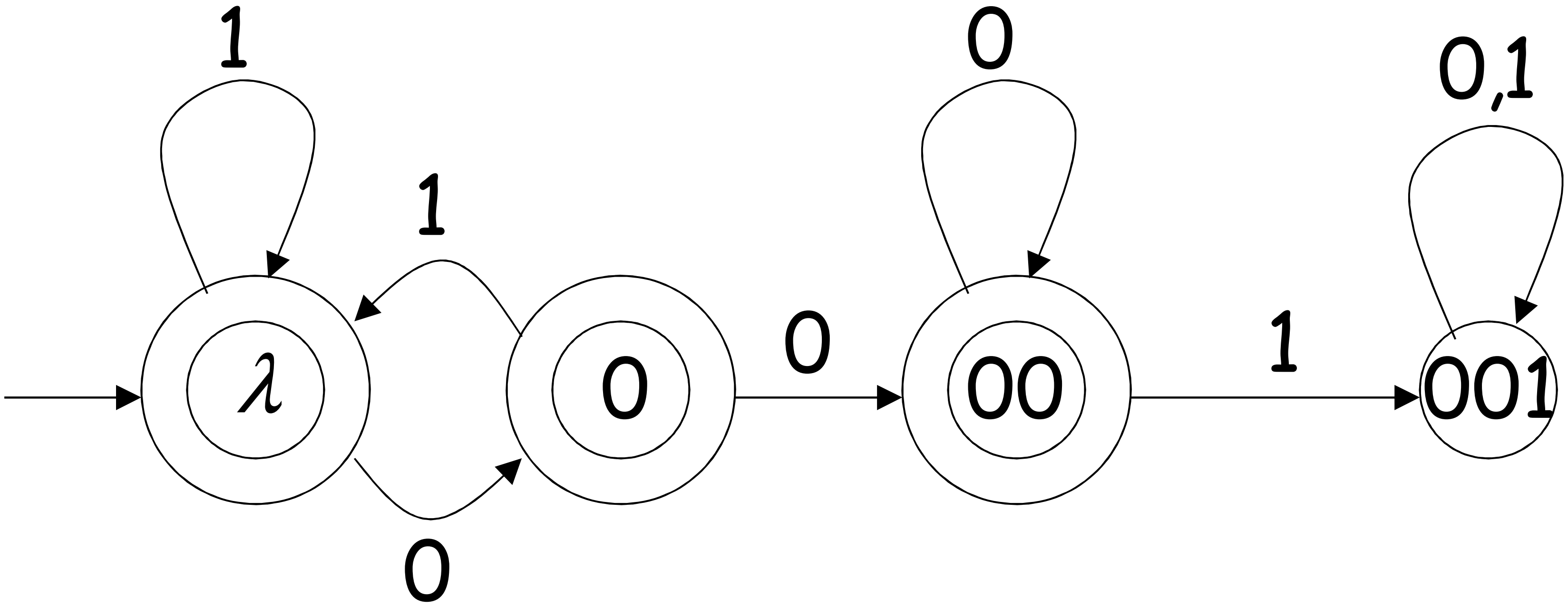
$$L(M) = \{\lambda\}$$

Linguaggio che riconosce le Stringa vuota

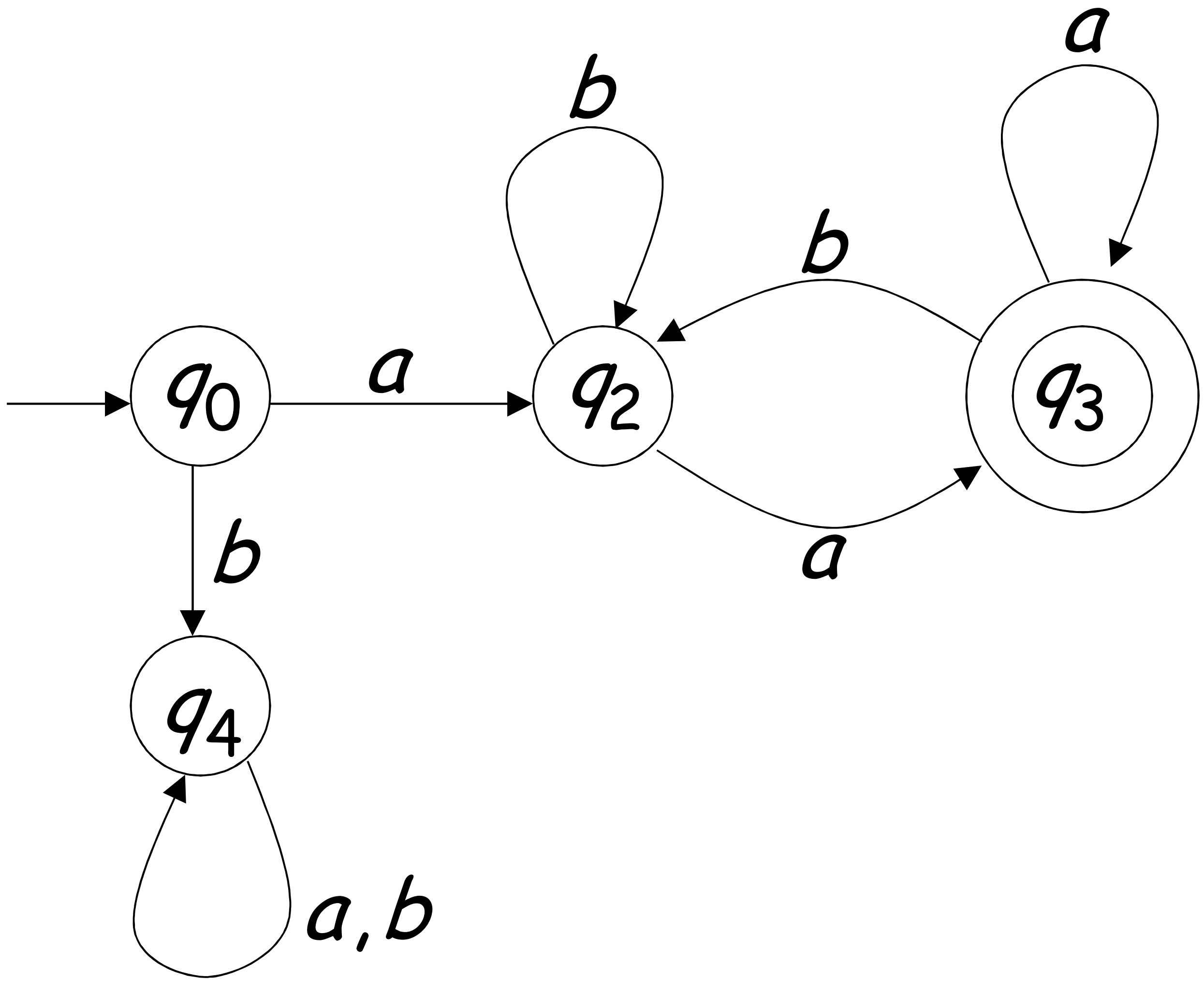
$L(M)$ = { tutte le stringhe binarie
che contengono 001
La sottostringa }



$L(M)$ = { tutte le stringhe binarie che non Contengono 001 }



$$L(M) = \left\{ awa : w \in \{a,b\}^* \right\}$$



Linguaggi regolari

Definizione:

Un linguaggio L è regolare se esiste un DFA M che lo accetta ($L(M) = L$)

I linguaggi accettati da tutti I DFA formano la famiglia dei linguaggi regolari

Esempi di linguaggi regolari:

$\{abba\}$ $\{\lambda, ab, abba\}$

$\{a^n b : n \geq 0\}$ $\{awa : w \in \{a,b\}^*\}$

{ tutte stringhe {a,b}* con prefisso ab }

{ all binary strings without substring 001}

$\{x : x \in \{1\}^*$ and x is even$\}$

$\{\ \}$ $\{\lambda\}$ $\{a,b\}^*$

Abbiamo visto in precedenza gli automi regolari che li definiscono

Esitono linguaggi che non sono regolari:

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

$$ADDITION = \{x + y = z \ : x = 1^n, y = 1^m, z = 1^k, \\ n + m = k\}$$

Non esiste nessun DFA che accetta
Questo linguaggio (vedremo più avanti)