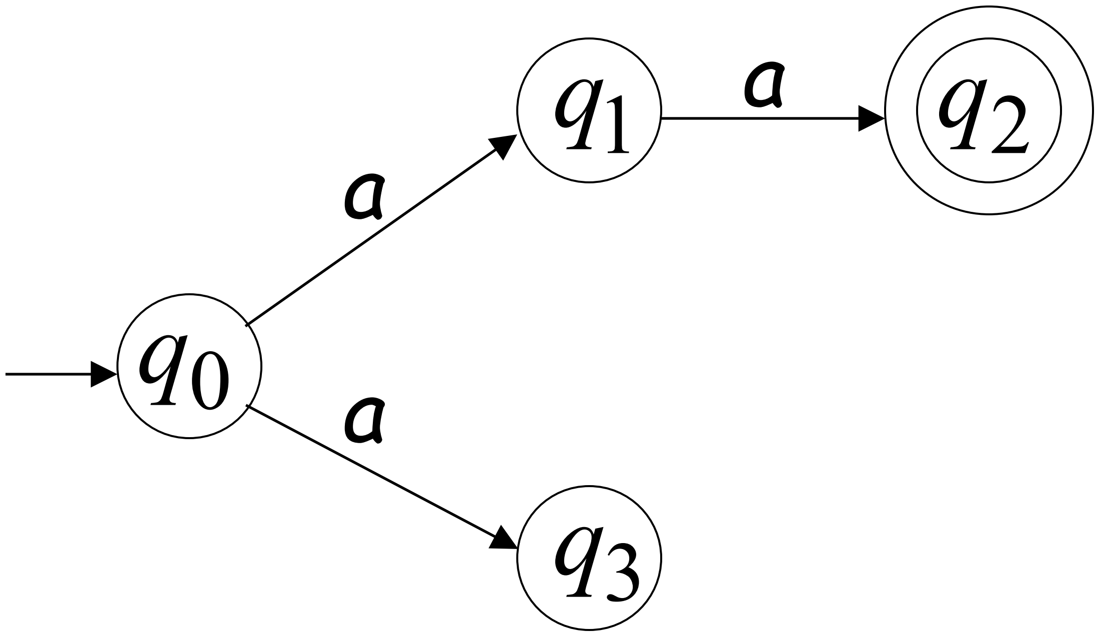
Non-Deterministic Finite Automata

Automa finito deterministico
calcolo finito e deterministico
sequenziale, un segmento di lng
dell'input

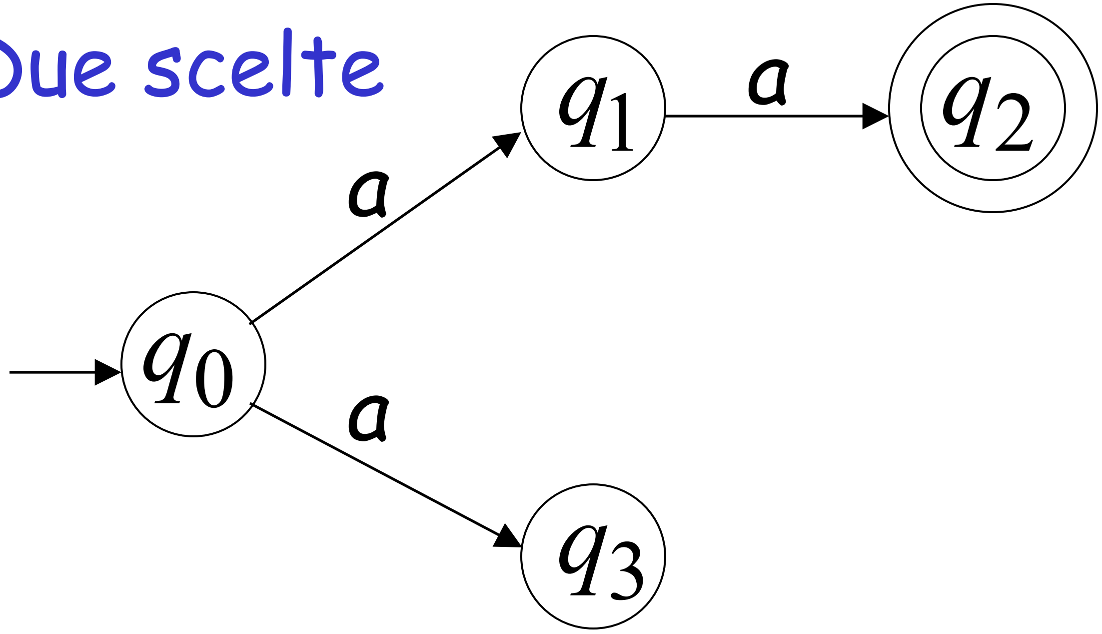
Automi non deterministici (NFA)

alfabeto = $\{a\}$

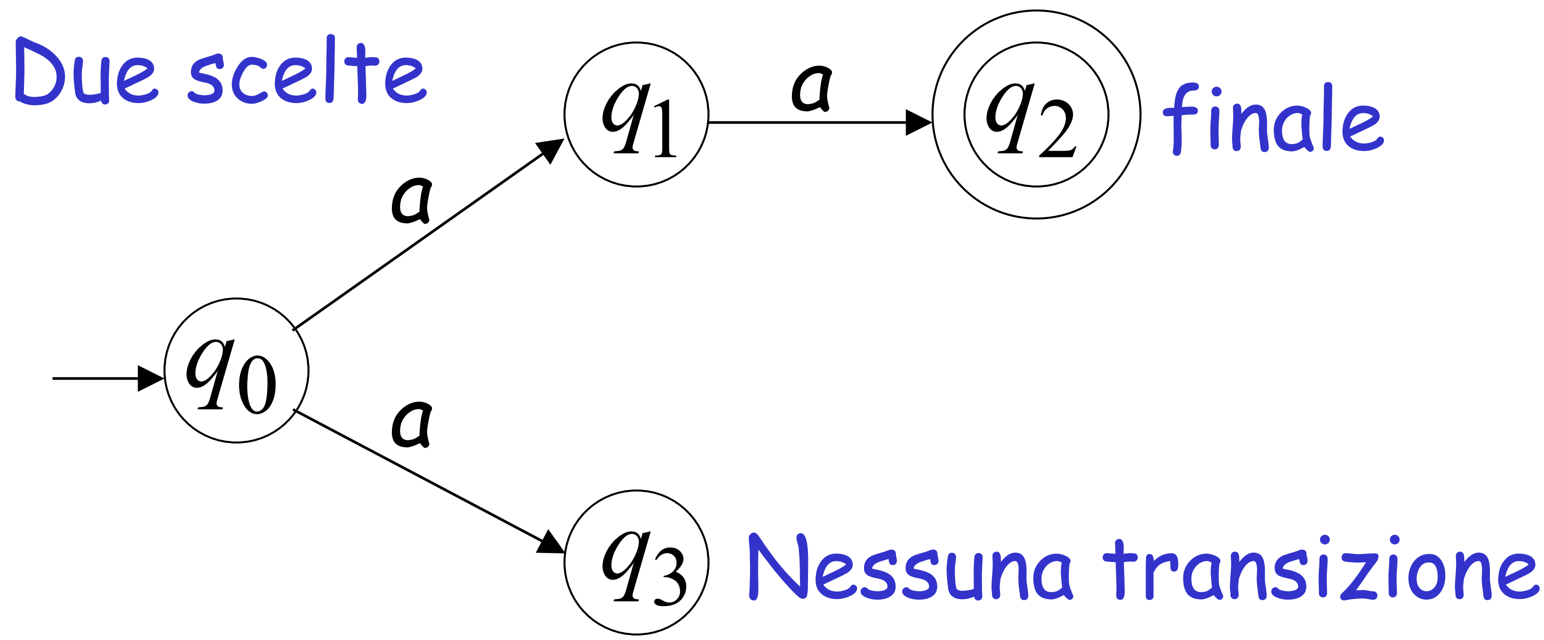


alfabeto = $\{a\}$

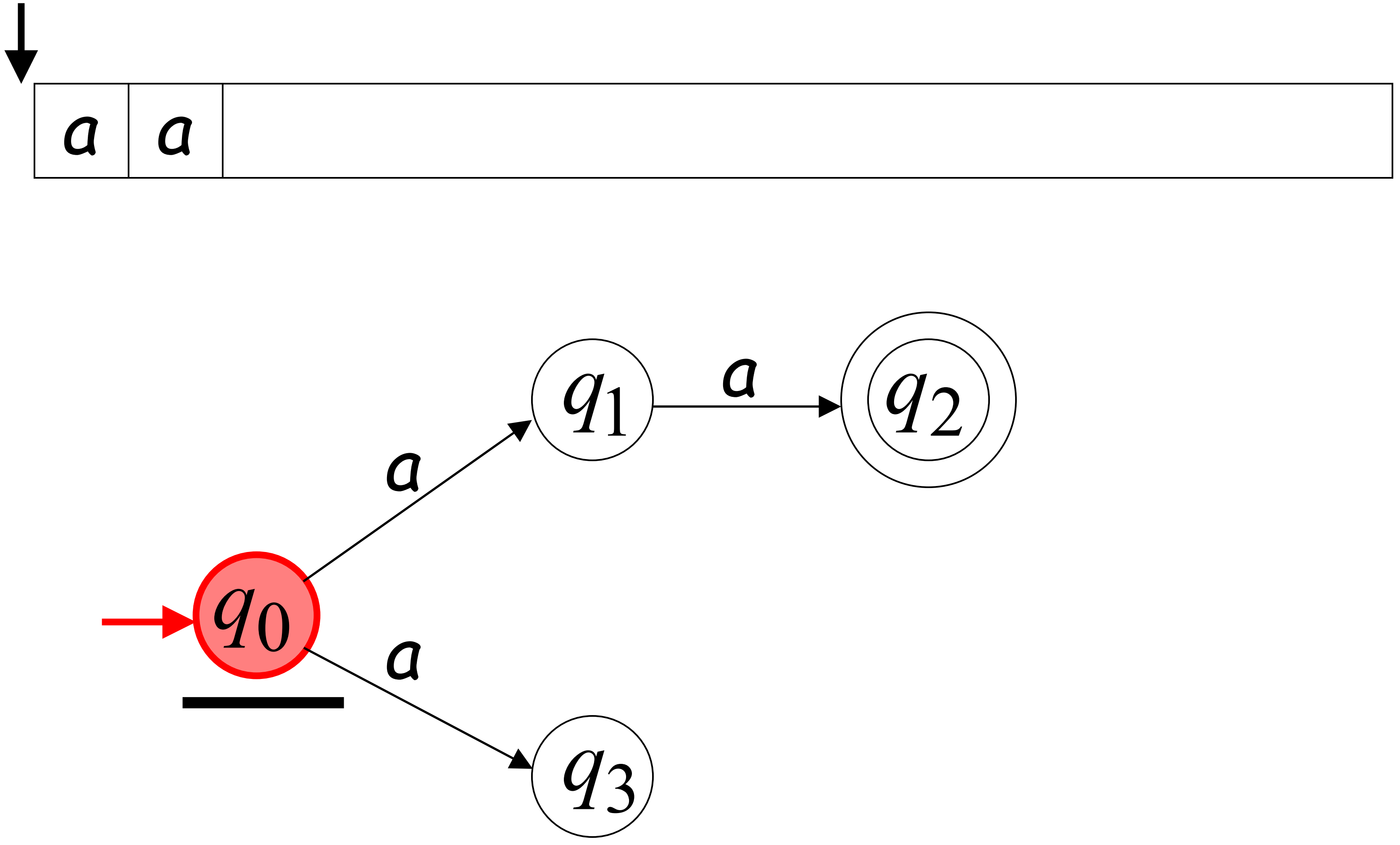
Due scelte



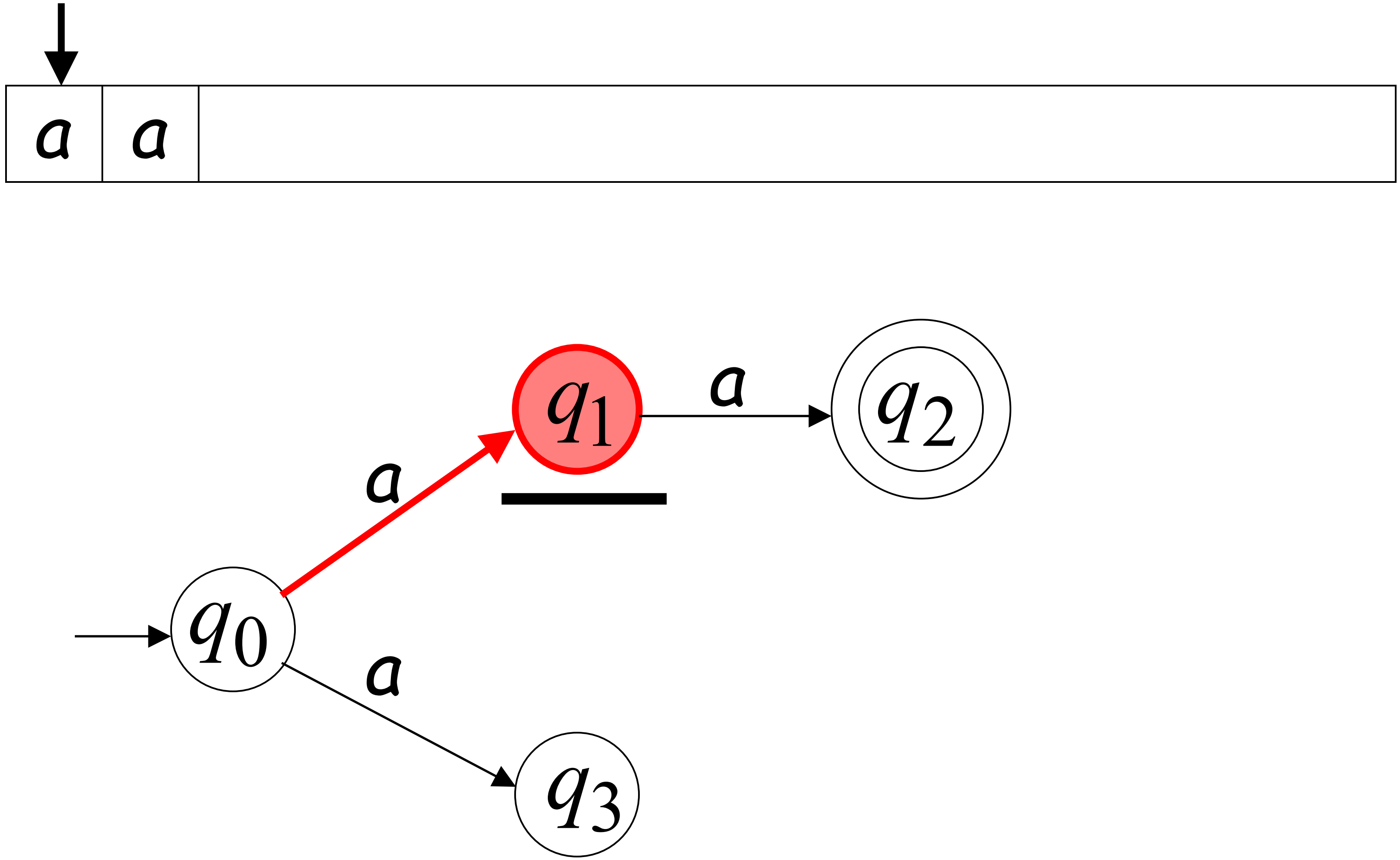
alfabeto = $\{a\}$



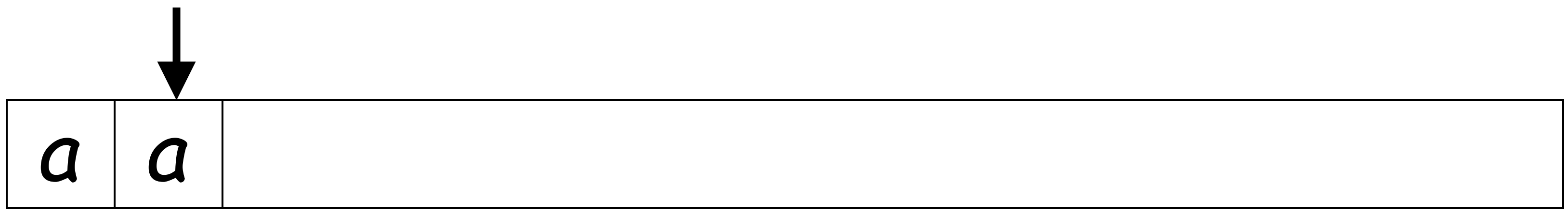
Prima delle due scelte



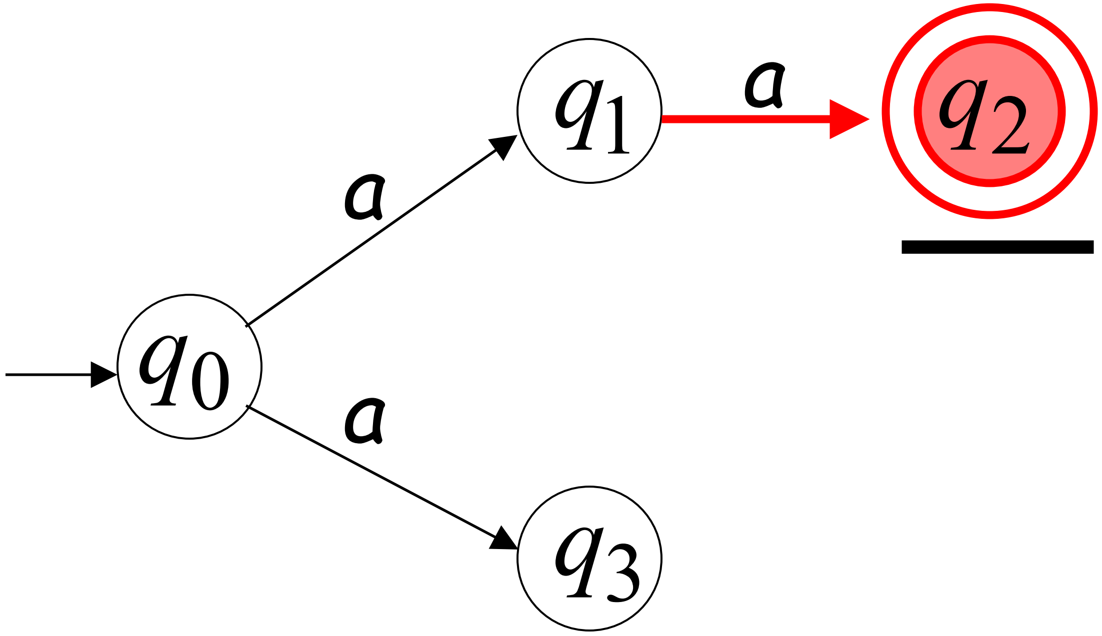
Prima scelta



Prima scelta

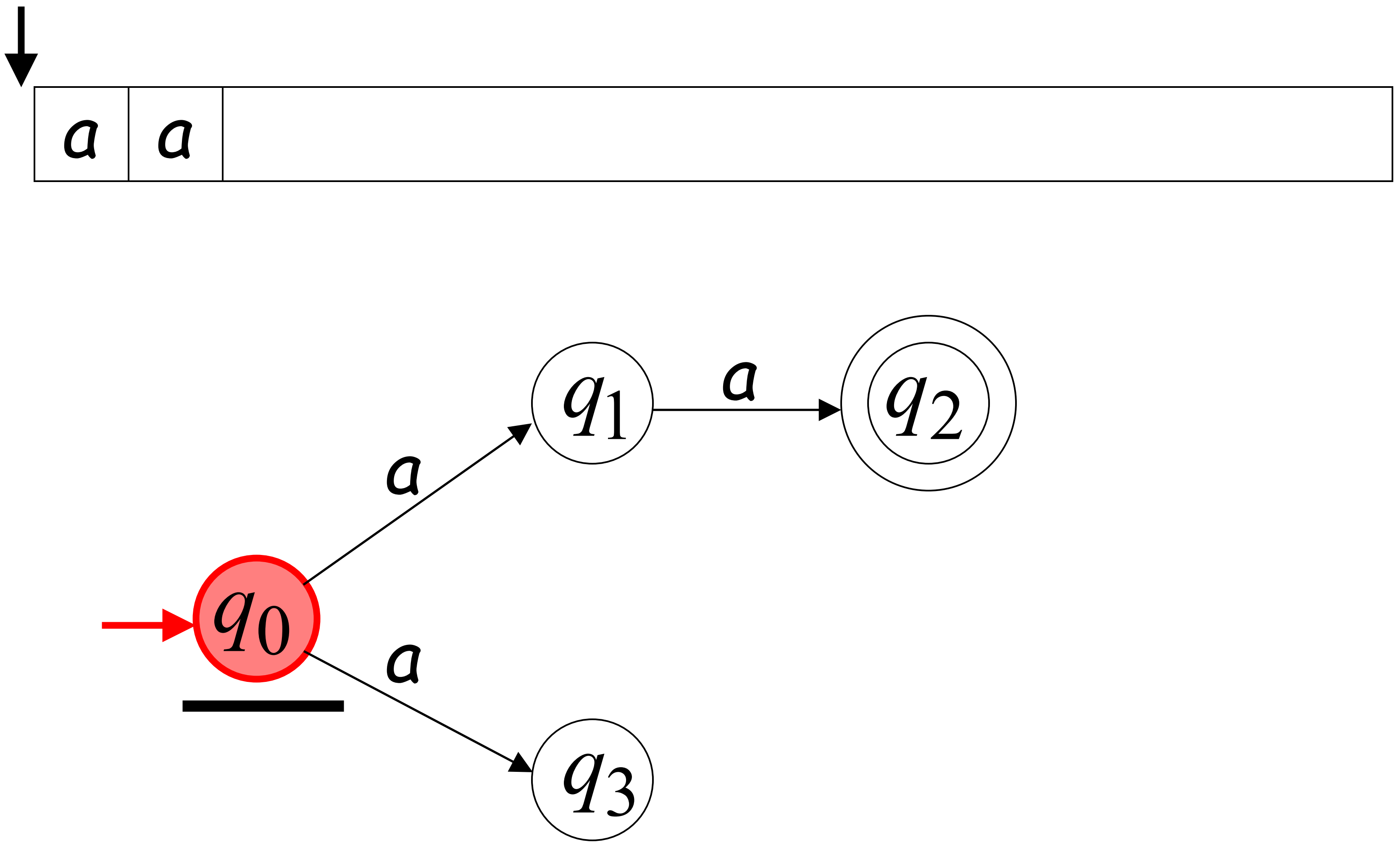


Abbiamo consumato tutto l'input

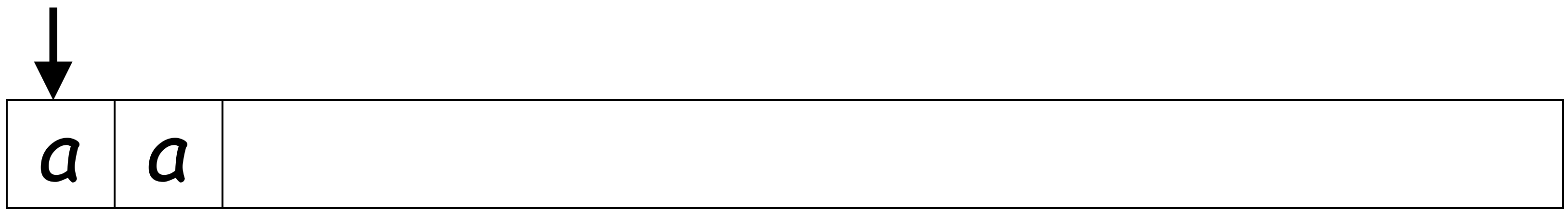


"accettato"

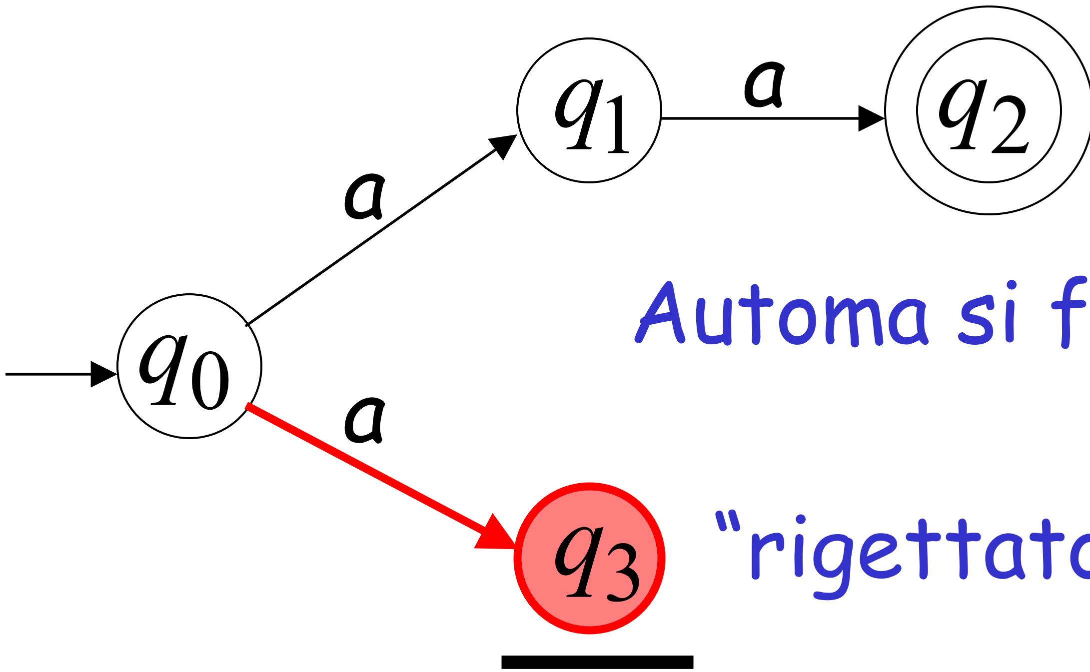
Seconda scelta



Seconda scelta



Input non può essere tutto usato



Automa si ferma

"rigettato"

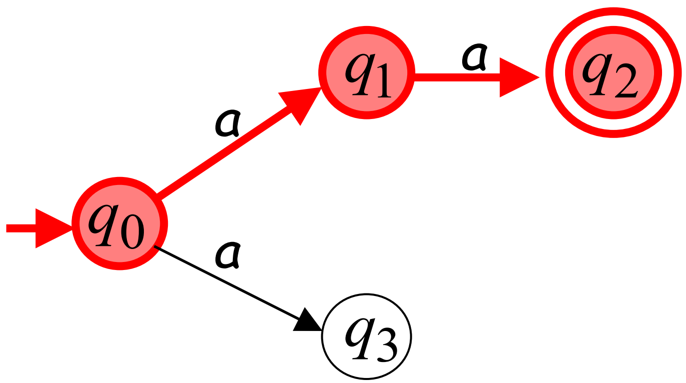
un NFA accetta una stringa:

Se esiste una computazione che accetta
la stringa

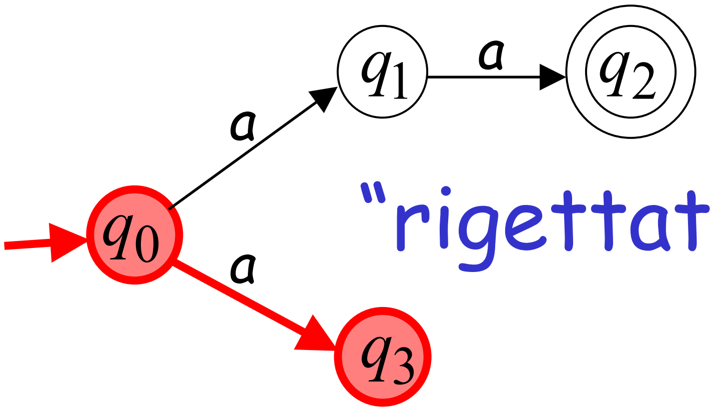
Tutta la stringa di input è stata letta e l’automa
Si trova in uno stato finale

aa È accettato dal NFA:

"accettato"



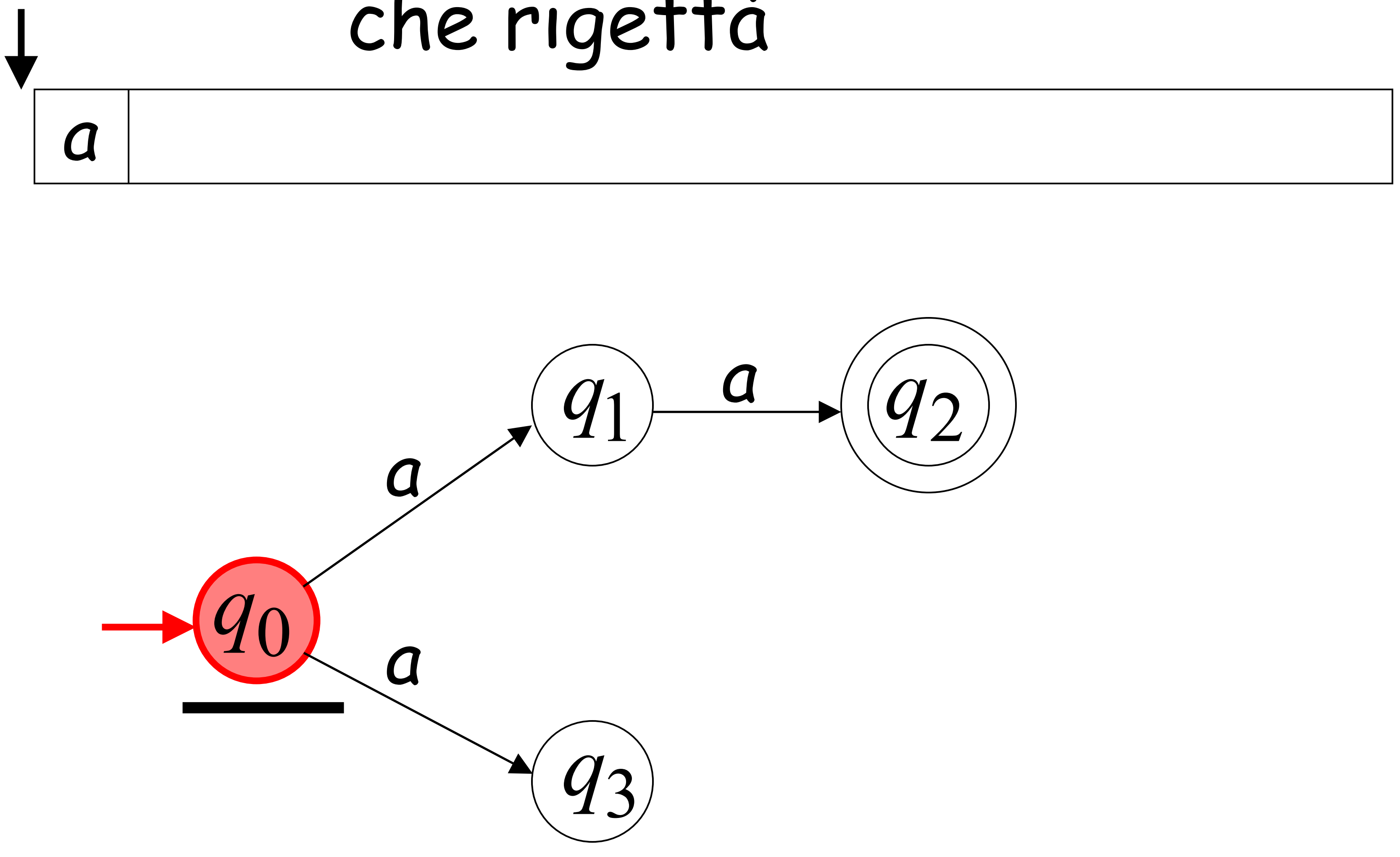
Perchè la Computazione accetta aa



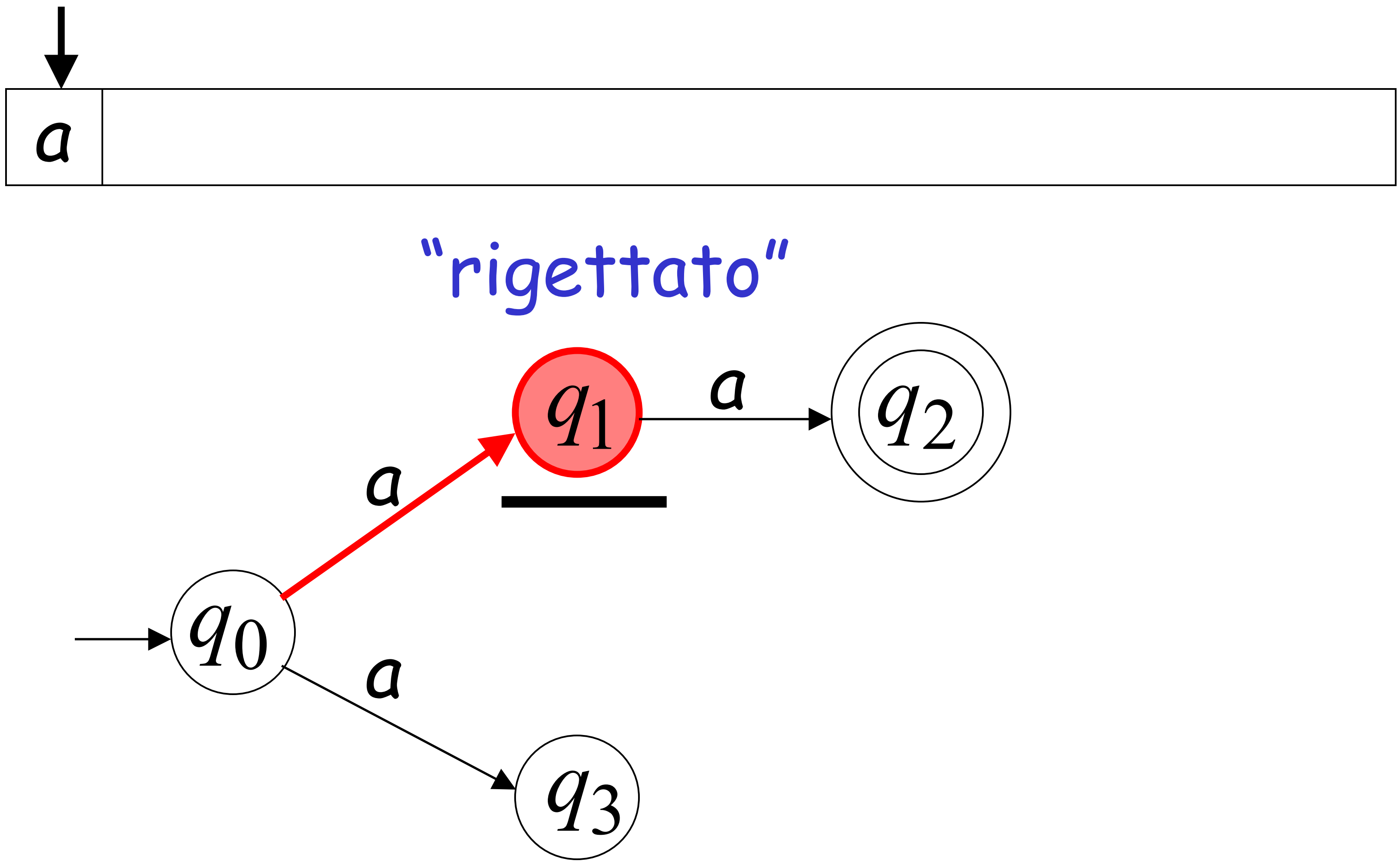
"rigettato"

Questa computazione è ignorata

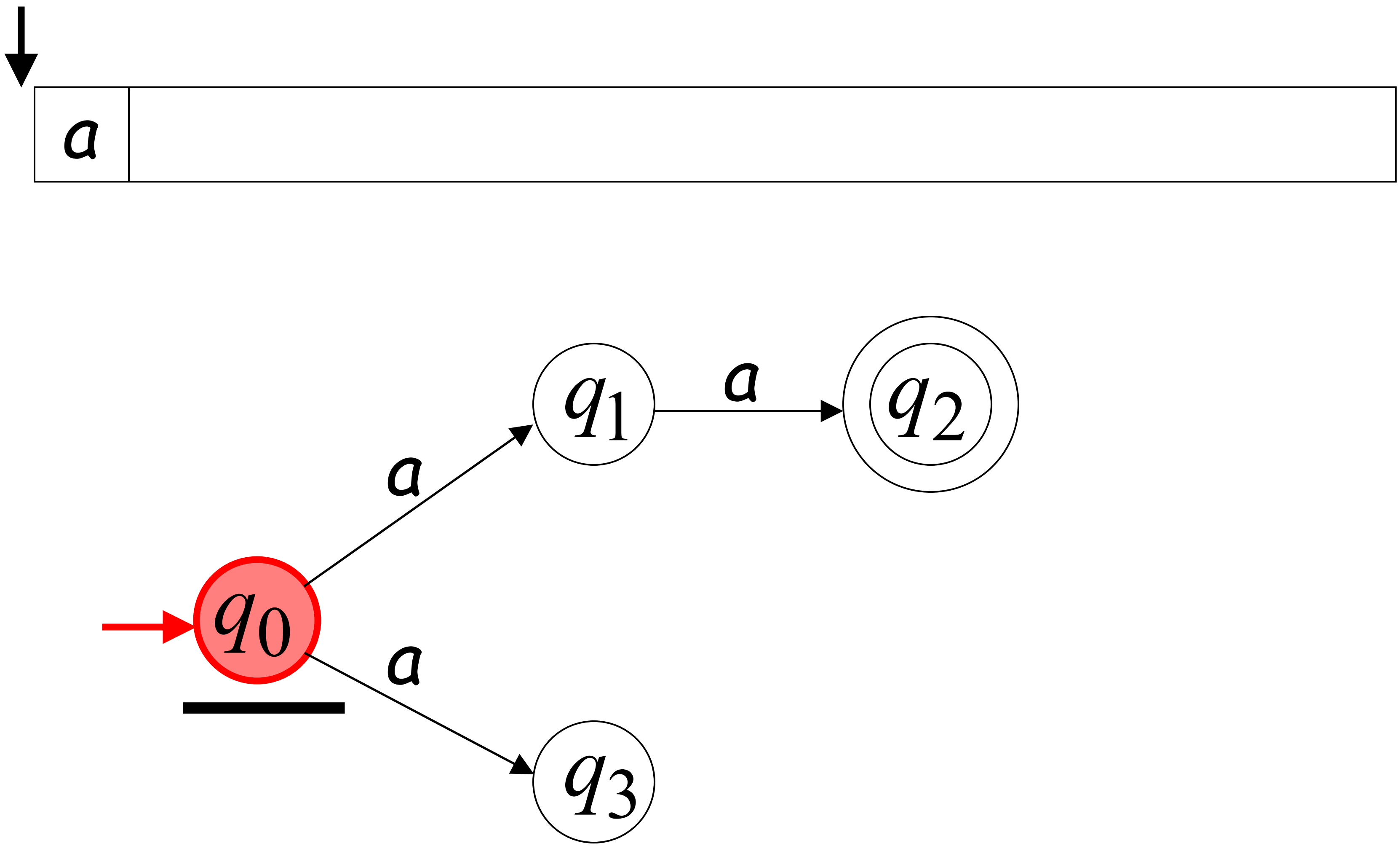
Esempio computazione che rigettà



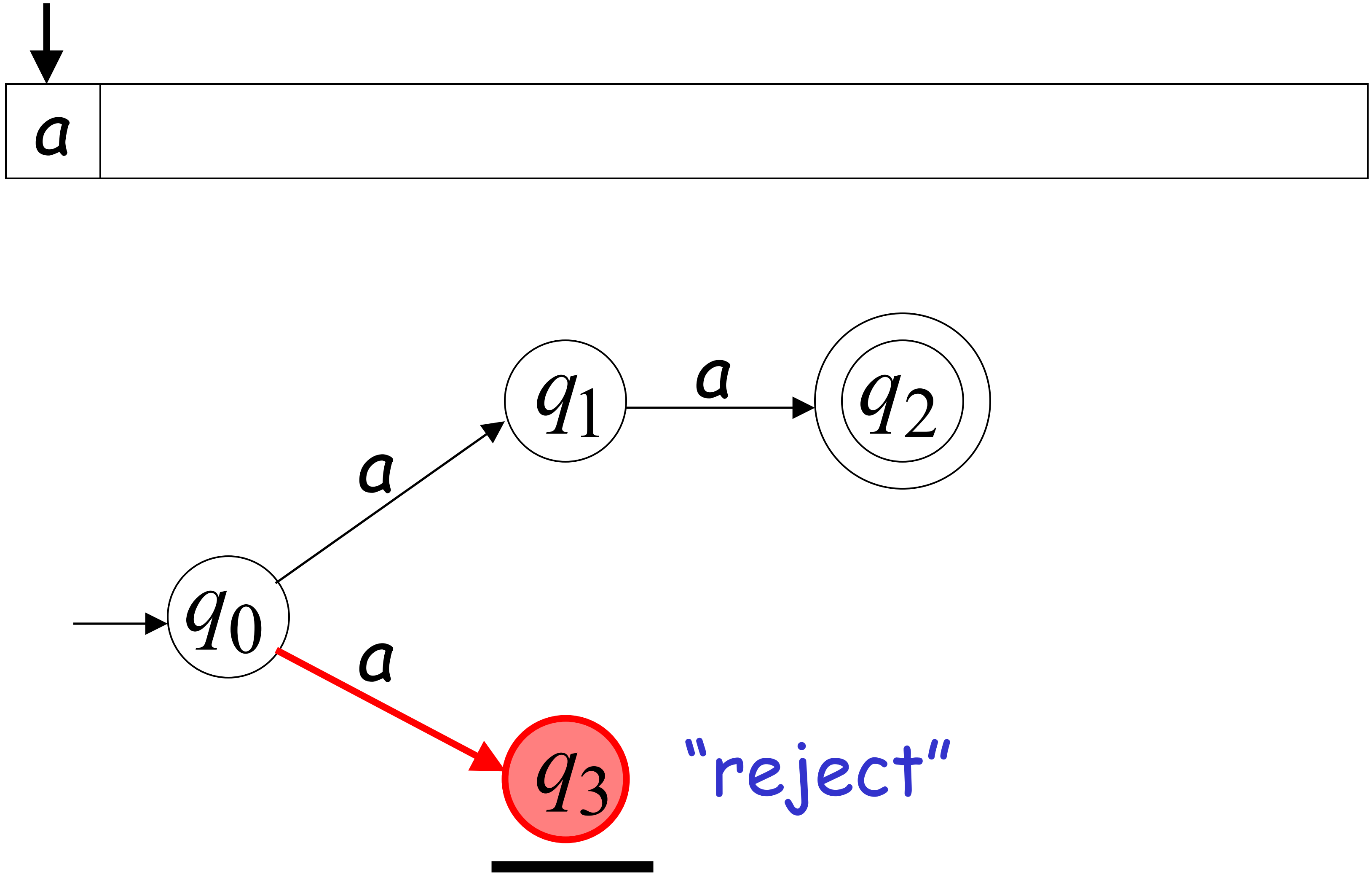
Prima scelta



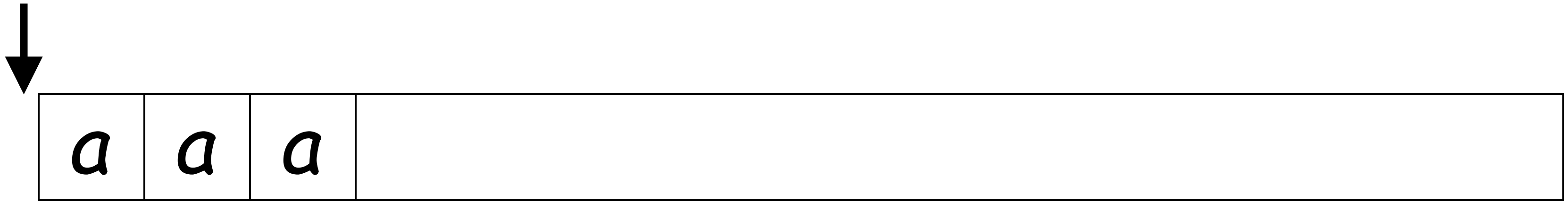
Seconda scelta

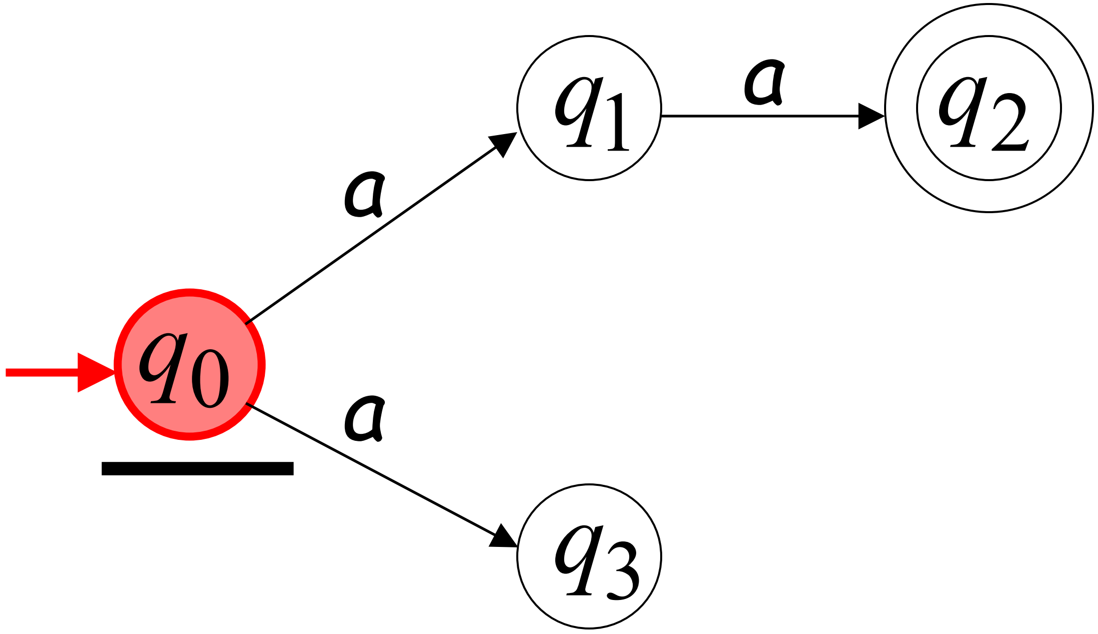


Seconda scelta

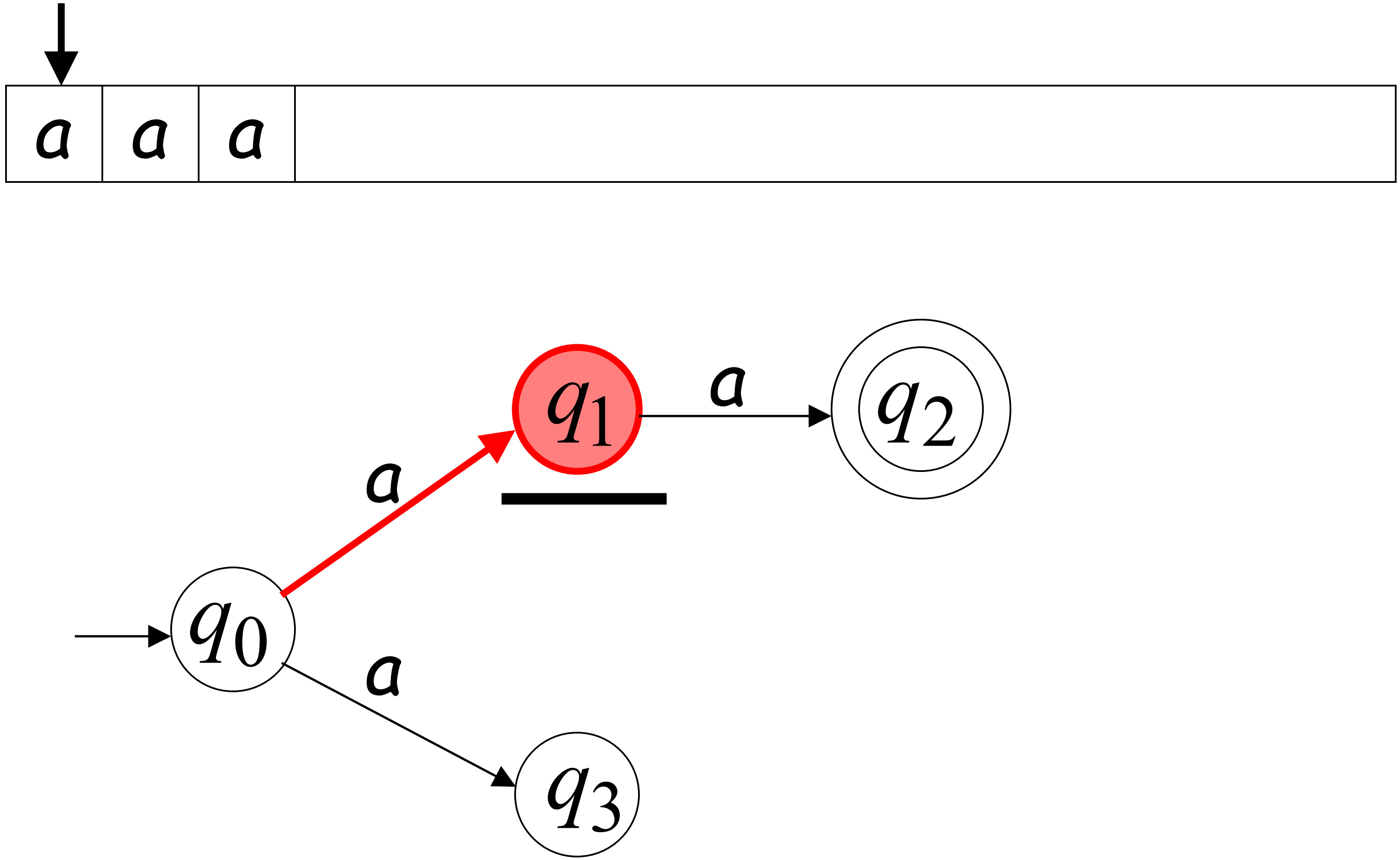


Un altro esempio

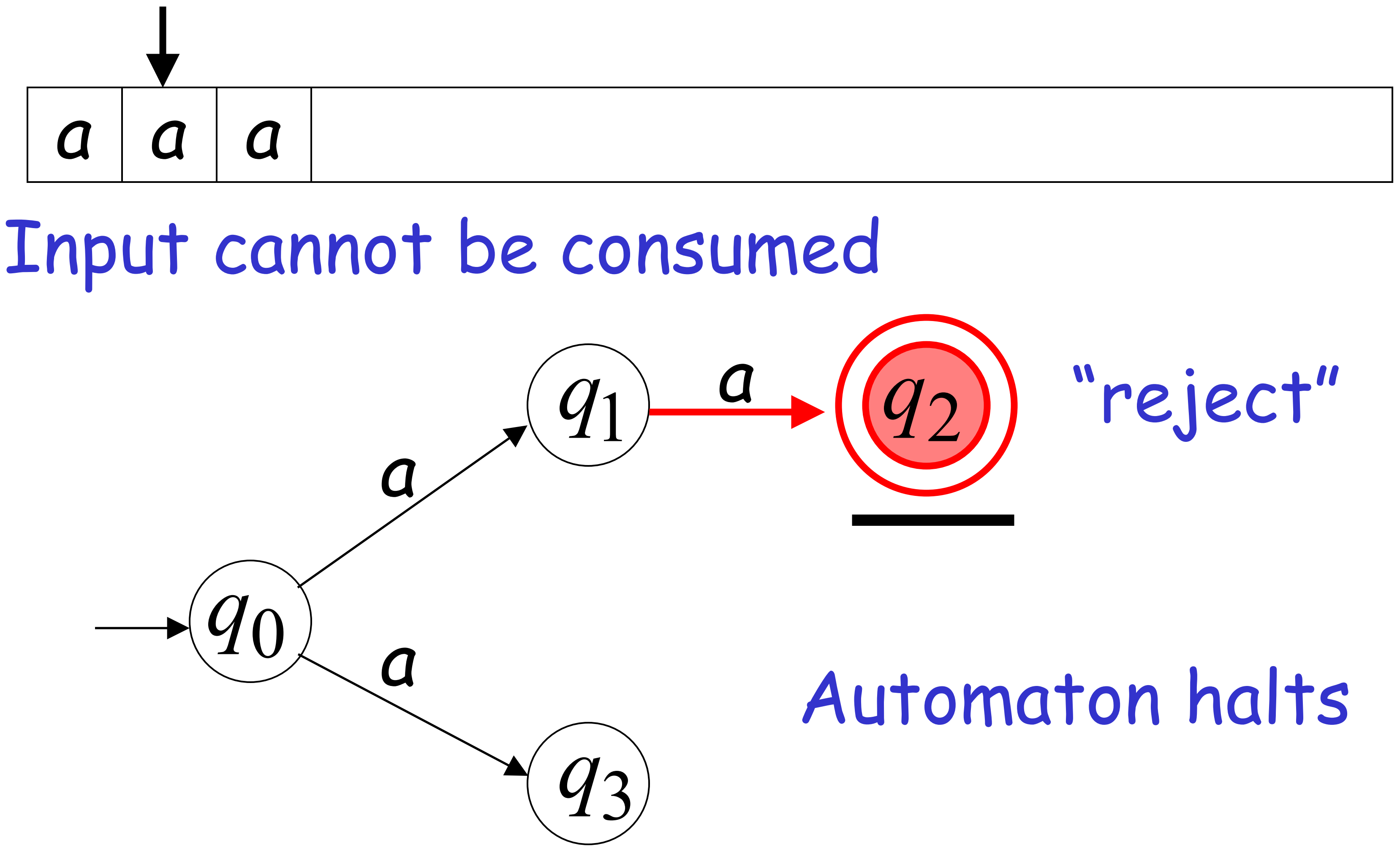




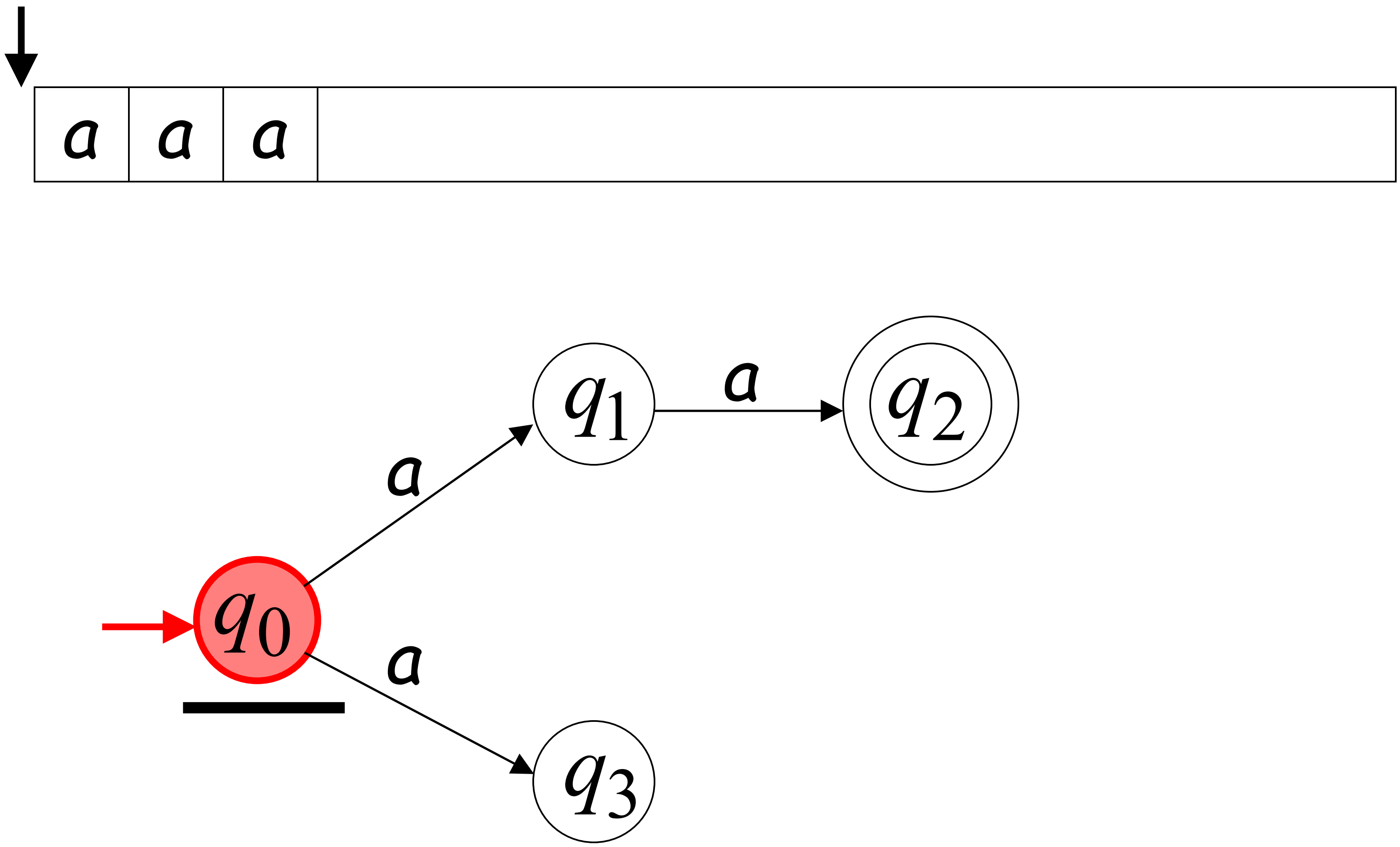
Prima scelta



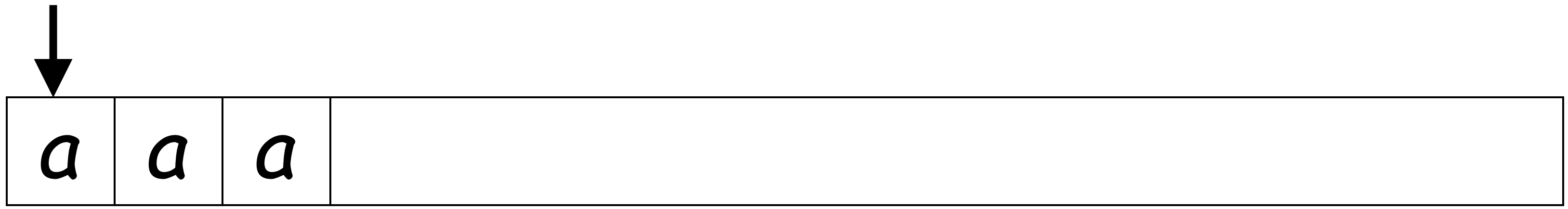
First Choice



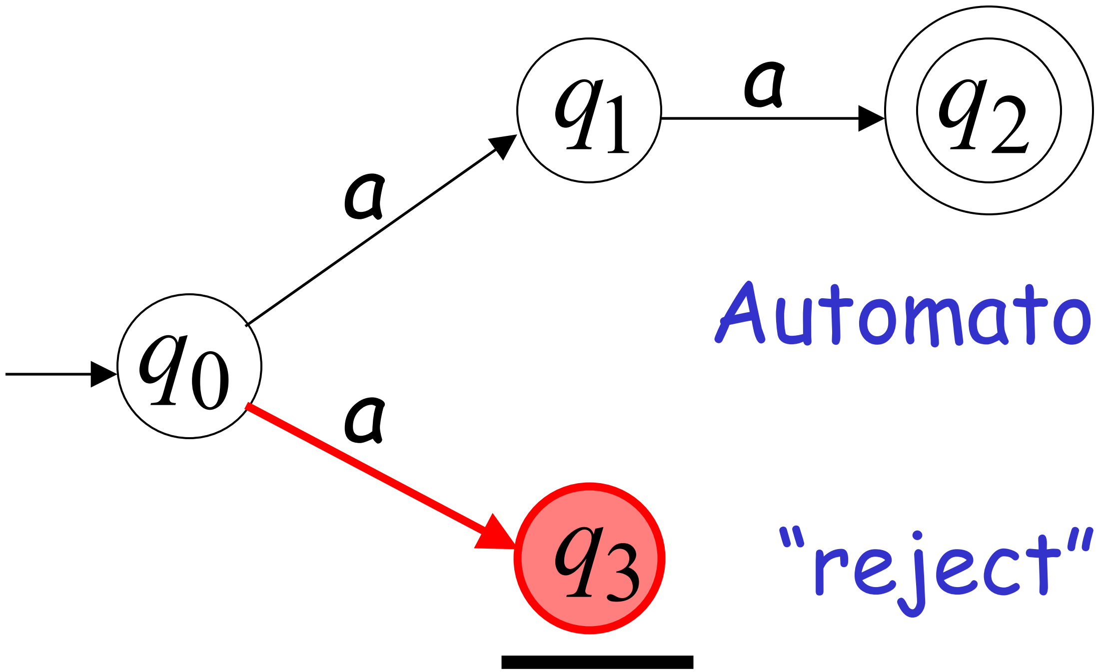
Second Choice



Second Choice



Input non viene tutto consumato



Automaton halts

“reject”

An NFA rejects a string:

Se non vi è una computazione del NFA che accetta la stringa.

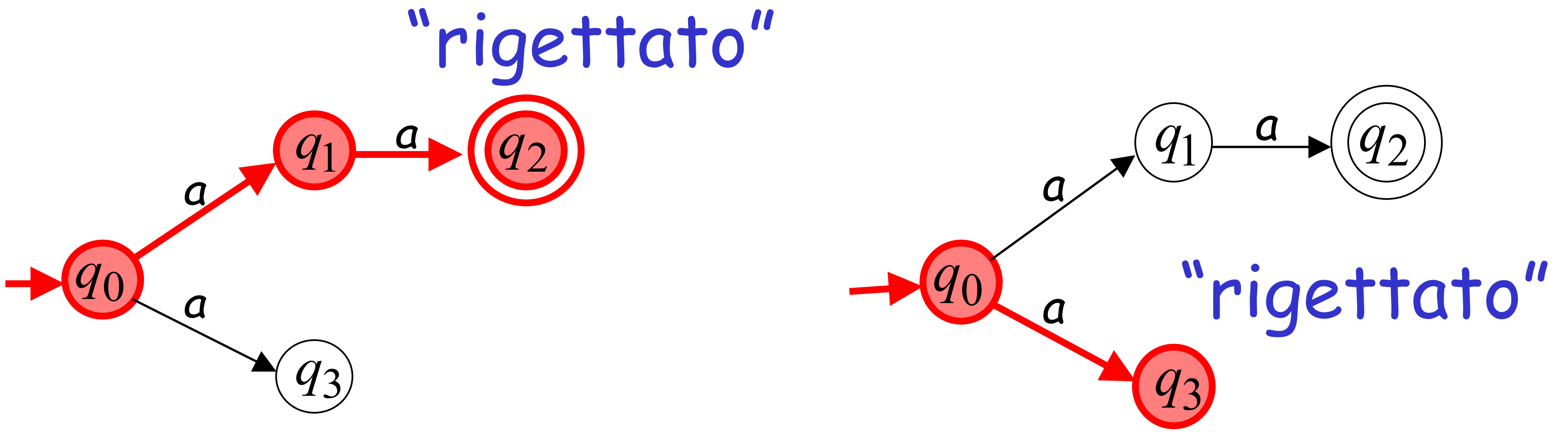
Per ogni computazione:

- tutto l'input è consumato e l'automa
- non ha raggiunto uno stato finale

O

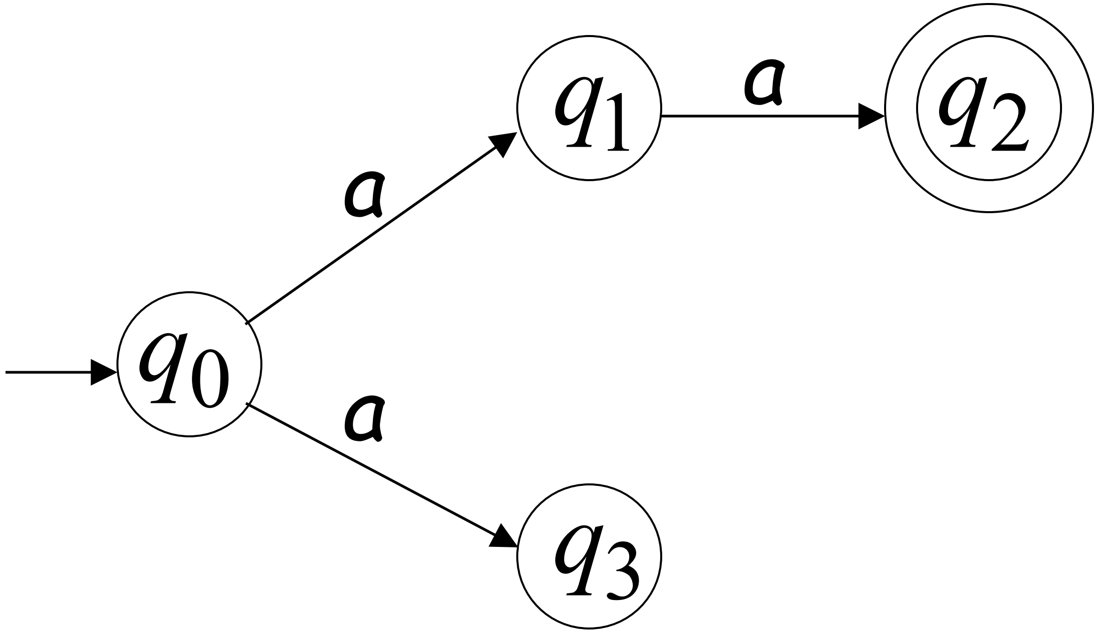
- L'input non è stato tutto consumato

aaa È rigettato dal NFA:

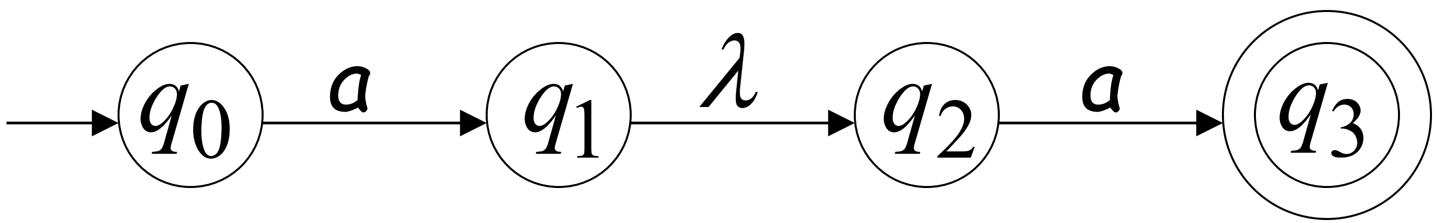


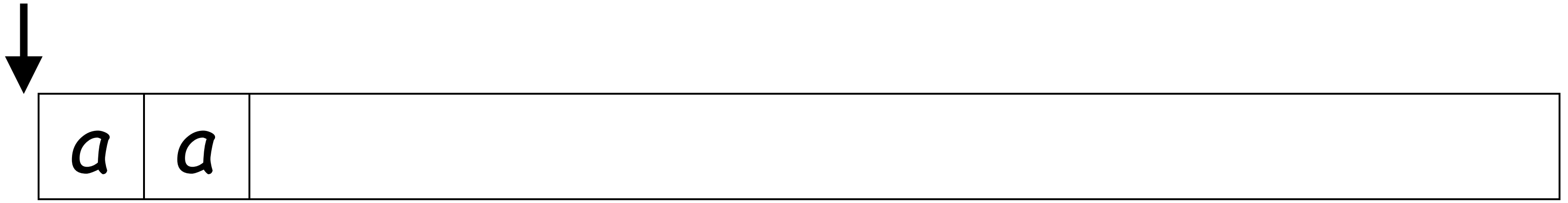
Tutte le possibili computazioni
non raggiungono uno stato finale

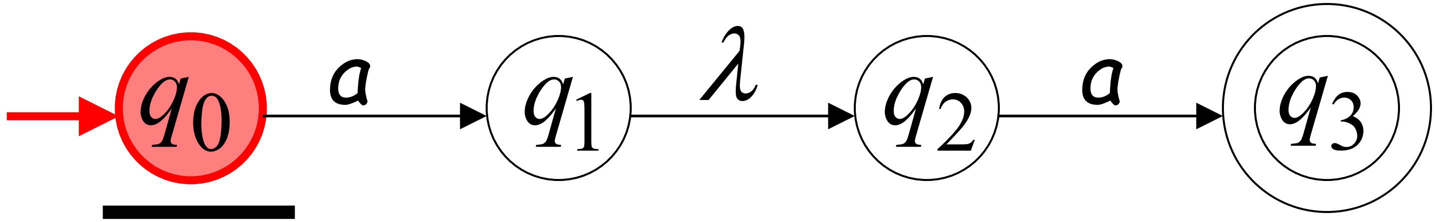
Linguaggio accettato: $L = \{aa\}$

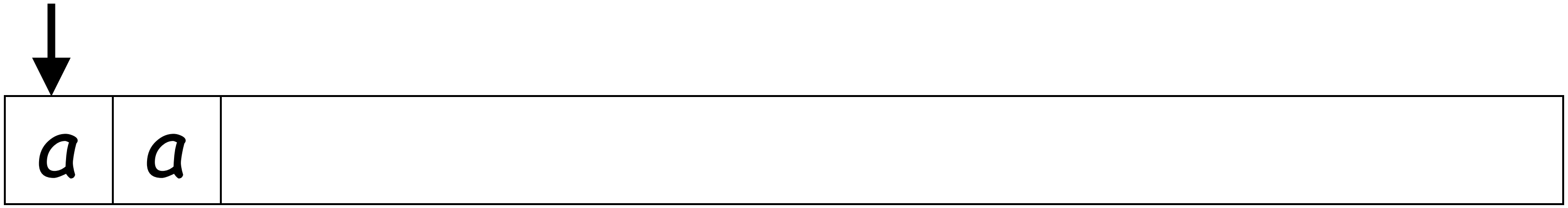


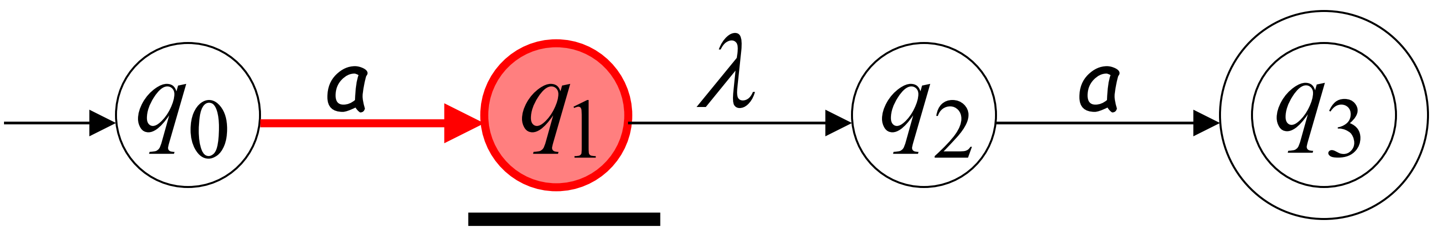
Lambda transizione



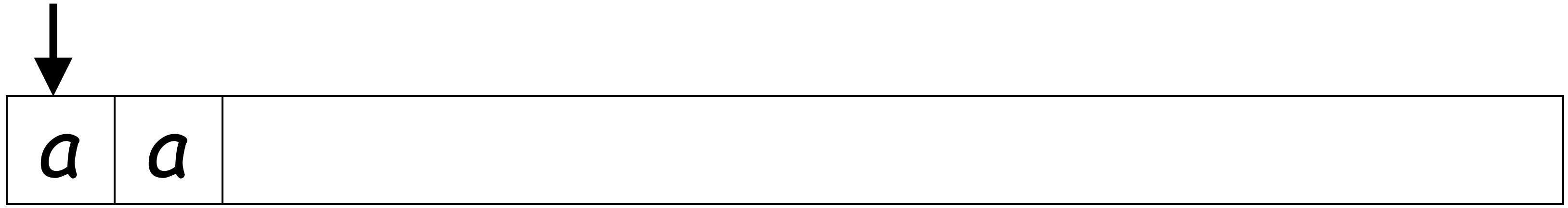

a
a

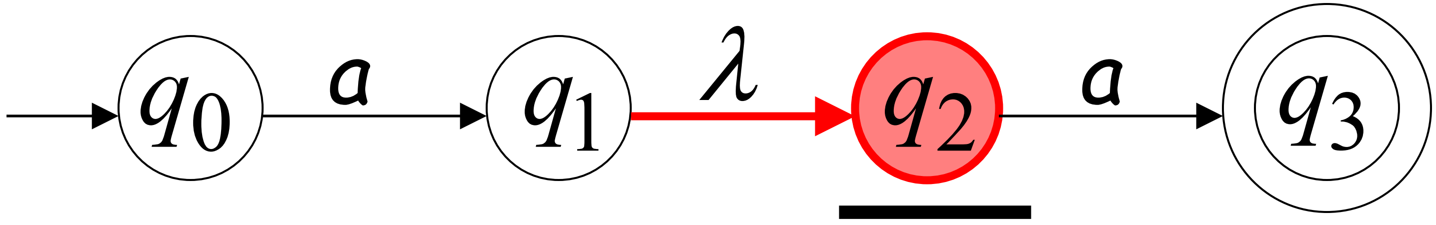

q_0
a
q_1
λ
q_2
a
q_3


a
a

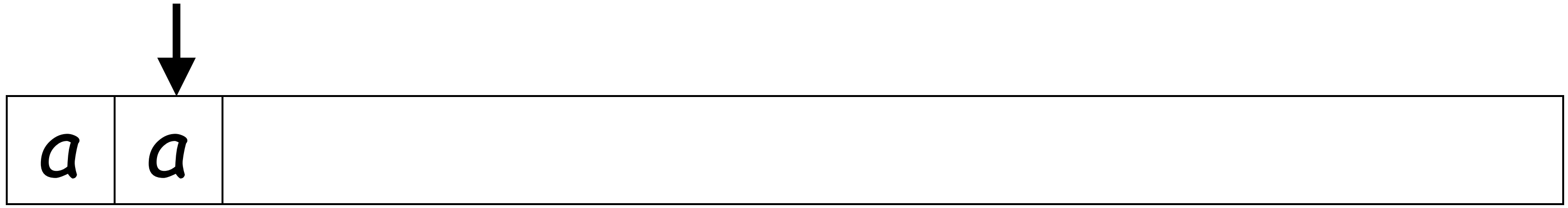

q_0
a
q_1
λ
q_2
a
q_3

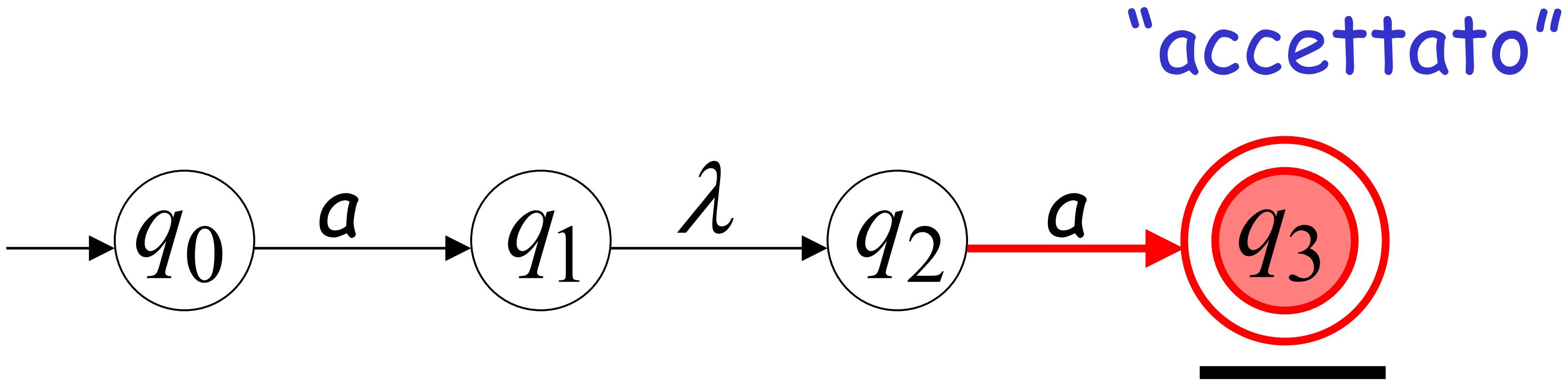
La testina dell'input non si muove





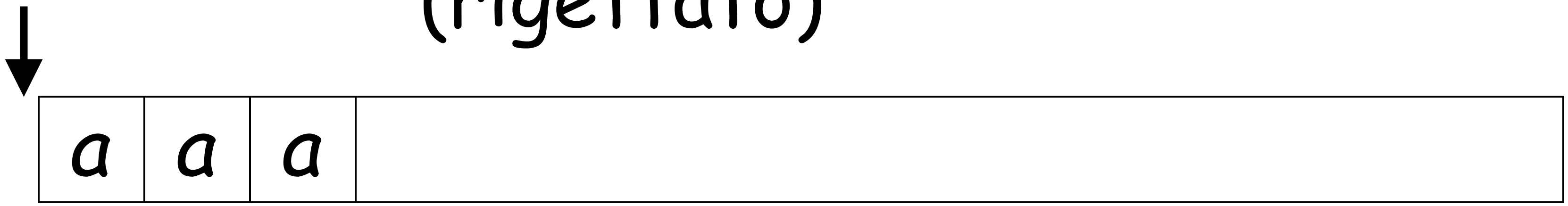
Tutto l'input è esaminato

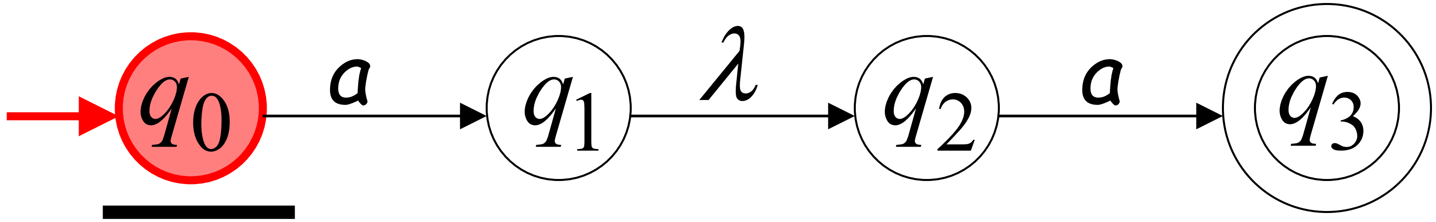


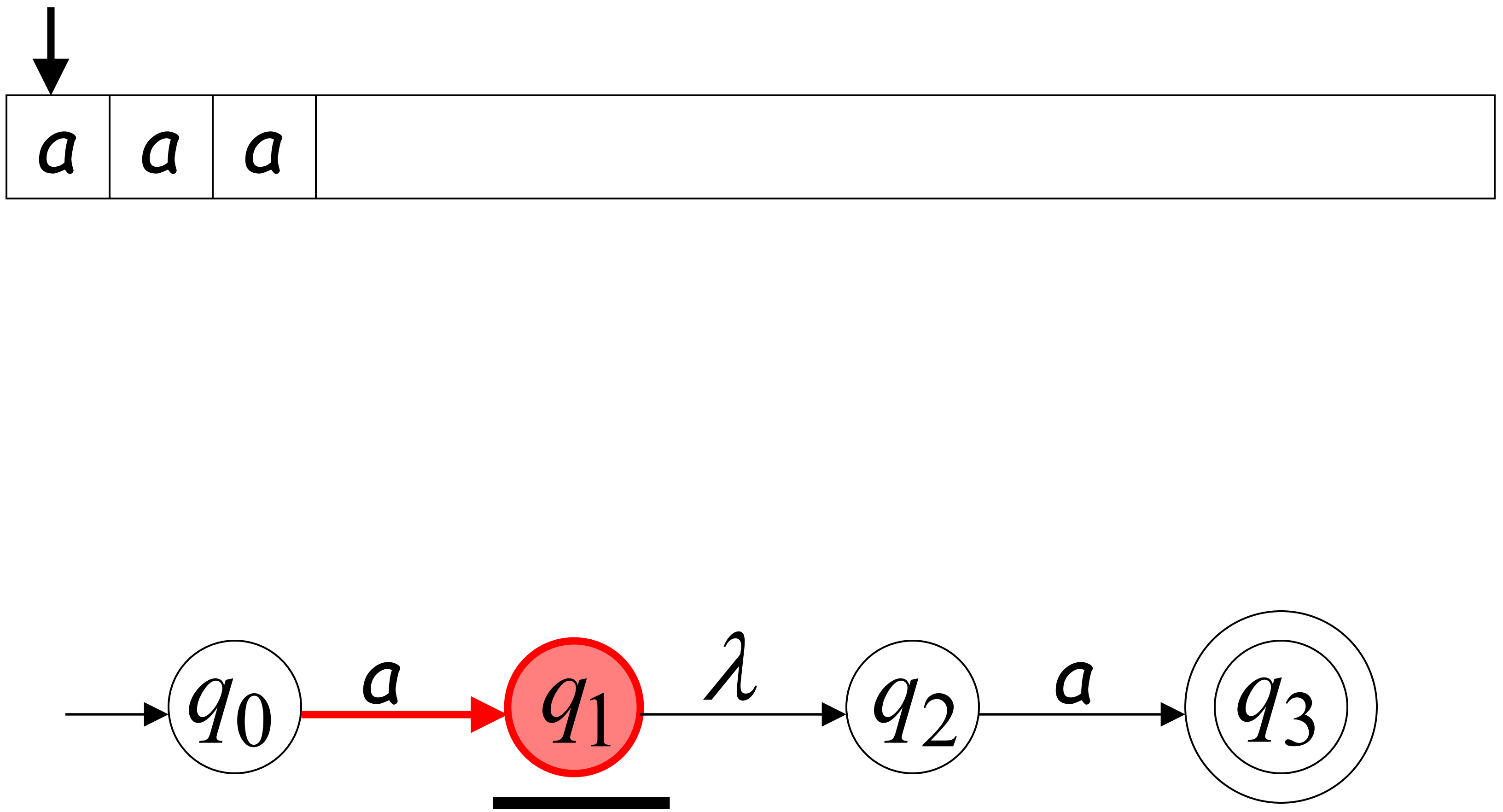


stringa aa è accettata

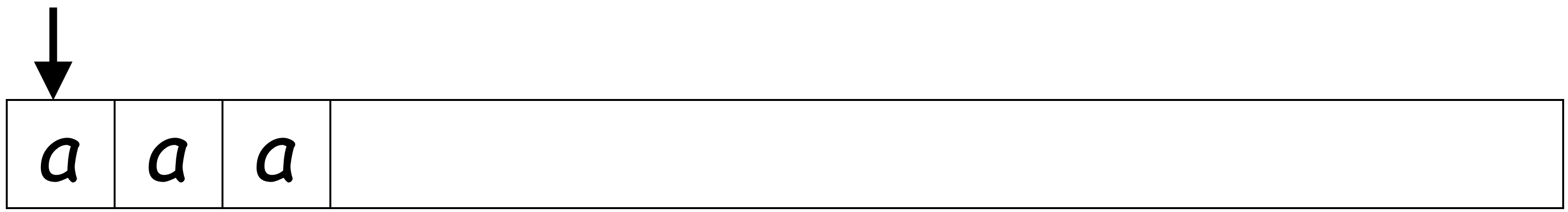
Esempio di non accettazione (rigettato)

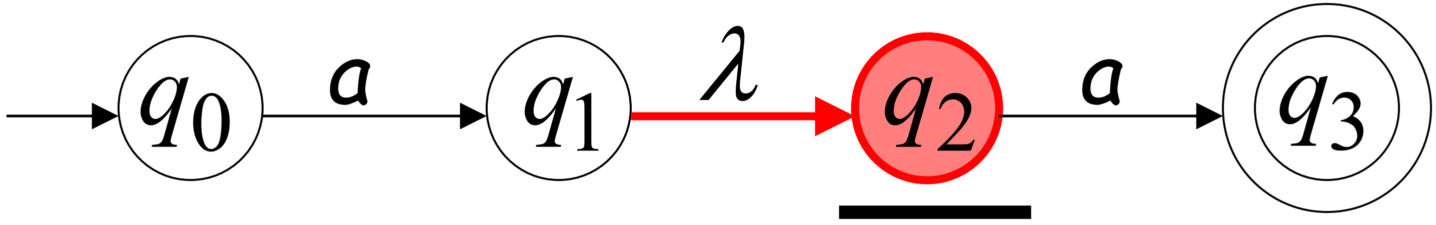




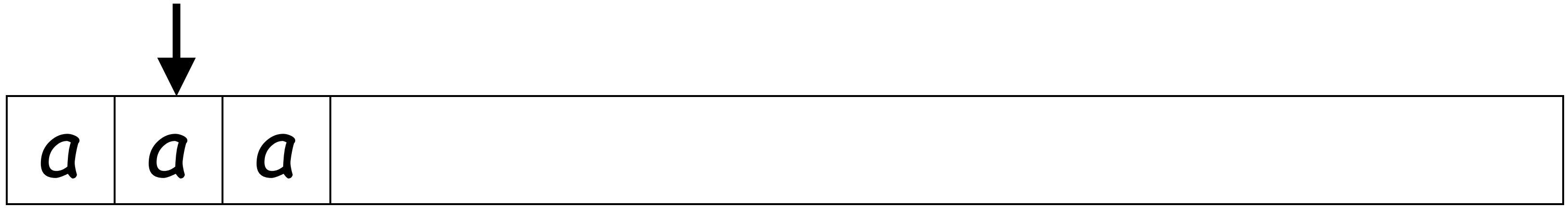

a
a
a
q_0
a
q_1
λ
q_2
a
q_3

(la testina non si muove)



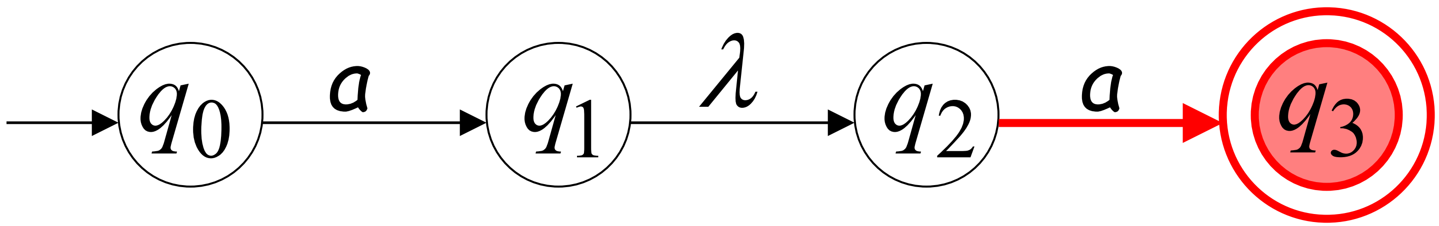


Input non viene analizzato tutto



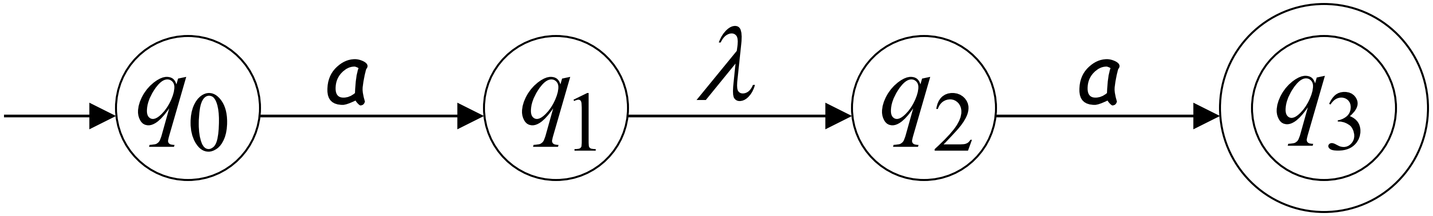
Automa si ferma

"rigettato"



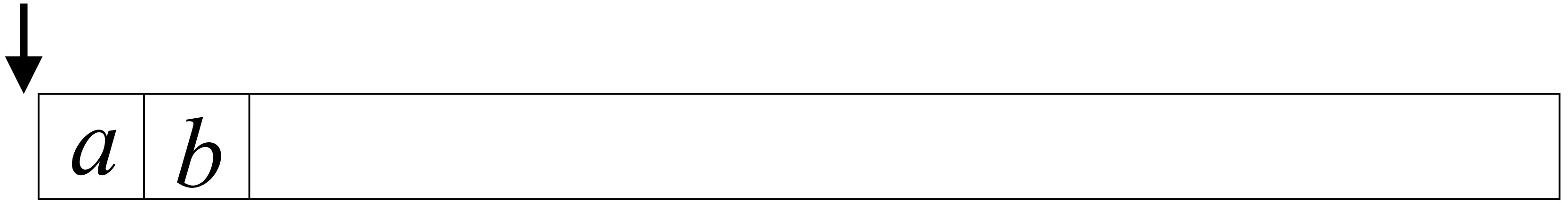
stringa **aaa** è rigettata

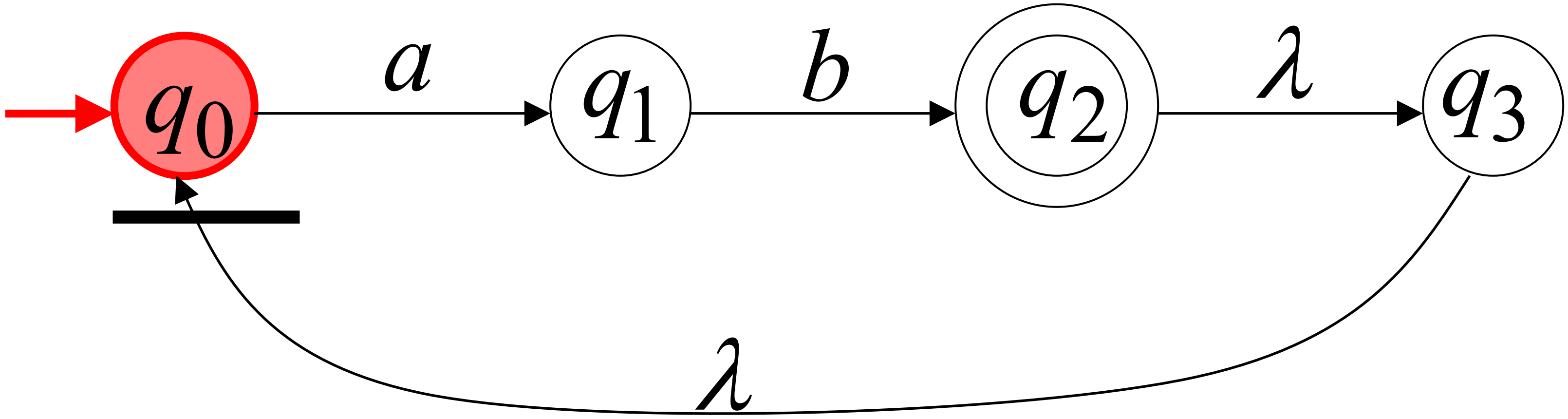
Linguaggio accettato: $L = \{aa\}$

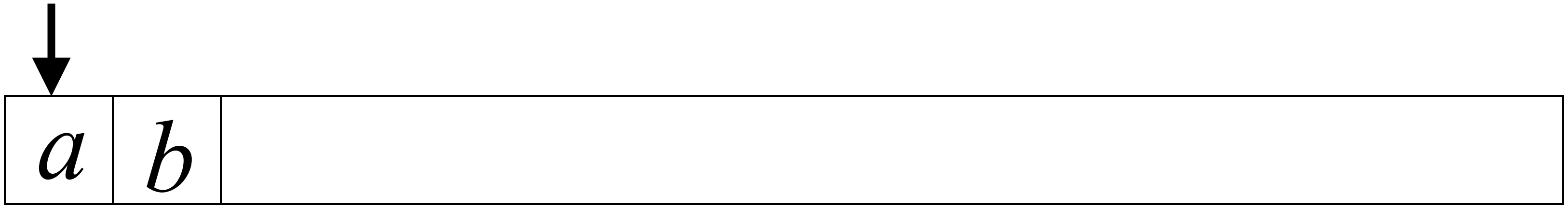


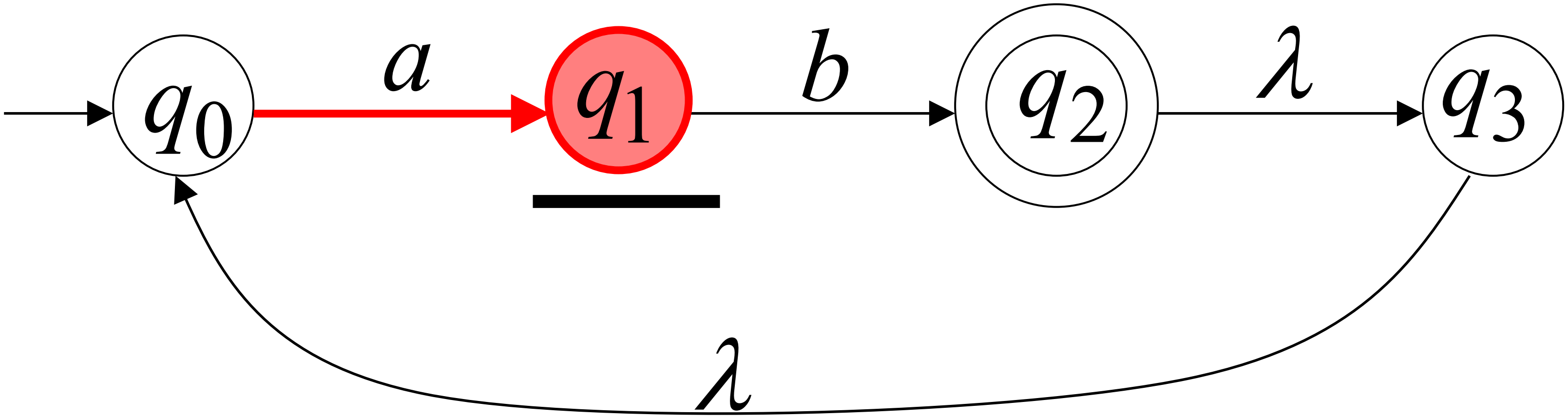
Esiste una computazione si
Per ogni computazione no

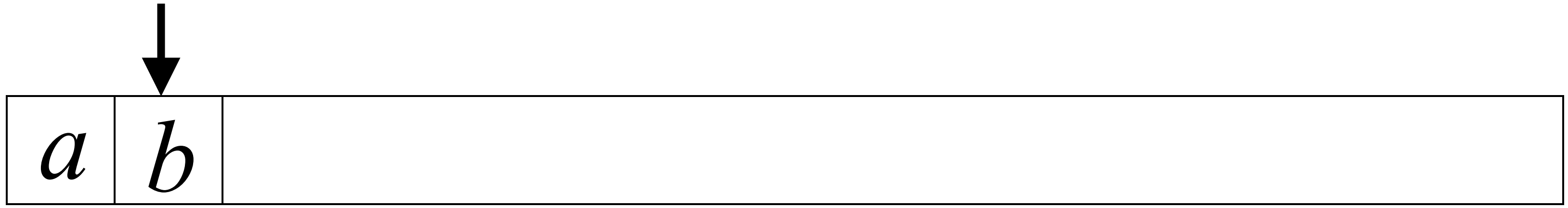
Un altro NFA

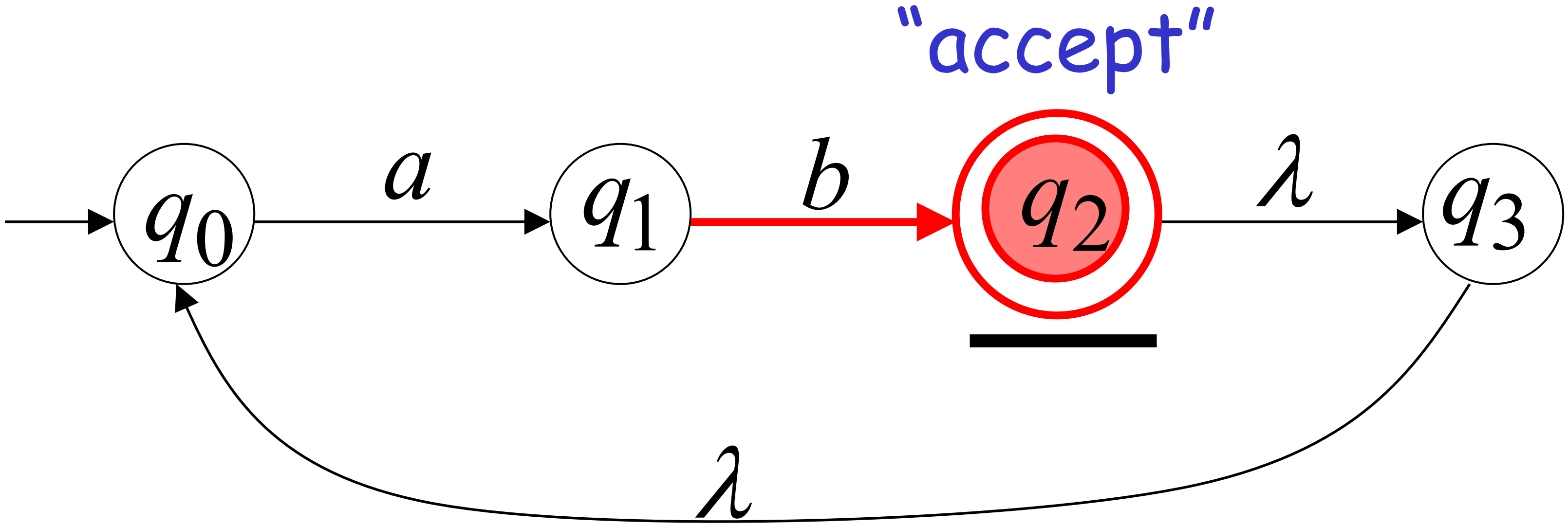




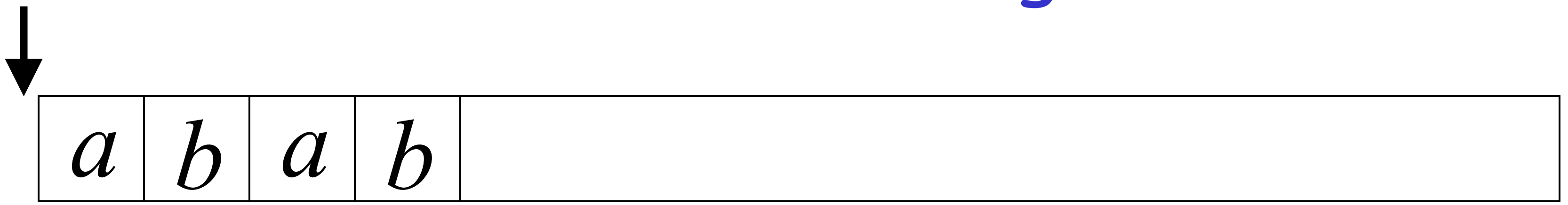

a
b

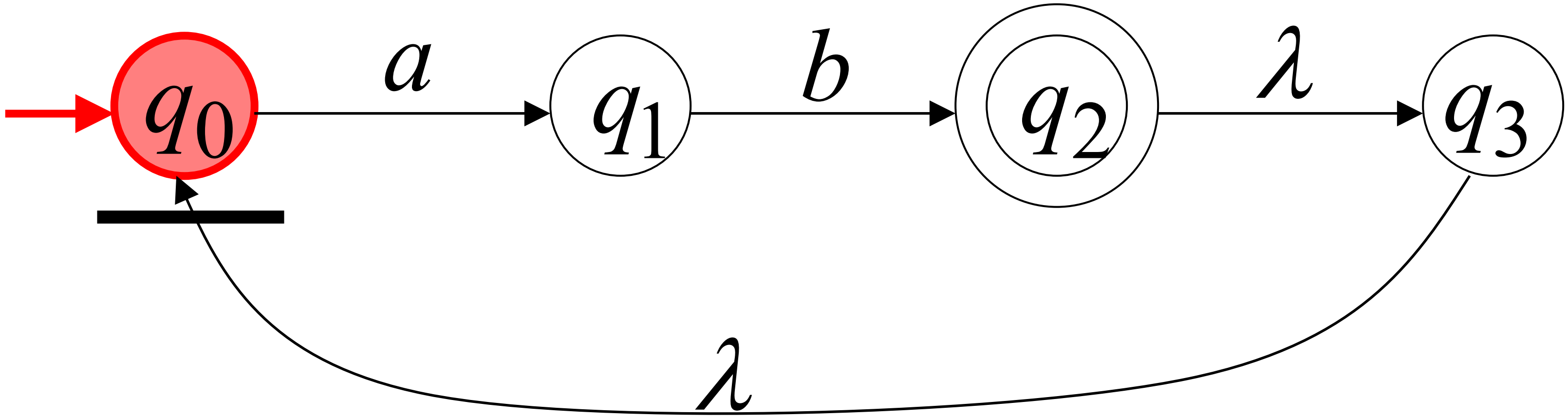

q0
a
q1
b
q2
λ
q3
λ

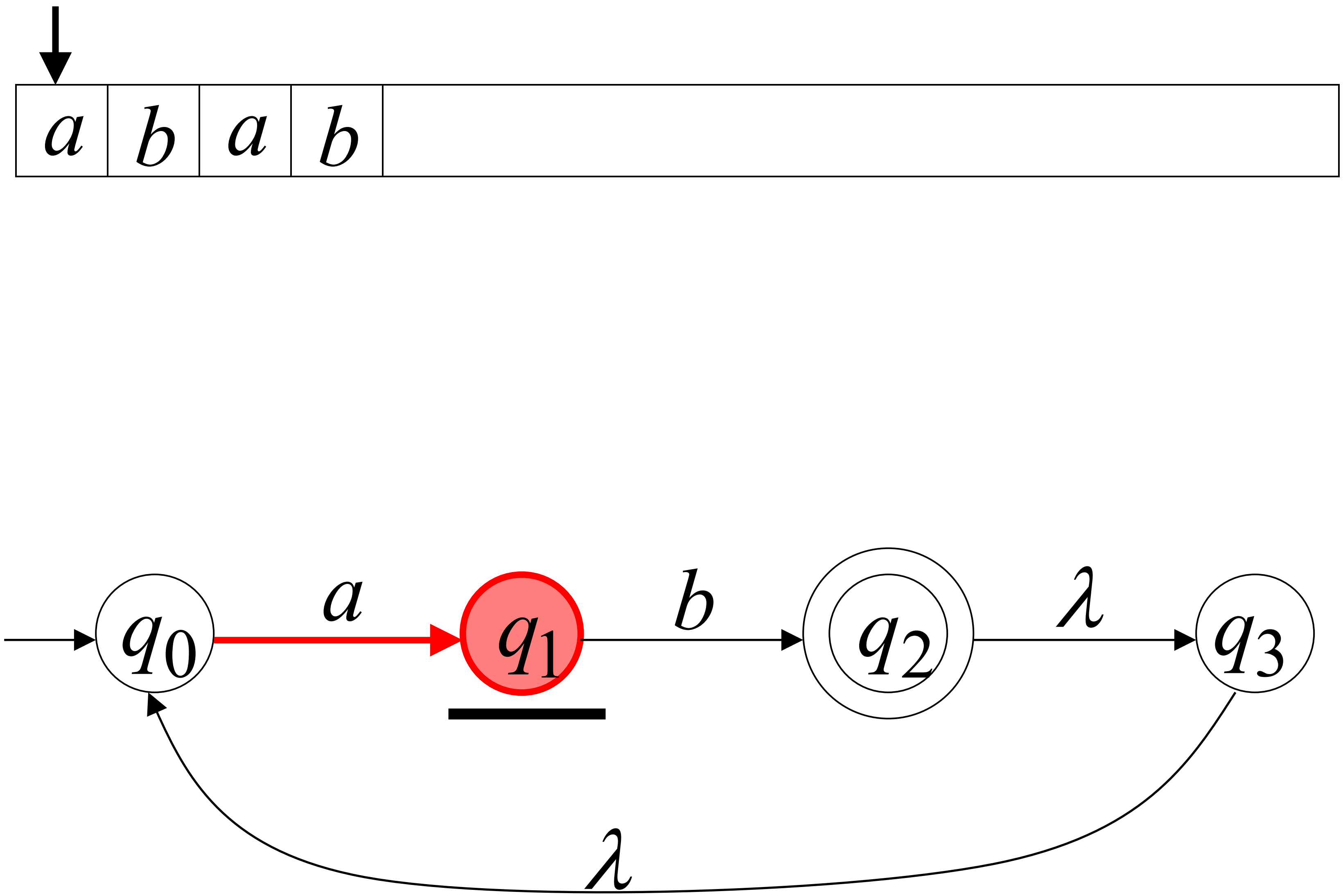

a
b

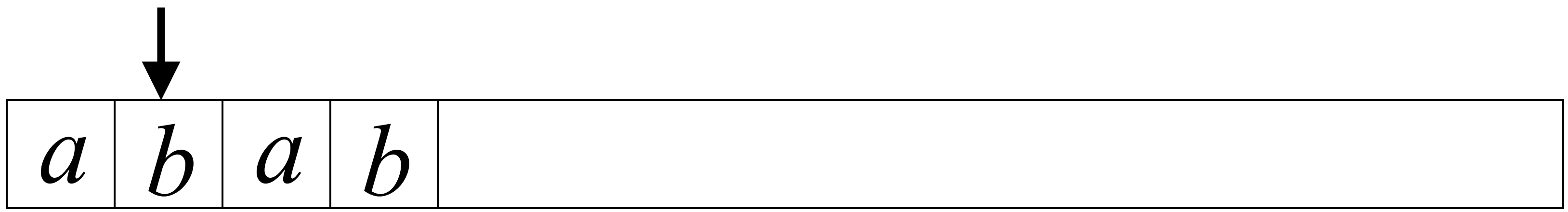

"accept"
q_0
q_1
q_2
q_3
a
b
λ
λ

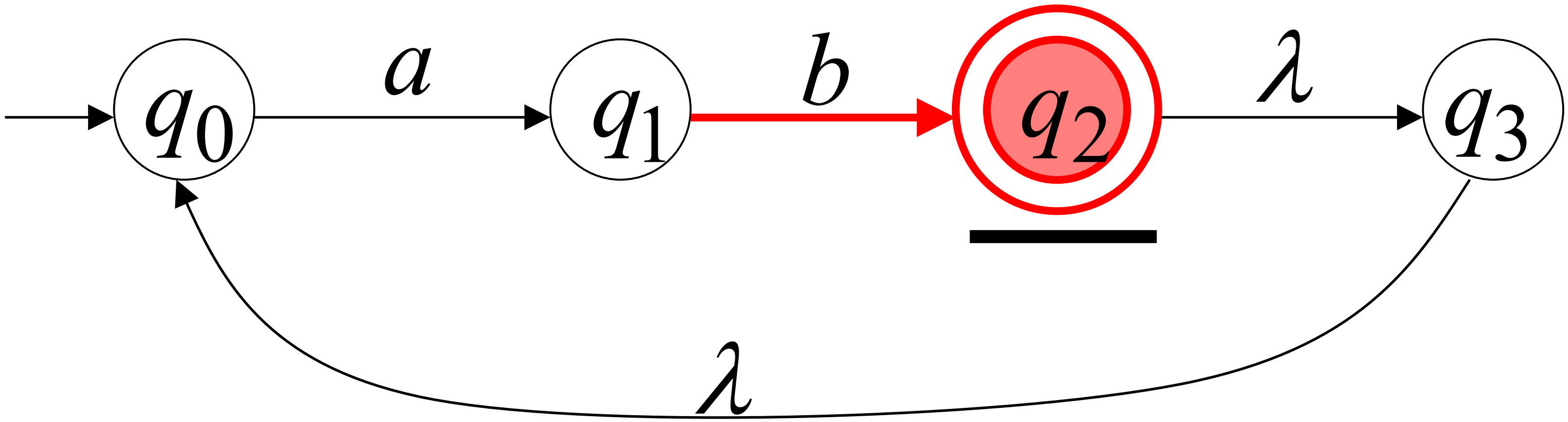
Un altra stringa

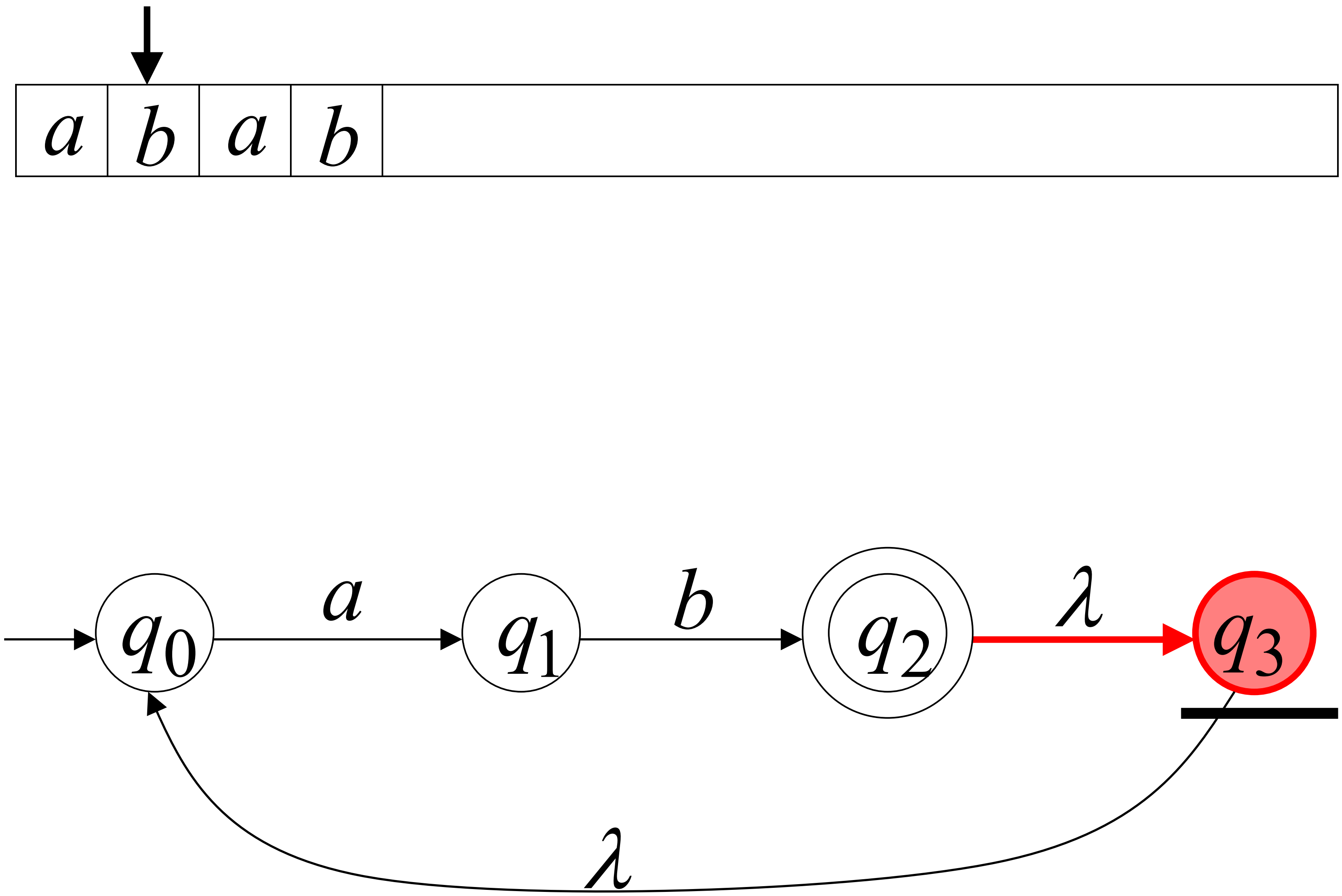


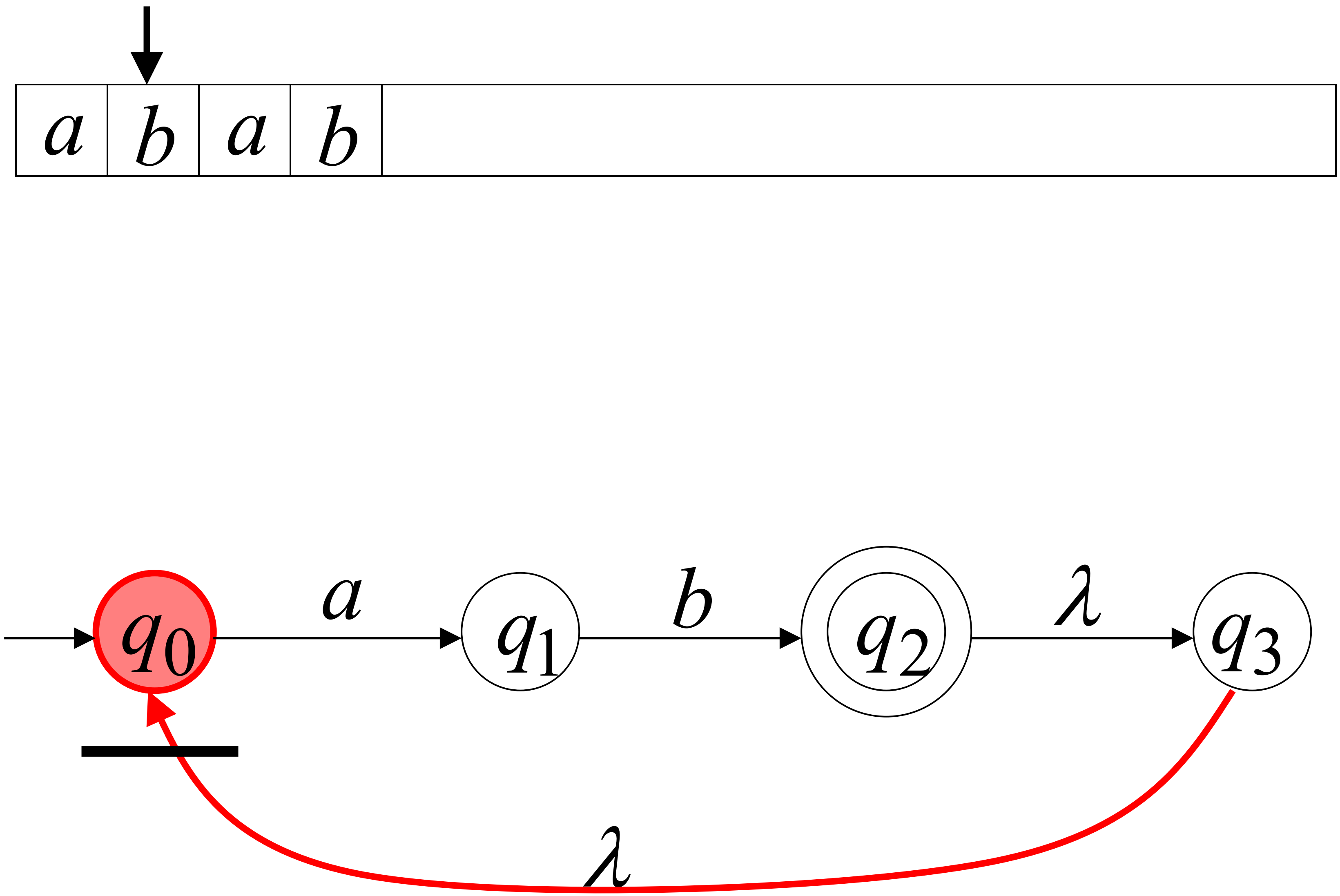


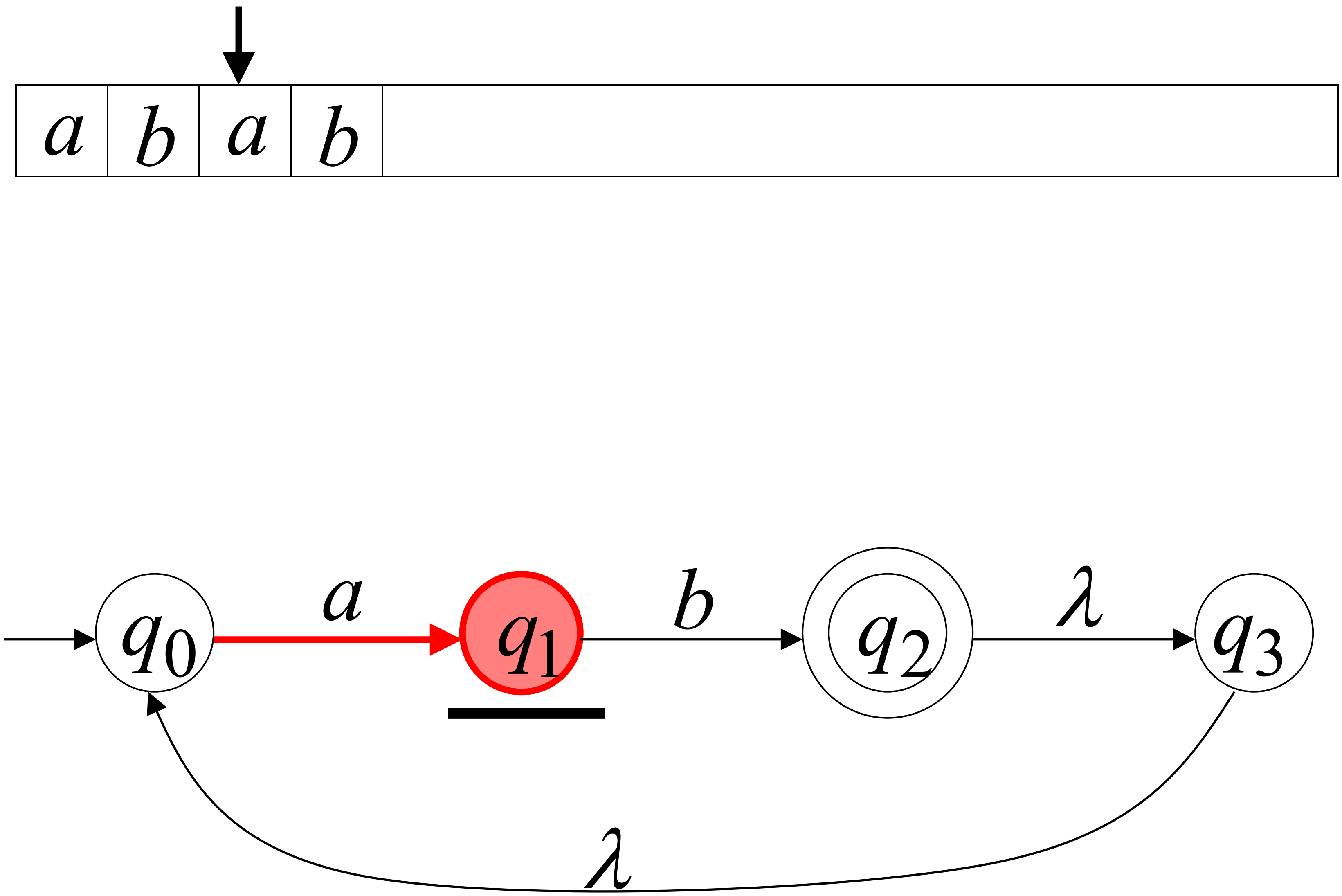

a
b
a
b
q_0
q_1
q_2
q_3
a
b
λ
λ

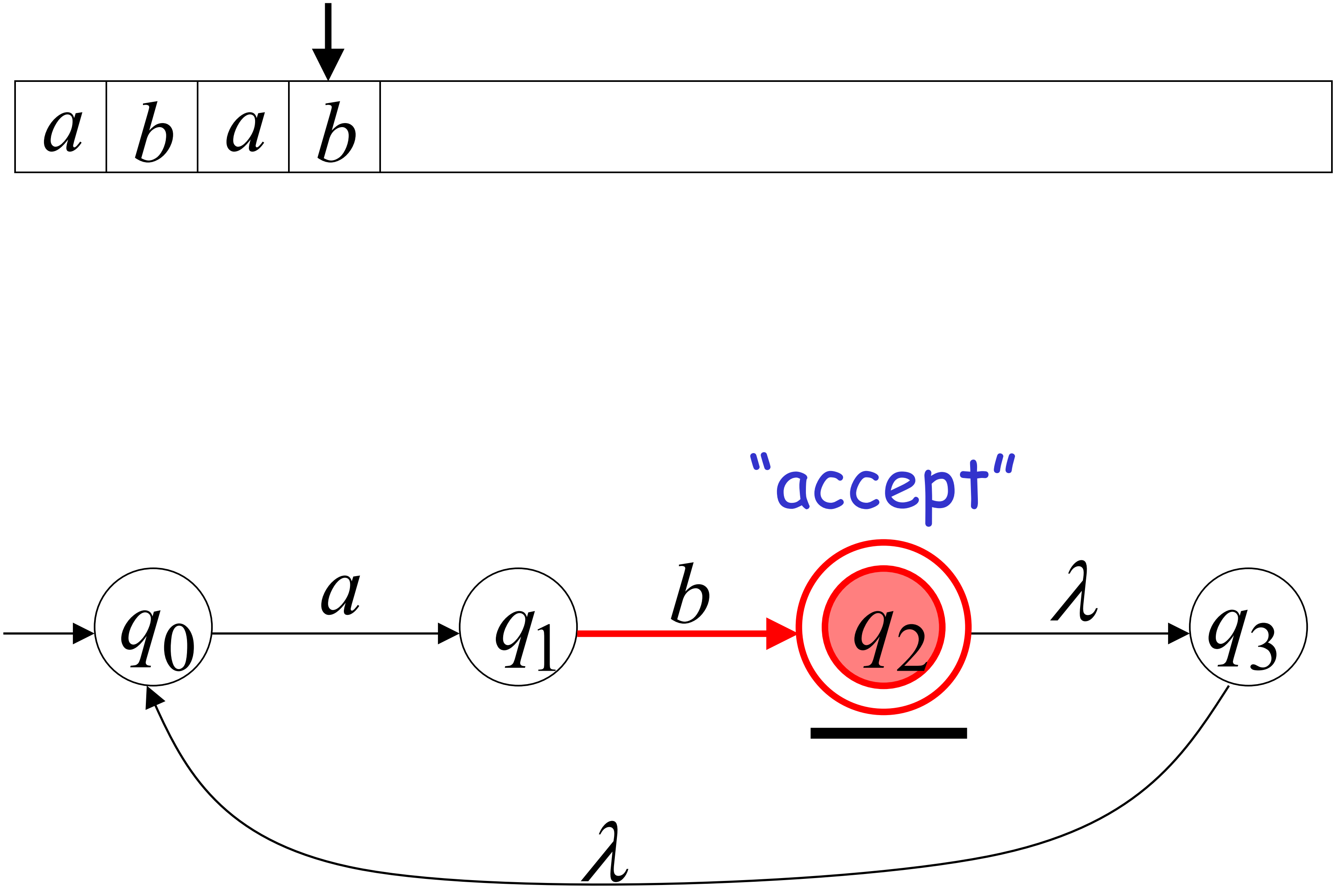

a
b
a
b


q_0
a
q_1
b
q_2
λ
q_3
λ


a
b
a
b
q_0
a
q_1
b
q_2
λ
q_3
λ


a
b
a
b
a
b
λ
λ
q_0
q_1
q_2
q_3


a b a b
a
b
λ
λ
q_0
q_1
q_2
q_3

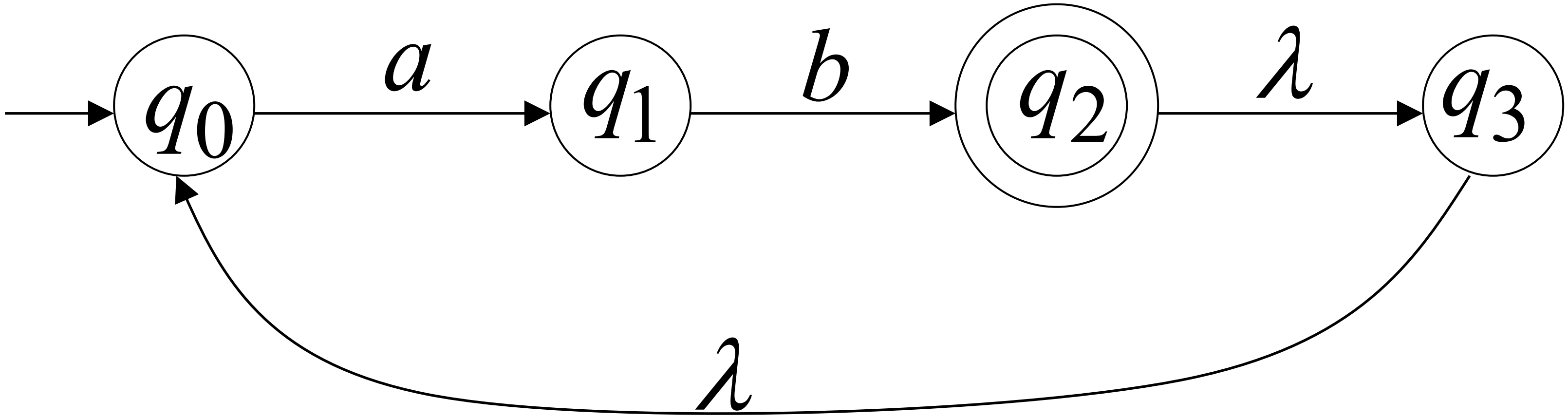


a
b
a
b
"accept"
q_0
a
q_1
b
q_2
λ
q_3
λ

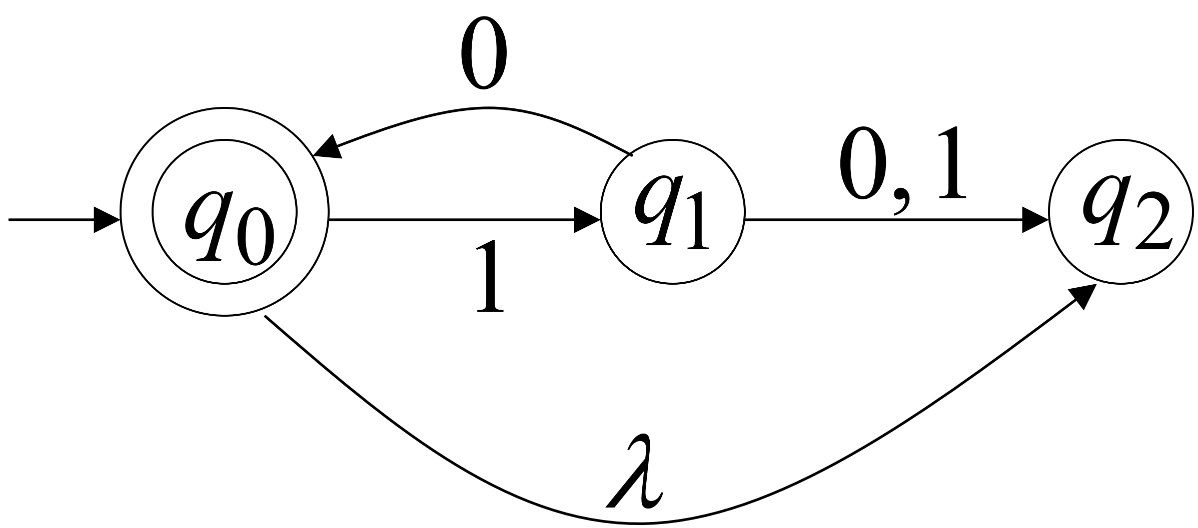
Linguaggio accettato

$$L = \{ab,\ abab,\ ababab,\ \ldots\}$$

$$= \{ab\}^{+}$$



NFA esempio

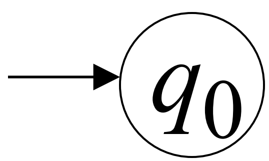


Remarks:

- Il simbolo λ non appare mai
- sul nastro di input

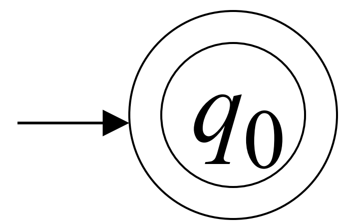
- Semplici automata:

M_1



$$L(M_1) = \{\}$$

M_2



$$L(M_2) = \{\lambda\}$$

Formal Definition of NFAs

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

Q : Set of states, i.e. $\{q_0, q_1, q_2\}$

Σ : Input aplhabet, i.e. $\{a, b\}$ $\lambda \notin \Sigma$

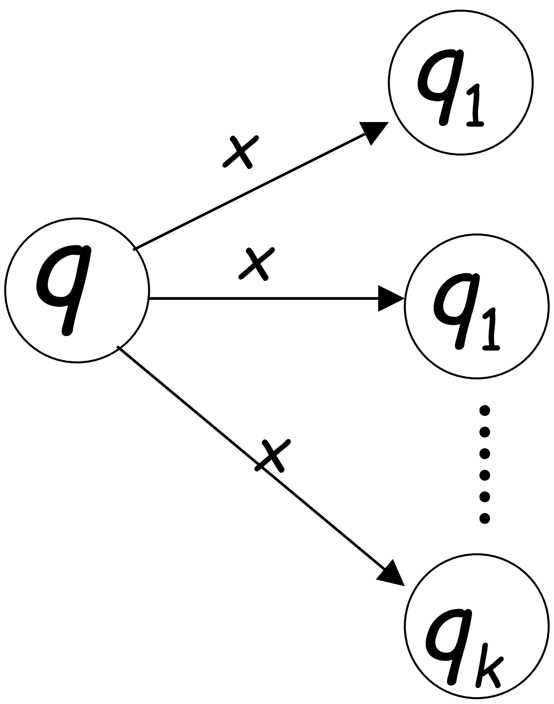
δ : Transition function

q_0 : Initial state

F : Accepting states

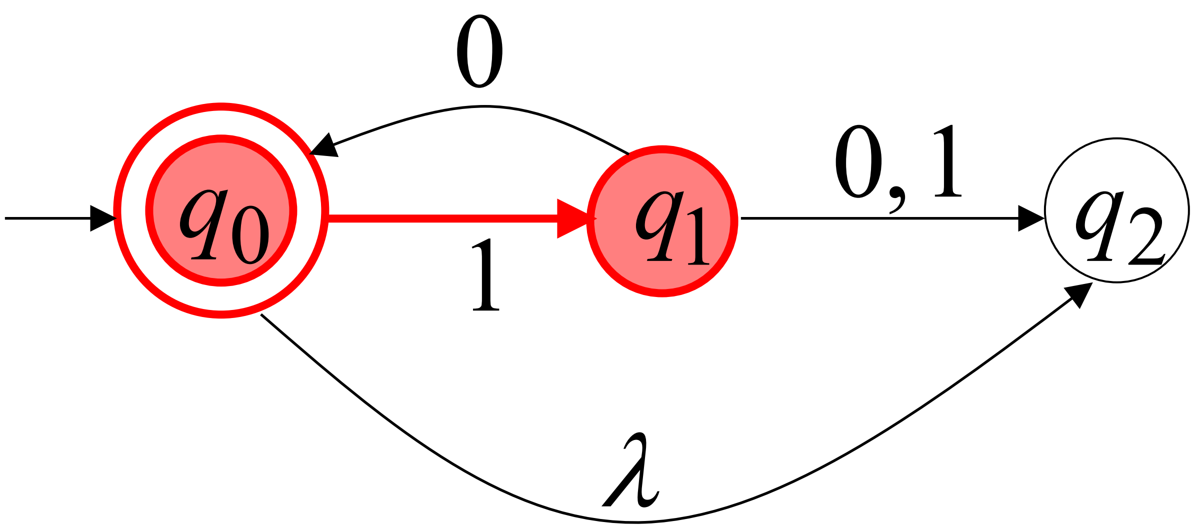
Funzione di transizione δ

$$\delta(q, x) = \{q_1, q_2, \ldots, q_k\}$$

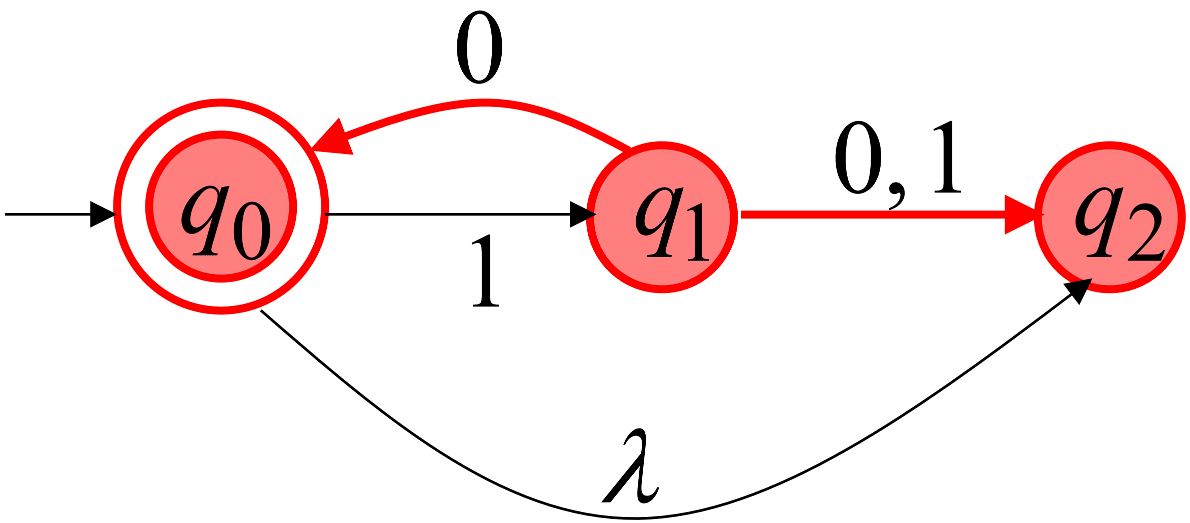


Stati risultanti con una transizione con simbolo x

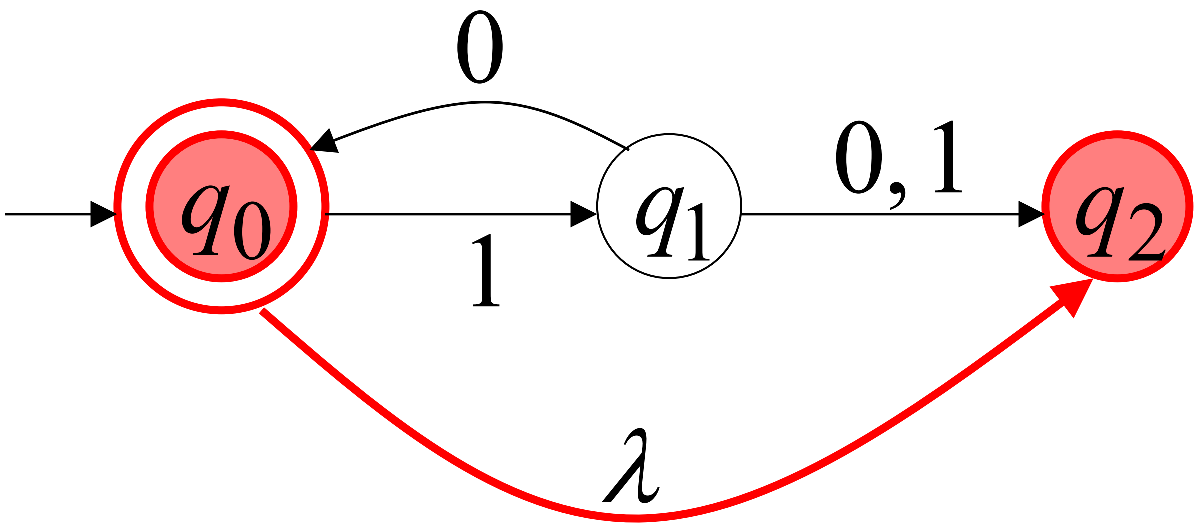
$$\delta(q_0, 1) = \{q_1\}$$



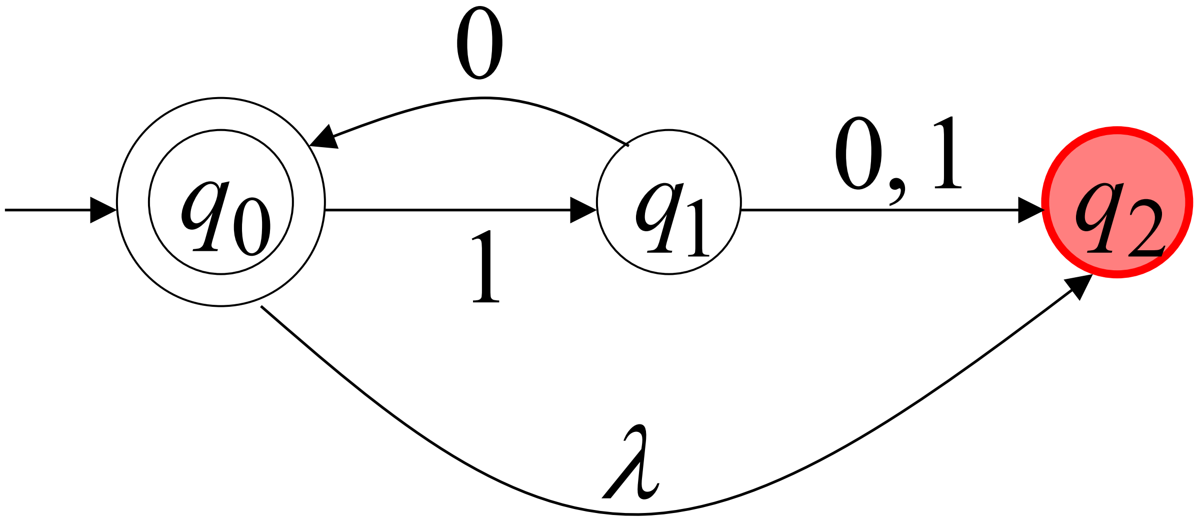
$$\delta(q_1, 0) = \{q_0, q_2\}$$



$$\delta(q_0, \lambda) = \{q_2\}$$



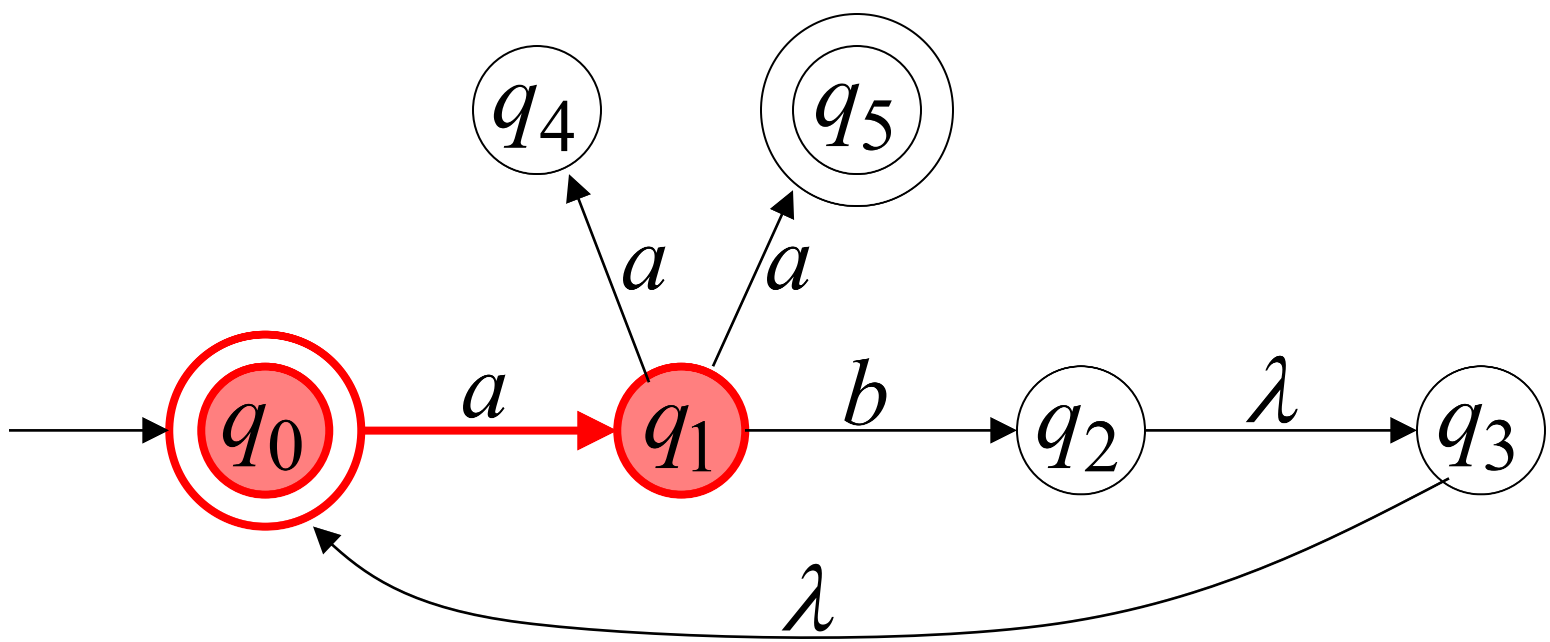
$$\delta(q_2, 1) = \emptyset$$



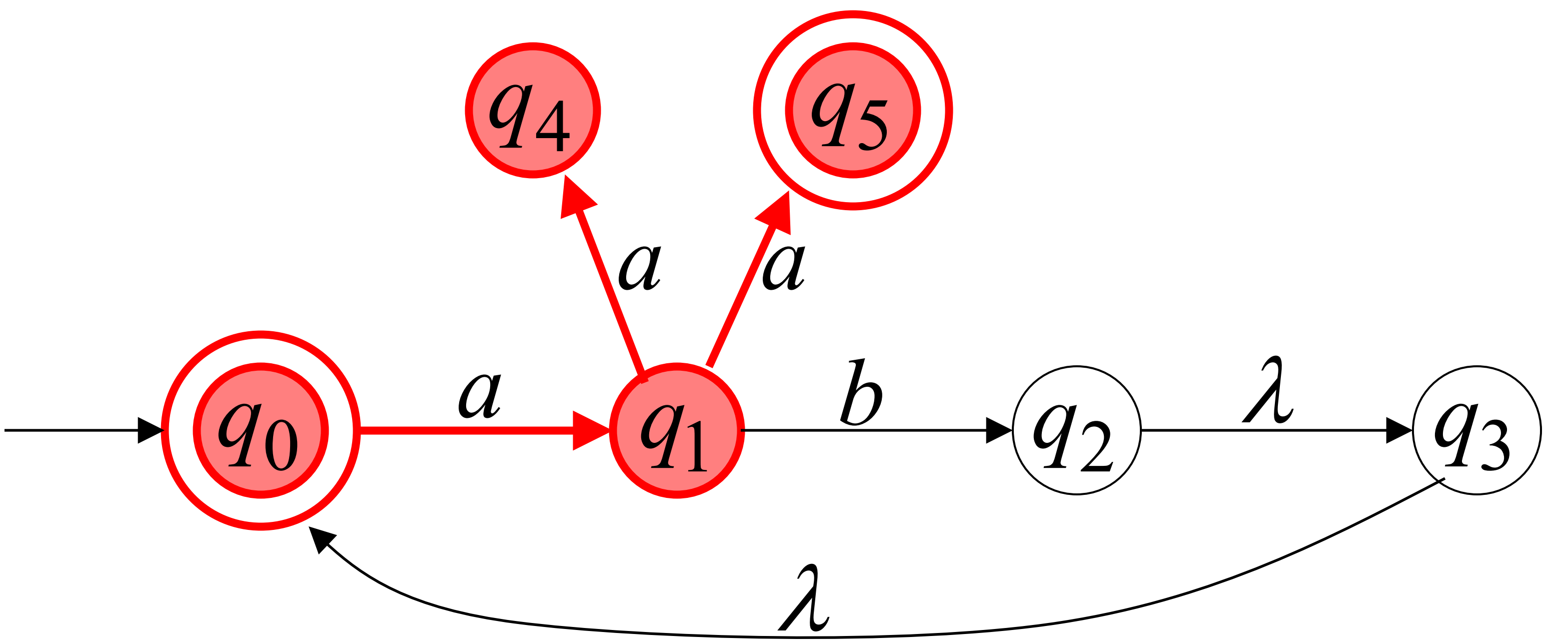
Funzione di transizione estesa δ^*

La stessa cosa δ ma applicata a stringhe

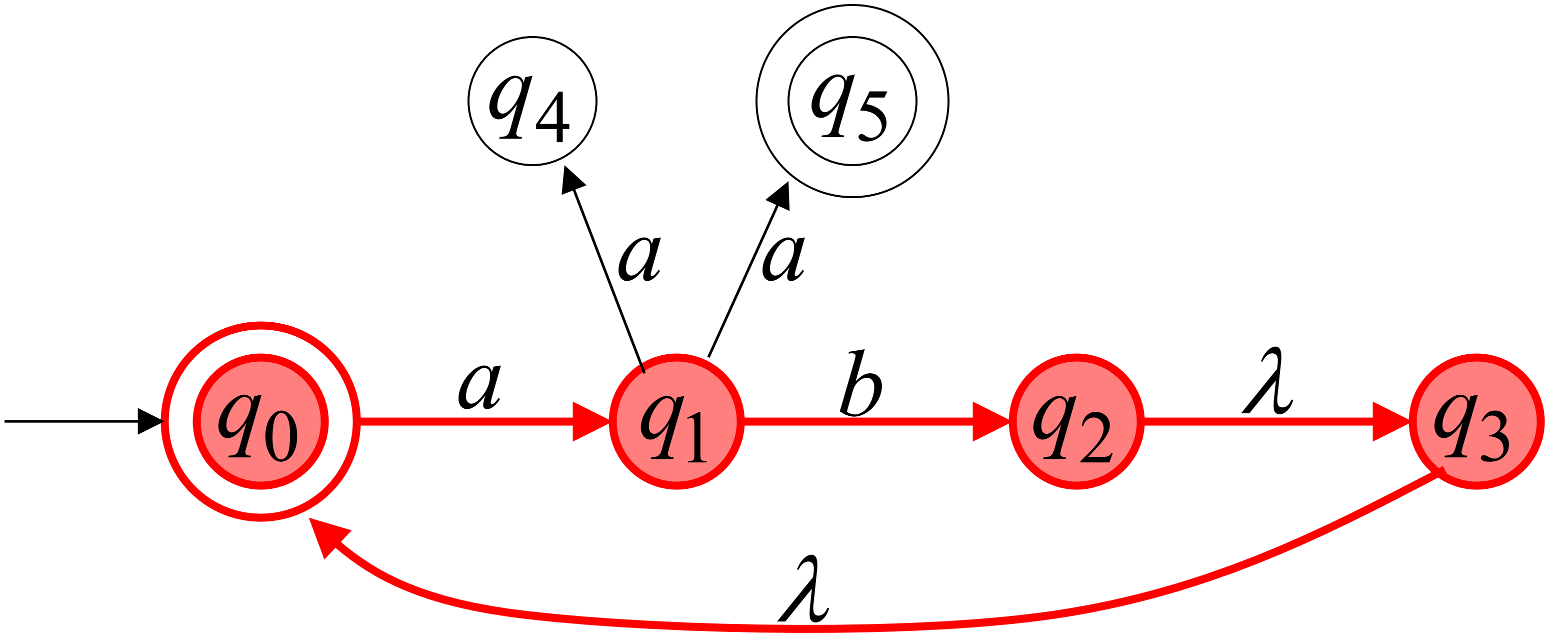
$$\delta^*(q_0, a) = \{q_1\}$$



$$\delta^*(q_0, aa) = \{q_4, q_5\}$$

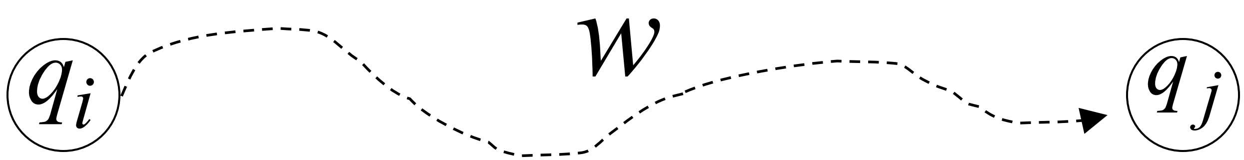


$$\delta^*(q_0, ab) = \{q_2, q_3, q_0\}$$

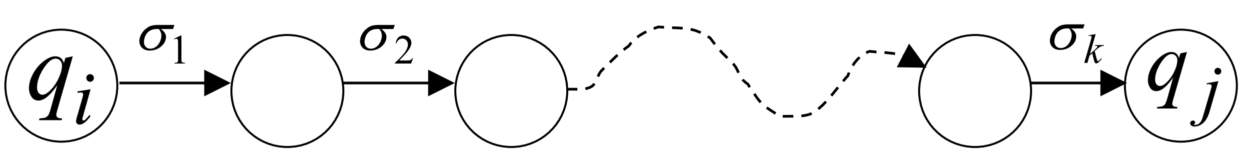


In generale

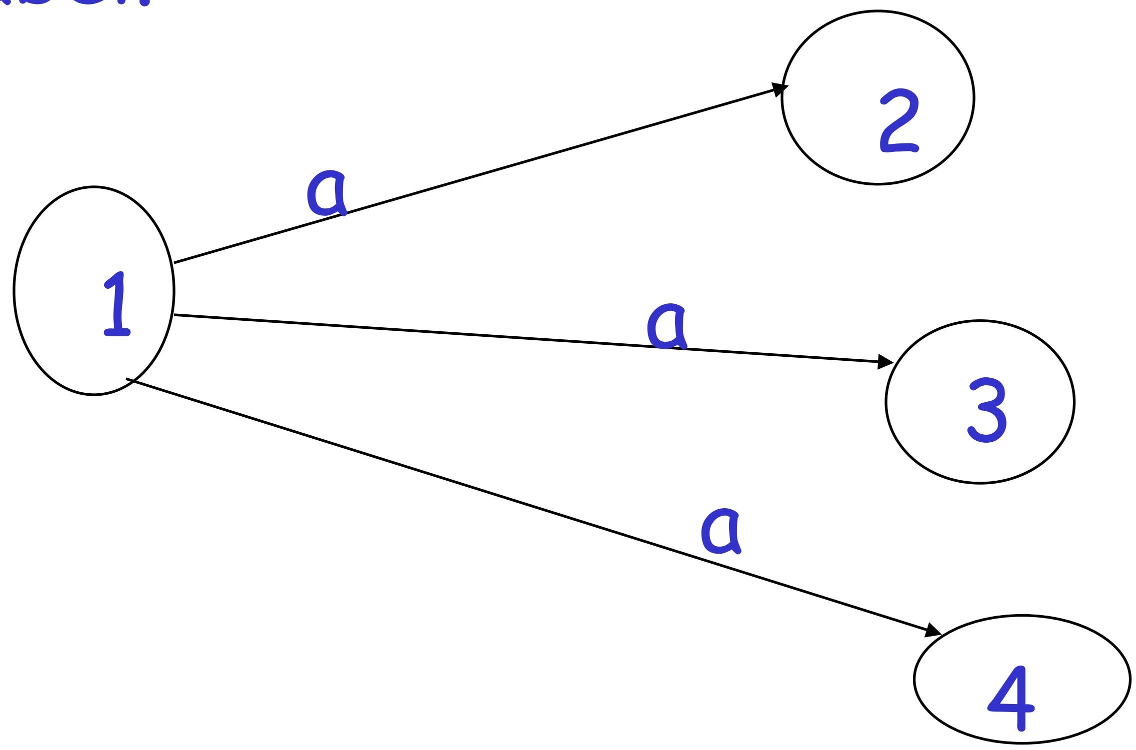
$q_j \in \delta^*(q_i, w)$: vi è un cammino da q_i a q_j con label w



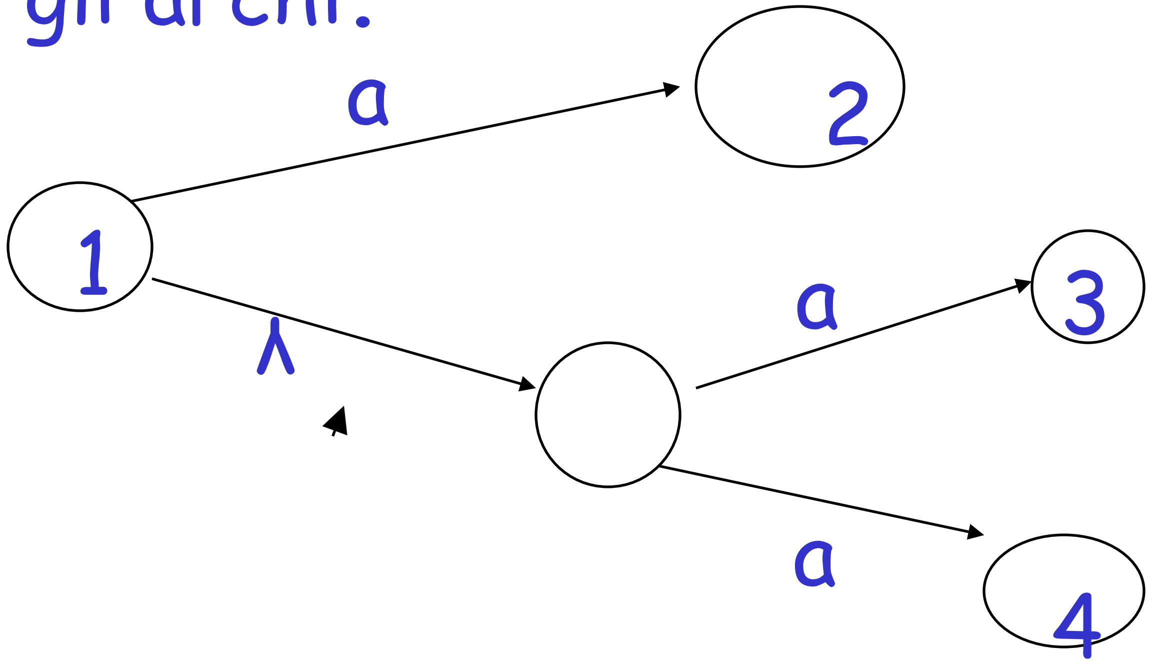
$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



Grado di non determinismo di un nodo per ogni nodo il numero di archi con la stessa label.



Grado di non determinismo di un automa, il grado massimo di non determinismo di tutti gli archi.



The Language of an NFA M

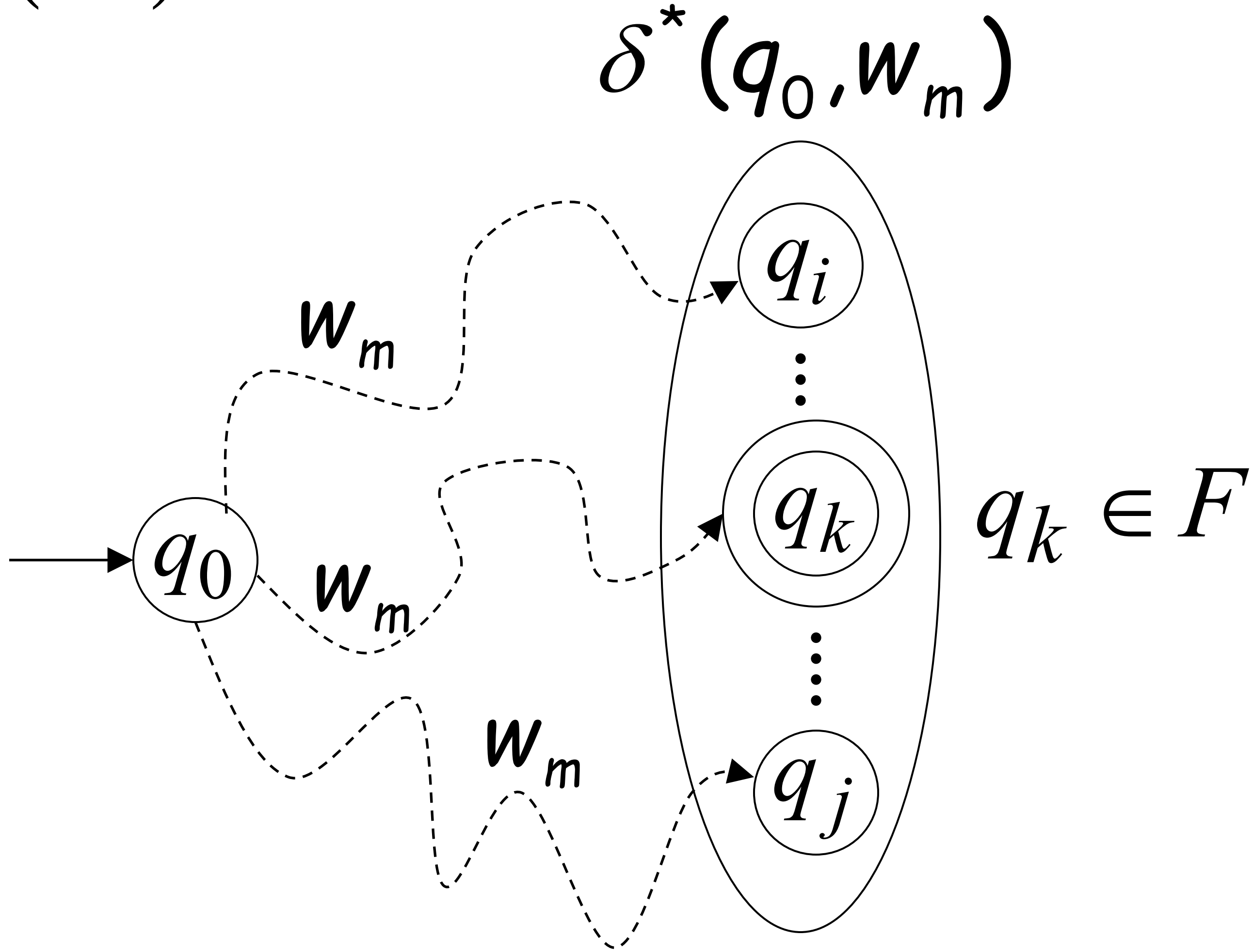
Il linguaggio accettato da M è:

$$L(M) = \{w_1, w_2, \ldots w_n\}$$

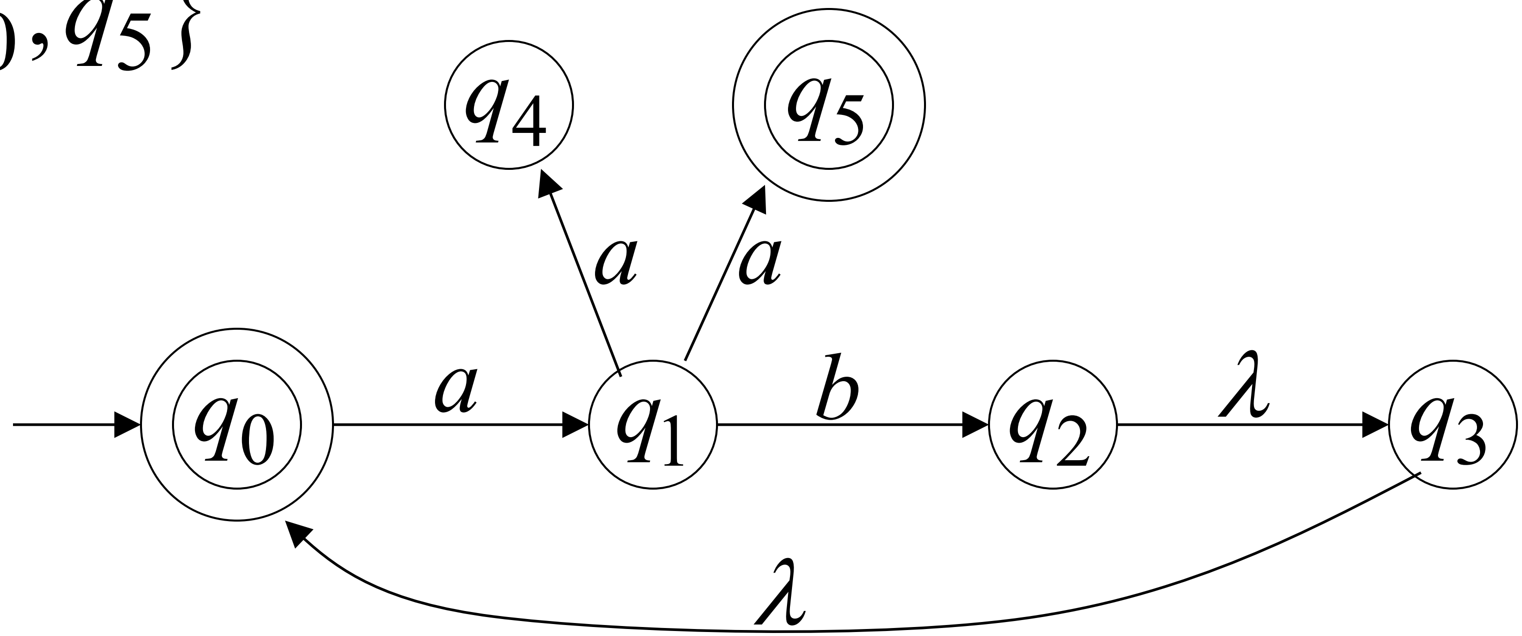
dove $\delta^*(q_0, w_m) = \{q_i, \ldots, q_k, \ldots, q_j\}$

E vi è un $q_k \in F$ (stato finale)

$$w_m \in L(M)$$



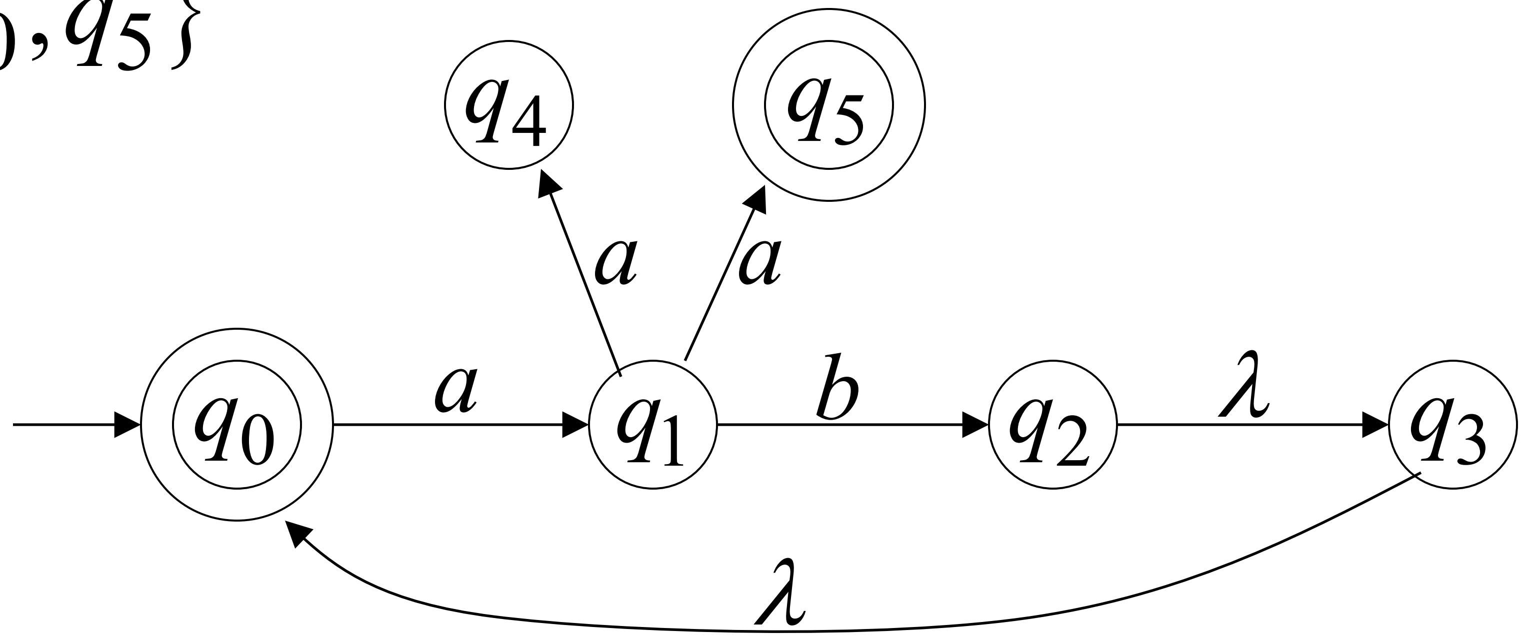
$$F = \{q_0, q_5\}$$



$$\delta^*(q_0, aa) = \{q_4, \underline{q_5}\} \quad \Longrightarrow \quad aa \in L(M)$$

$$q_5 \in F$$

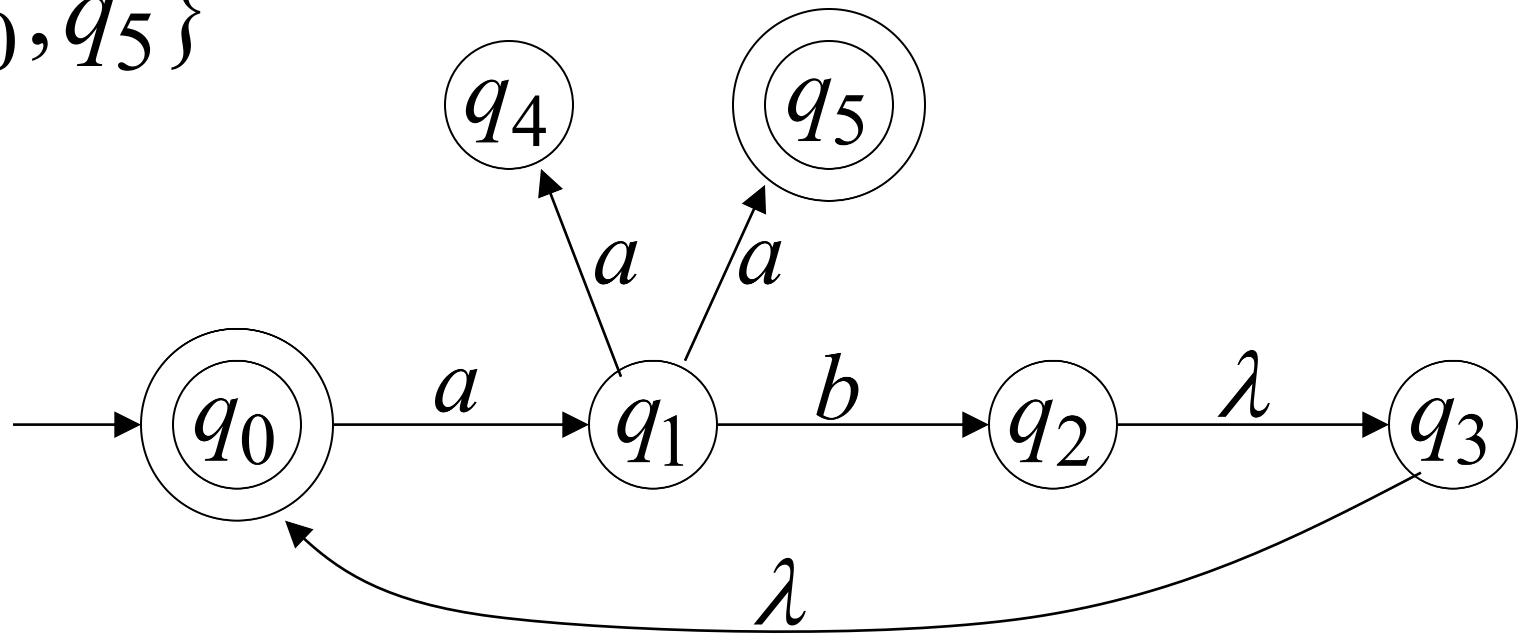
$$F = \{q_0, q_5\}$$



$$\delta^*(q_0, ab) = \{q_2, q_3, \underline{q_0}\} \implies ab \in L(M)$$

$$\searrow \in F$$

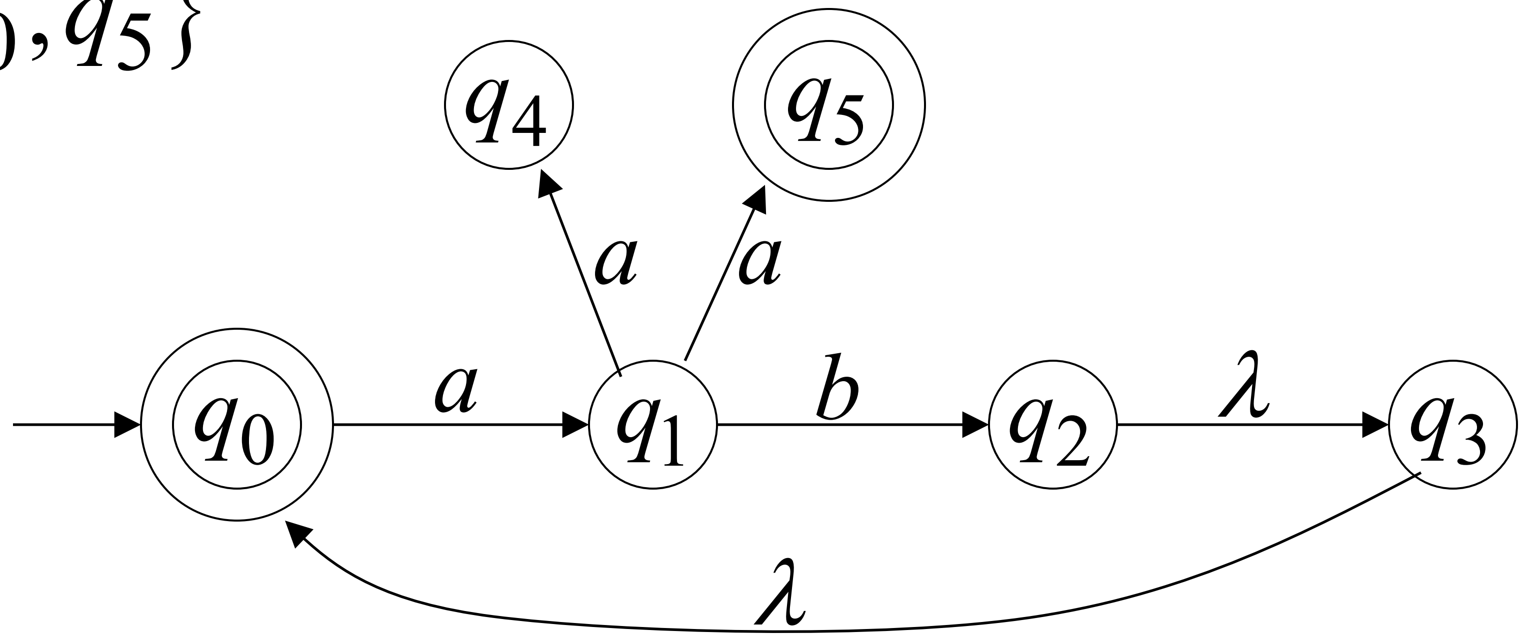
$F = \{q_0, q_5\}$



$\delta^*(q_0, abaa) = \{q_4, \underline{q_5}\}$ ⇨ $aaba \in L(M)$

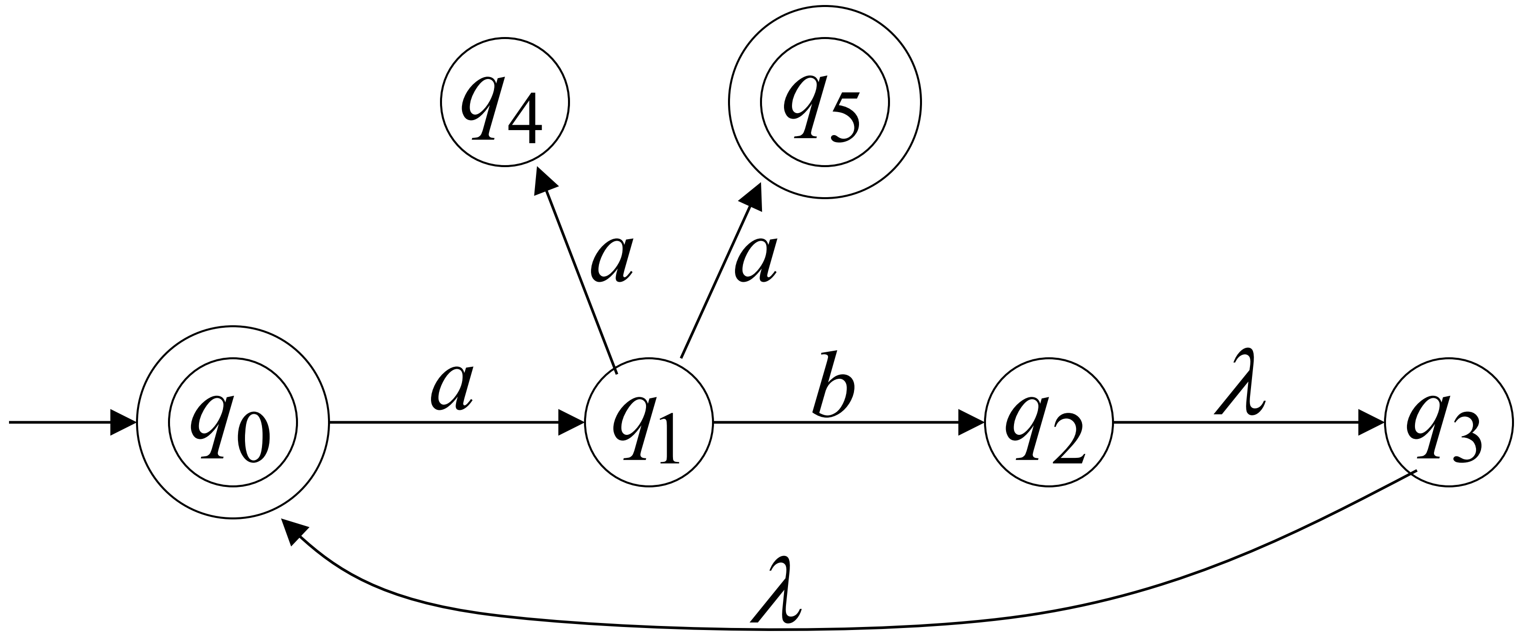
$\in F$

$F = \{q_0, q_5\}$



$\delta^*(q_0, aba) = \{q_1\}$ $\Longrightarrow$ $aba \notin L(M)$

$\notin F$



$$L(M) = \{ab\}^* \cup \{ab\}^* \{aa\}$$

δ*(stato, cW)=

{

delta_*(q,W)

con q elemento dell'insieme {delta(stato, c)}

}

$$q \in \delta^*(q, \lambda)$$ Per ogni stato

1 δ *(stato, cW)=

{

δ*(q,W)

con q $\in$ {δ (stato, c)}

}

2 $$q \in \delta^{*}(q, \lambda)$$ Per ogni stato

NFA accettano i linguaggi regolari

Equivalenza tra macchine

Definizione:

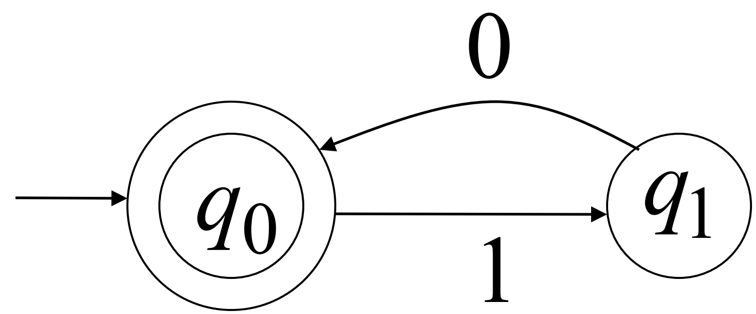
macchina M_1 è equivalente alla macchina M_2

se $L(M_1) = L(M_2)$

Esempio di macchine equivalenti

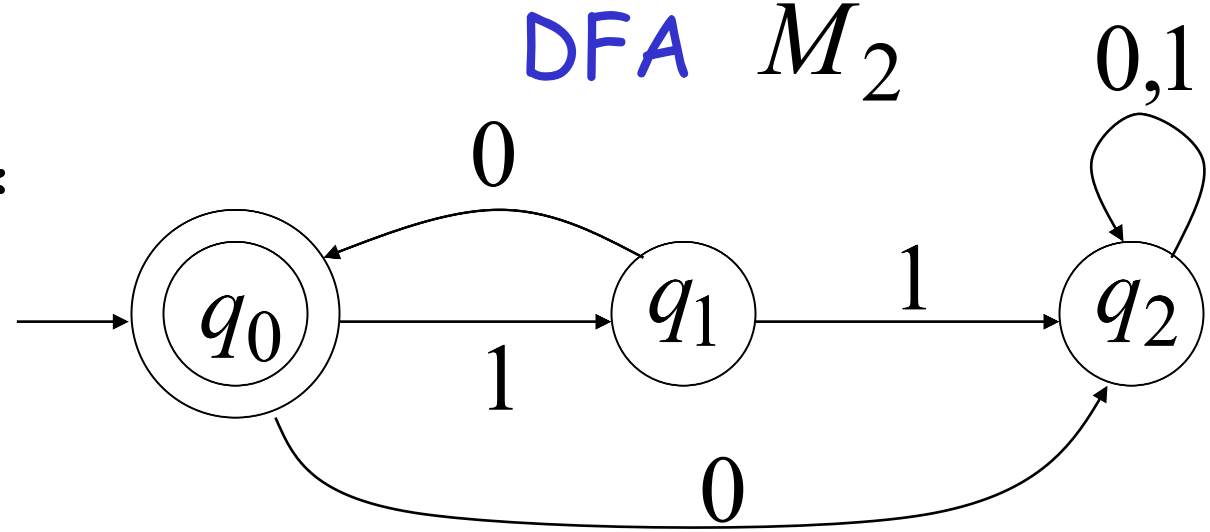
NFA M_1

$$L(M_1) = \{10\}^*$$



DFA M_2

$$L(M_2) = \{10\}^*$$



Teorema:

{ Linguaggi Accettati da NFA } = { Linguaggi regolari }

Linguaggi Accettati da un DFA

NFA e DFA hanno lo stesso potere di computazione, Accettano gli stessi inguaggi.

dimostrazione: mostreremo

$$\left\{\begin{array}{l}\text{Linguaggi a}\\ \text{Accettati}\\ \text{da NFA}\end{array}\right\} \supseteq \left\{\begin{array}{l}\text{Linguaggi}\\ \text{regolari}\end{array}\right\}$$

AND

$$\left\{\begin{array}{l}\text{Linguaggi a}\\ \text{Accettati}\\ \text{da NFA}\end{array}\right\} \subseteq \left\{\begin{array}{l}\text{Linguaggi}\\ \text{regolari}\end{array}\right\}$$

Parte prima

$$\left\{\begin{array}{l}\text{Linguaggi a} \\ \text{Accettati} \\ \text{da NFA}\end{array}\right\} \supseteq \left\{\begin{array}{l}\text{Linguaggi} \\ \text{regolari}\end{array}\right\}$$

ogni DFA è banalmente un NFA

$\Downarrow$

Ogni linguaggio L accettato da un DFA
È anche accettato da un NFA

Parte seconda

$$\left\{\begin{array}{l}\text{Linguaggi a}\\ \text{Accettati}\\ \text{da NFA}\end{array}\right\} \subseteq \left\{\begin{array}{l}\text{Linguaggi}\\ \text{regolari}\end{array}\right\}$$

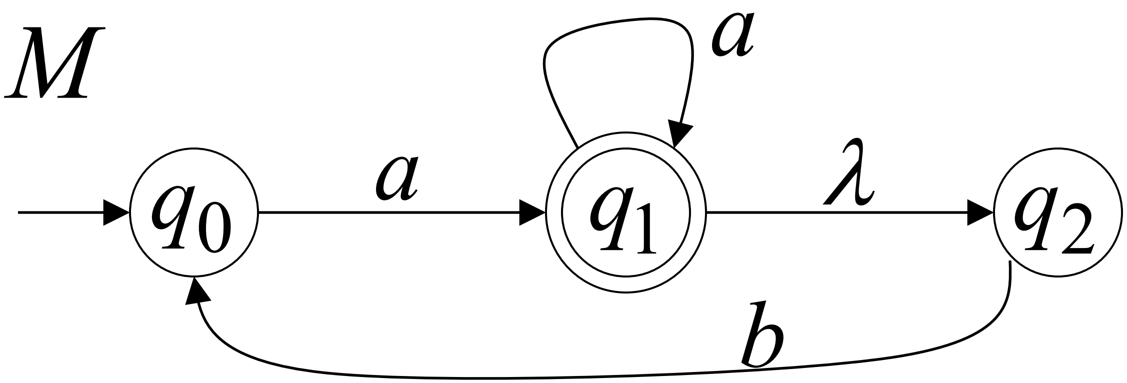
Ogni nfa può essere tradotto in un nfa

$\Downarrow$

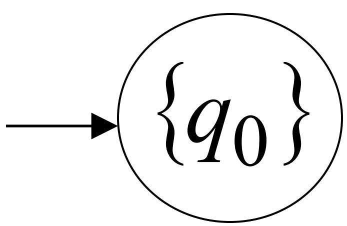
Ogni linguaggio L accettato da un NFA
È anche accettato da un DFA

Conversione da NFA a DFA

NFA M

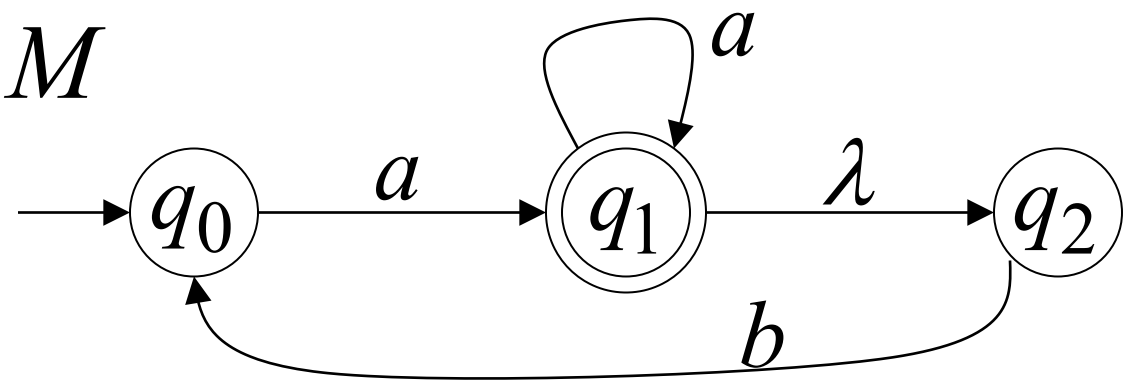


DFA M'

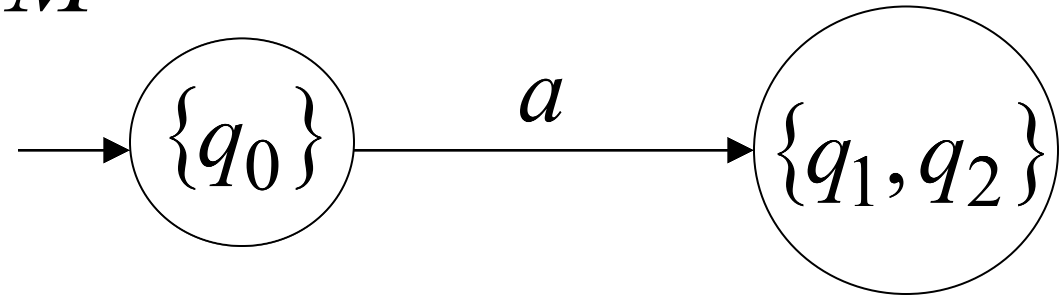


$$\delta^*(q_0, a) = \{q_1, q_2\}$$

NFA M

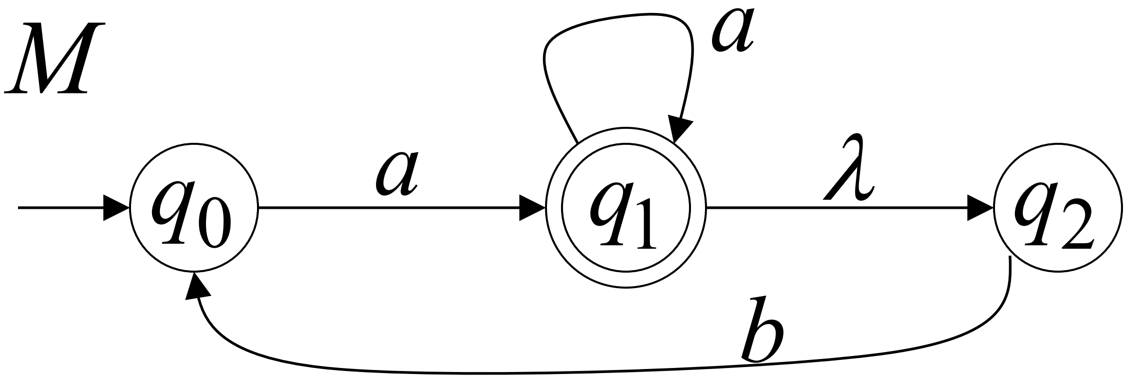


DFA M'

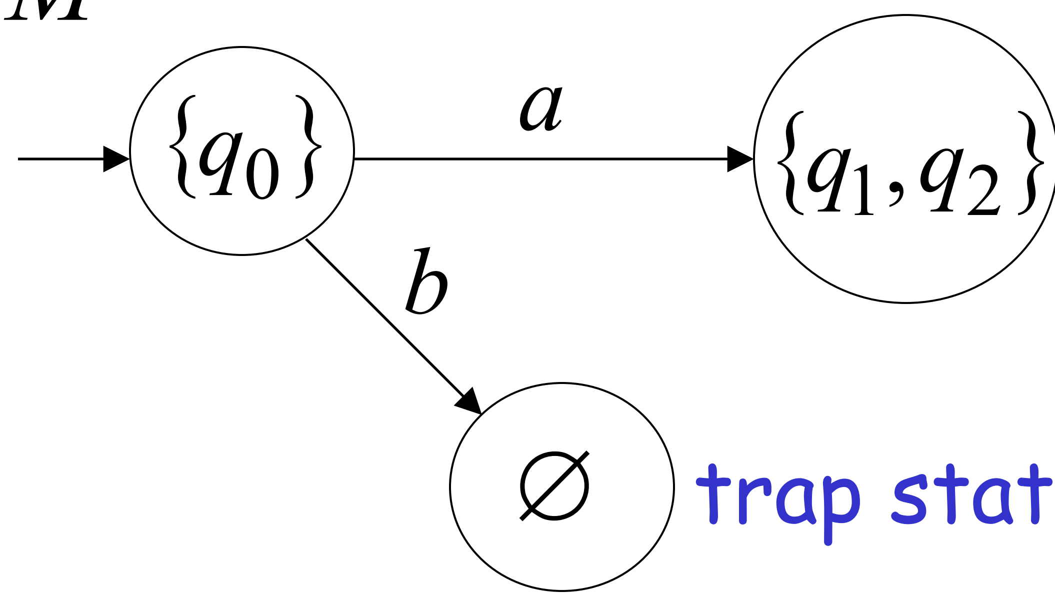


$\delta^*(q_0, b) = \varnothing$ Insieme vuoto

NFA M



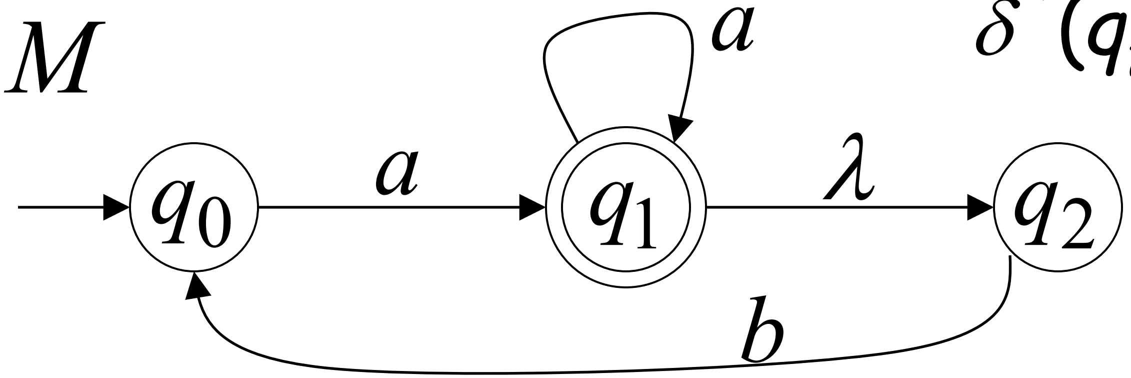
DFA M'



trap state

$$\delta^*(q_1, a) = \{q_1, q_2\}$$
$$\delta^*(q_2, a) = \varnothing$$

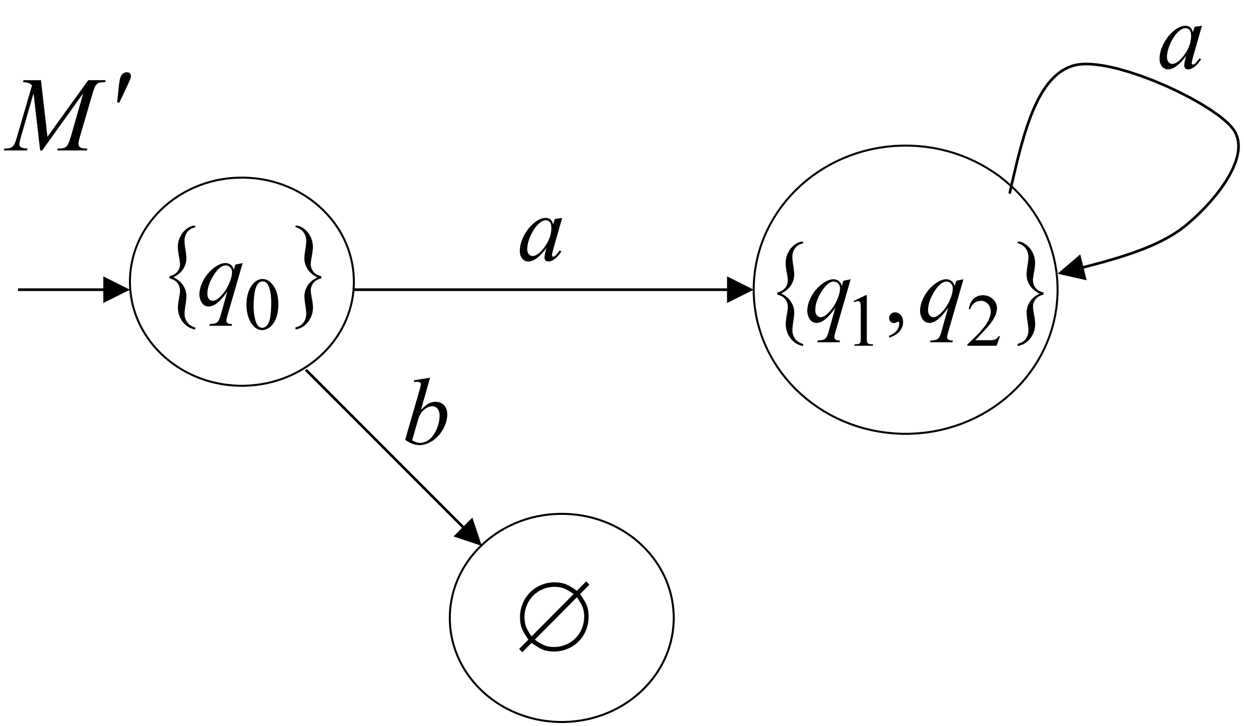
NFA M



unione

$\{q_1, q_2\}$

DFA M'

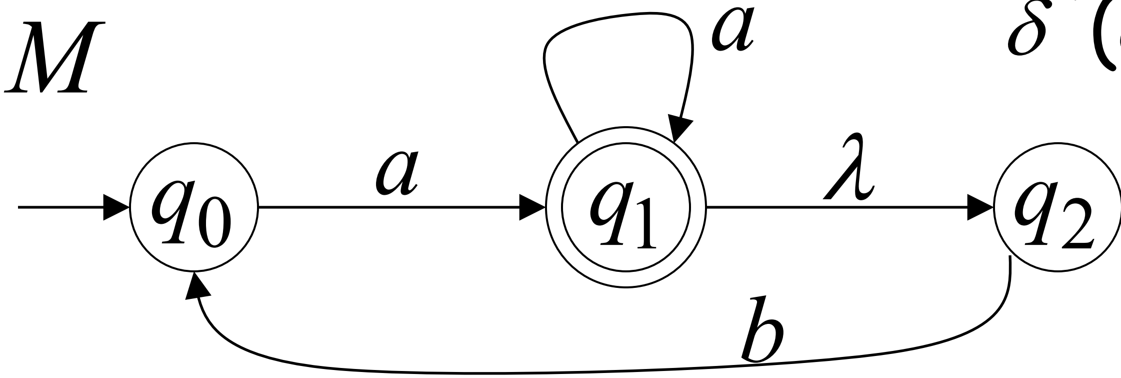


$$\delta^*(q_1, b) = \{q_0\}$$
$$\delta^*(q_2, b) = \{q_0\}$$

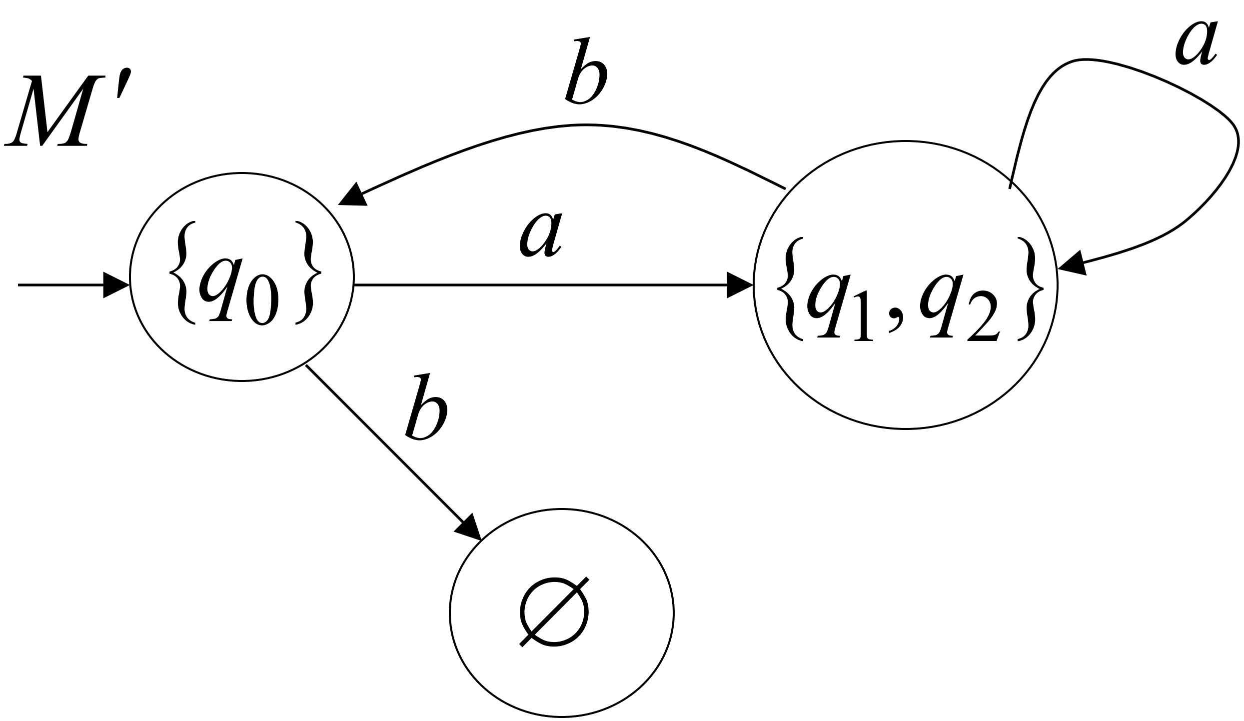
unione

$\{q_0\}$

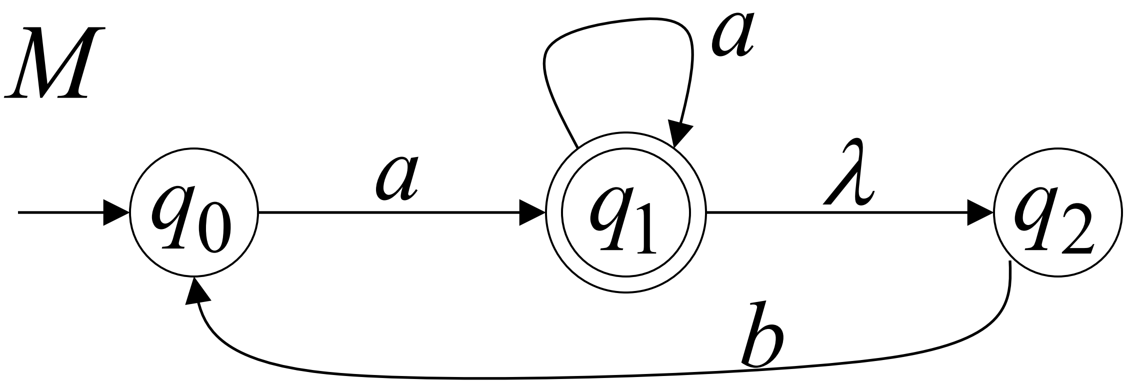
NFA M



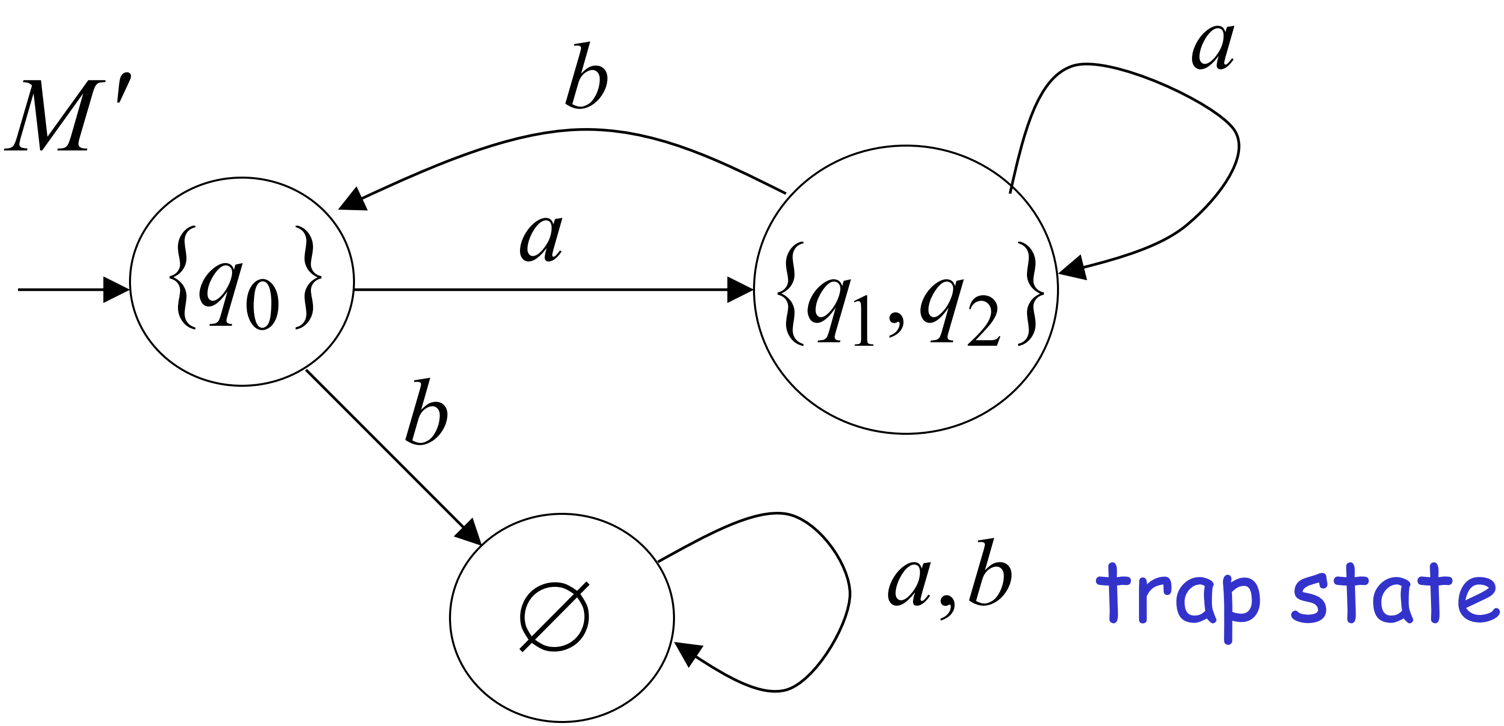
DFA M'



NFA M

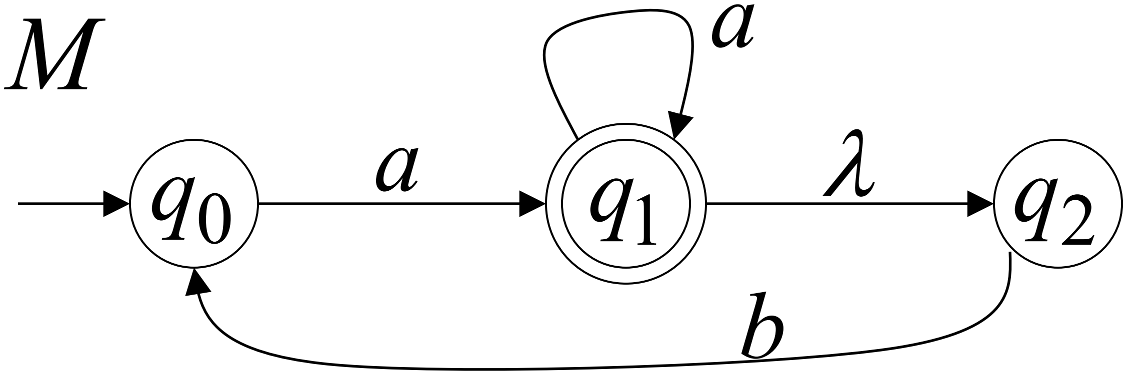


DFA M'



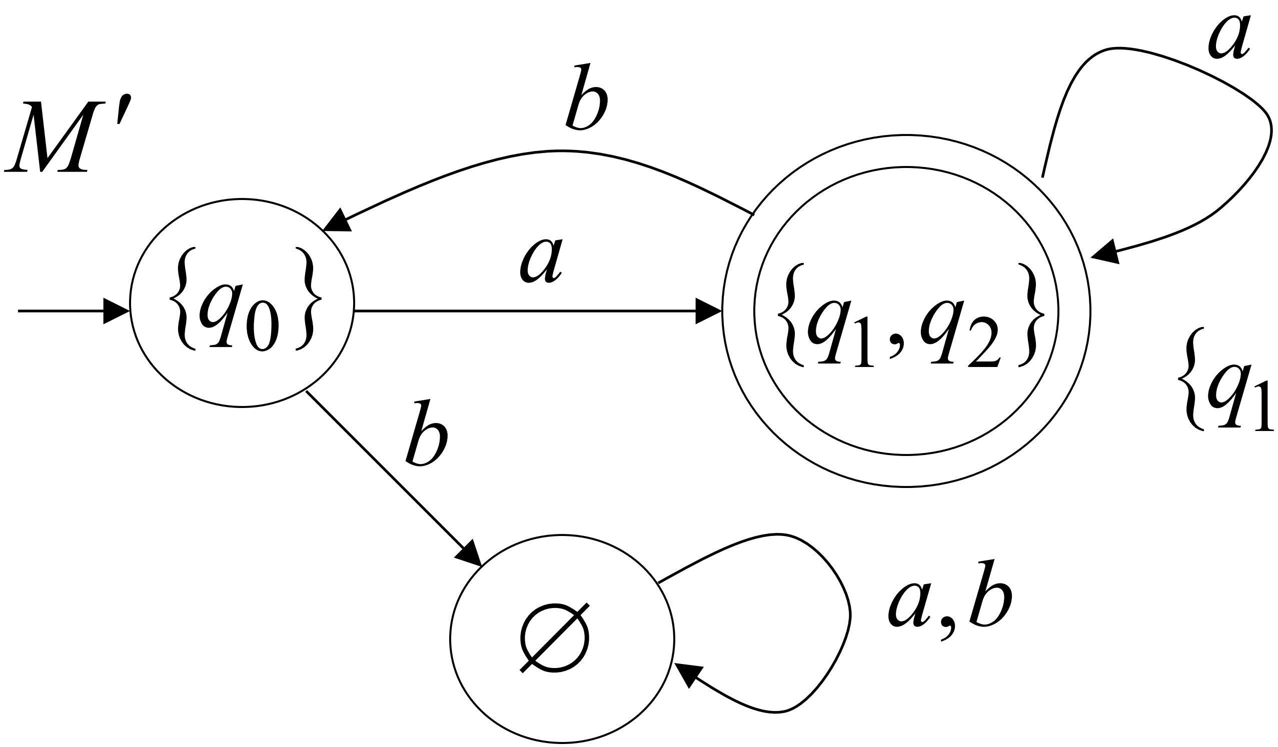
Fine della costruzione

NFA M



$q_1 \in F$

DFA M'



$\{q_1, q_2\} \in F'$

Procedura generale

Input: NFA M

Output: un equivalente DFA M'

con $L(M) = L(M')$

NFA ha gli stati $q_0, q_1, q_2, \ldots$

DFA ha gli stati definiti dall'insieme delle parti

$$\emptyset, \{q_0\}, \{q_1\}, \{q_0, q_1\}, \{q_1, q_2, q_3\}, \ldots.$$

Step della procedura

step

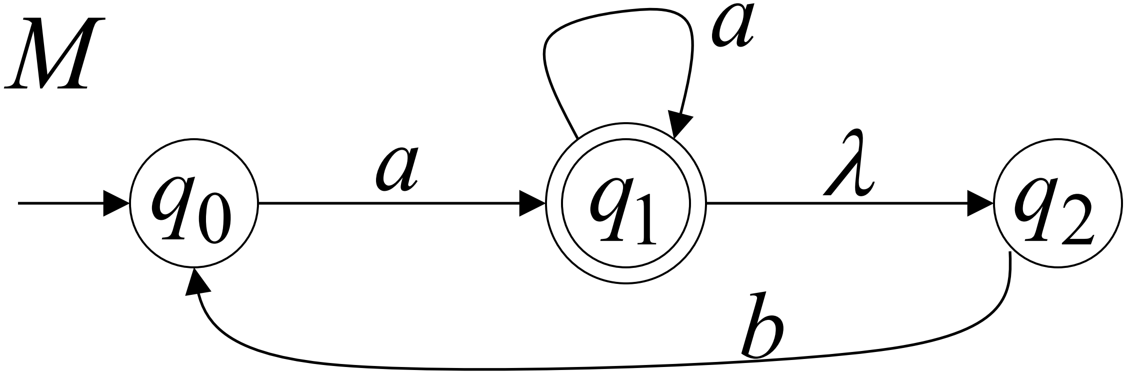
1. Stato iniziale NFA: q_0

$\Downarrow$

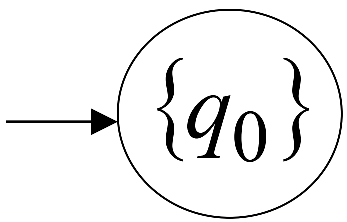
stato iniziale del DFA: $\{q_0\}$

esempio

NFA M



DFA M'



step

2. per ogni stato DFA $\{q_i, q_j, \ldots, q_m\}$

calcolo nel NFA

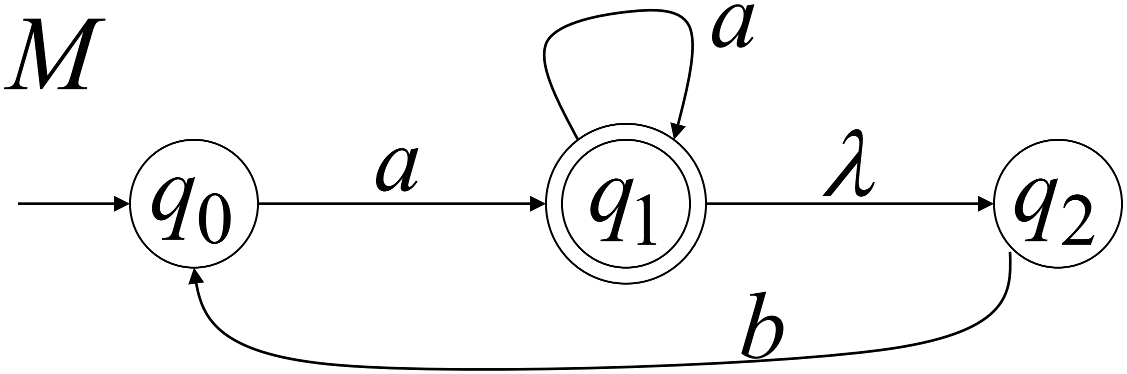
$$\left.\begin{array}{r} \delta^*(q_i, a) \\ \cup\, \delta^*(q_j, a) \\ \ldots \\ \cup\, \delta^*(q_m, a) \end{array}\right\} \overset{\text{unione}}{=} \{q'_k, q'_l, \ldots, q'_n\}$$

addiziona questa nuova transizione al DFA

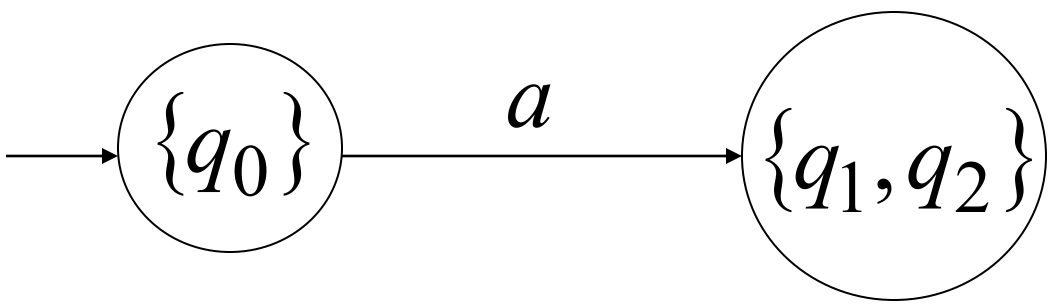
$$\delta(\{q_i, q_j, \ldots, q_m\},\ a) = \{q'_k, q'_l, \ldots, q'_n\}$$

esempio $\delta^*(q_0, a) = \{q_1, q_2\}$

NFA M



DFA M' $\delta(\{q_0\}, a) = \{q_1, q_2\}$

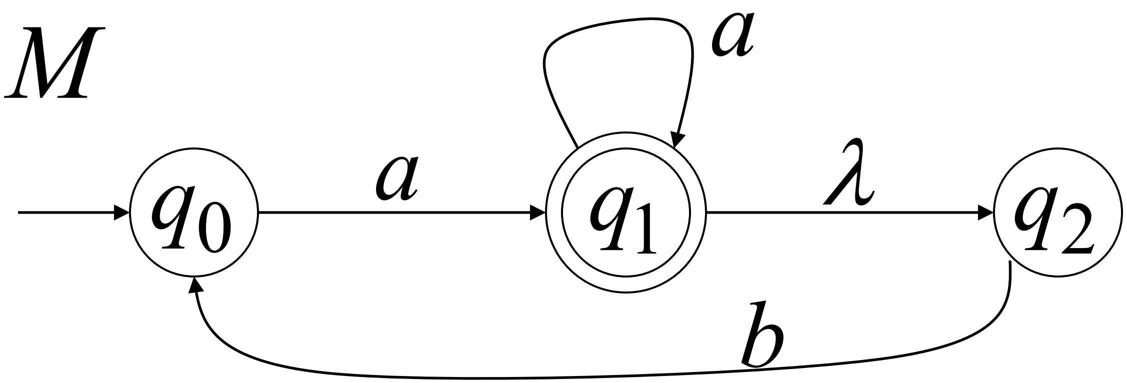


step

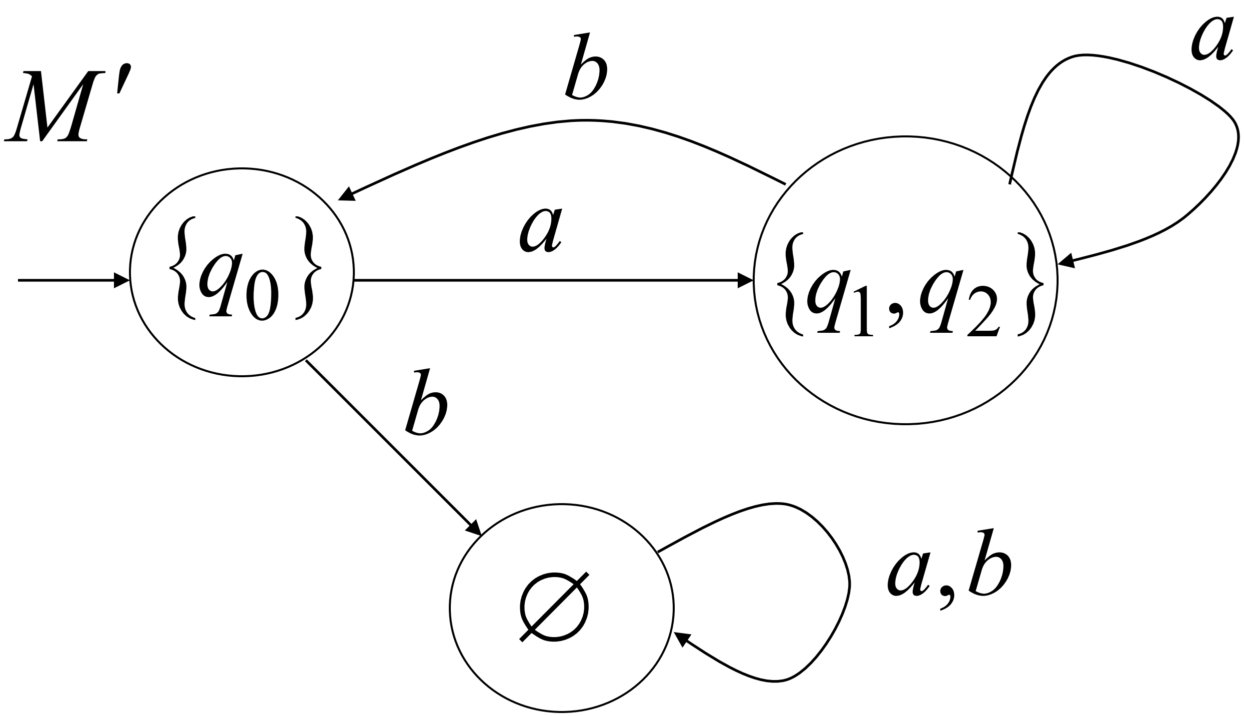
3. Ripeti lo step 2 per ogni stato nel DFA e simboli nell'alfabeto finchè non vi sono più stati che possono essere addizionati al DFA

esempio

NFA M



DFA M'



step

4.

$\{q_i, q_j, \ldots, q_m\}$

Per ogni stato DFA

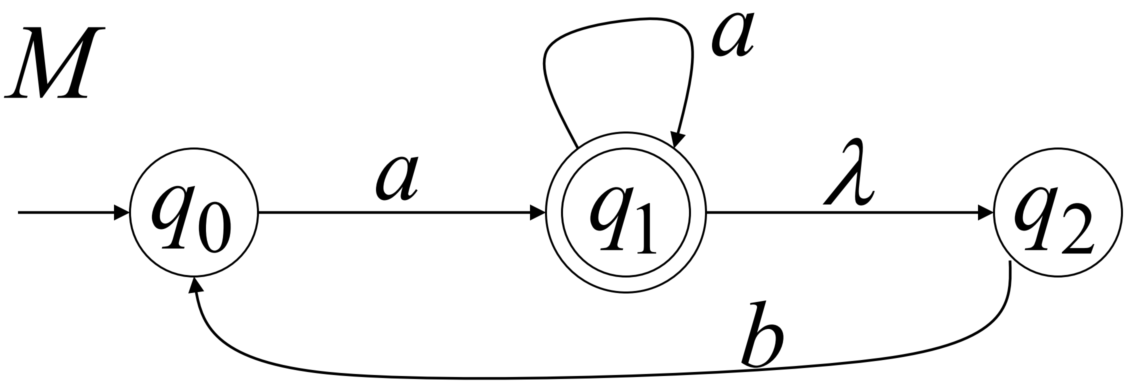
Se qualche q_j è uno stato di accettazione del NFA

Allora $\{q_i, q_j, \ldots, q_m\}$

è uno stato di accettazione del DFA

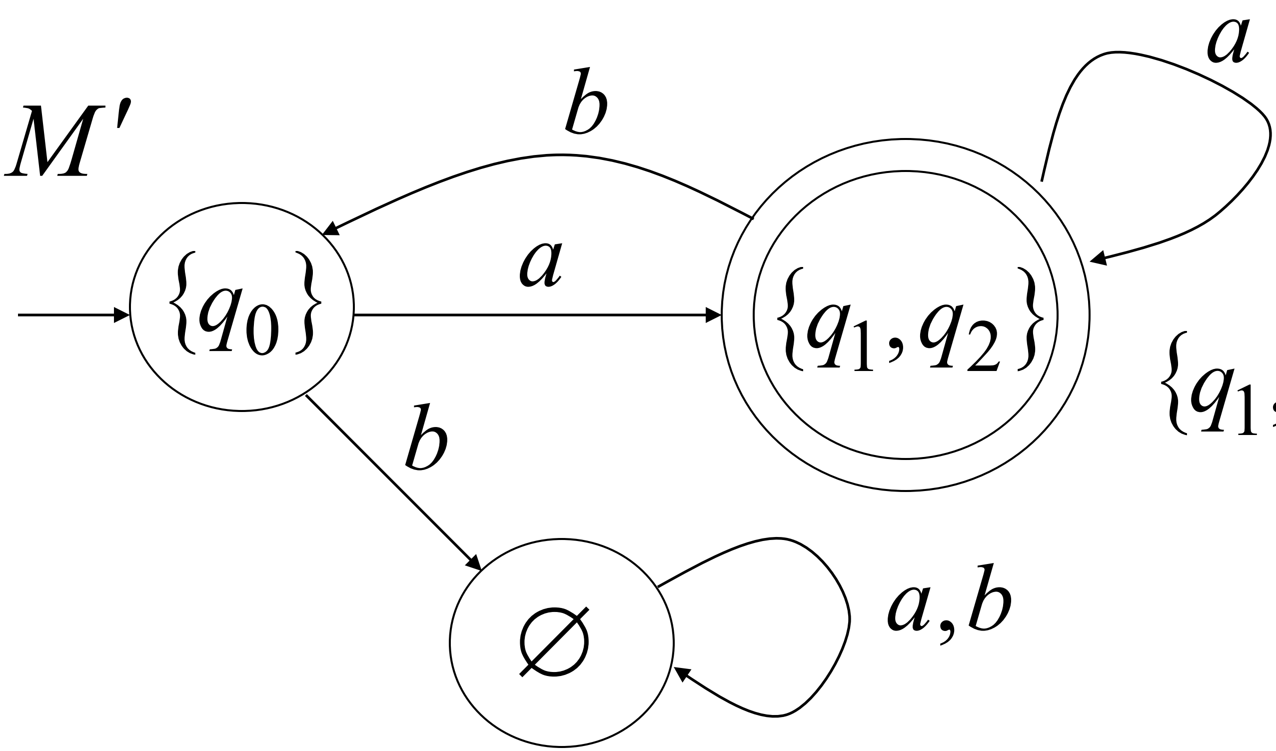
Example

NFA M



$q_1 \in F$

DFA M'



$\{q_1, q_2\} \in F'$

Step della procedura

step

1. Stato iniziale NFA: q_0

stato iniziale del DFA: $\{q_0\}$

step

2. per ogni stato DFA $\{q_i, q_j, \ldots, q_m\}$

calcolo nel NFA

$$\left.\begin{array}{r} \delta^*(q_i, a) \\ \cup\, \delta^*(q_j, a) \\ \ldots \\ \cup\, \delta^*(q_m, a) \end{array}\right\} \overset{\text{unione}}{=} \{q'_k, q'_l, \ldots, q'_n\}$$

addiziona questa nuova transizione al DFA

$$\delta(\{q_i, q_j, \ldots, q_m\},\ a) = \{q'_k, q'_l, \ldots, q'_n\}$$

step

3. Ripeti lo step 2 per ogni stato nel DFA e simboli nell'alfabeto finchè non vi sono più stati che possono essere addizionati al DFA

step

4. Per ogni stato del DFA $\{q_i, q_j, \ldots, q_m\}$

se è presente uno stato q_j finale, accettante, del NFA

allora, $\{q_i, q_j, \ldots, q_m\}$ è uno stato accettante del DFA

Lemma:

Se traduciamo un NFA M in un DFA M'

Allora i due automata sono equivalenti:

$$L(M) = L(M')$$

dimostrazione:

Dobbiamo dimostrare che: $L(M) \subseteq L(M')$

e

$$L(M) \supseteq L(M')$$

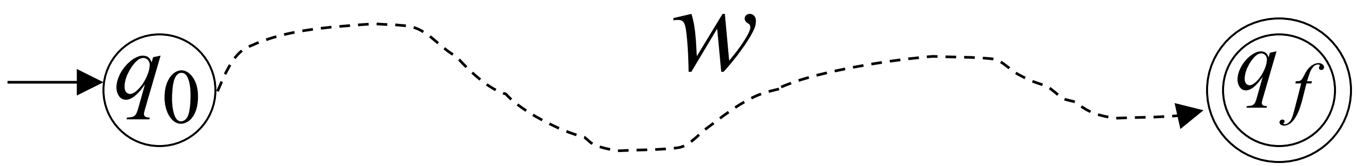
Mostriamo che: $L(M) \subseteq L(M')$

NFA contenuto in DFA

Dobbiamo provare che:

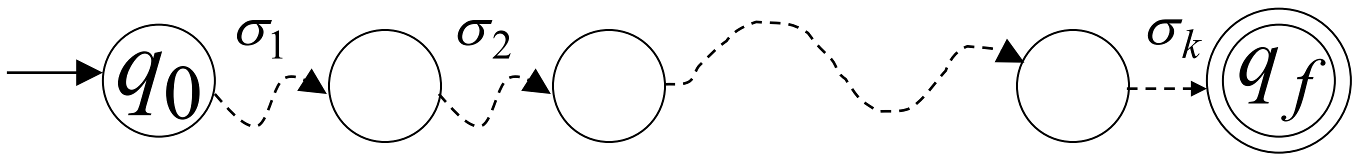
$w \in L(M)$ ⇒ $w \in L(M')$

considera $w \in L(M)$ NFA



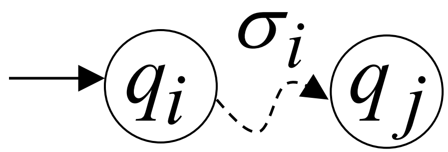
simboli

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



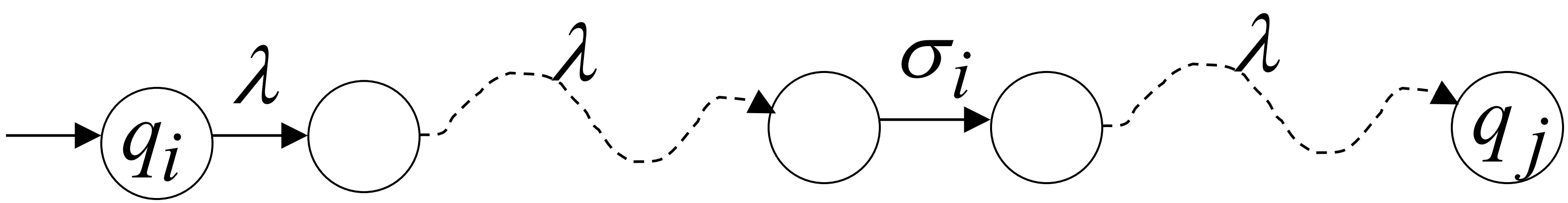
ricordiamo

Simboli, lng 1



Denota un sotto cammino tale che

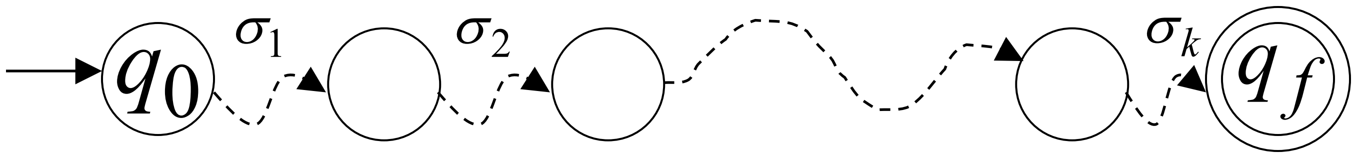
simboli



Mostriamo che se $w \in L(M)$

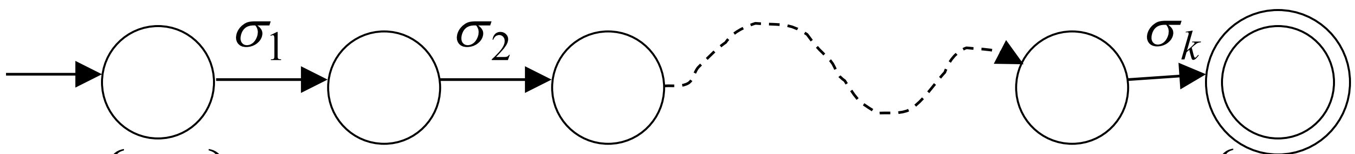
$w = \sigma_1 \sigma_2 \cdots \sigma_k$

NFA $M:$



allora

DFA $M':$



$\{q_0\}$ state label

$\{q_f, \ldots\}$ state label

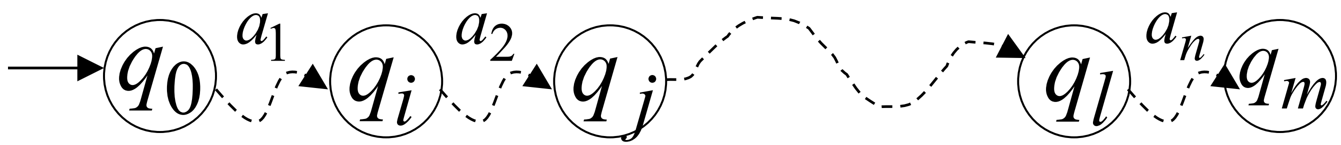
$w \in L(M')$

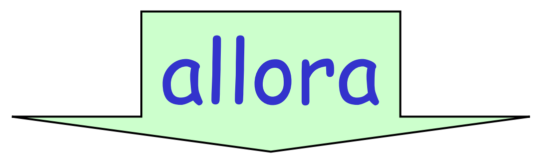
In modo piu generale ,

mostreremo che se in M :

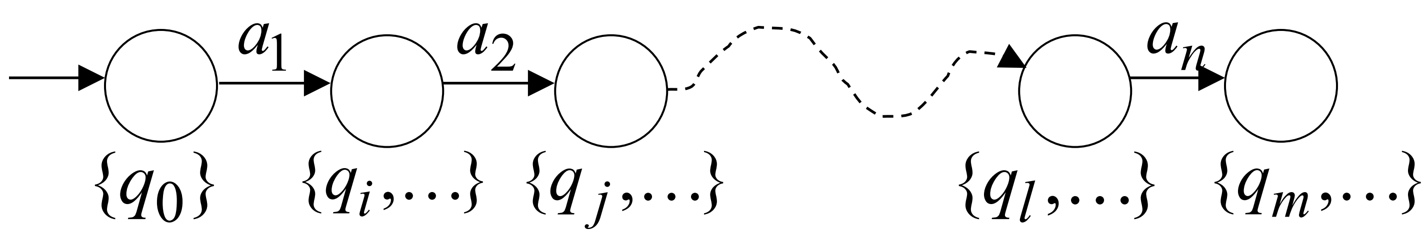
(stringa arbitraria) $v = a_1 a_2 \cdots a_n$

NFA M :



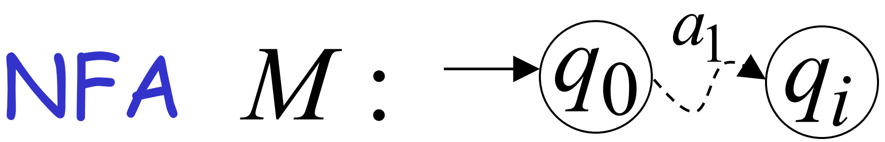


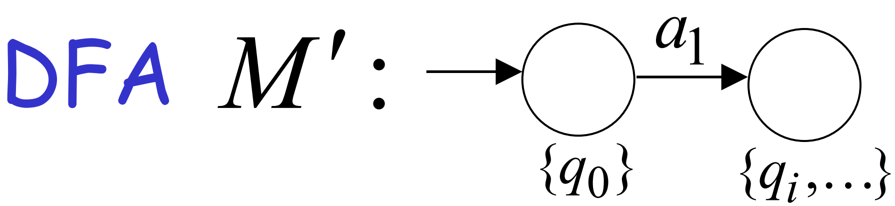
DFA M' :



Dimostrazione per induzione su $|v|$

Base induzione: $|v| = 1$ $\qquad v = a_1$





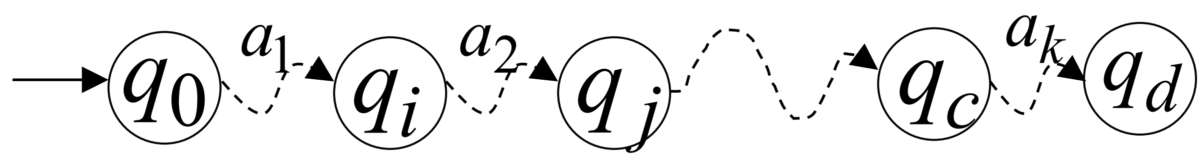
[vero per come costruito M']

Ipotesi induttiva: $1 \le |v| \le k$

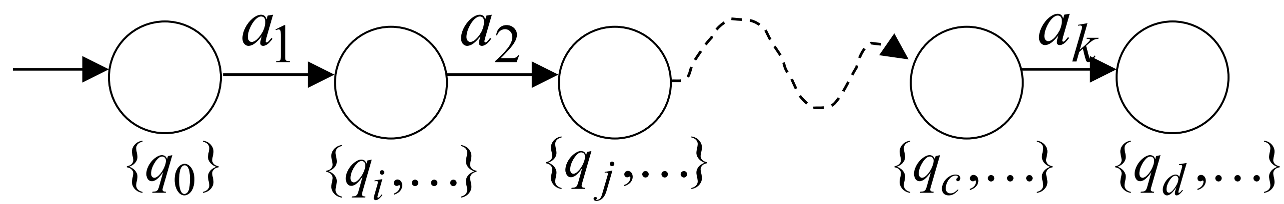
$$v = a_1 a_2 \cdots a_k$$

Supponiamo valga

NFA $M:$



DFA $M':$

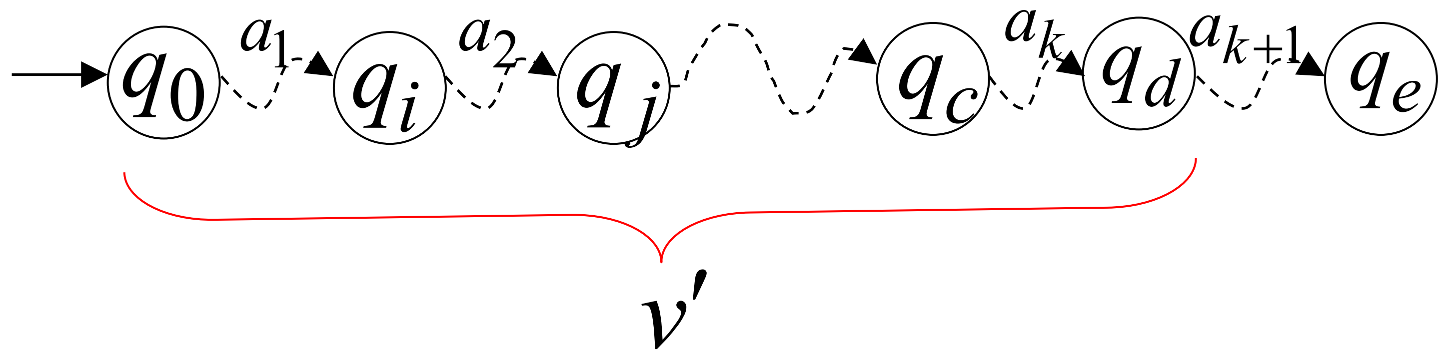


Step induttivo: $|v| = k+1$

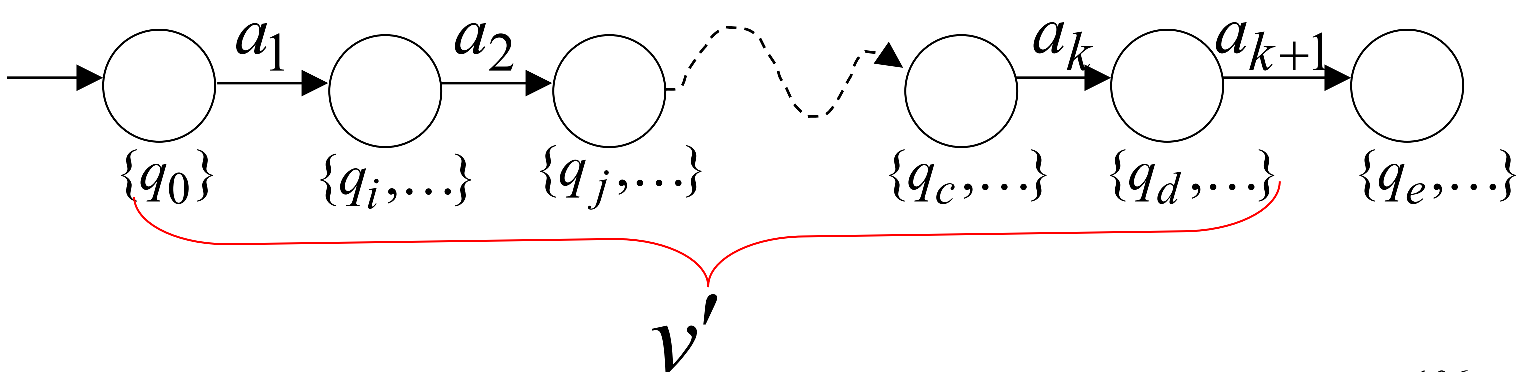
$$v = \underbrace{a_1 a_2 \cdots a_k}_{v'} a_{k+1} = v' a_{k+1}$$

Vero per costruzione di M'

NFA M :



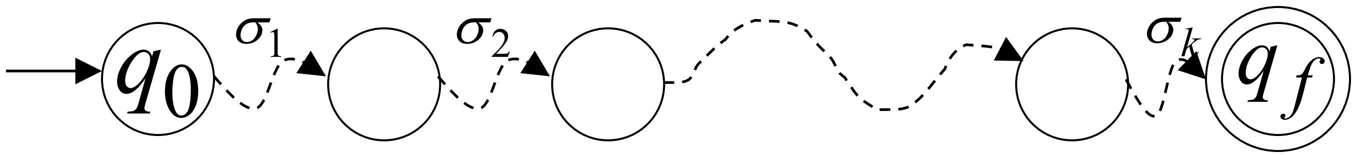
DFA M' :



Quindi se $w \in L(M)$

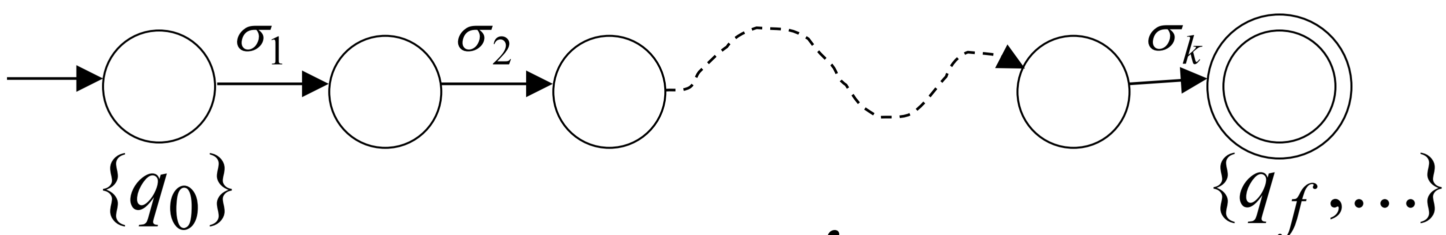
$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$

NFA $M:$



allora

DFA $M':$



$$w \in L(M')$$

allora: $L(M) \subseteq L(M')$ dimostrato

e $L(M) \supseteq L(M')$ banale

quindi: $L(M) = L(M')$

Fine lemma