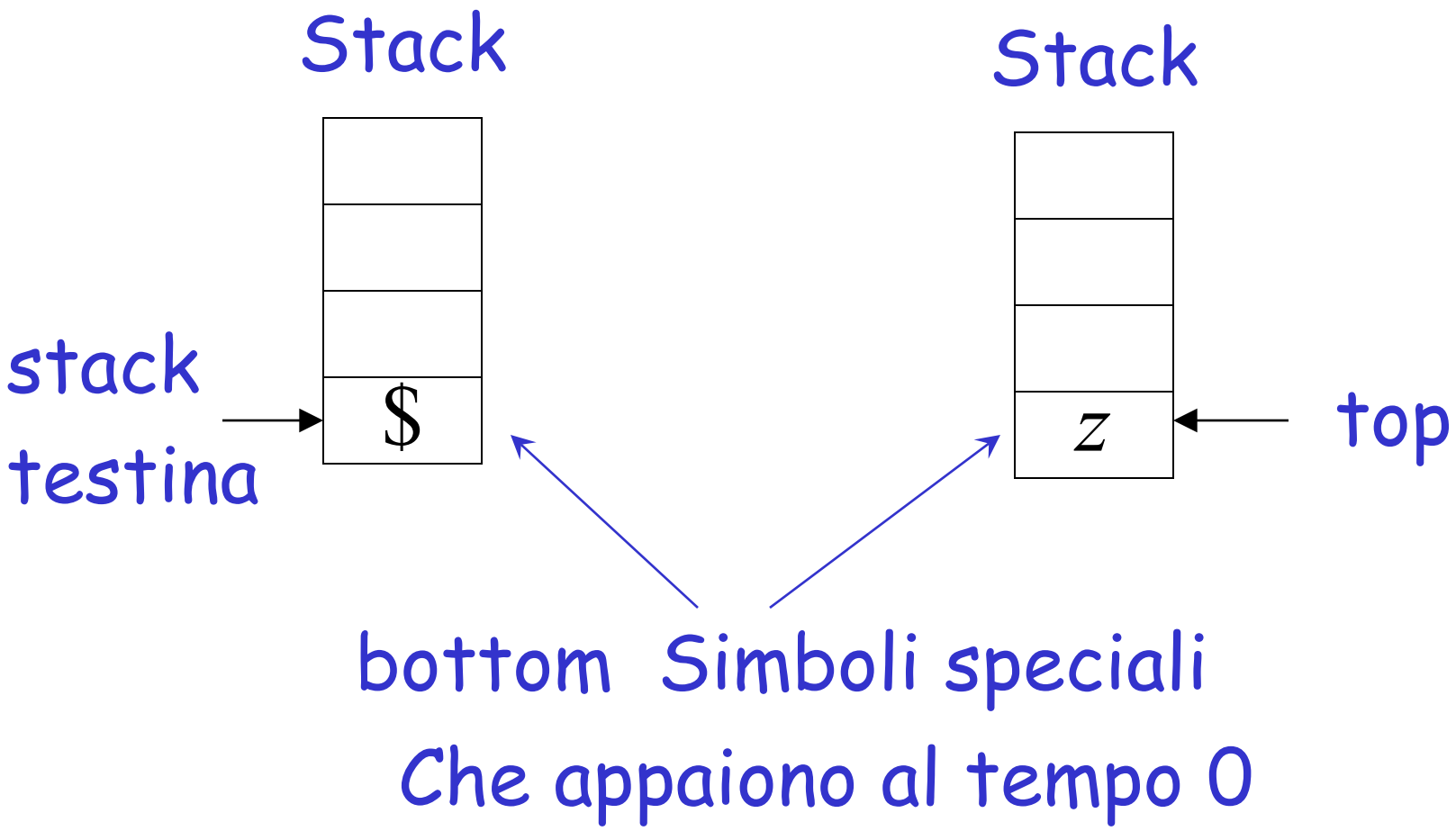
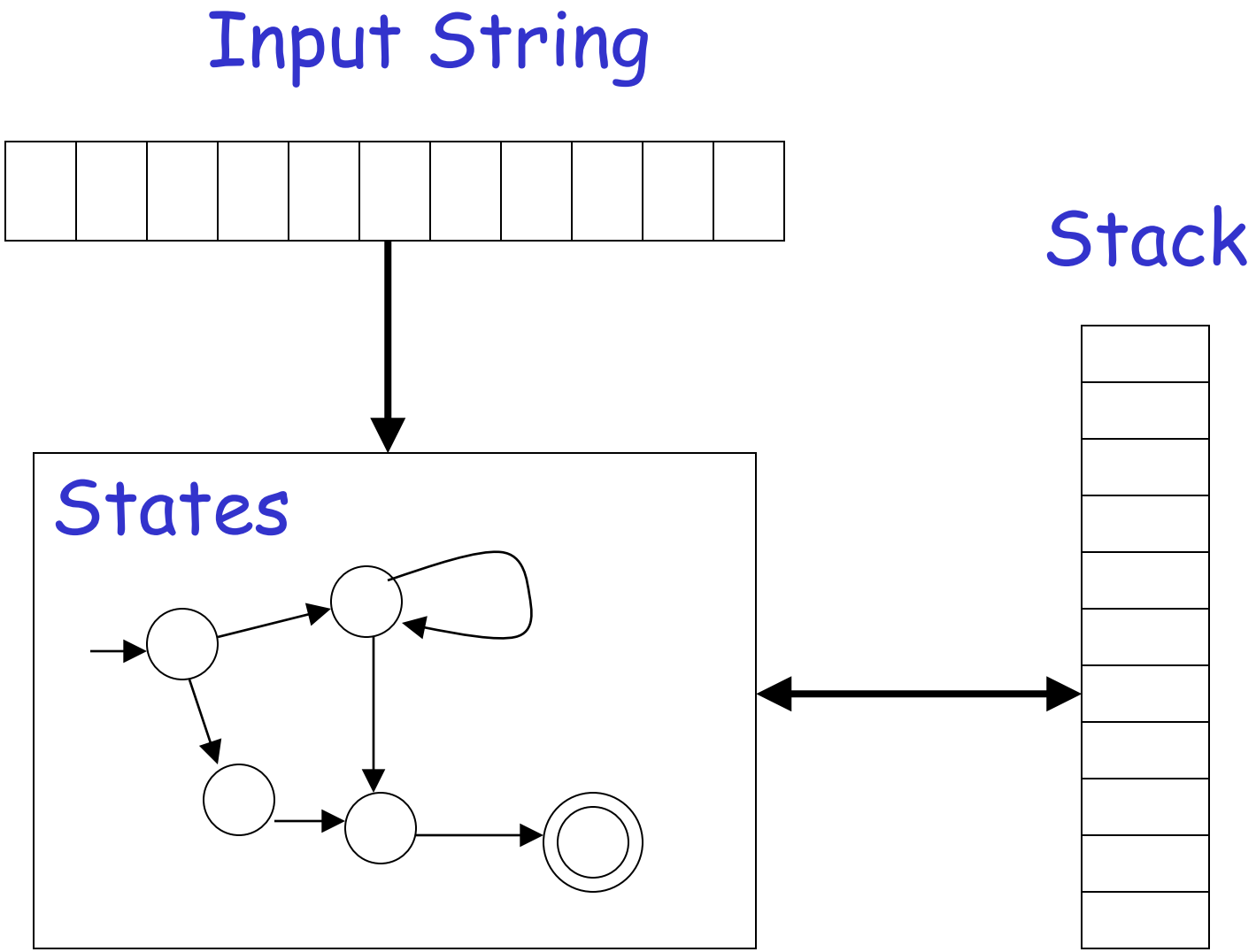
Pushdown Automata
PDA

Initial Stack Symbol



Pushdown Automaton -- PDA

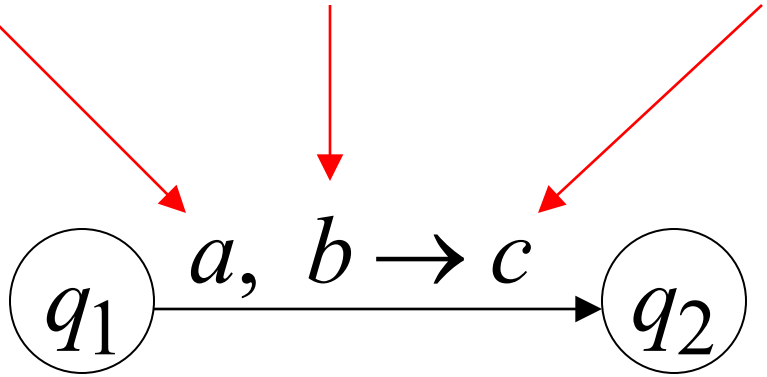


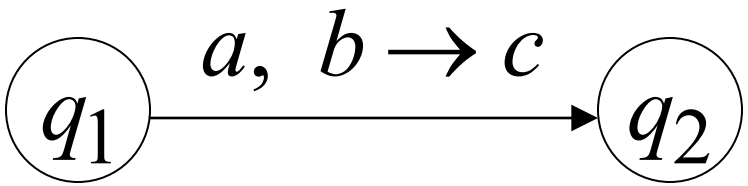
Gli stati

Input simbolo

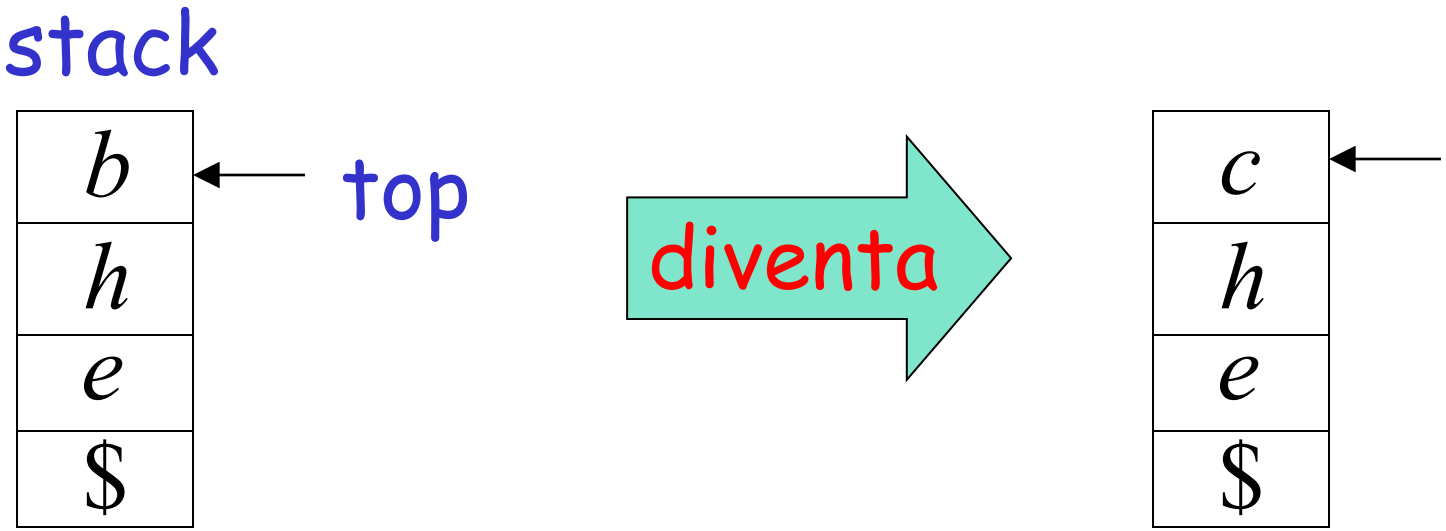
Pop simbolo

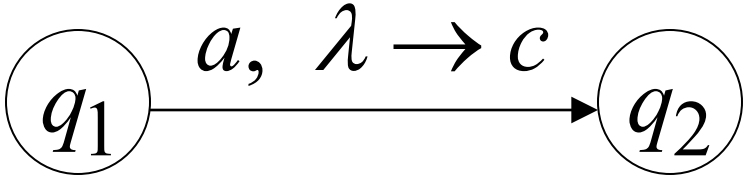
Push simbolo



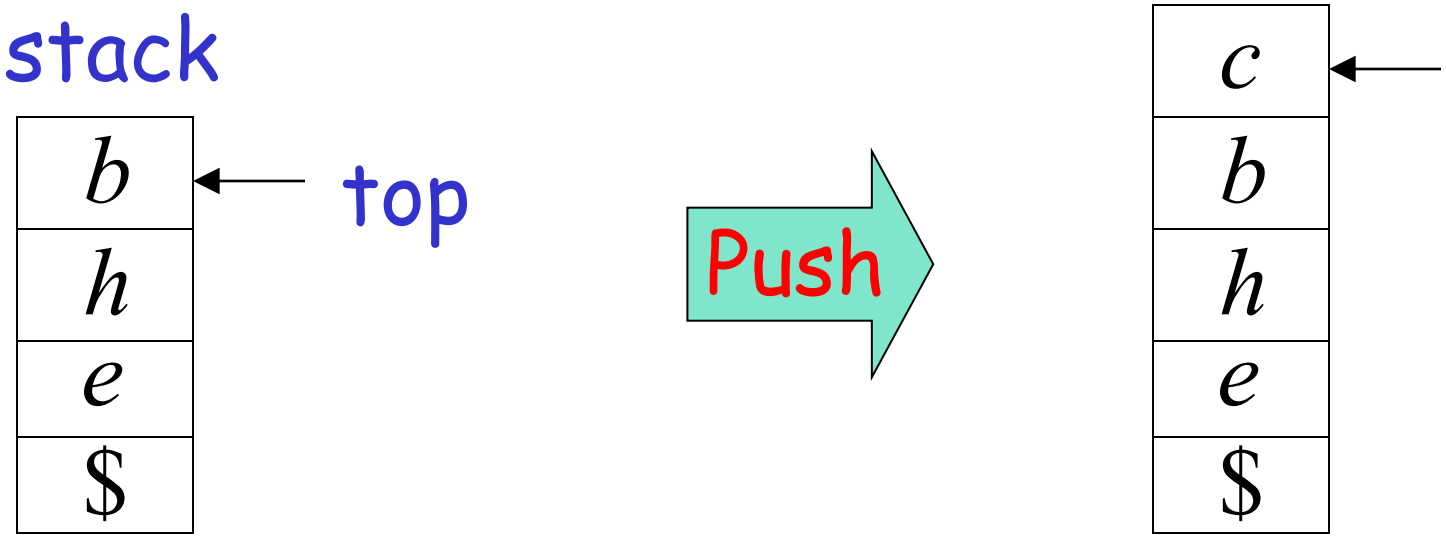

$a,\ b \to c$
q_1
q_2

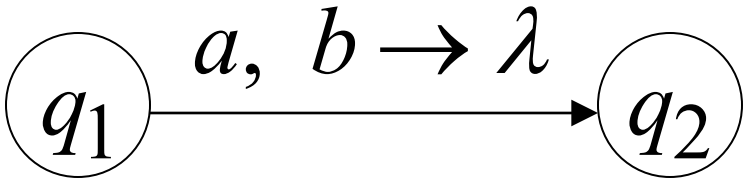

input
…
a
…
…
a
…


stack
b
top
h
e
$\$$
diventa
c
h
e
$\$$

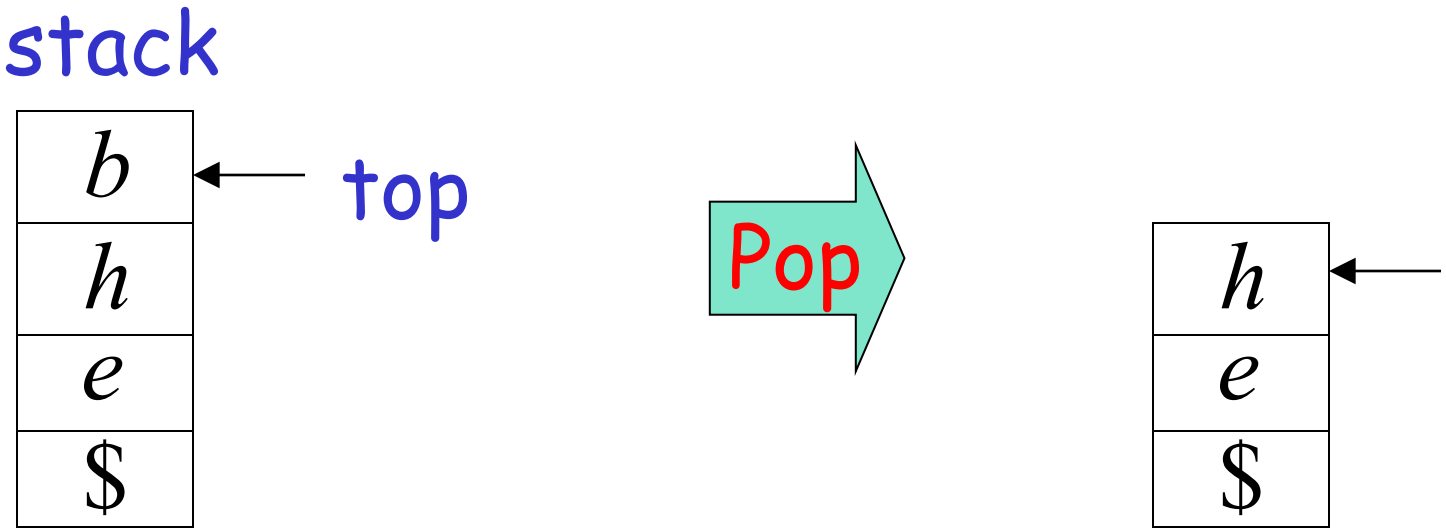

$a,\ \lambda \rightarrow c$
q_1
q_2

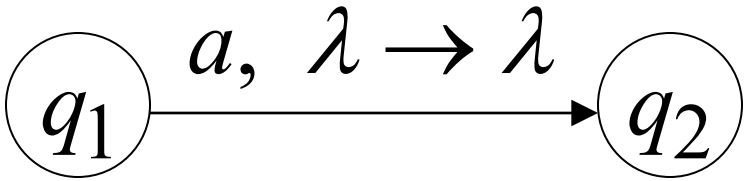

input
… a …
… a …


stack
b
h
e
$
top
Push
c
b
h
e
$

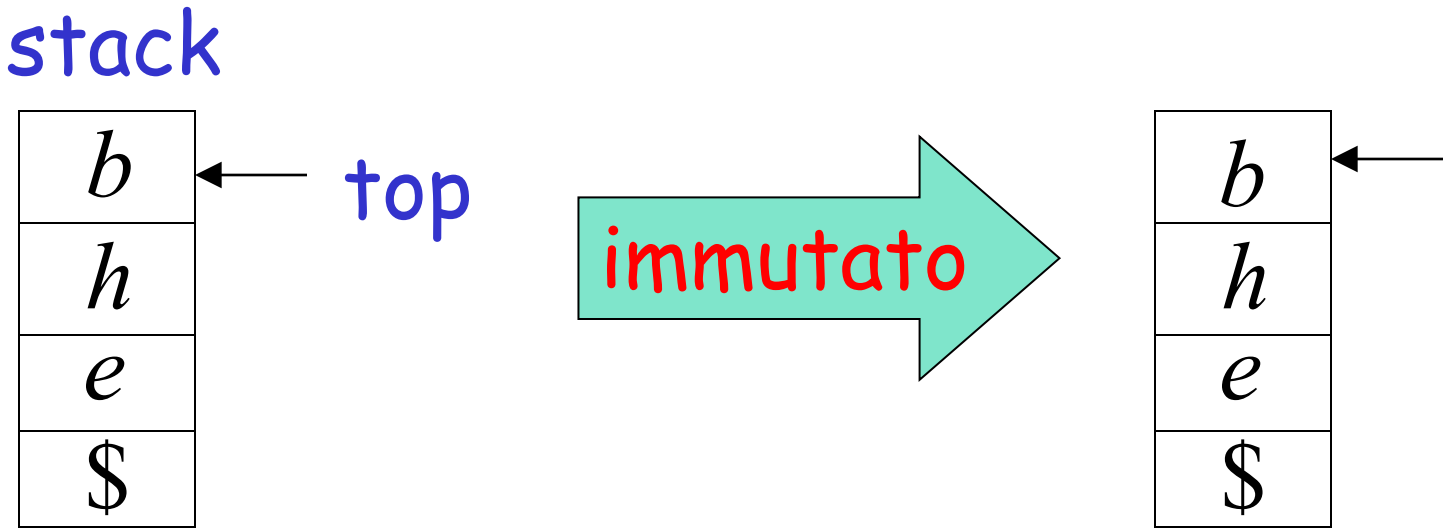

$a,\ b \to \lambda$
q_1
q_2


input
… a …
… a …

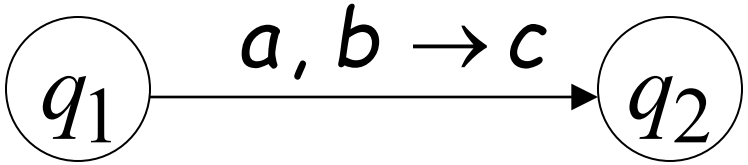

stack
b
h
e
$
top
Pop
h
e
$

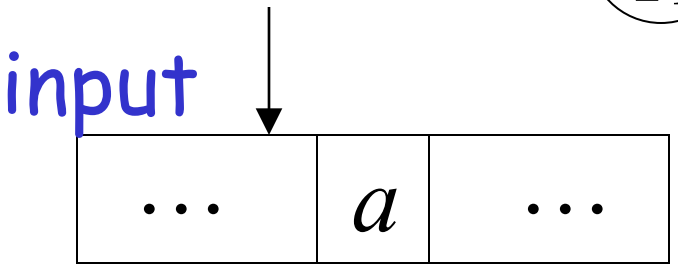

$a,\ \lambda \rightarrow \lambda$
q_1
q_2


input
…
a
…
…
a
…

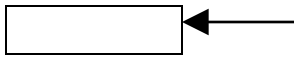

stack
b
h
e
$
top
immutato
b
h
e
$

Pop da uno stack vuoto





stack



Pop

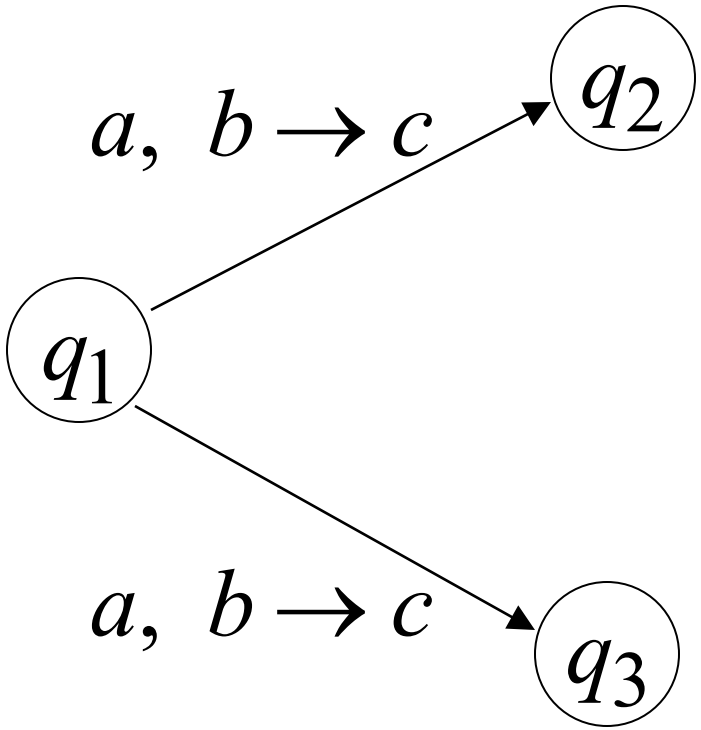
Automa si ferma!

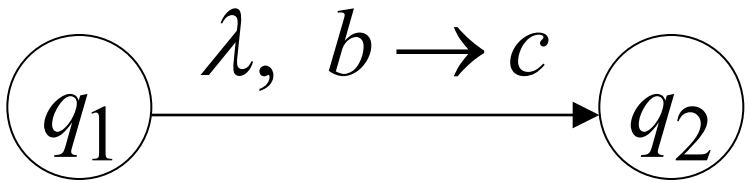
Se l'automata tenta di fare un pop da uno stack vuoto allora si ferma la computazione e rigetta l'input

Non-Determinismo

PDAs sono non-deterministici

Permettono transizioni non deterministiche

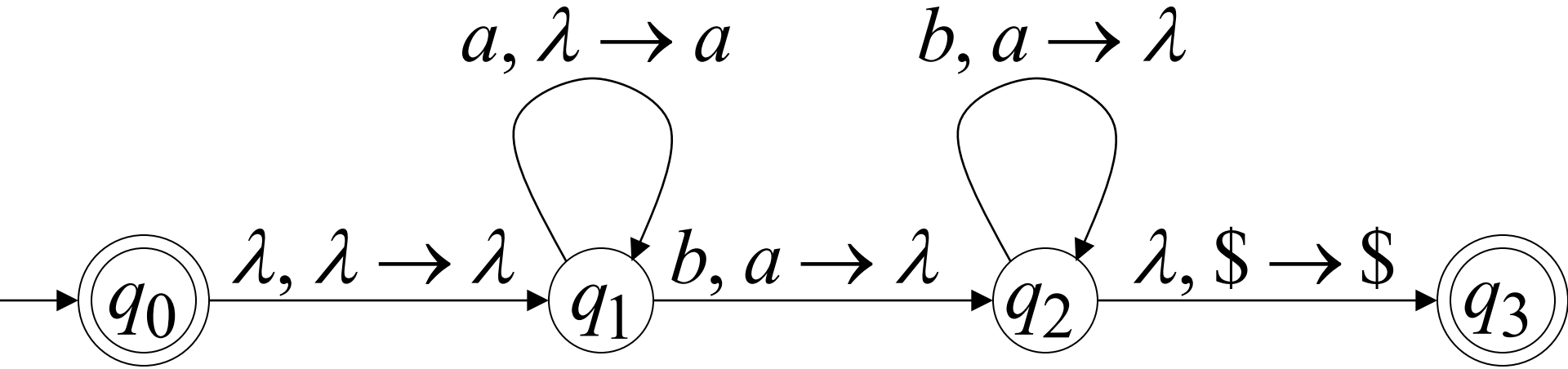




λ – transition

Esempio di PDA

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



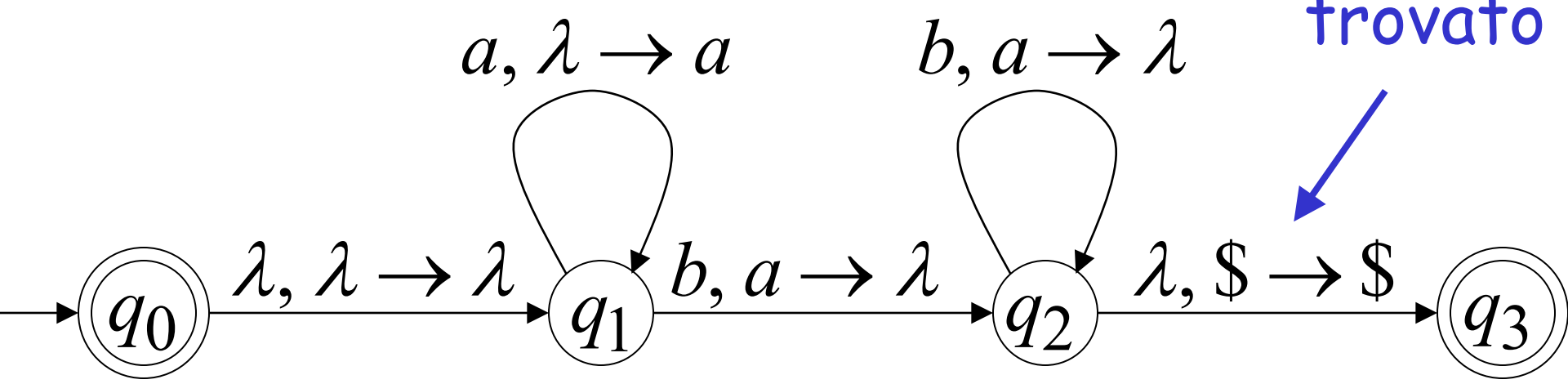
$$L(M) = \{a^n b^n : n \geq 0\}$$

Idea di base:

1. Push le a's nello stack

2. Verifica le b's in input con le a's nello stack

3. Match trovato



Esempio di esecuzione:

Time 0

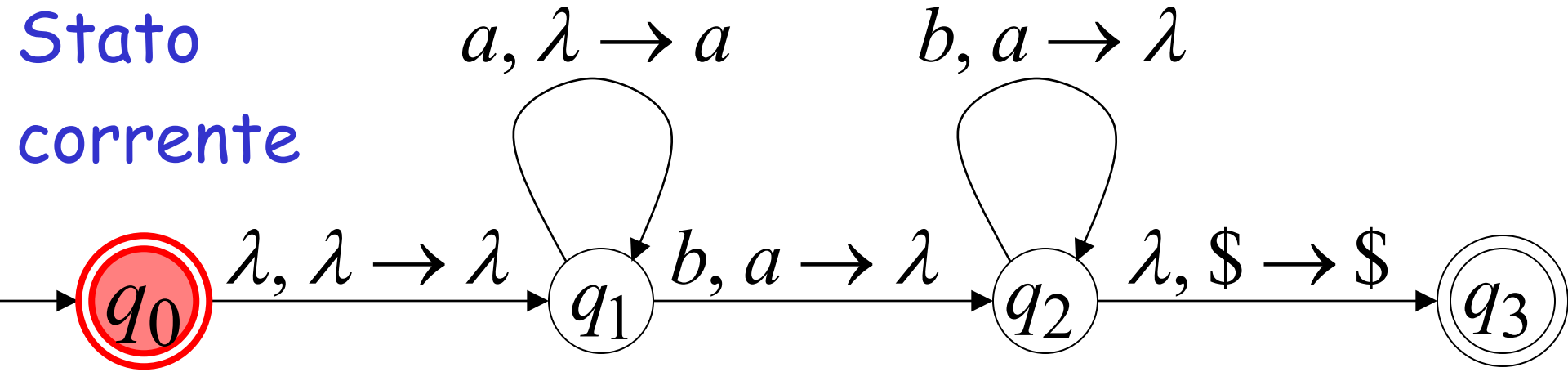
Input

a	a	a	b	b	b

$

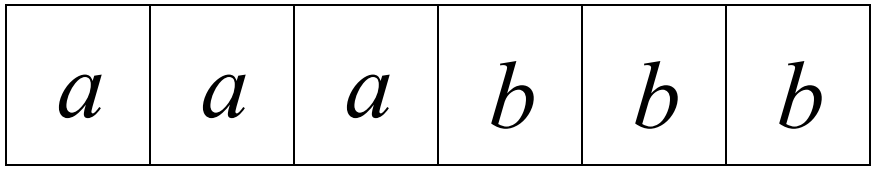
Stack

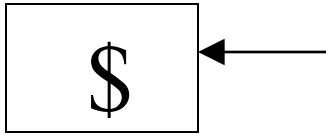
Stato corrente



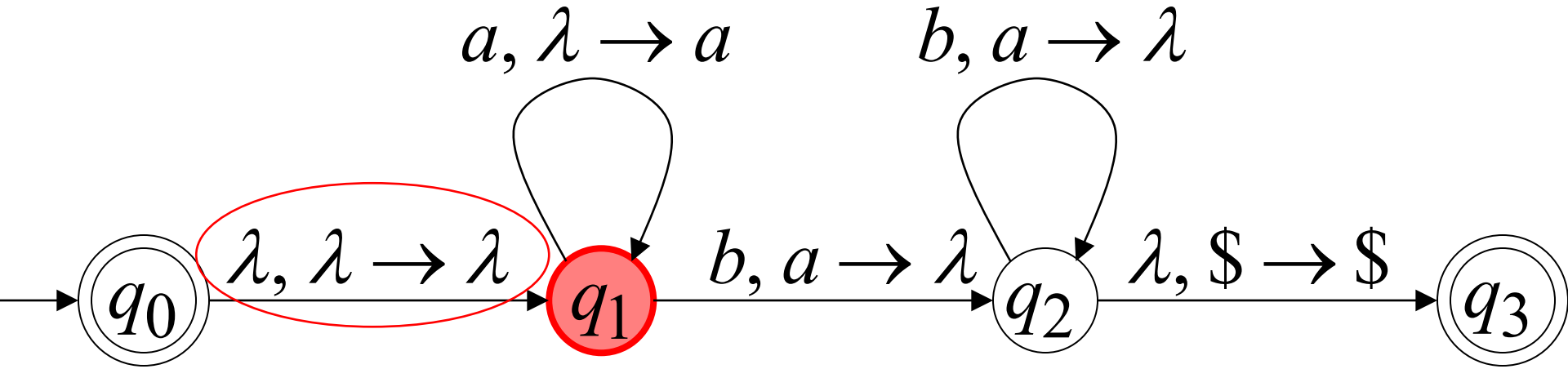
Time 1

Input



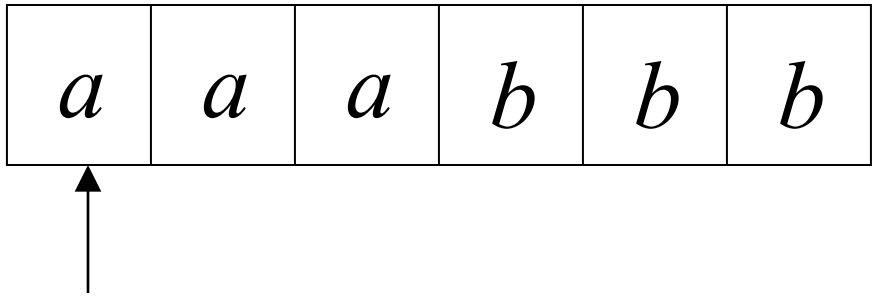


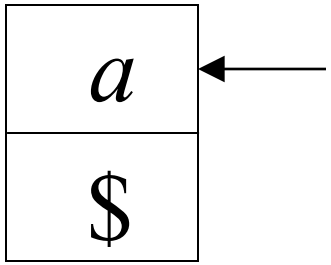
Stack



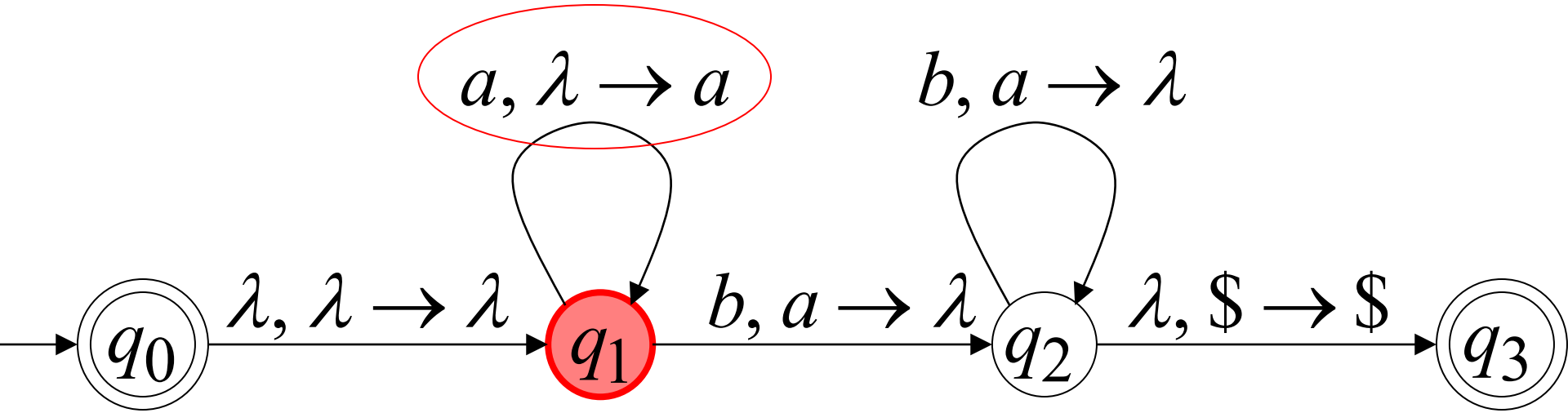
Time 2

Input





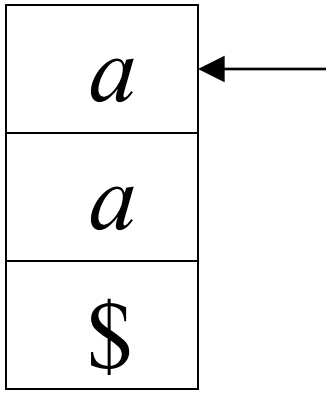
Stack



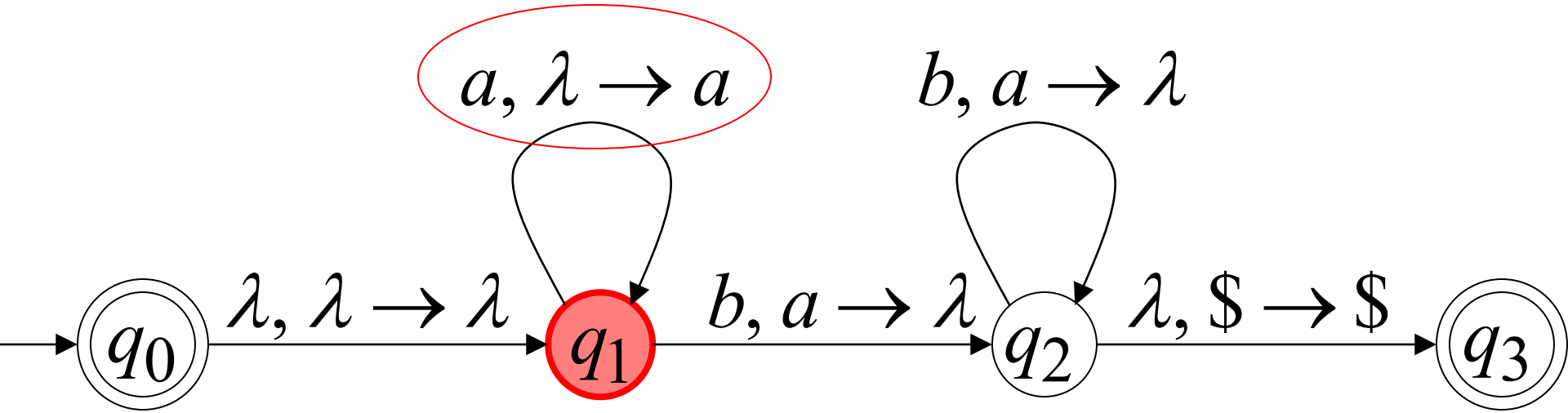
Time 3

Input

a	a	a	b	b	b

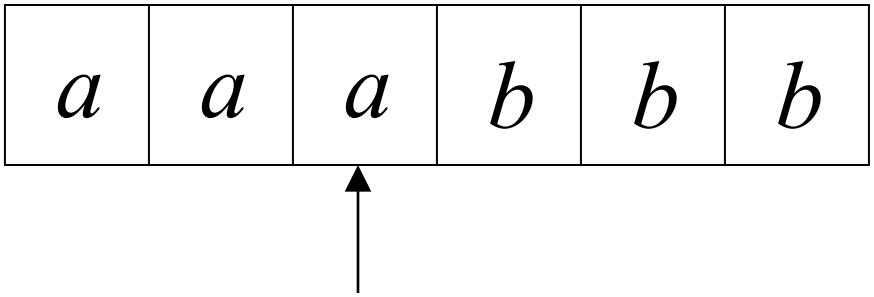


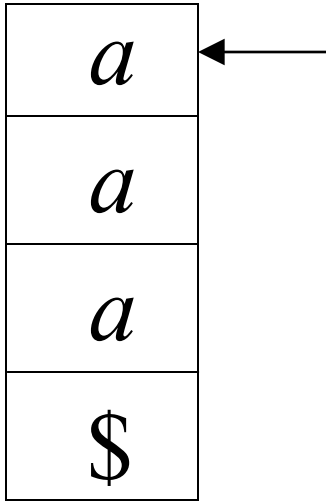
Stack



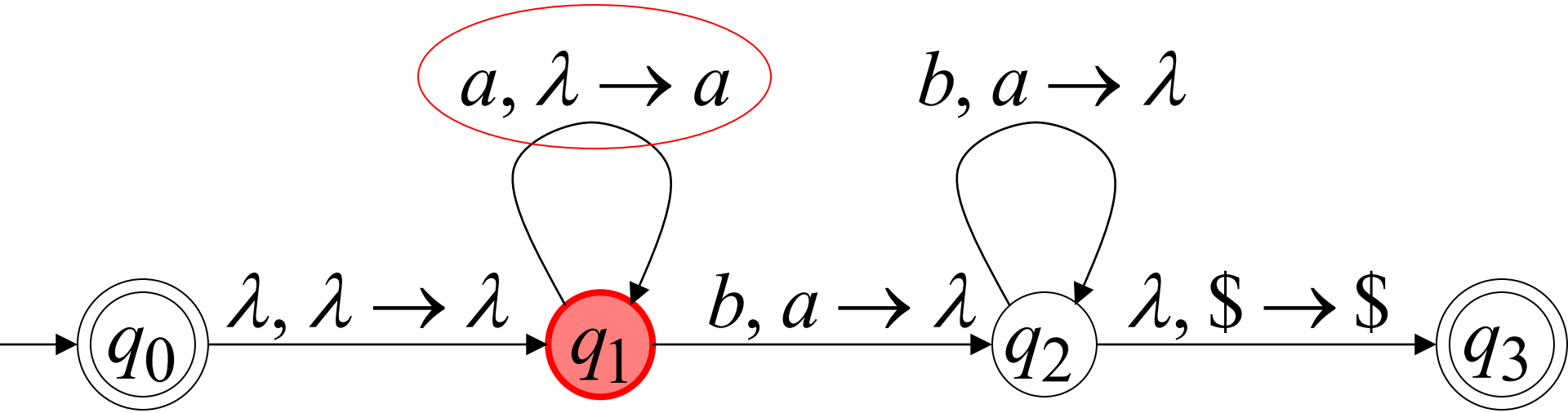
Time 4

Input



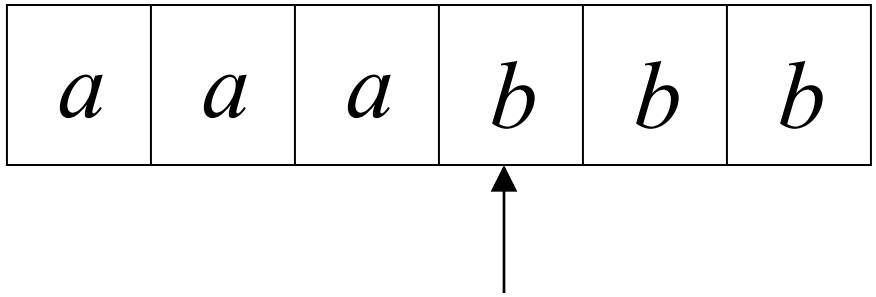


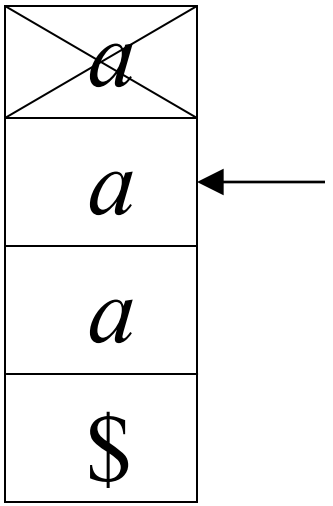
Stack



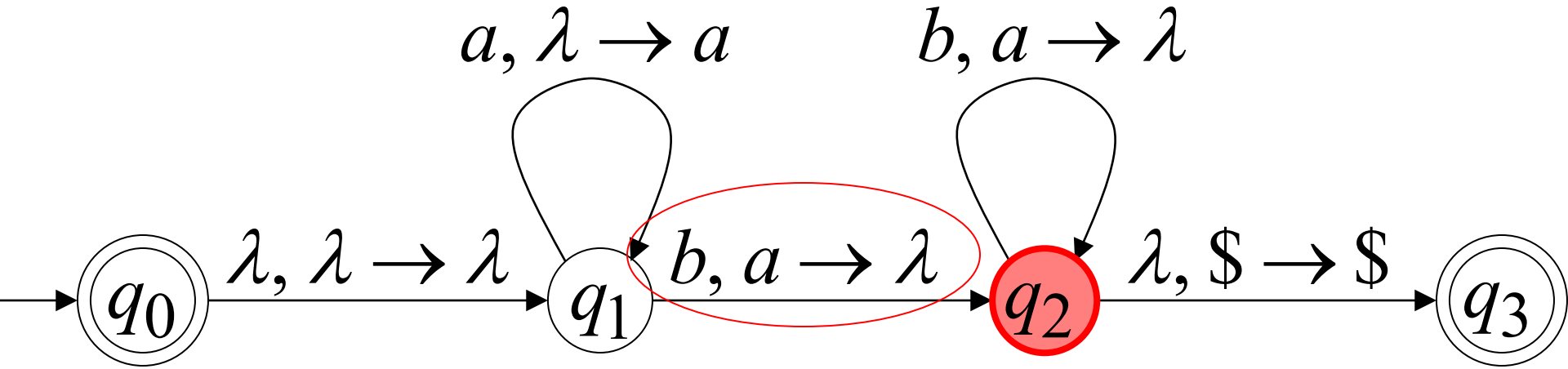
Time 5

Input



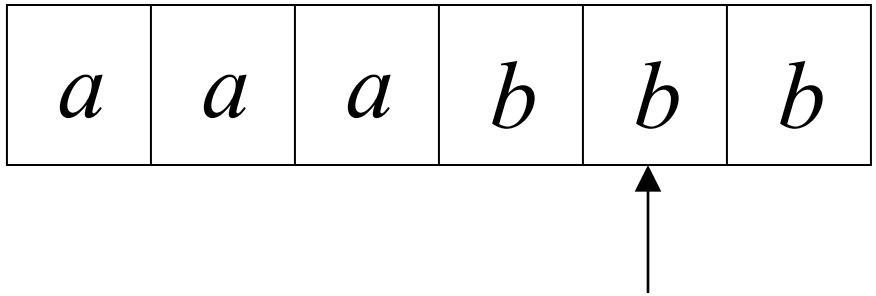


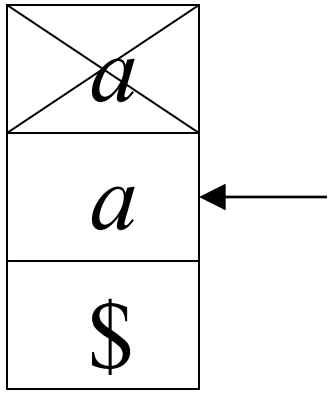
Stack



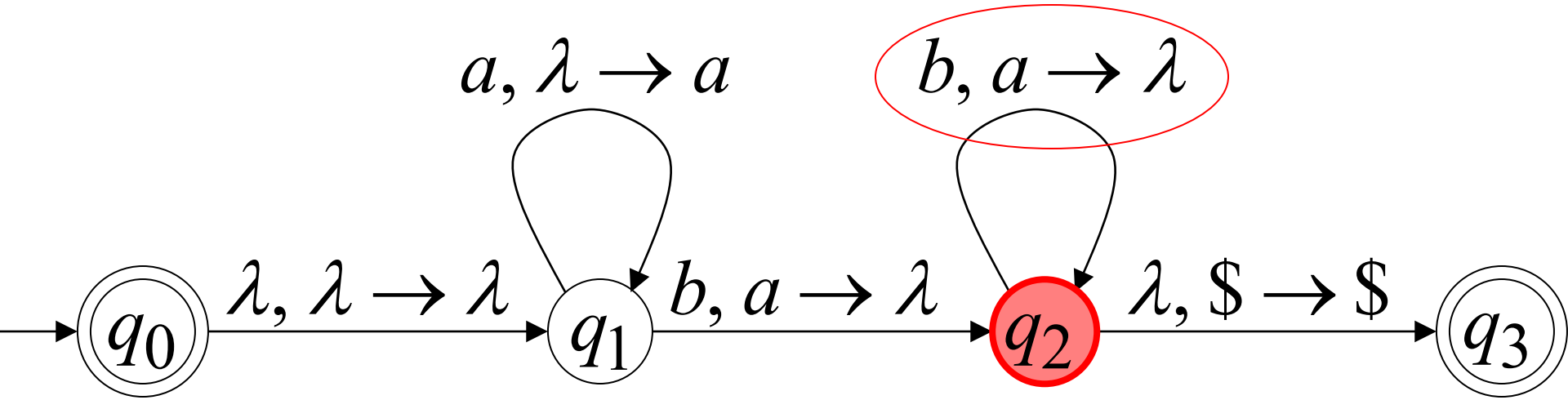
Time 6

Input



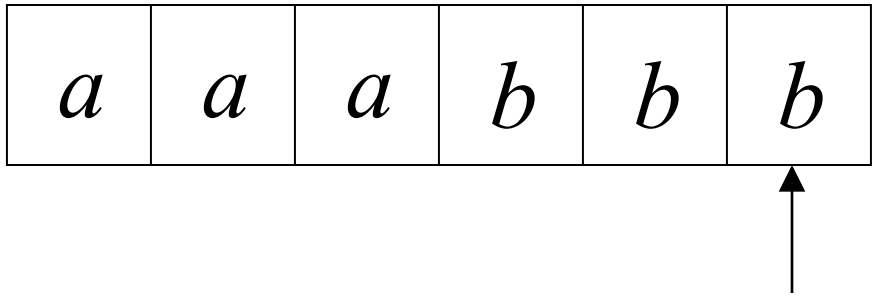


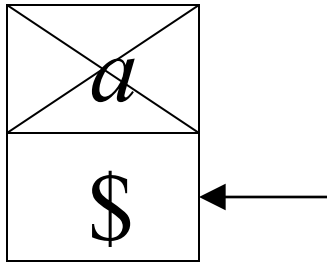
Stack



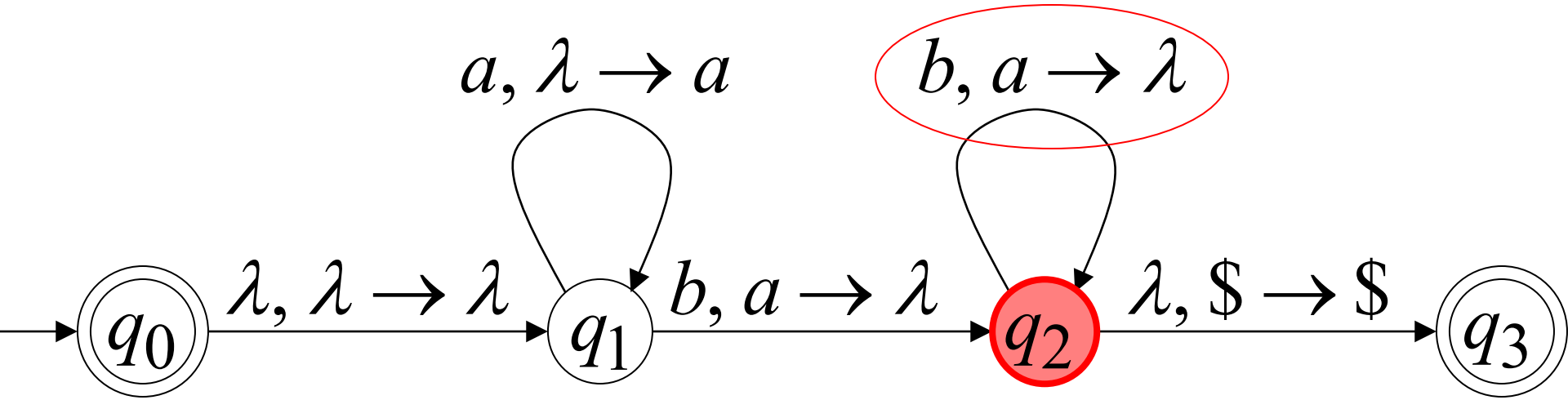
Time 7

Input



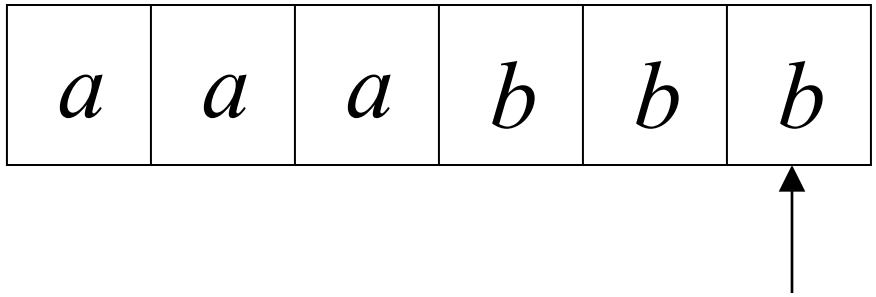


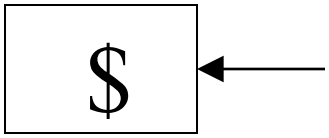
Stack



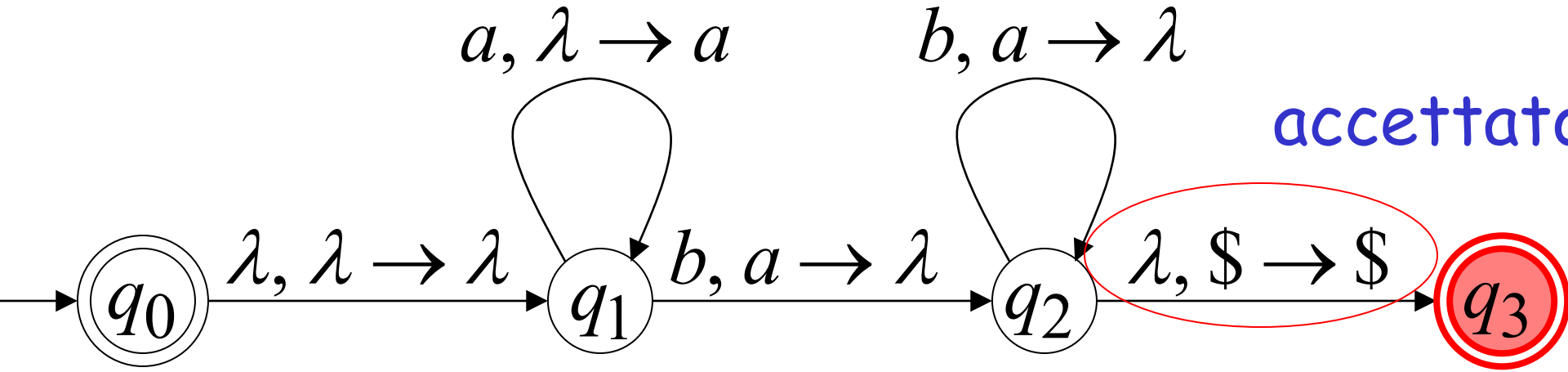
Time 8

Input





Stack



accettato

Una stringa è accettata se
vi è una computazione tale che:

tutti gli input sono "consumati"

E

lo stato raggiunto è uno stato di accettazione

Non teniamo conto di quello che c'è nello Stack alla fine dello stato di accettazione

Esempio di non acettazione:

Time 0

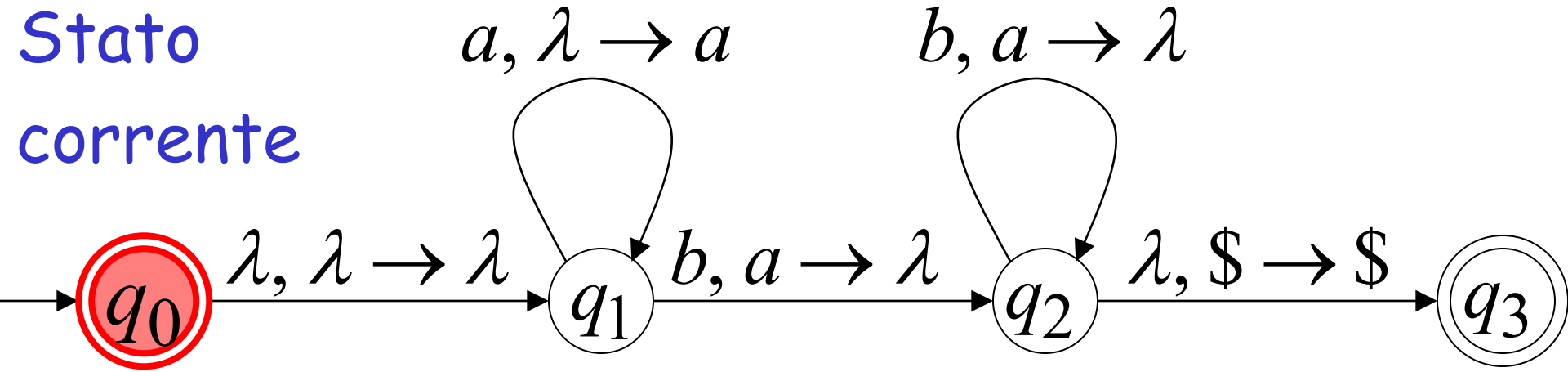
a	a	b

Input

$

Stack

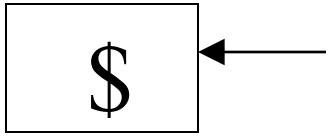
Stato corrente



non acettazione : Time 1

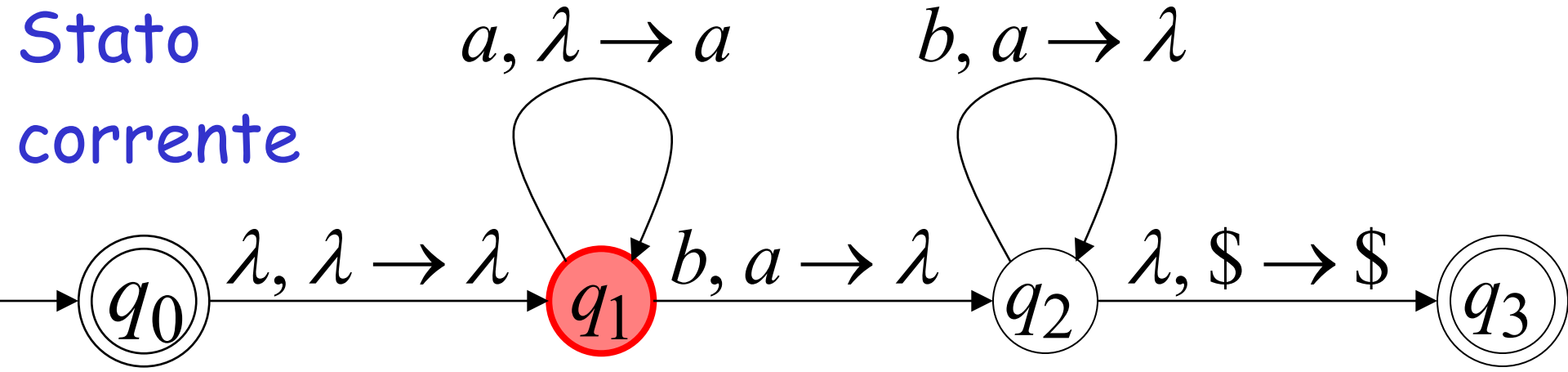
Input

a	a	b



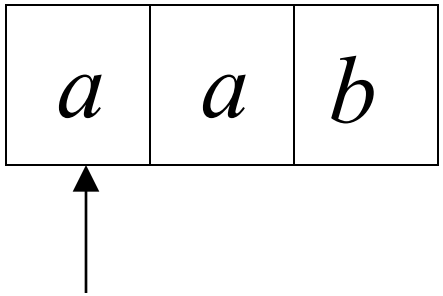
Stack

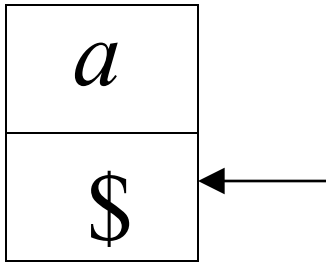
Stato corrente



non acettazione : Time 2

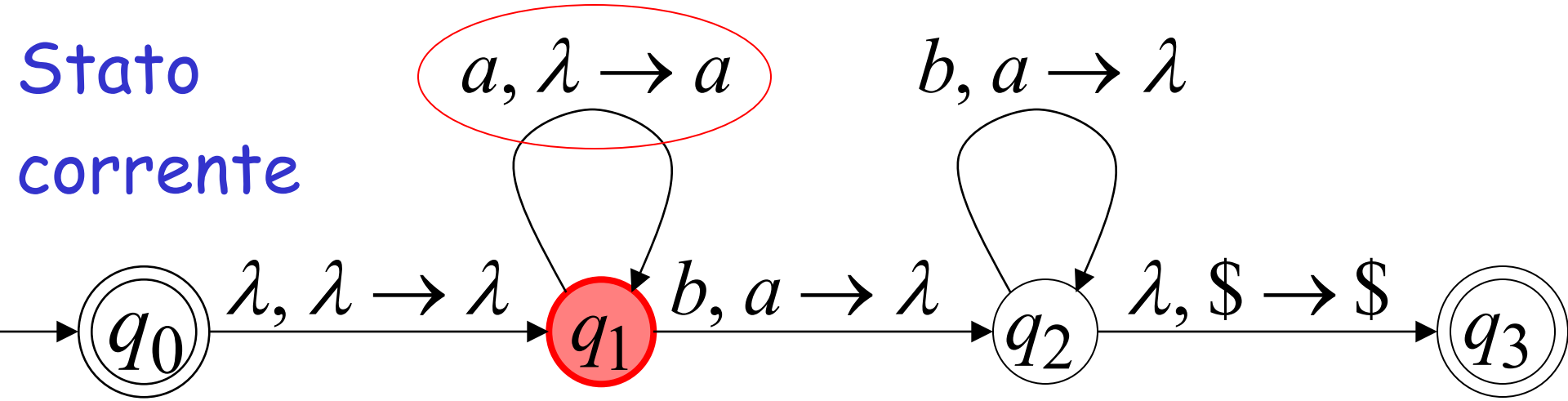
Input





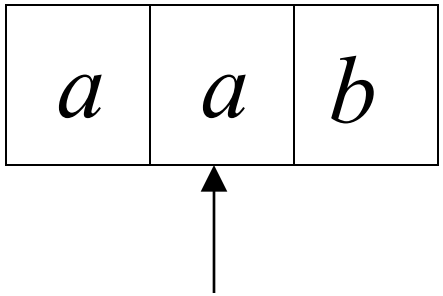
Stack

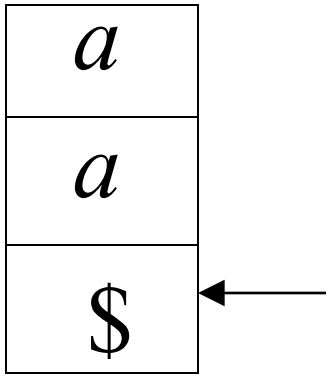
Stato corrente



non acettazione : Time 3

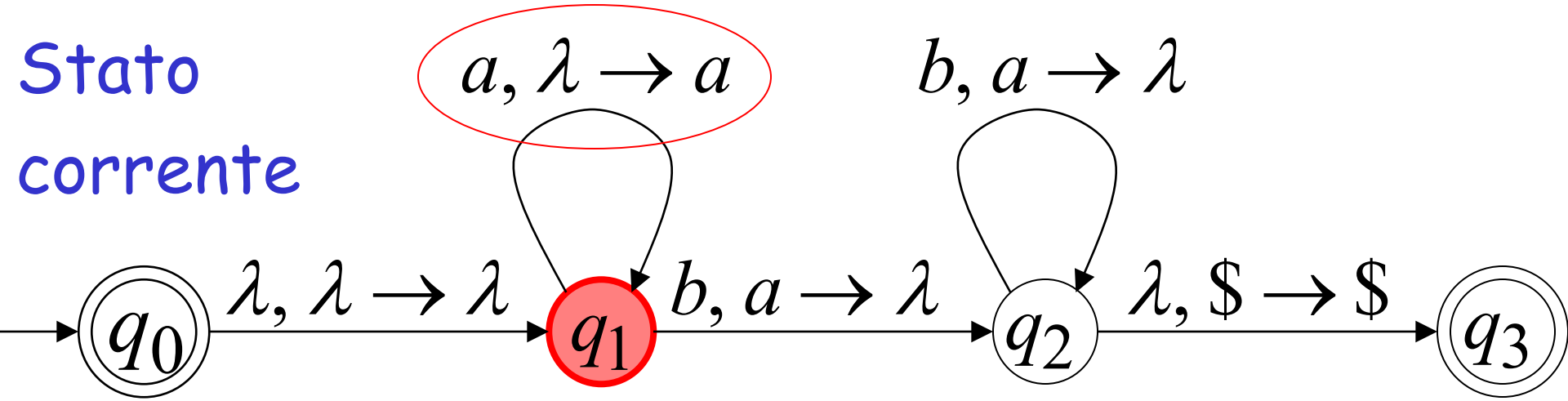
Input





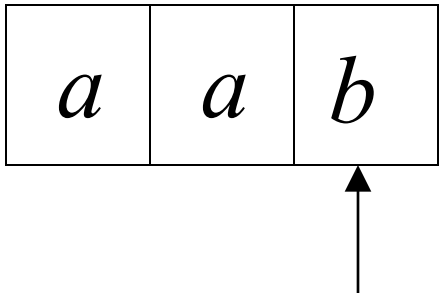
Stack

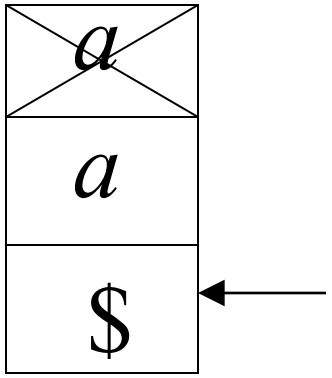
Stato corrente



non acettazione : Time 4

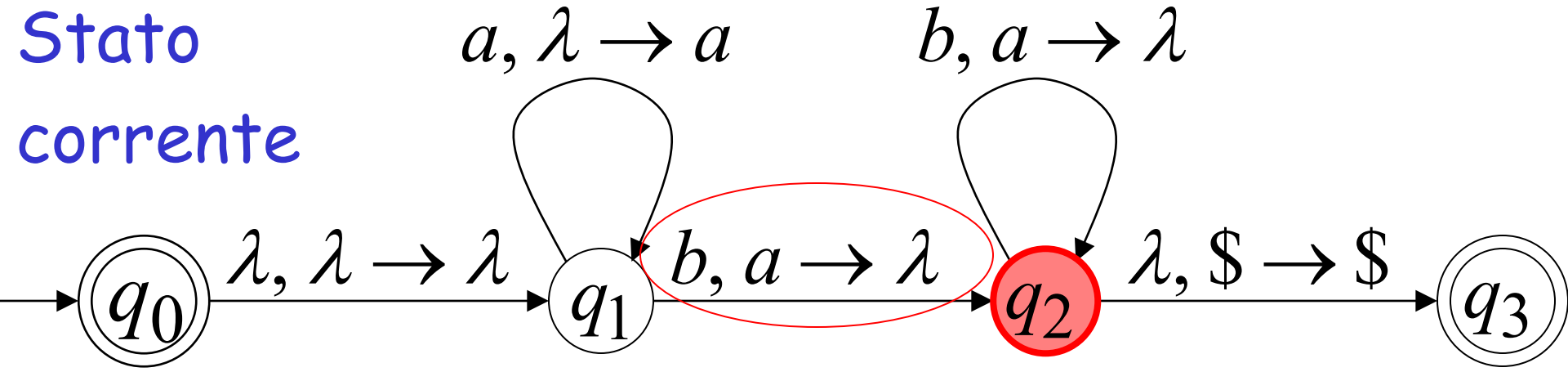
Input





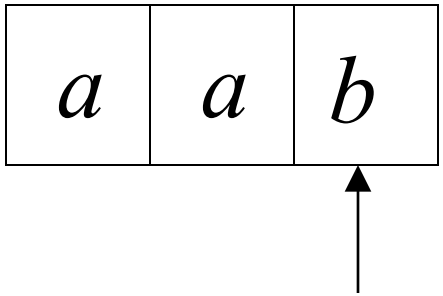
Stack

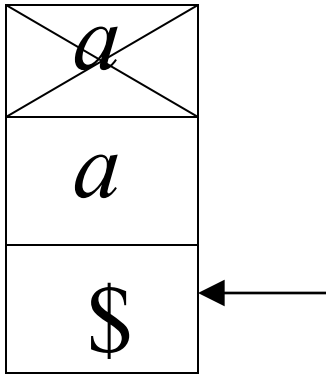
Stato corrente



non accettazione : Time 4

Input

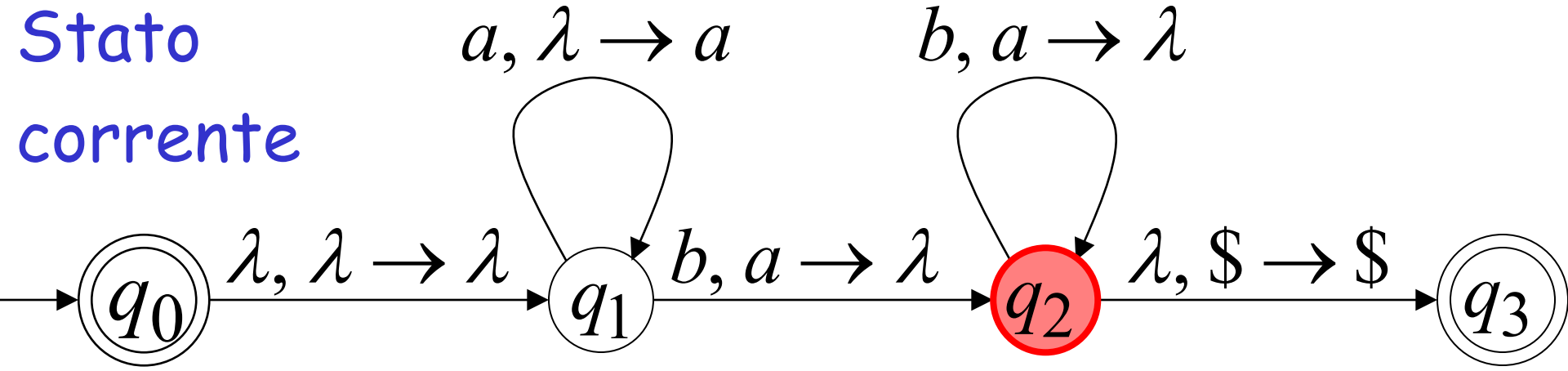




Stack

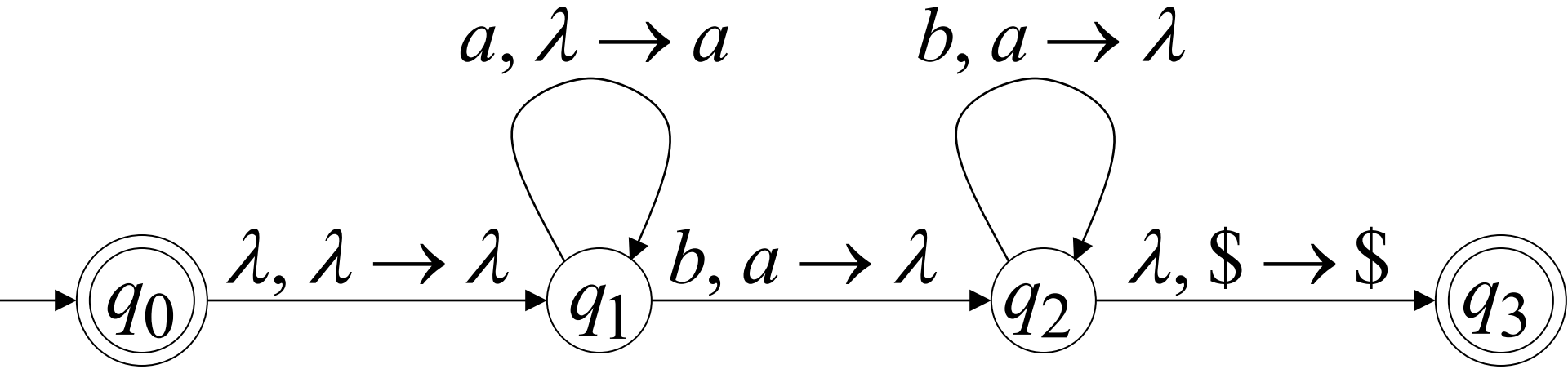
reject

Stato corrente



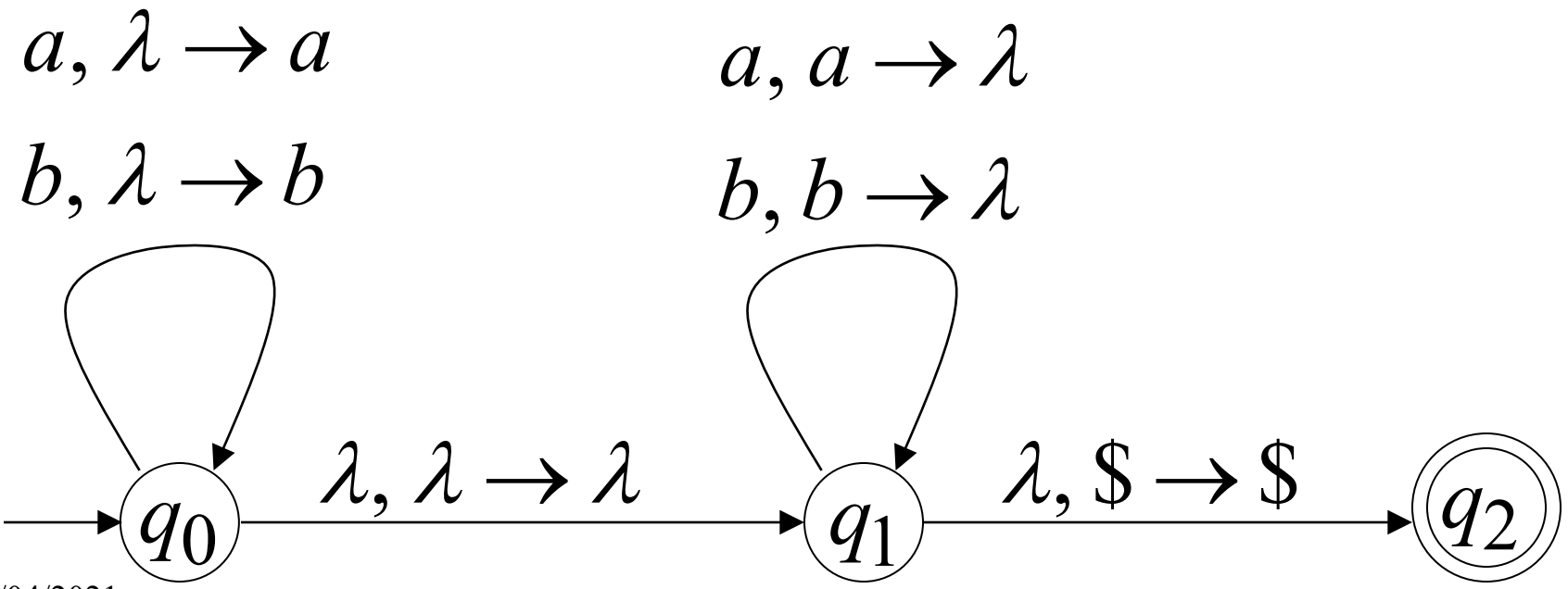
Non esiste una computazione accettante Per aab .

La stringa aab è rigettata dal PDA.



Un altro esempio di PDA

PDA M: $L(M) = \{vv^R : v \in \{a,b\}^*\}$



aby aa y a y aba=xyz

abcy cba

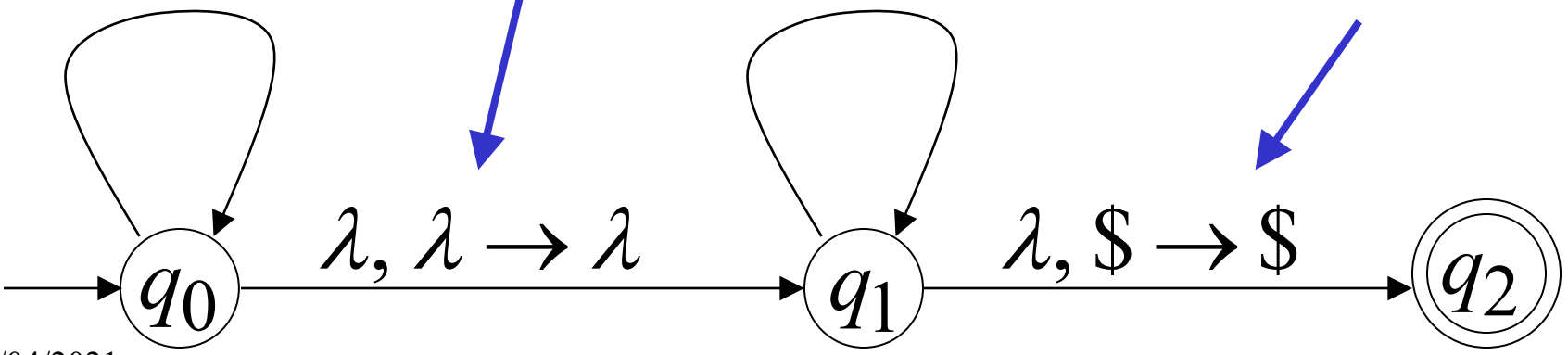
Basic Idea: $L(M) = \{vv^R : v \in \{a,b\}^*\}$

1. Push v
Nello stack

2. Congettura
metà
Dell'input

3. verifica v^R in
Input con v nello
stack

4. Verifica
trovata

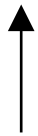


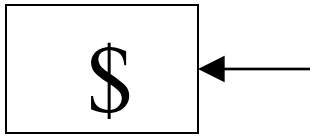
esecuzione:

Time 0

Input

a	b	b	a





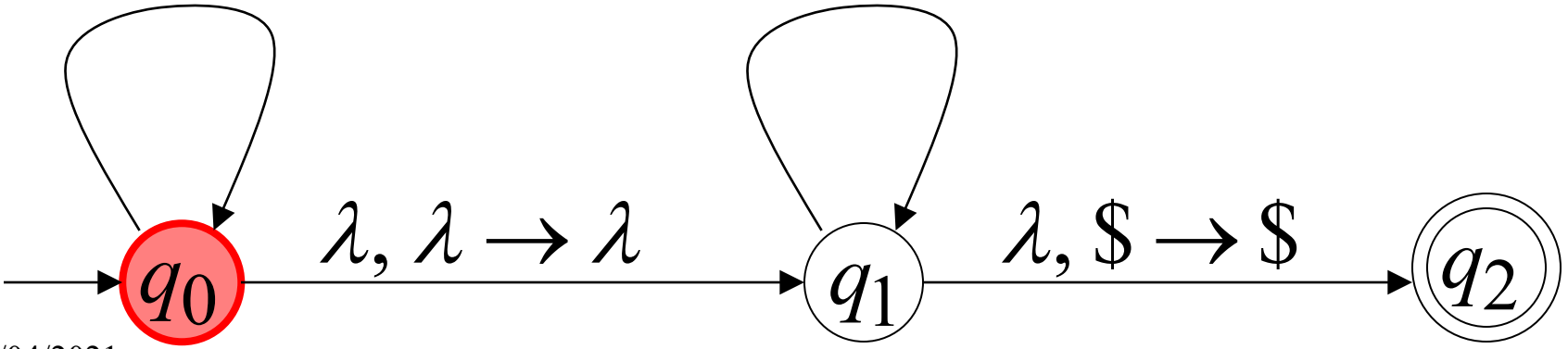
Stack

$a, \lambda \to a$

$b, \lambda \to b$

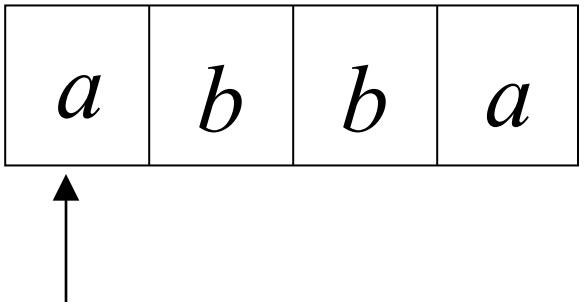
$a, a \to \lambda$

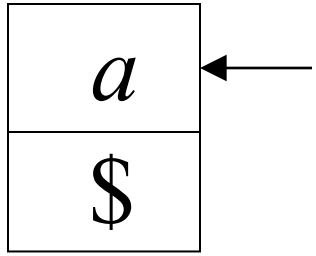
$b, b \to \lambda$



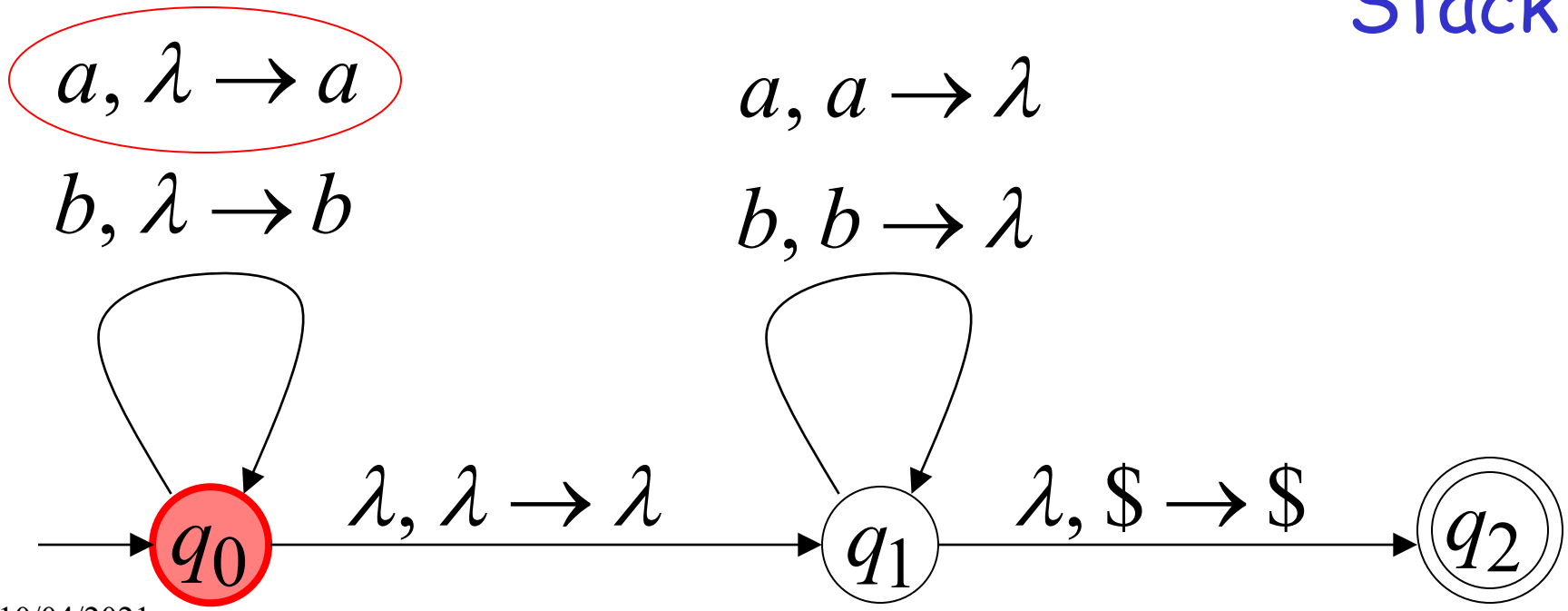
Time 1

Input



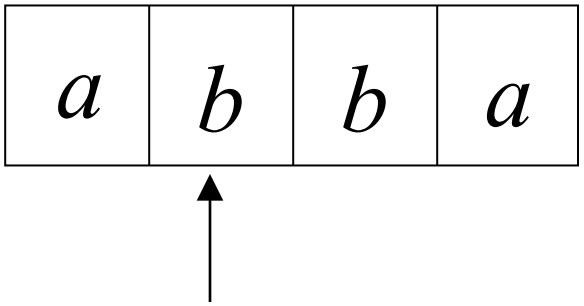


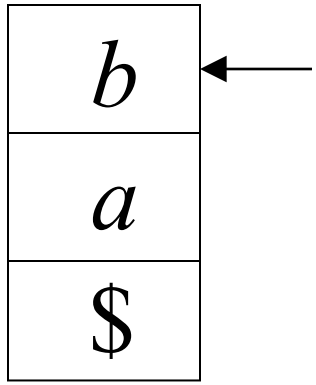
Stack



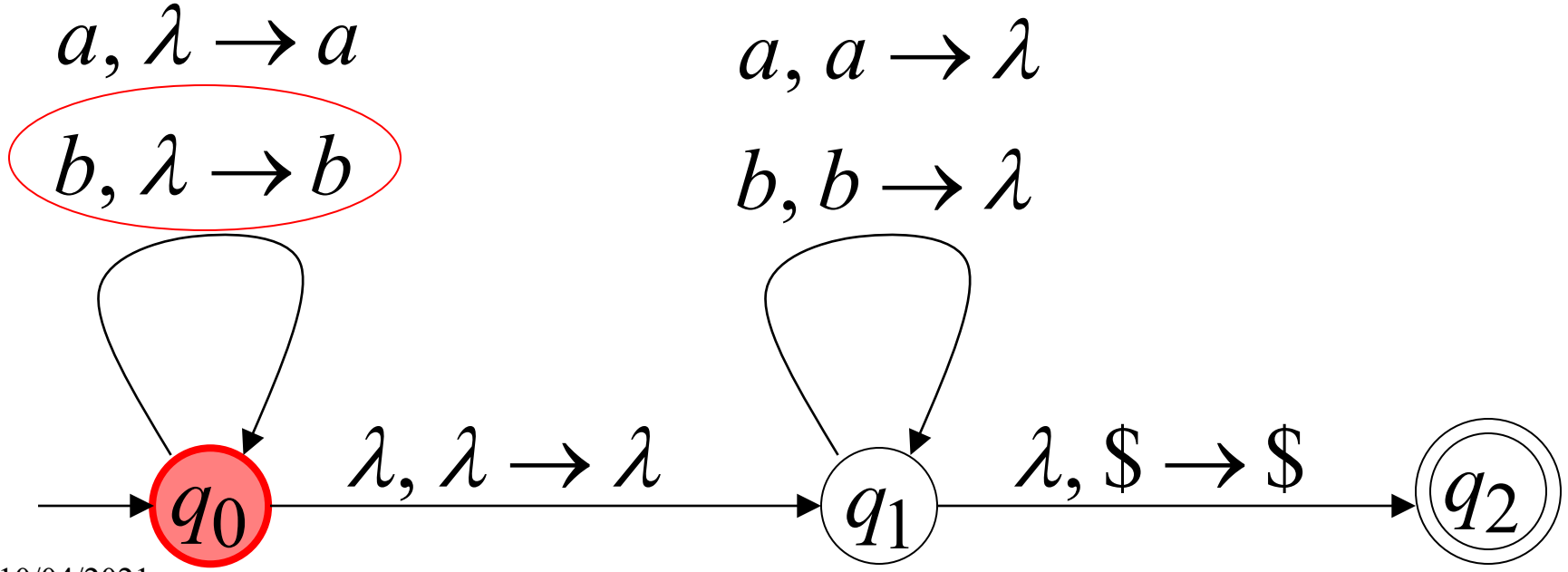
Time 2

Input



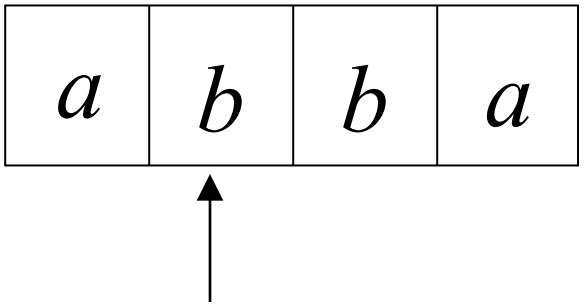


Stack

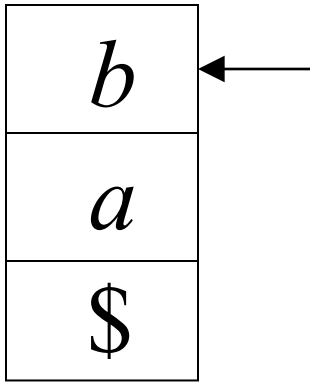


Time 3

Input



Congettura sei
Metà input



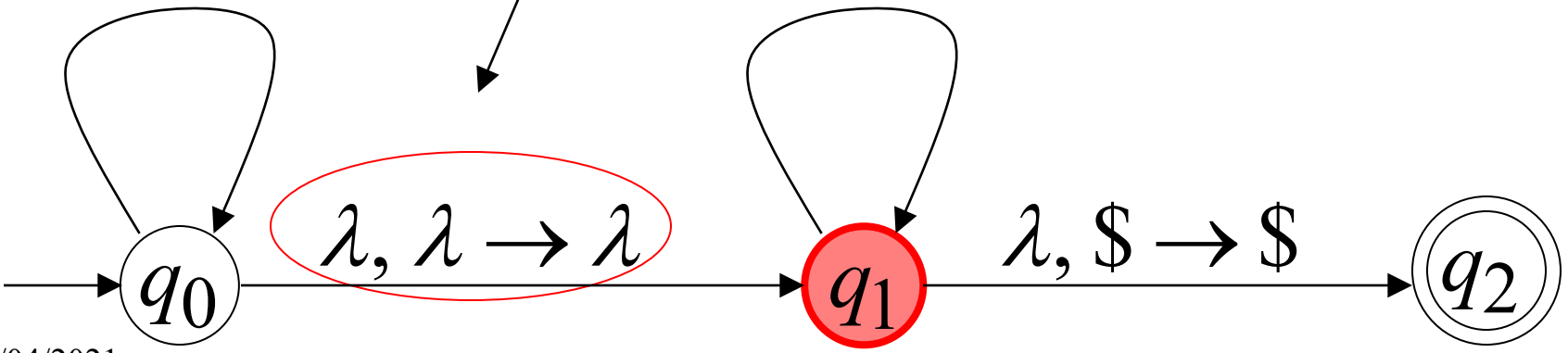
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

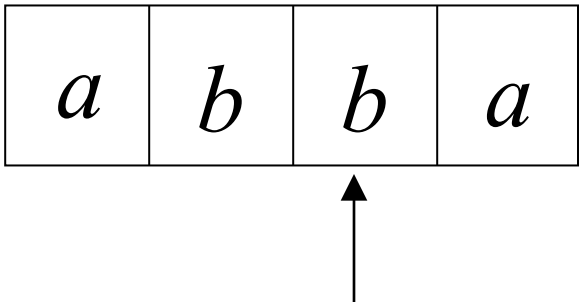
$a, a \rightarrow \lambda$

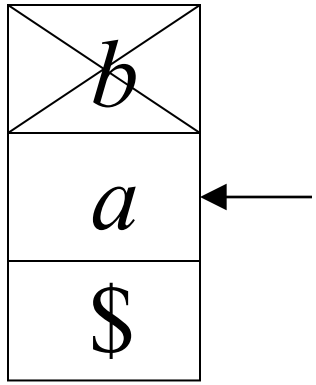
$b, b \rightarrow \lambda$



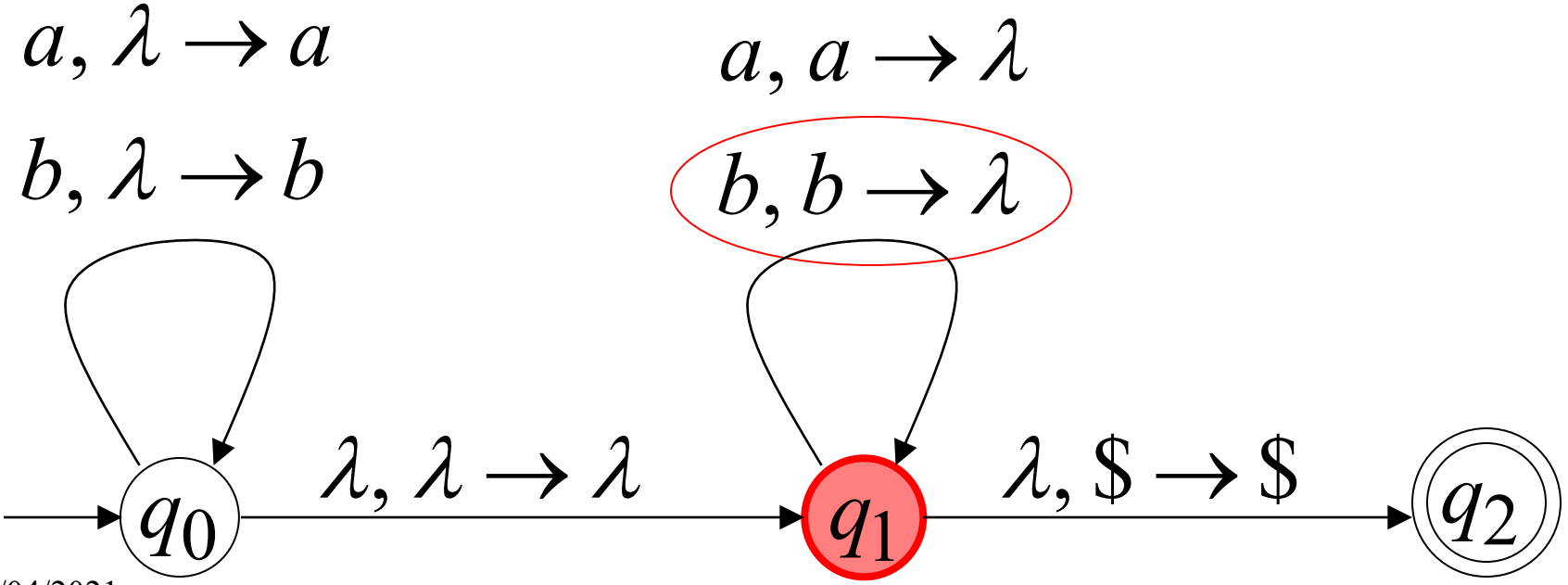
Time 4

Input



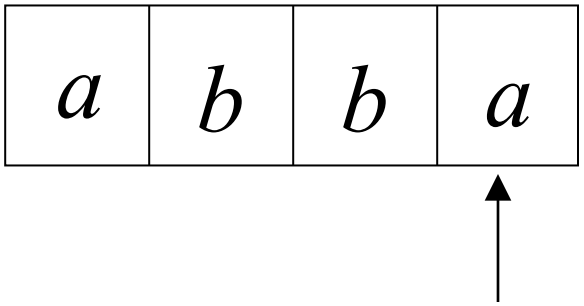


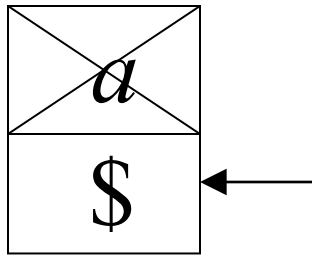
Stack



Time 5

Input





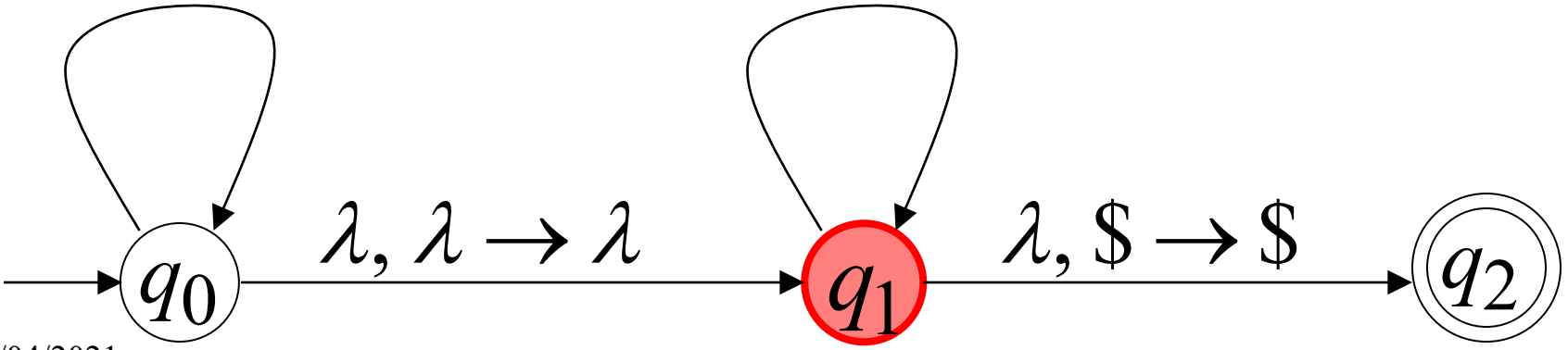
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

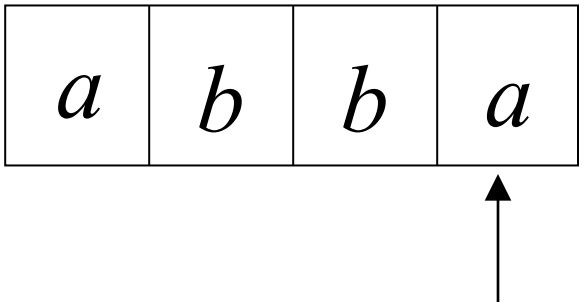
$a, a \rightarrow \lambda$

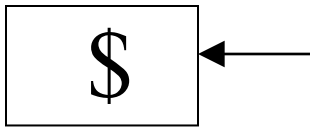
$b, b \rightarrow \lambda$



Time 6

Input





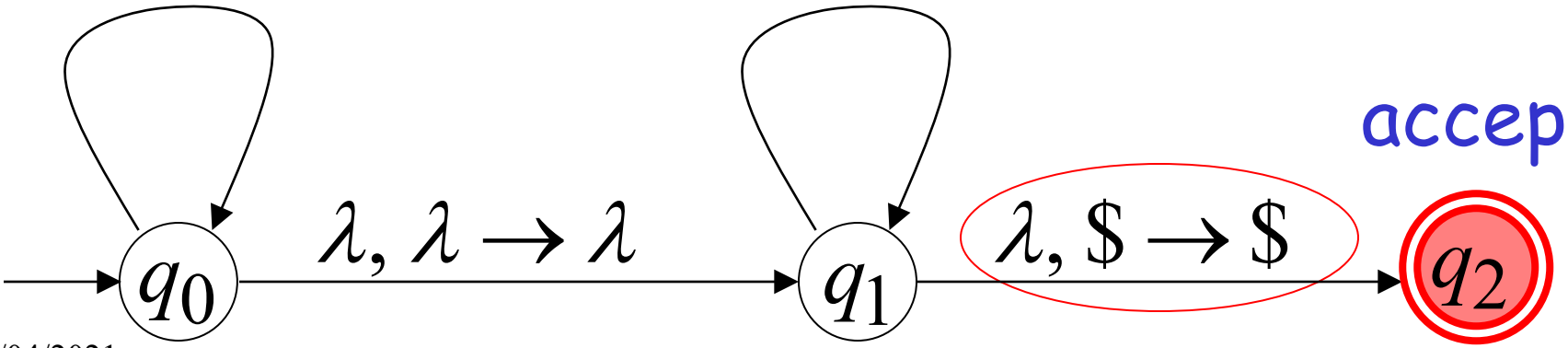
Stack

$a, \lambda \to a$

$b, \lambda \to b$

$a, a \to \lambda$

$b, b \to \lambda$



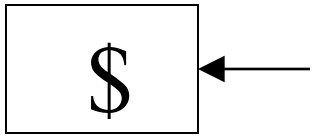
accept

esecuzione:

Time 0

Input

a	b	b	a



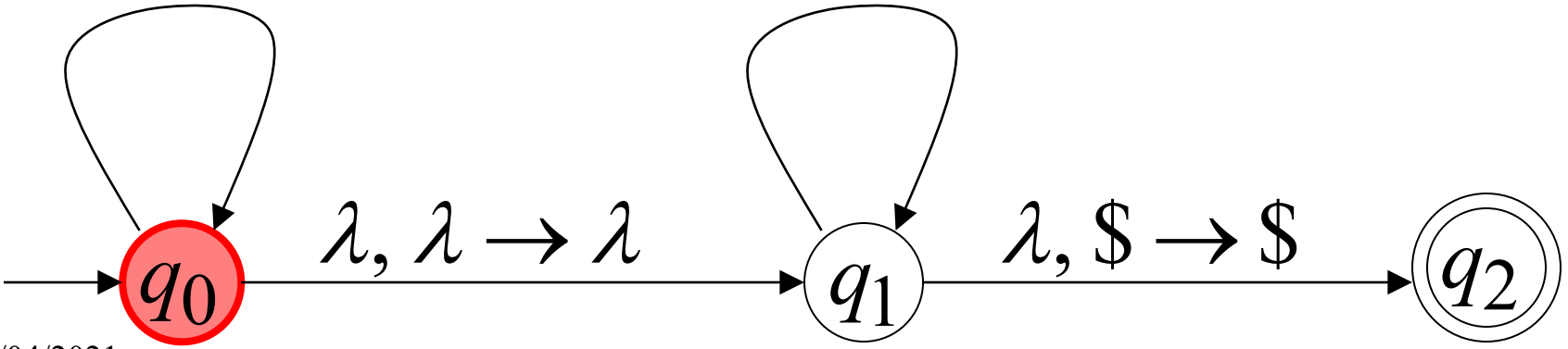
Stack

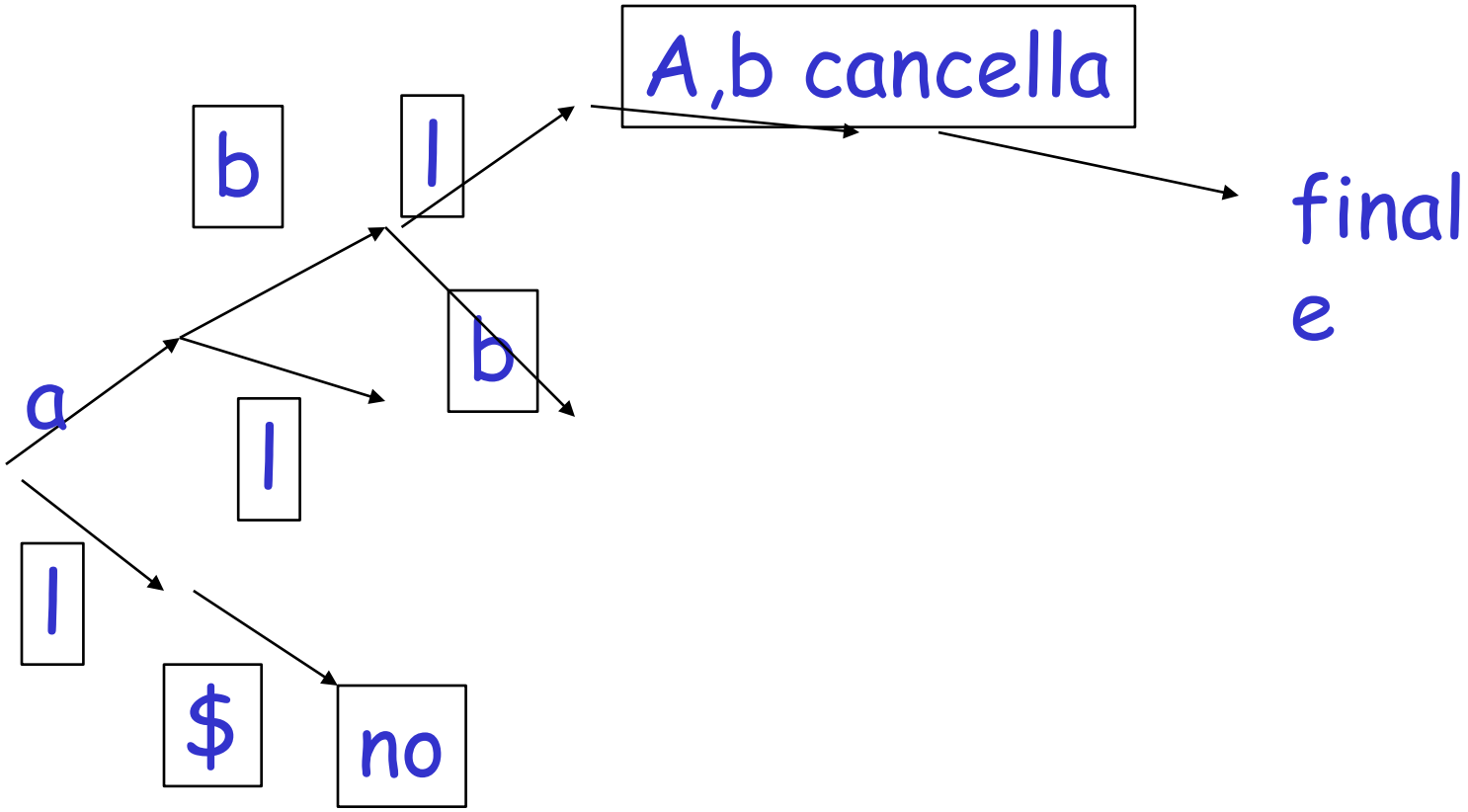
$a, \lambda \to a$

$b, \lambda \to b$

$a, a \to \lambda$

$b, b \to \lambda$

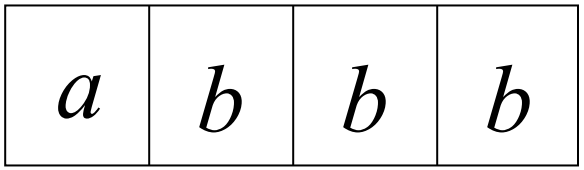


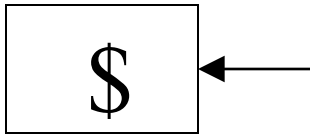

A,b cancella
b
l
b
a
l
l
$
no
final
e

Altro esempio:

Time 0

Input





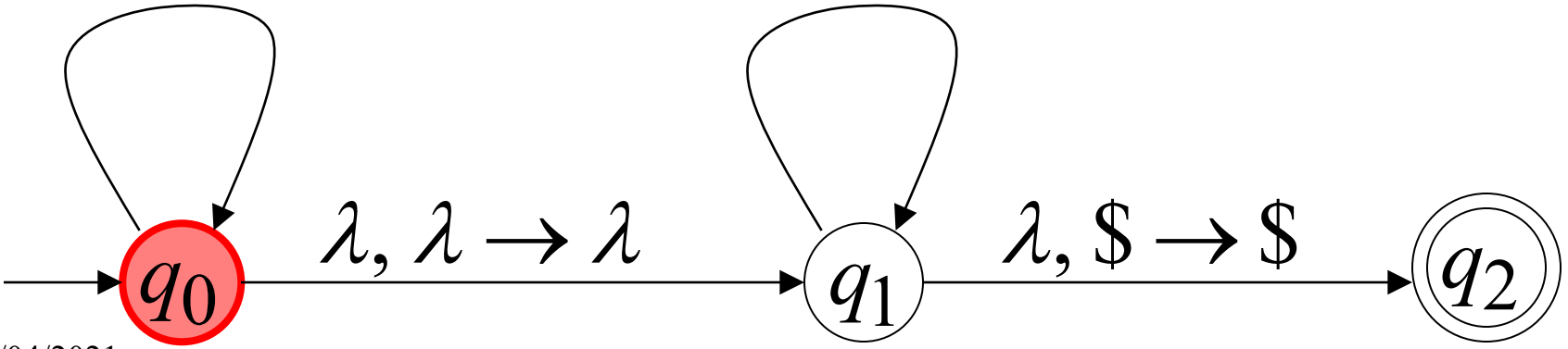
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

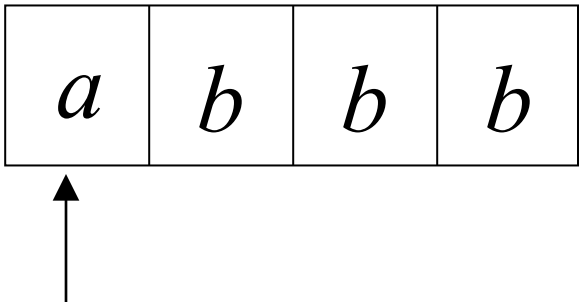
$a, a \rightarrow \lambda$

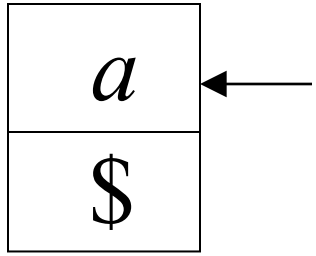
$b, b \rightarrow \lambda$



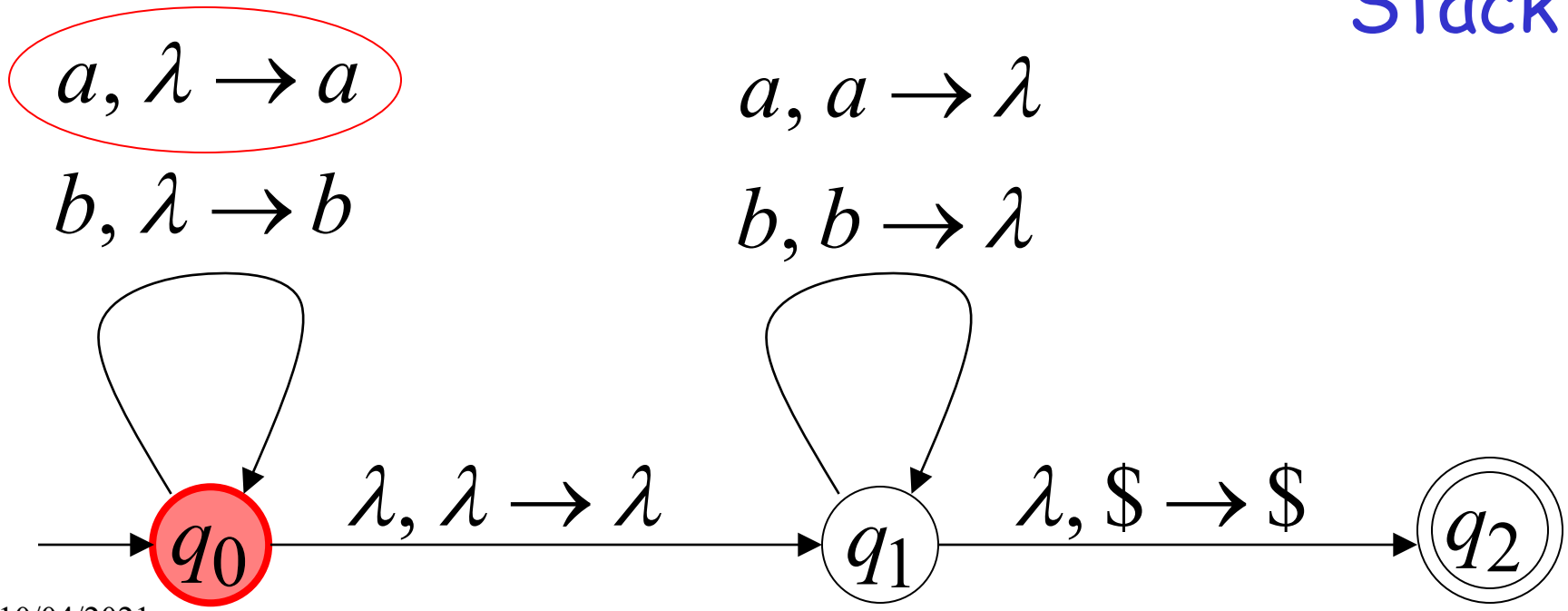
Time 1

Input

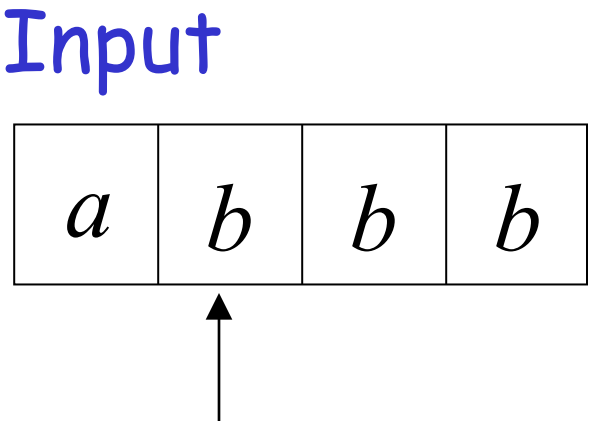


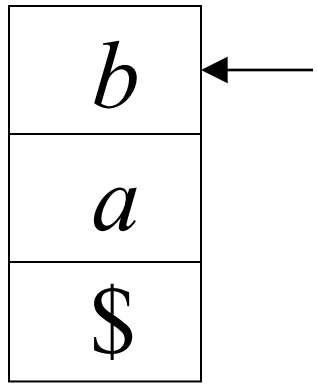


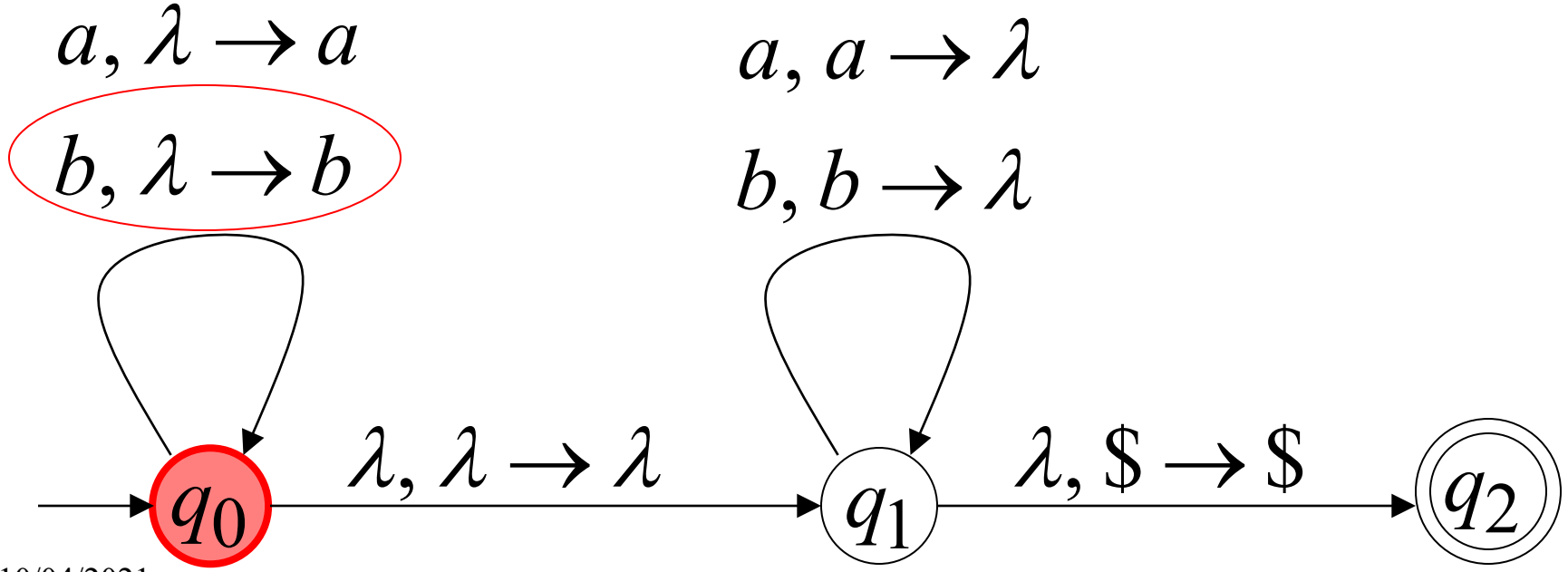
Stack



Time 2







Time 3

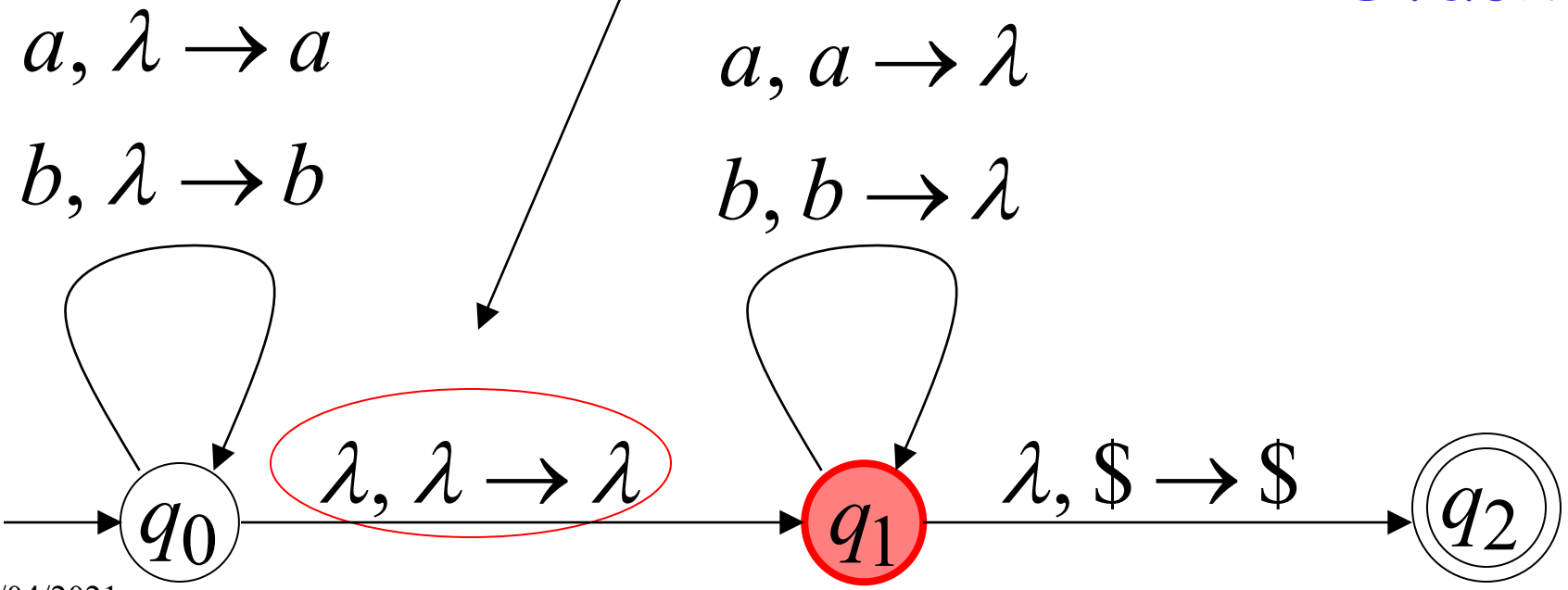
Input

a	b	b	b

Congettura sei
A metà dell'input

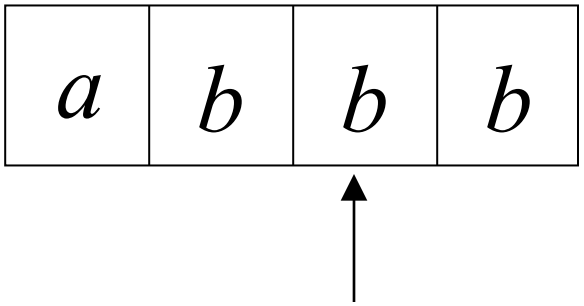
b
a
$

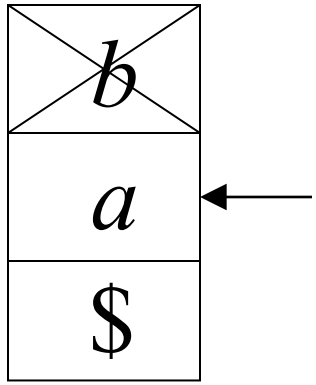
Stack



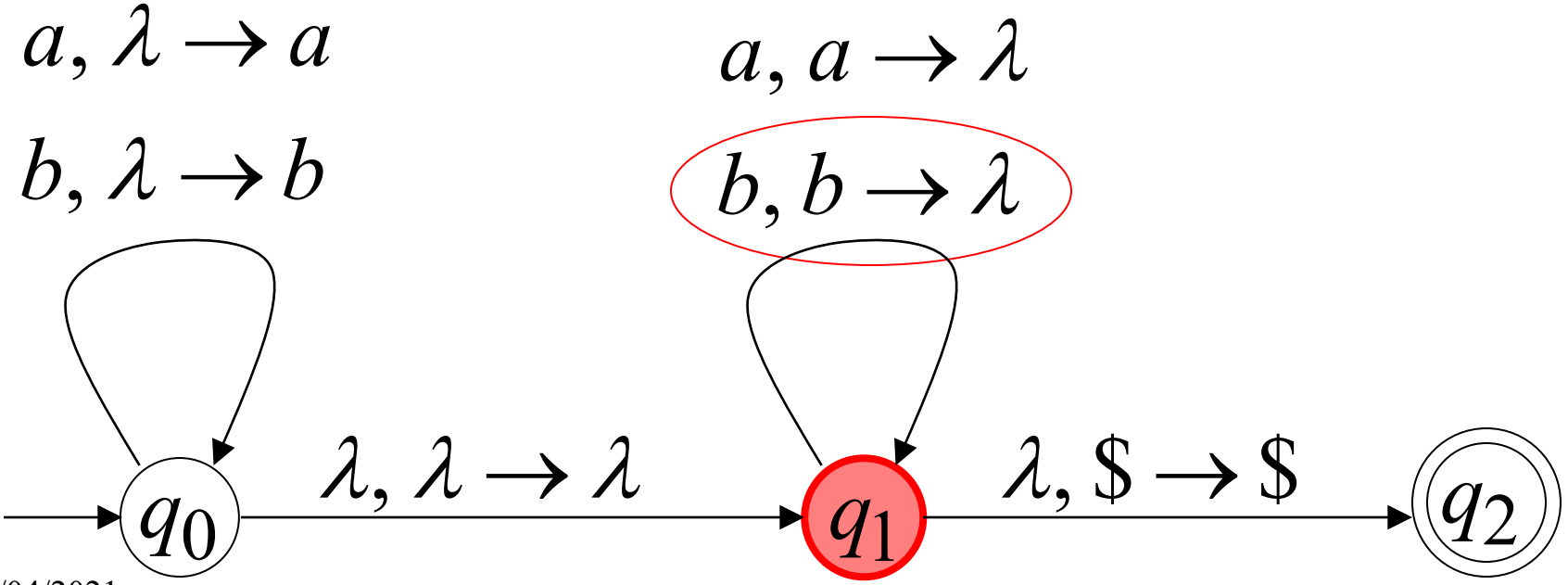
Time 4

Input





Stack



Time 5

Input

Non ci sono possibili transizioni.

a	b	b	b

Input non è stato consumato

a
$

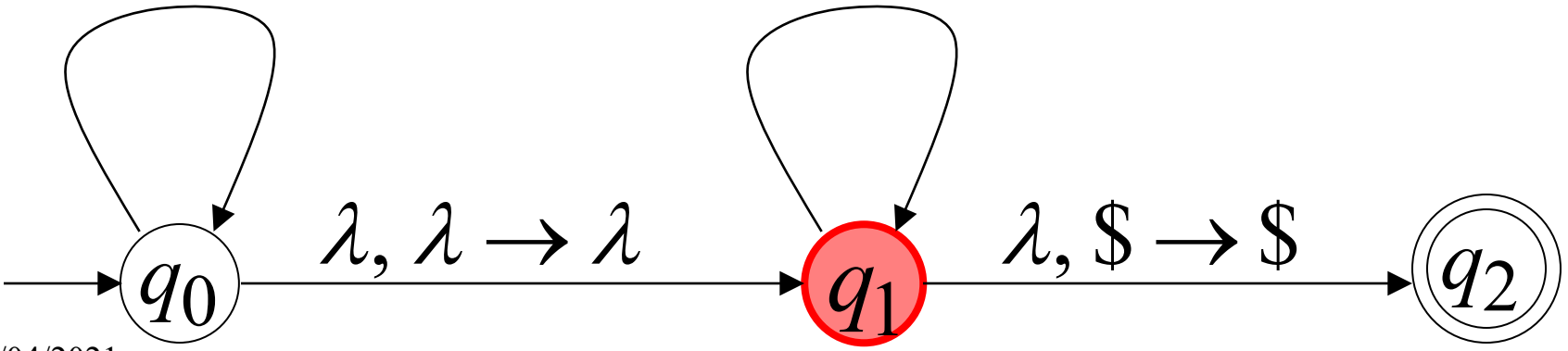
Stack

$a, \lambda \to a$

$b, \lambda \to b$

$a, a \to \lambda$

$b, b \to \lambda$

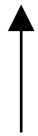


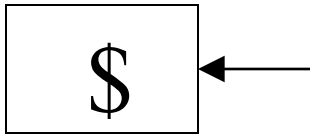
Un altra computazione sulla stessa stringa:

Input

Time 0

a	b	b	b





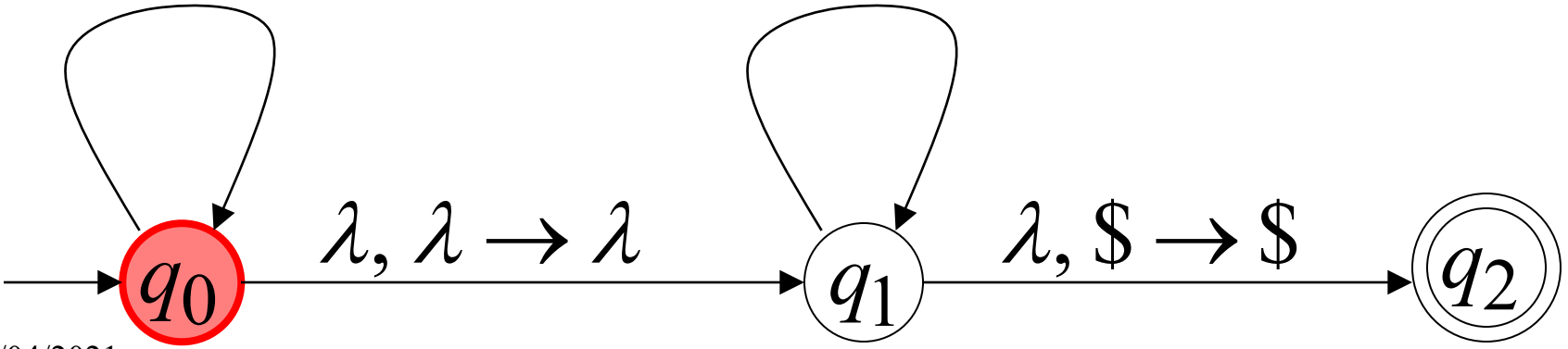
Stack

$a, \lambda \to a$

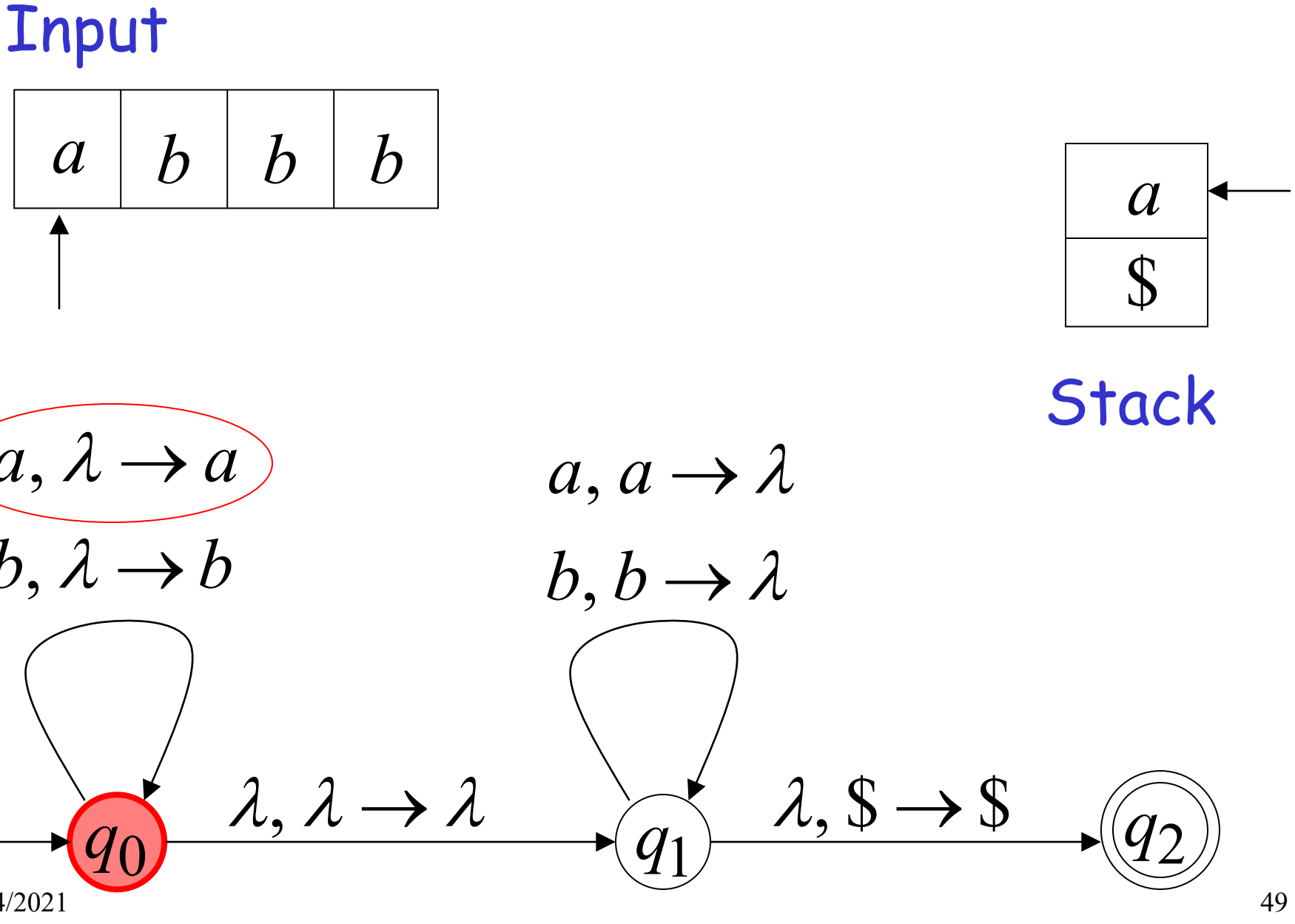
$b, \lambda \to b$

$a, a \to \lambda$

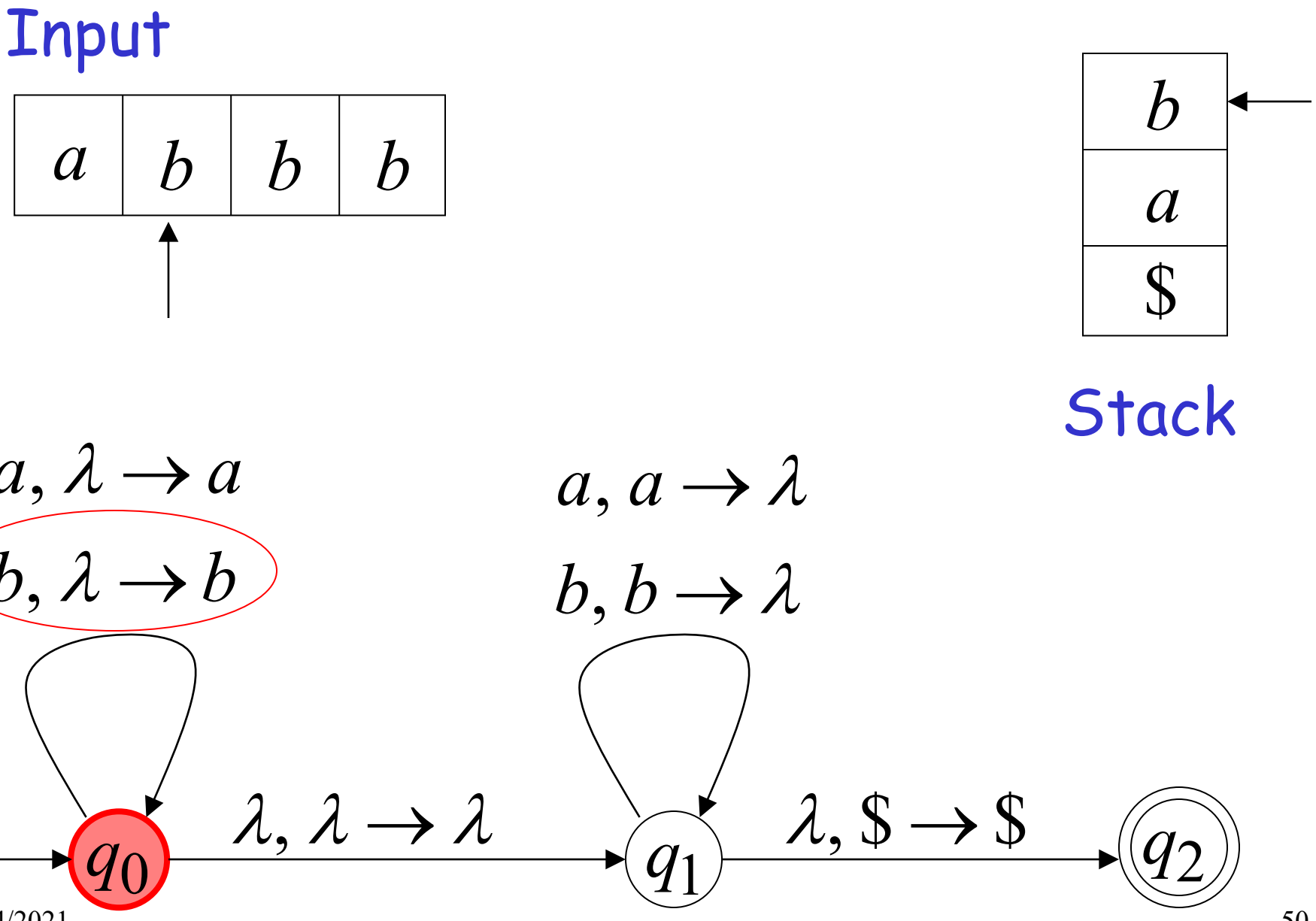
$b, b \to \lambda$



Time 1

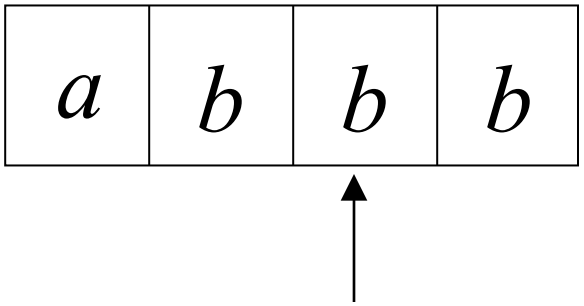


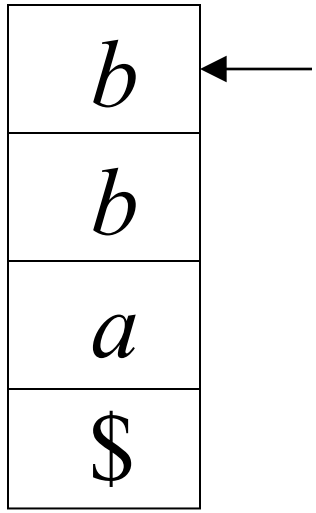
Time 2



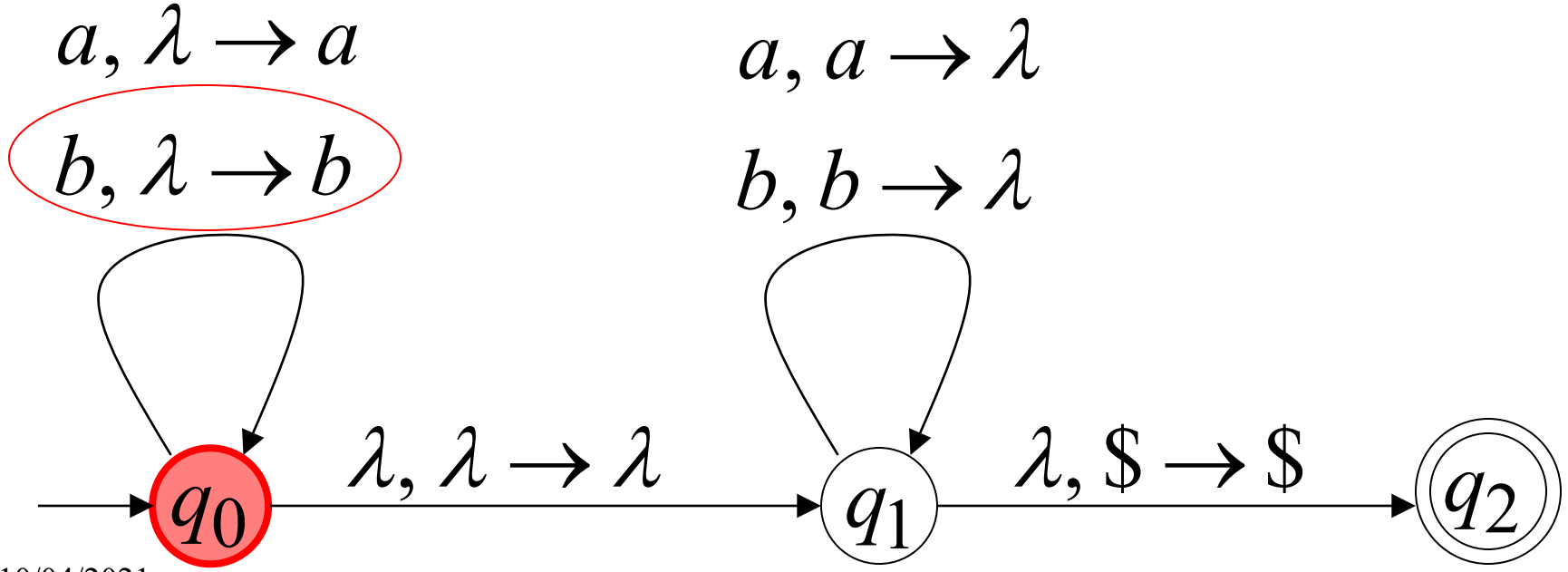
Time 3

Input





Stack



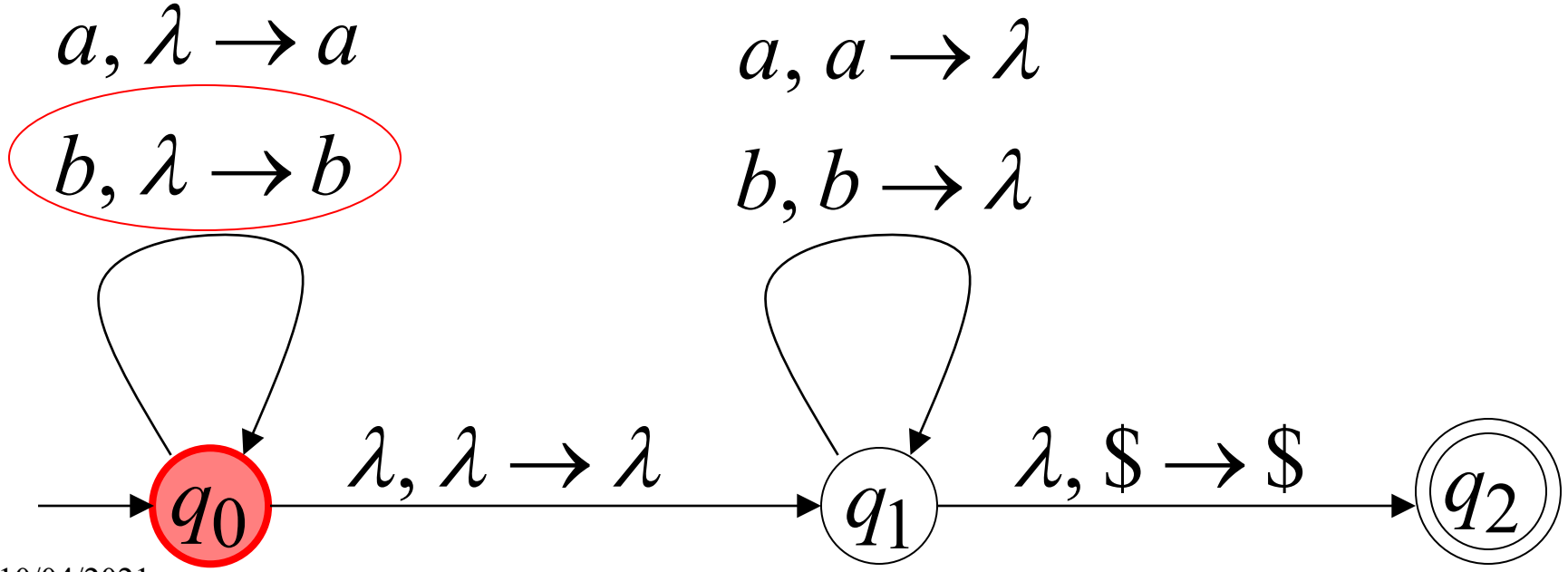
Time 4

Input

a	b	b	b

Stack

b
b
b
a
\$



Time 5

Input

a	b	b	b

No stato di accettazione

b
b
b
a
\$

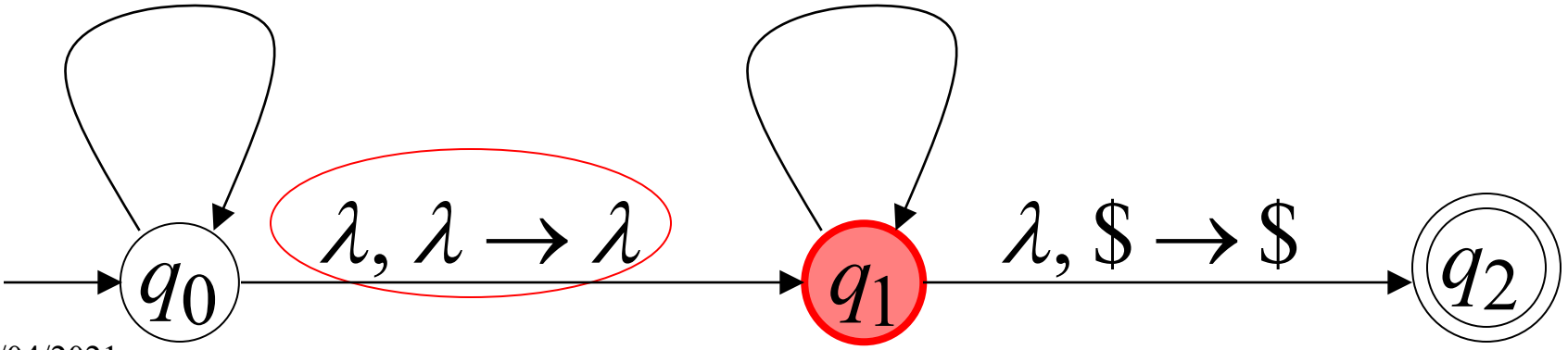
Stack

$a, \lambda \to a$

$b, \lambda \to b$

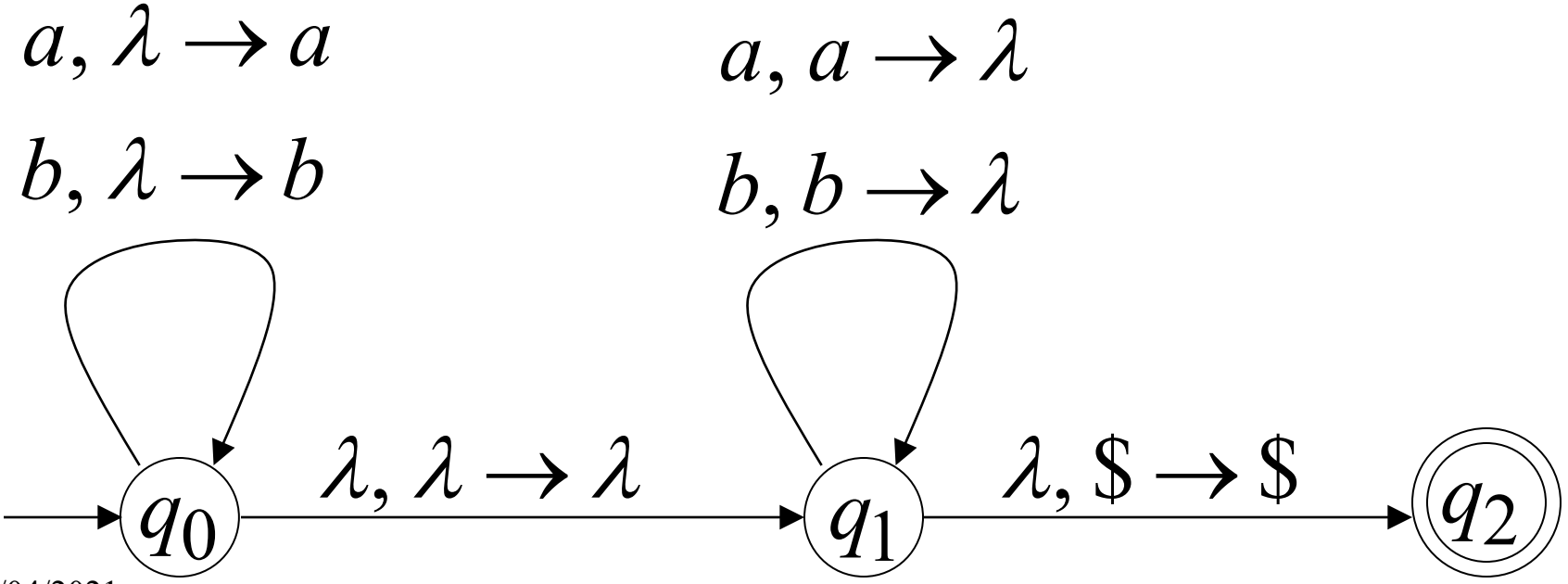
$a, a \to \lambda$

$b, b \to \lambda$



Non esiste nessuna computazione che accetta $abbb$

$$abbb \notin L(M)$$



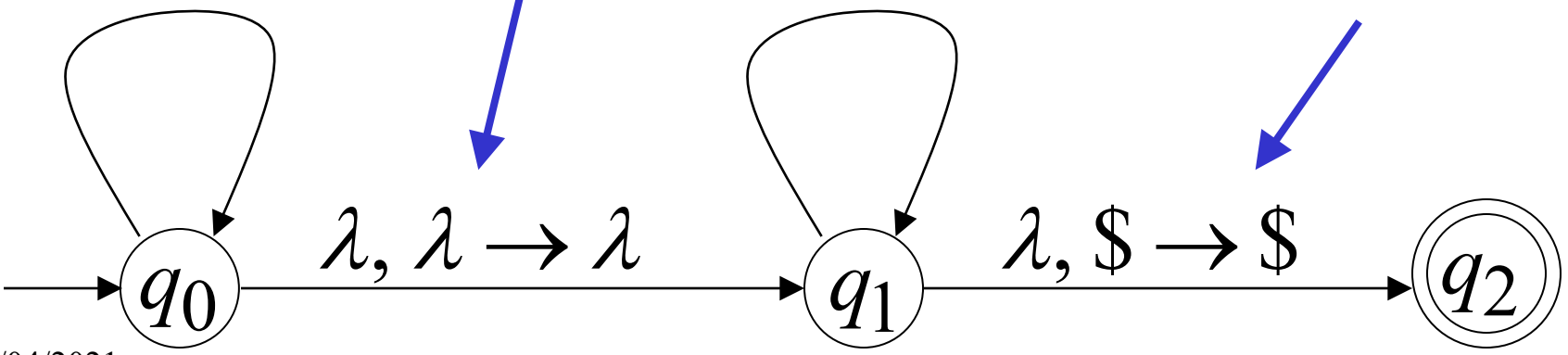
Basic Idea: $L(M) = \{vv^R : v \in \{a,b\}^*\}$

1. Push v Nello stack

2. Congettura metà Dell'input

3. verifica v^R in Input con v nello stack

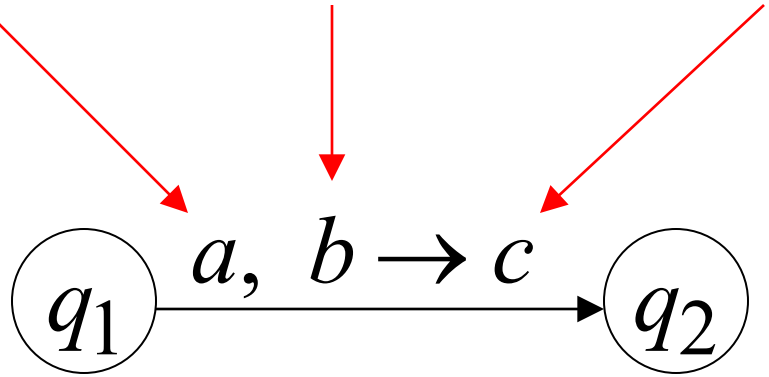
4. Verifica trovata



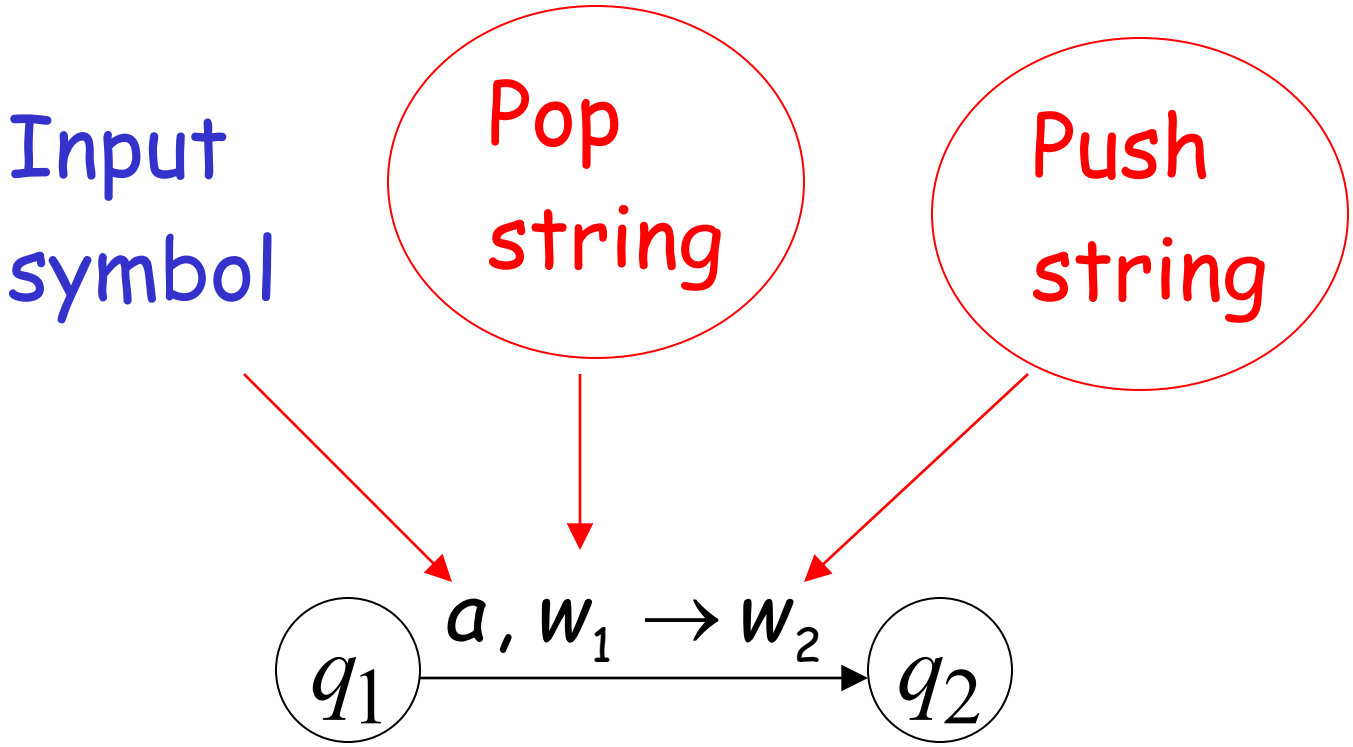
Input simbolo

Pop simbolo

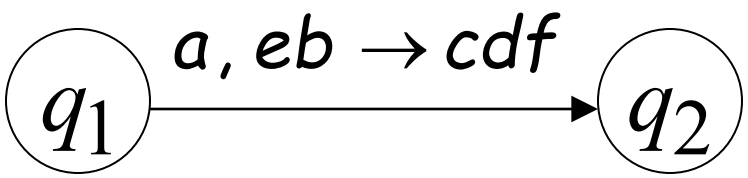
Push simbolo



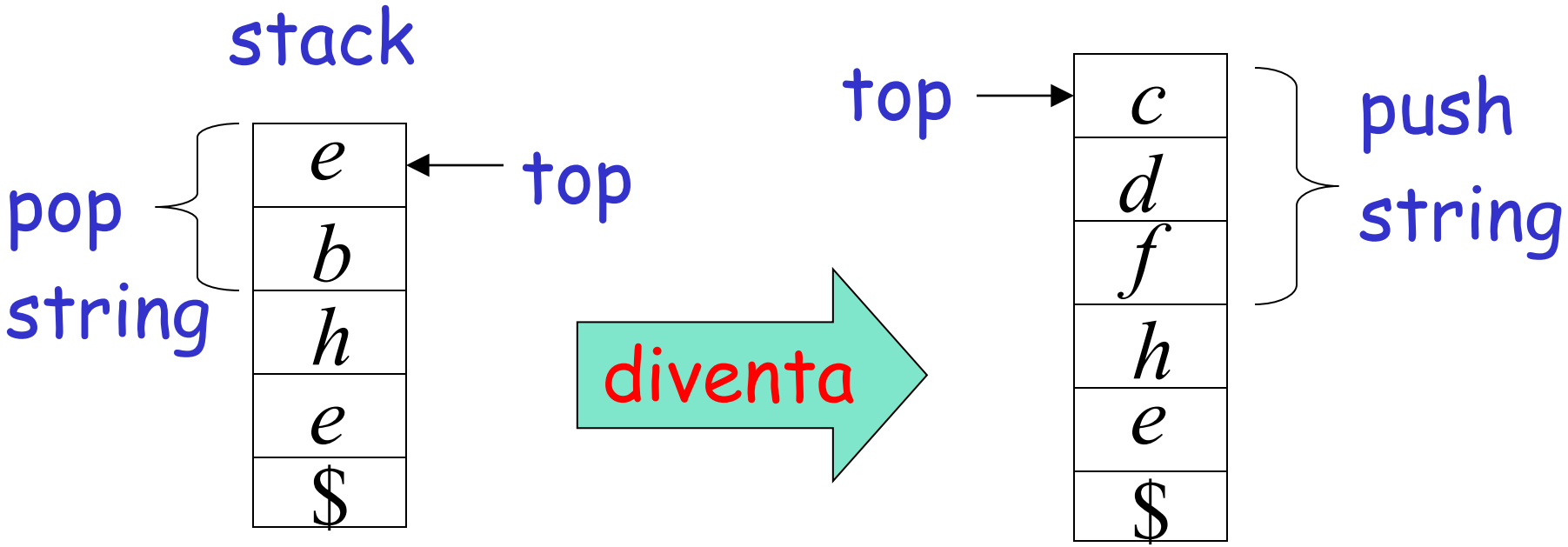
Pushing & Popping Strings

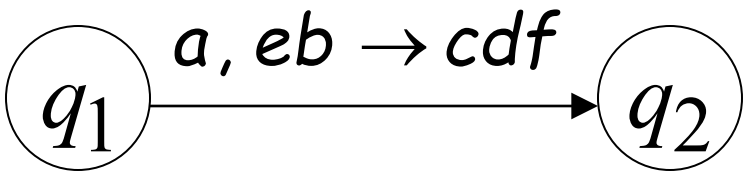


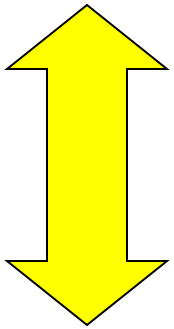
Esempio:


a, eb → cdf
q_1
q_2


input
…
a
…
…
a
…

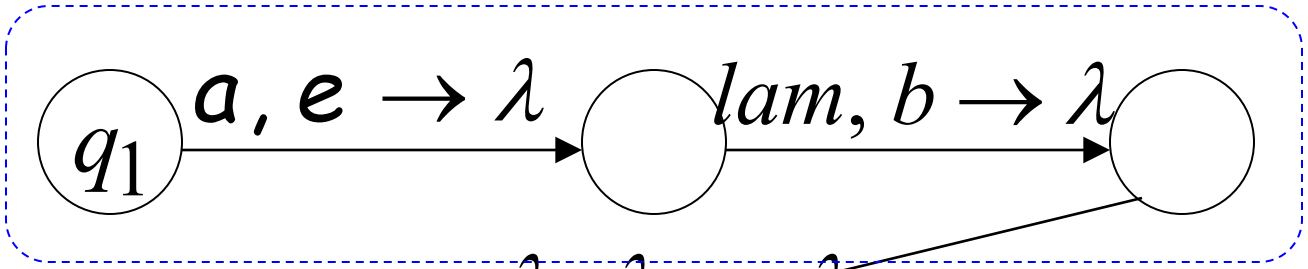

stack
pop
string
e
b
h
e
$
top
diventa
top
c
d
f
h
e
$
push
string



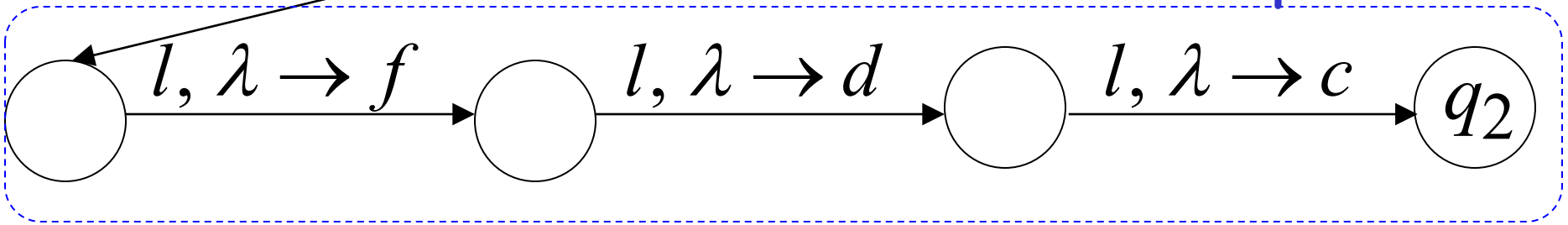


Transizioni equivalenti

pop



push



altro PDA esempio

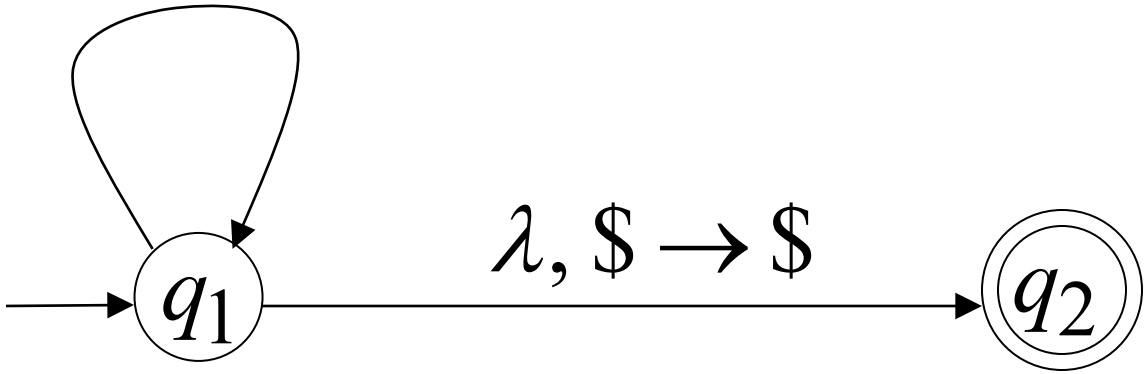
$$L(M) = \{w \in \{a,b\}^* : \quad n_a(w) = n_b(w)\}$$

PDA M

$a, \$ \to 0\$$ $\quad b, \$ \to 1\$$

$a, 0 \to 00$ $\quad b, 1 \to 11$

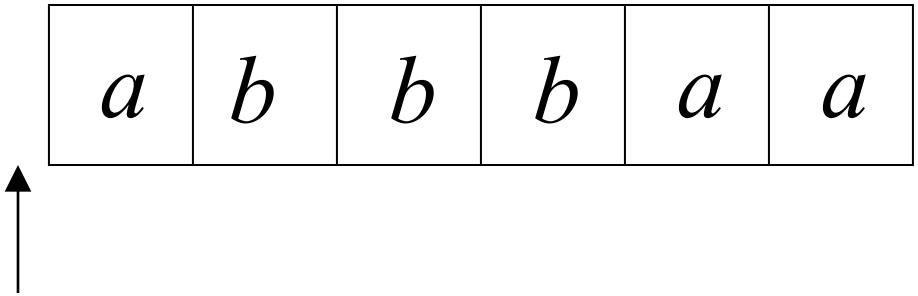
$a, 1 \to \lambda$ $\quad b, 0 \to \lambda$

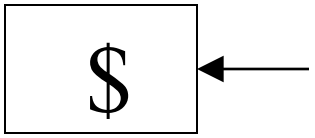


esecuzione:

Time 0

Input





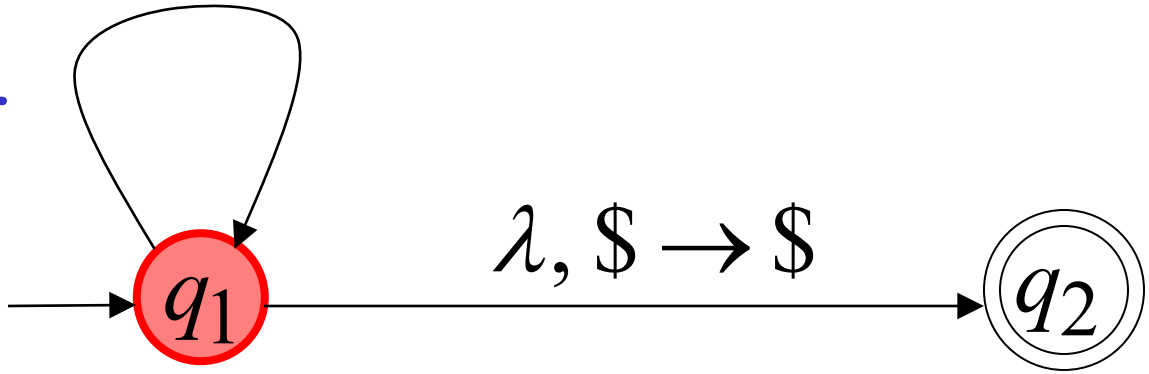
Stack

$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

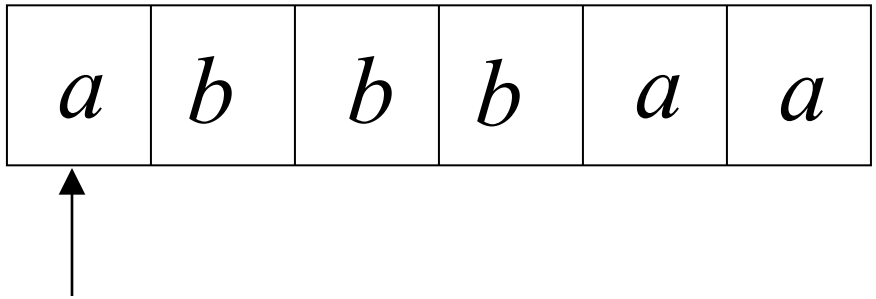
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$

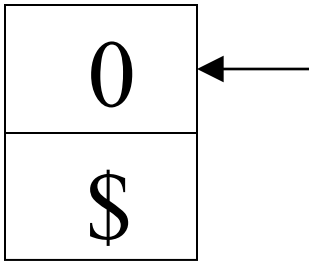
current state



Time 1

Input



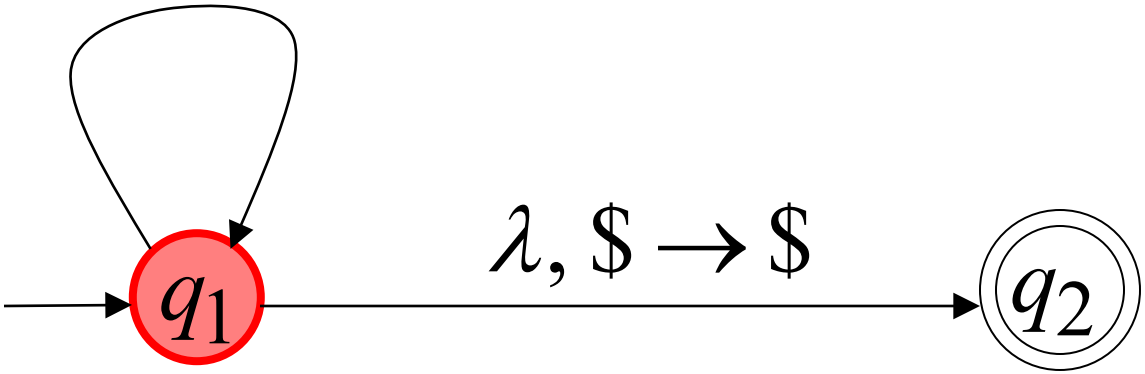


Stack

$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

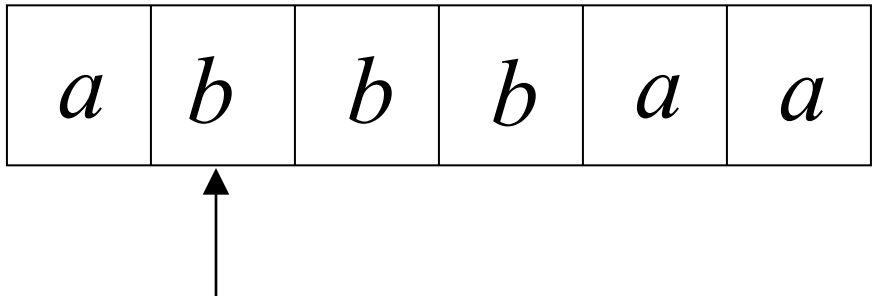
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

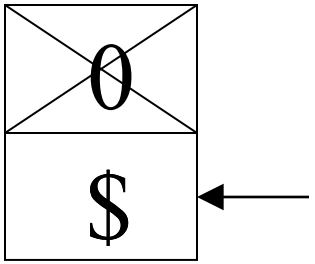
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 3

Input



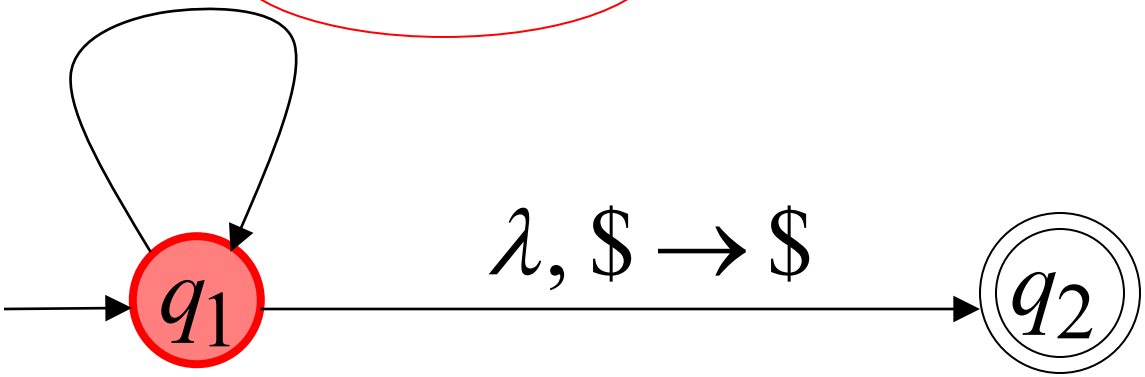


Stack

$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

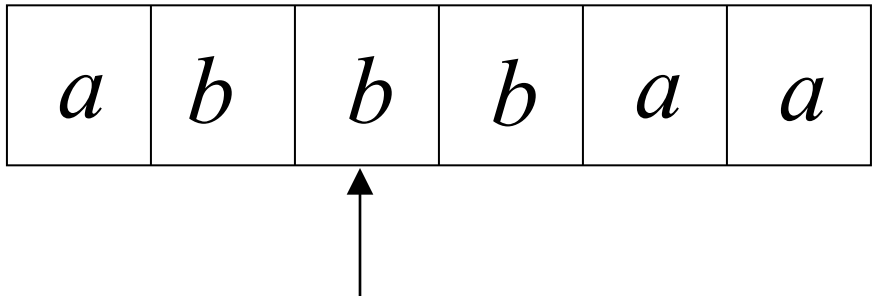
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

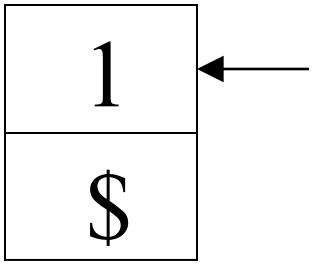
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 4

Input



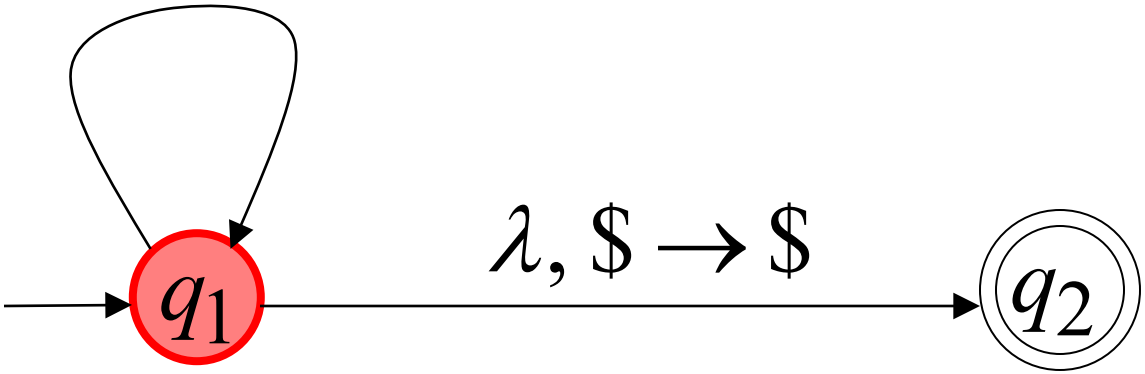


Stack

$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

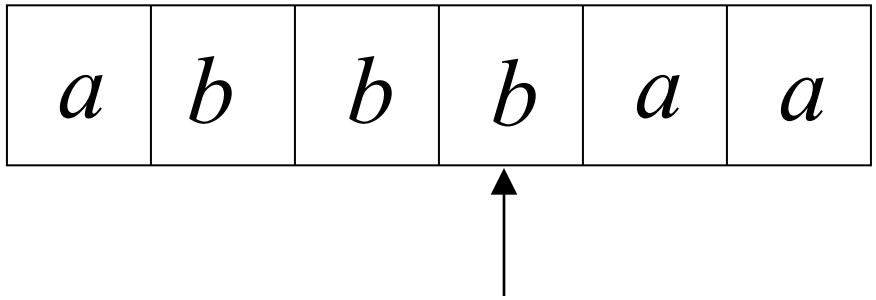
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

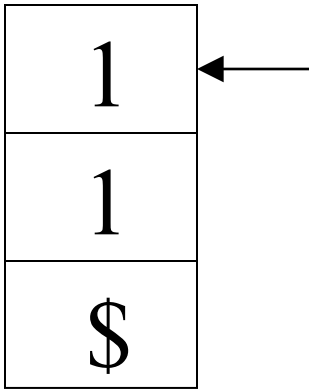
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 5

Input



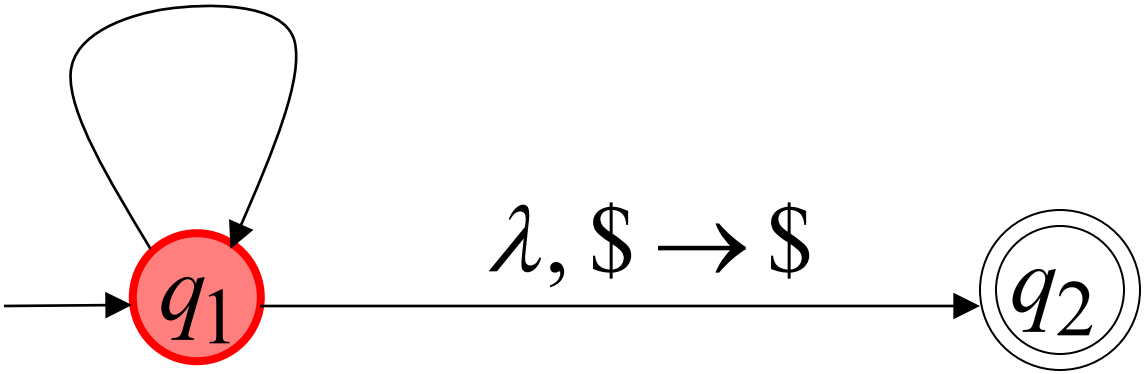


Stack

$$a, \$ \rightarrow 0\$ \qquad b, \$ \rightarrow 1\$$$

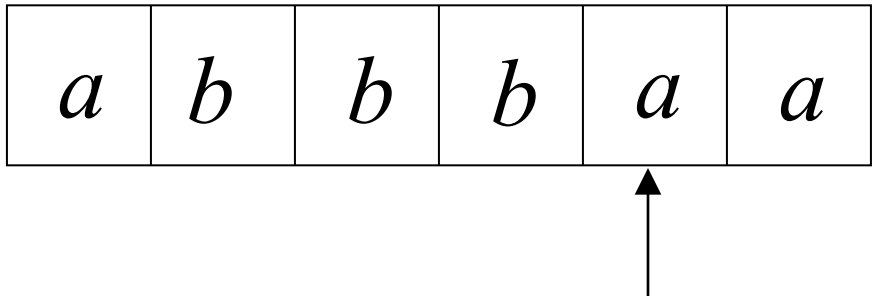
$$a, 0 \rightarrow 00 \qquad b, 1 \rightarrow 11$$

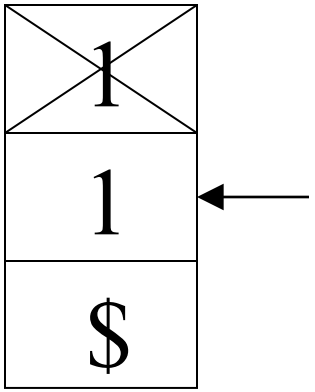
$$a, 1 \rightarrow \lambda \qquad b, 0 \rightarrow \lambda$$



Time 6

Input



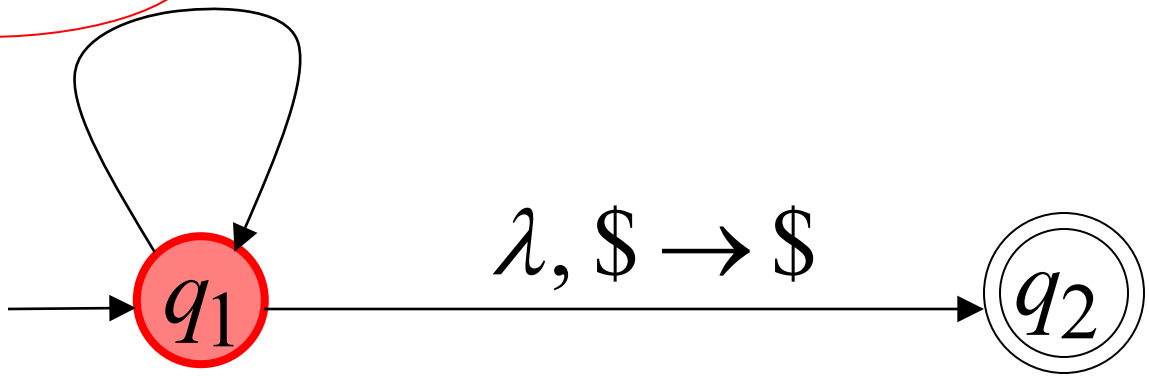


Stack

$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

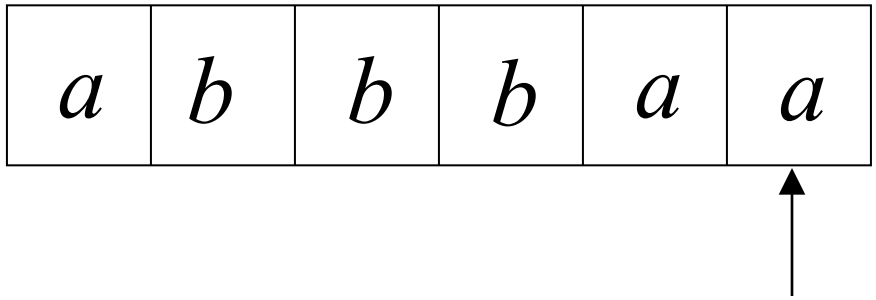
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

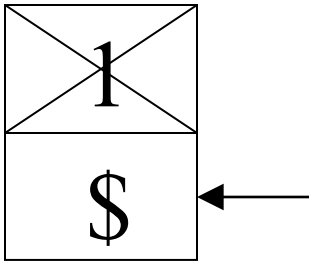
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 7

Input



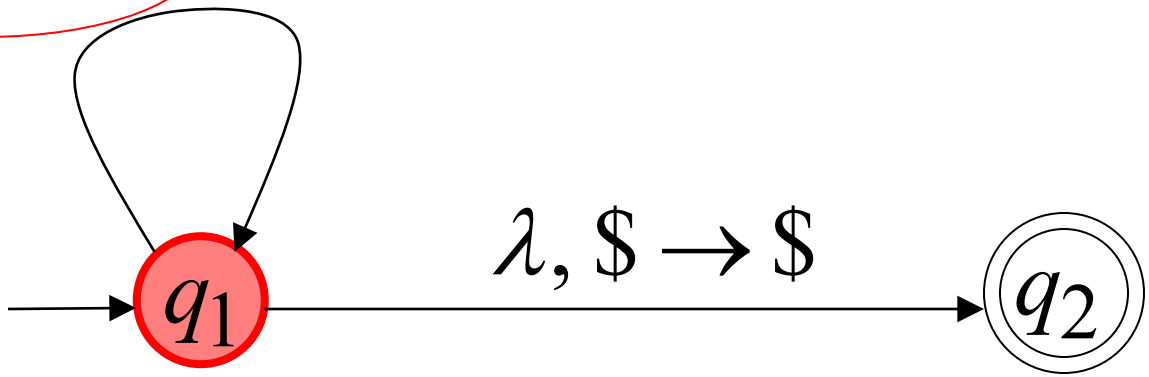


Stack

$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

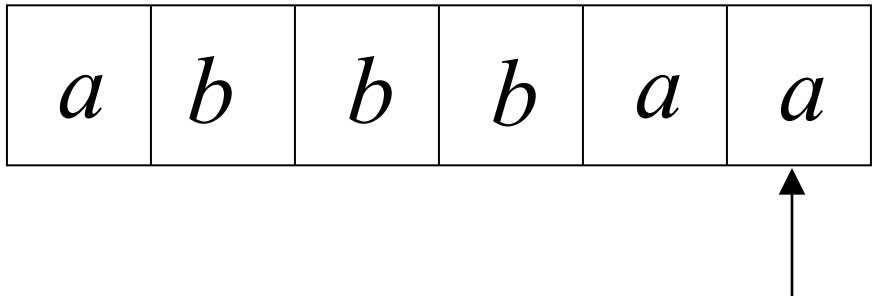
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

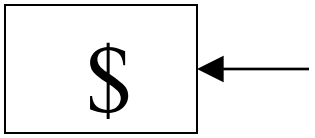
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 8

Input



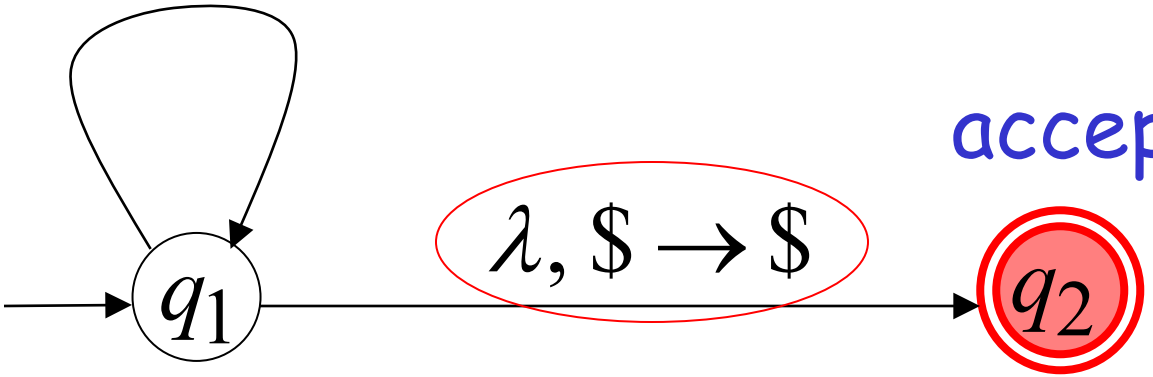


Stack

$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

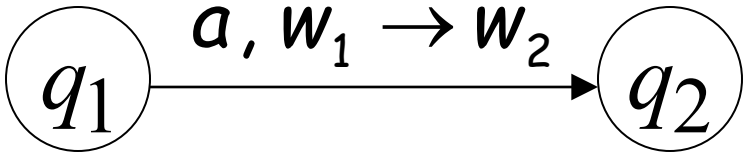
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



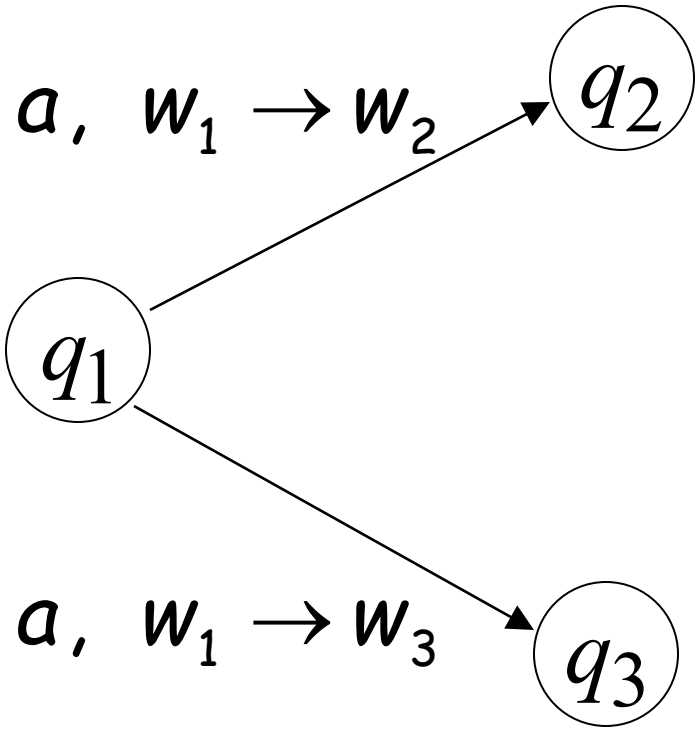
accept

Formalismo per I PDA



Funzione di transizione:

$$\delta(q_1, a, w_1) = \{(q_2, w_2)\}$$



Funzione di transizione :

$$\delta(q_1, a, w_1) = \{(q_2, w_2), (q_3, w_3)\}$$

Formal Definition

Pushdown Automaton (PDA)

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

States

Input alphabet

Stack alphabet

Transition function

Initial state

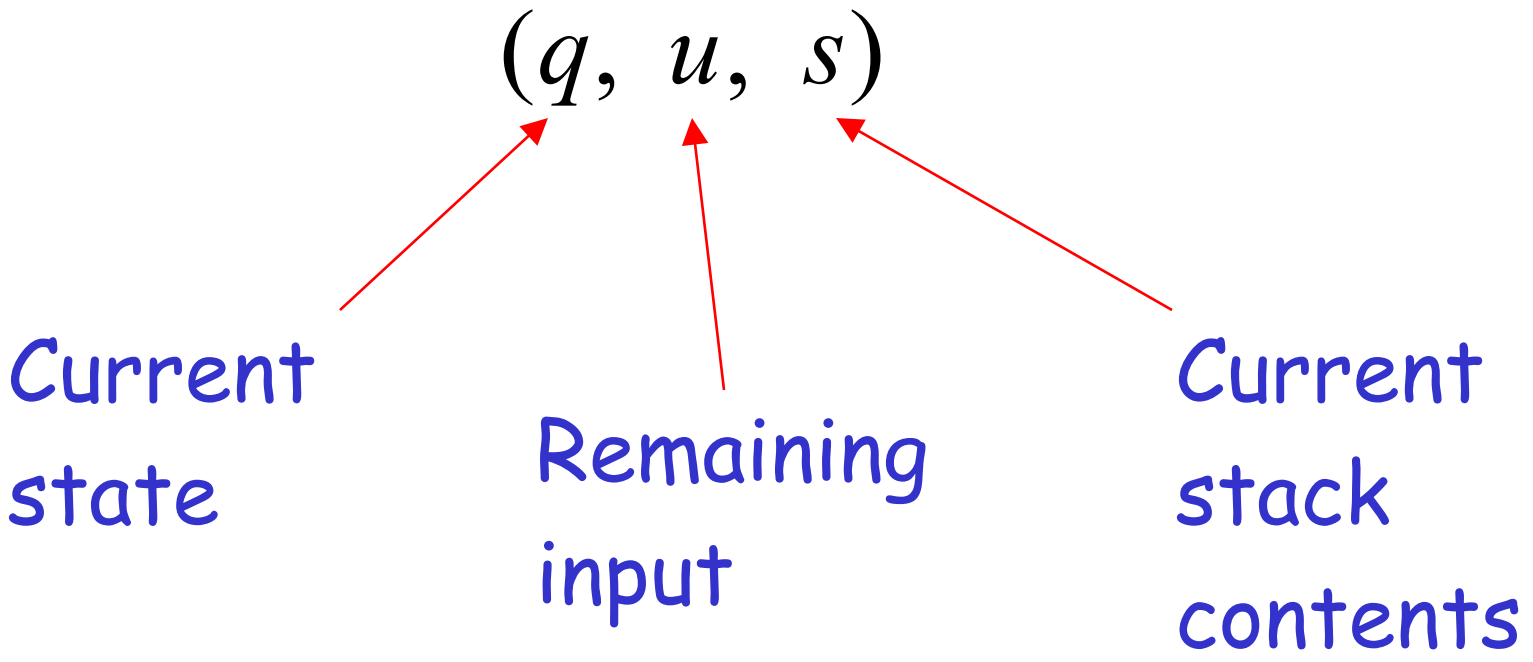
Stack start symbol

Accept states

Delta

: Stato x Inputx Stack -> P {(Stato, Stack)}

Instantaneous Description

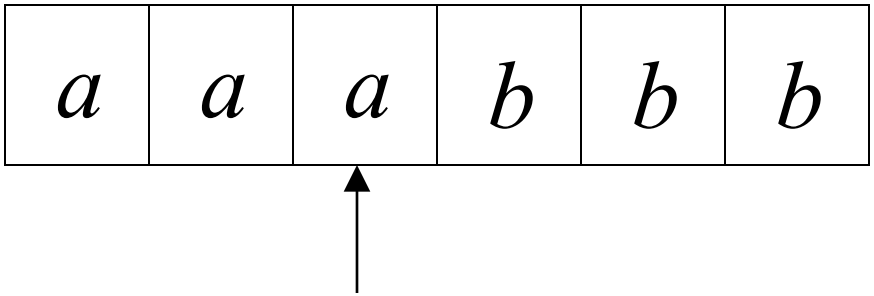


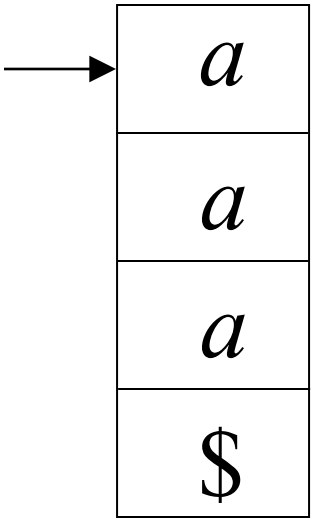
Example: Instantaneous Description

$$(q_1, bbb, aaa\$)$$

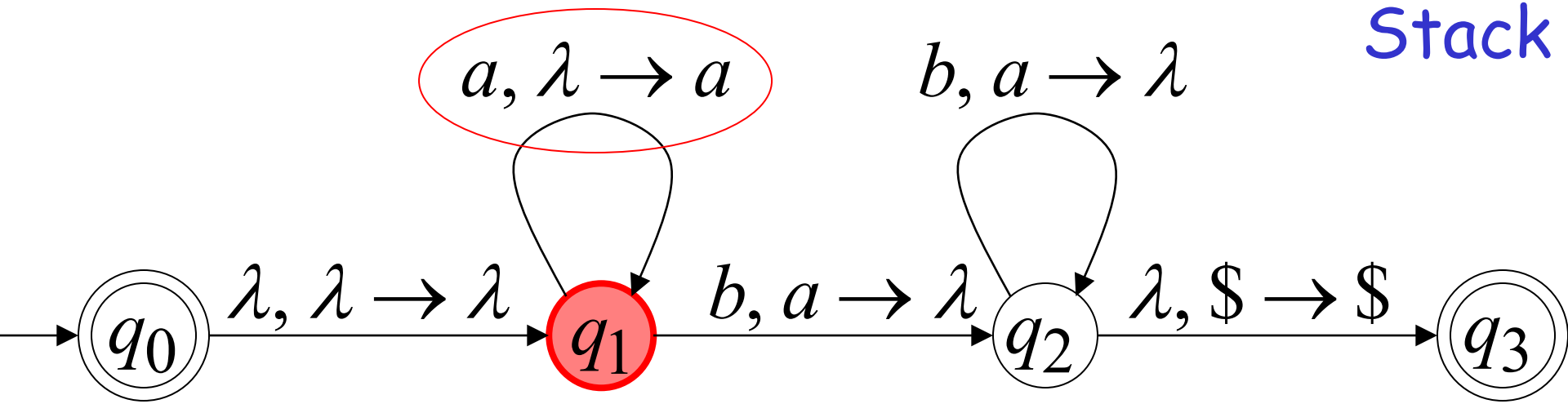
Time 4:

Input





Stack

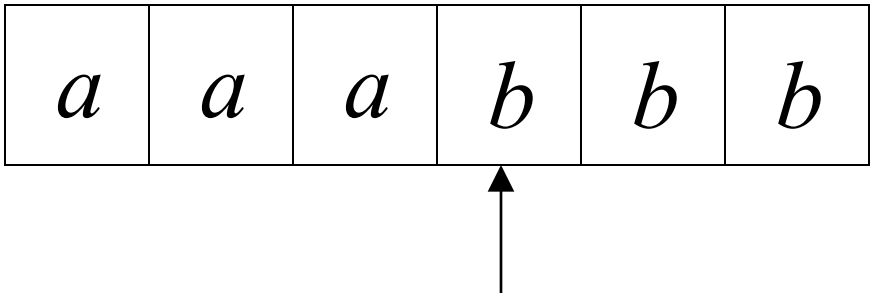


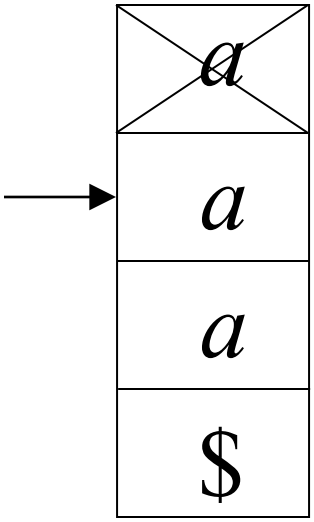
Example: Instantaneous Description

$$(q_2, bb, aa\$)$$

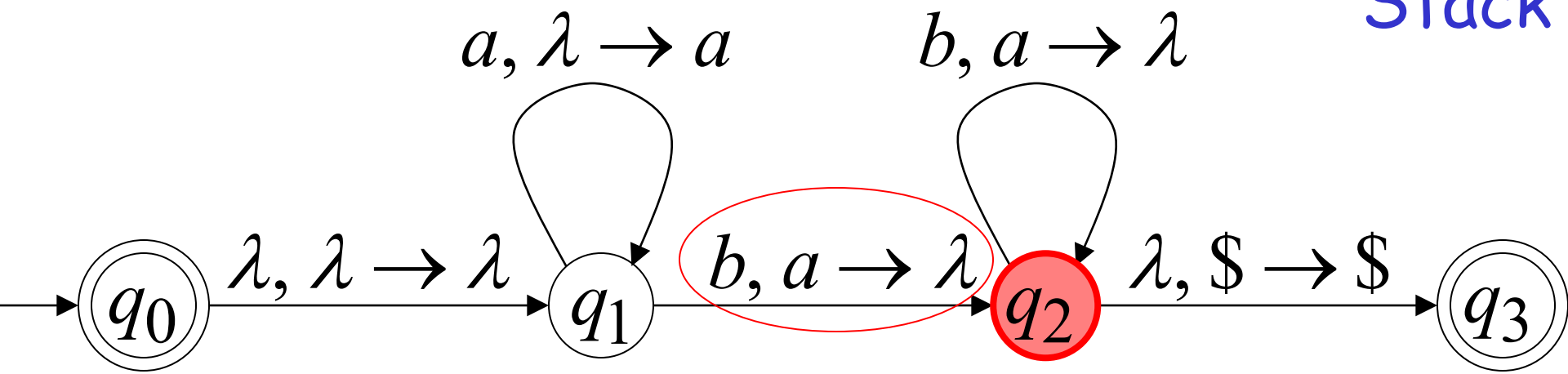
Time 5:

Input





Stack



scriviamo:

$$(q_1, bbb, aaa\$) \succ (q_2, bb, aa\$)$$

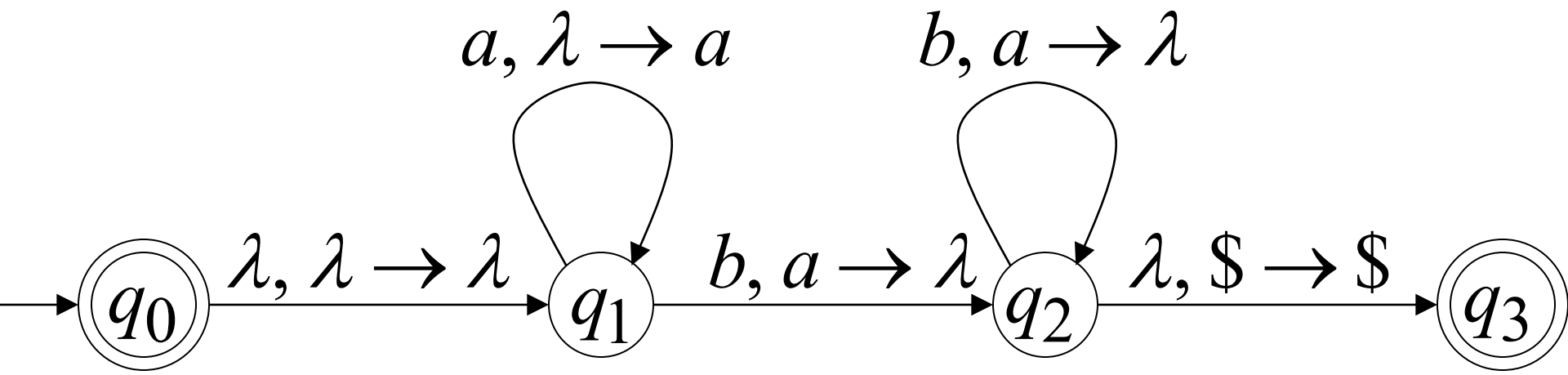
Time 4 Time 5

Una computazione:

$$(q_0, aaabbb, \$) \succ (q_1, aaabbb, \$) \succ$$

$$(q_1, aabbb, a\$) \succ (q_1, abbb, aa\$) \succ (q_1, bbb, aaa\$) \succ$$

$$(q_2, bb, aa\$) \succ (q_2, b, a\$) \succ (q_2, \lambda, \$) \succ (q_3, \lambda, \$)$$



$$(q_0, aaabbb, \$) \succ (q_1, aaabbb, \$) \succ$$
$$(q_1, aabbb, a\$) \succ (q_1, abbb, aa\$) \succ (q_1, bbb, aaa\$) \succ$$
$$(q_2, bb, aa\$) \succ (q_2, b, a\$) \succ (q_2, \lambda, \$) \succ (q_3, \lambda, \$)$$

Per convenienza scriviamo:

$$(q_0, aaabbb, \$) \overset{*}{\succ} (q_3, \lambda, \$)$$

Language of PDA

Linguaggio $L(M)$ accettato da PDA M :

$$L(M) = \{w: \ (q_0, w, z) \overset{*}{\succ} (q_f, \lambda, s)\}$$

Initial state

Accept state

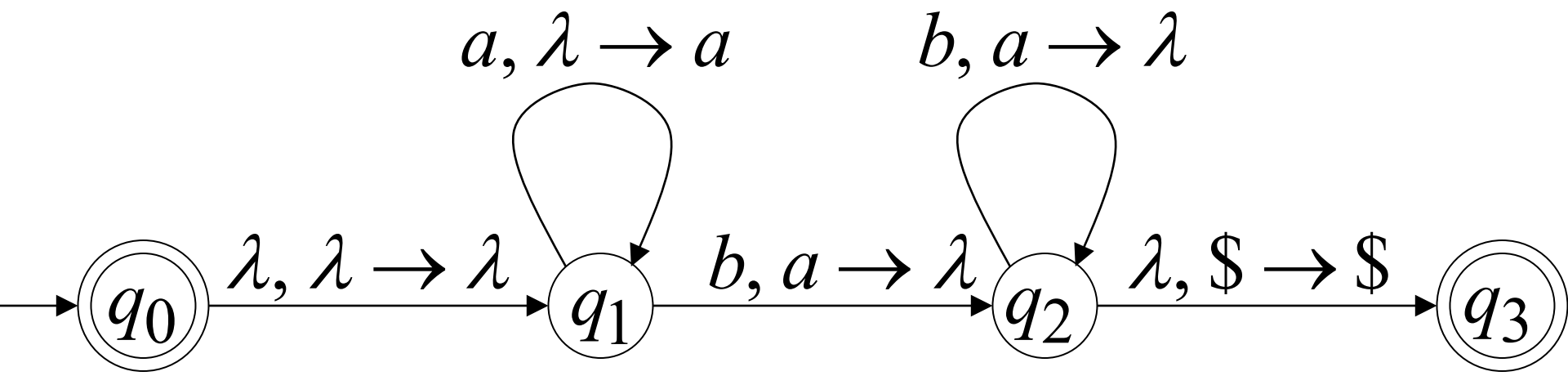
Stack può essere anche non vuoto, quindi s qualsiasi.

Esempio:

$$(q_0, aaabbb, \$) \overset{*}{\succ} (q_3, \lambda, \$)$$

$$aaabbb \in L(M)$$

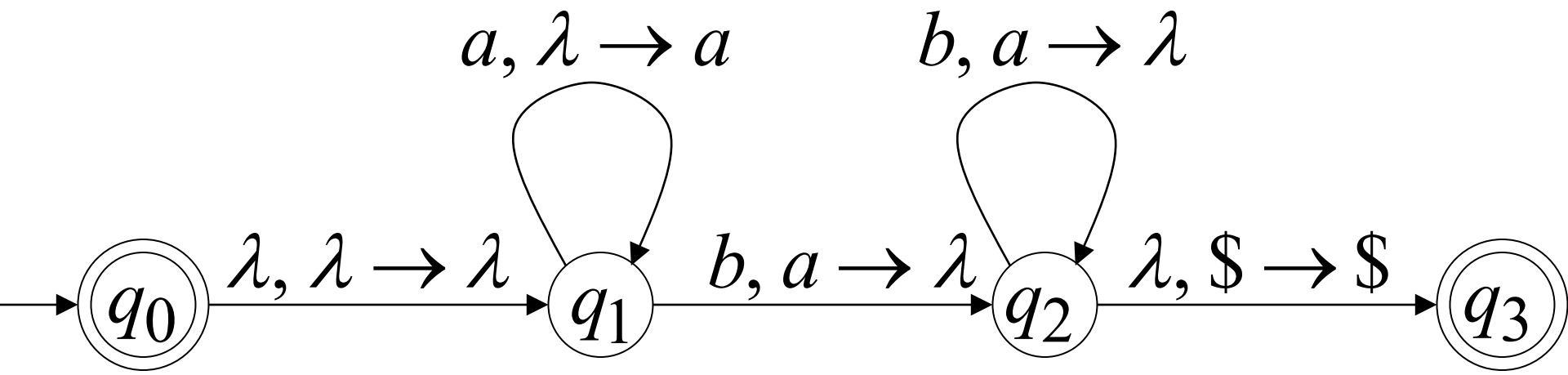
PDA M:



$$(q_0, a^n b^n, \$) \overset{*}{\succ} (q_3, \lambda, \$)$$

$$a^n b^n \in L(M)$$

PDA M :



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

PDA M :