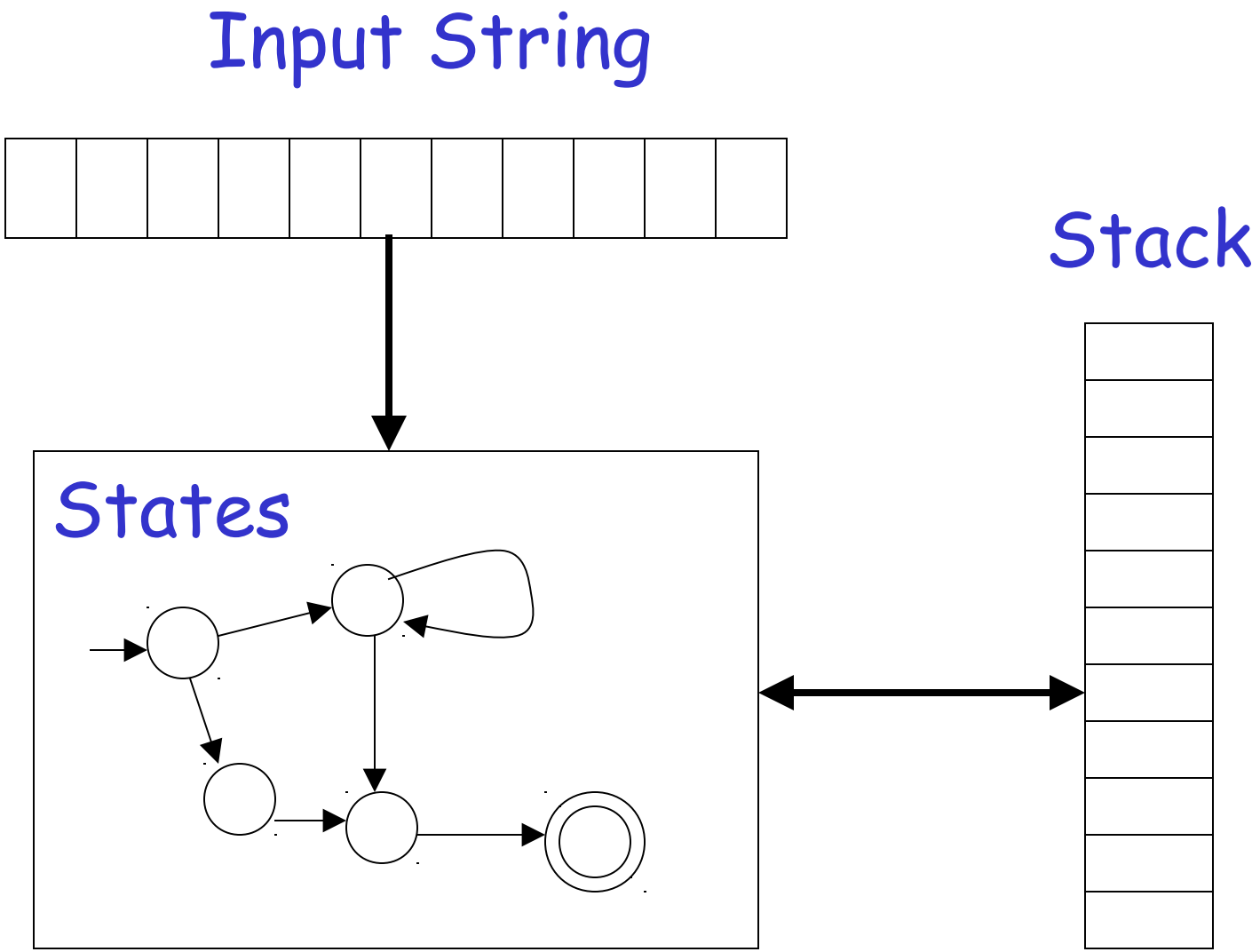
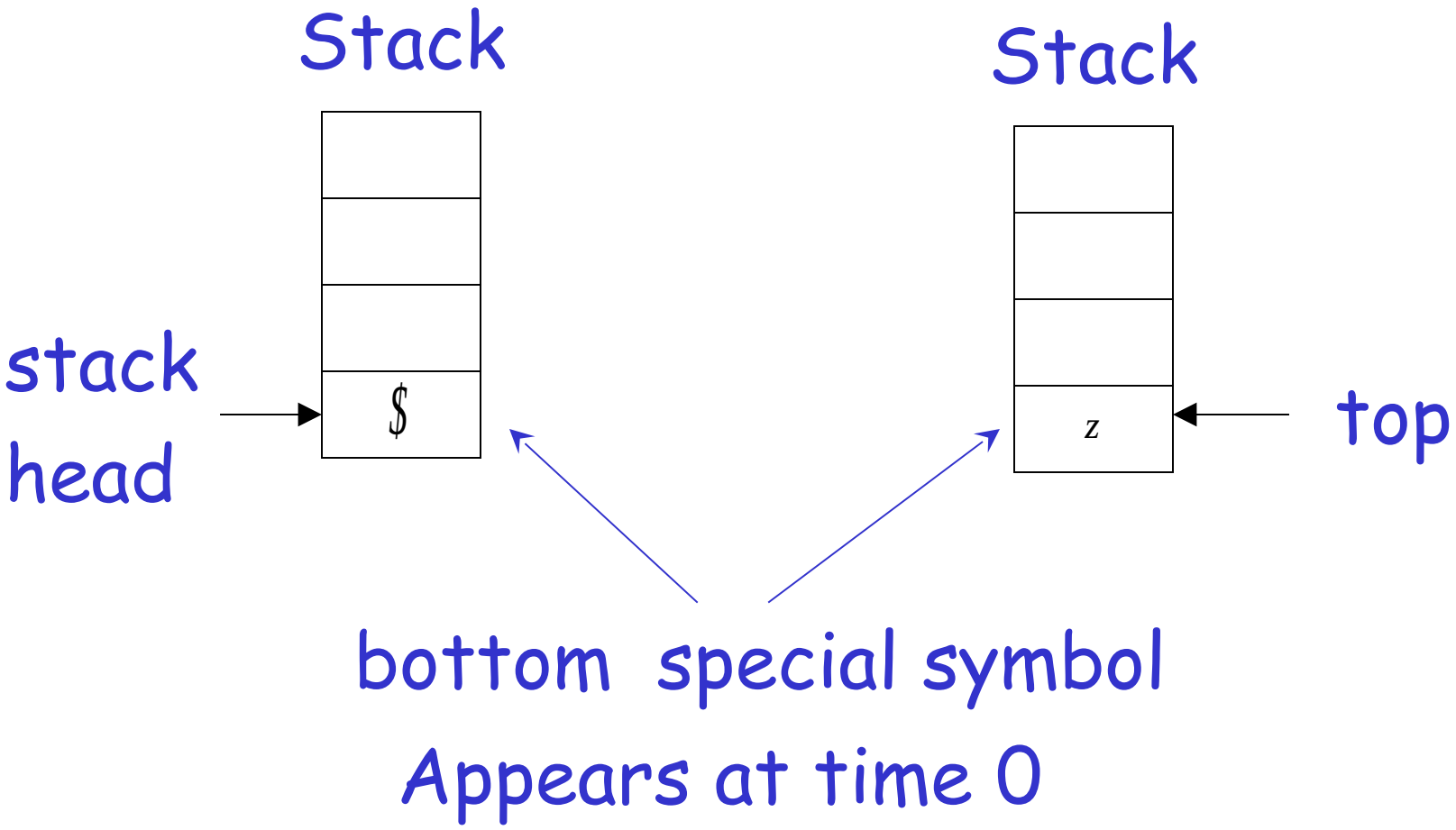
TIEI 2018
Formal Languages and Automata

Pushdown Automata Examples

Pushdown Automaton -- PDA



Initial Stack Symbol

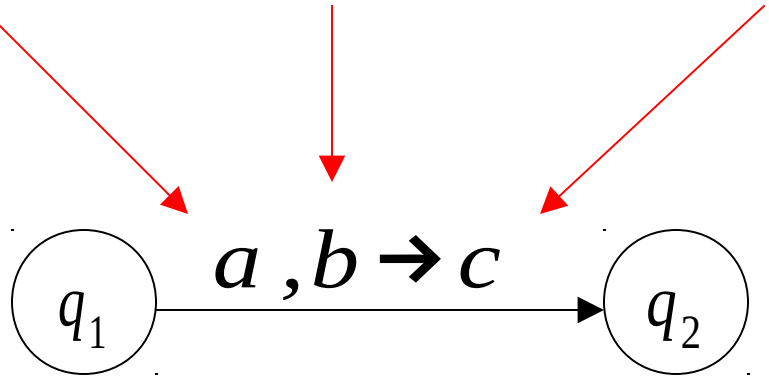


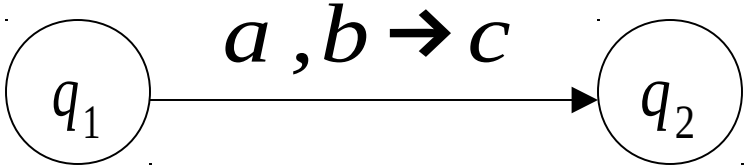
The States

Input symbol

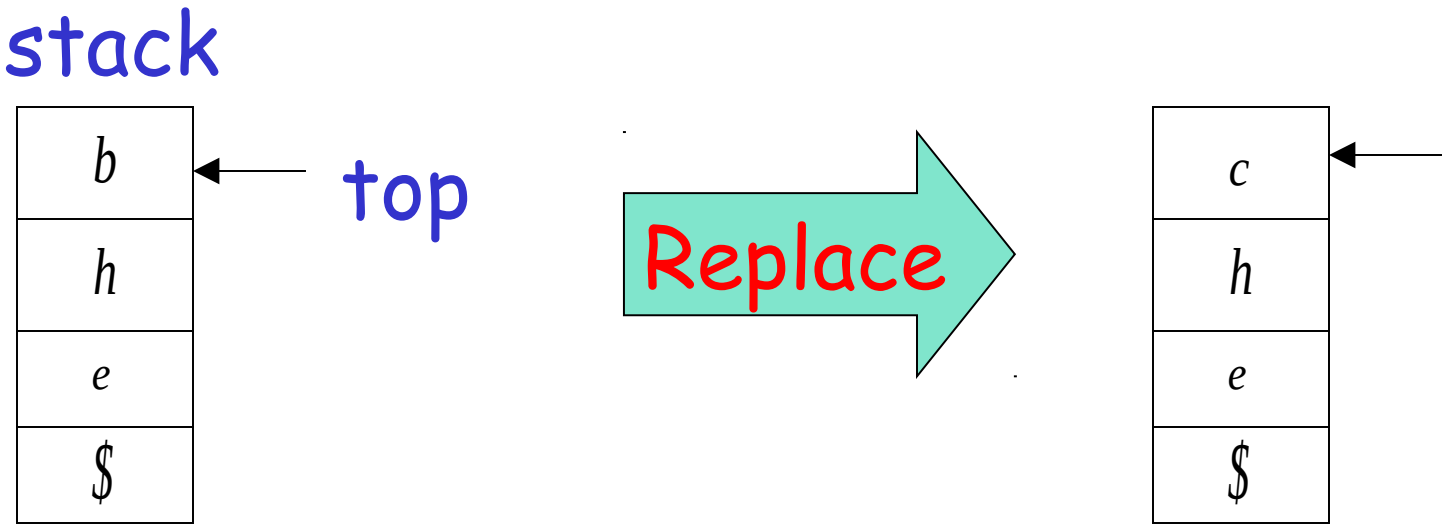
Pop symbol

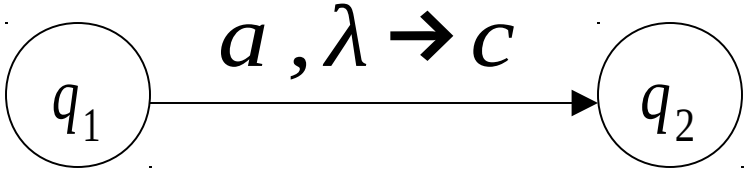
Push symbol



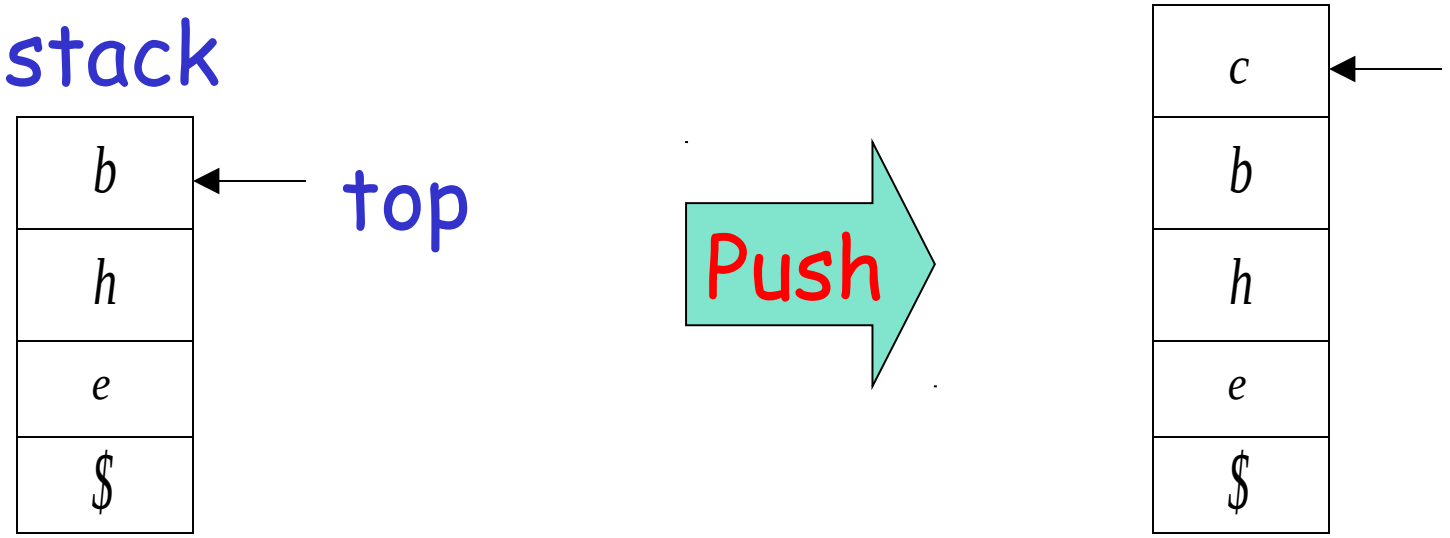

$a, b \rightarrow c$
q_1
q_2

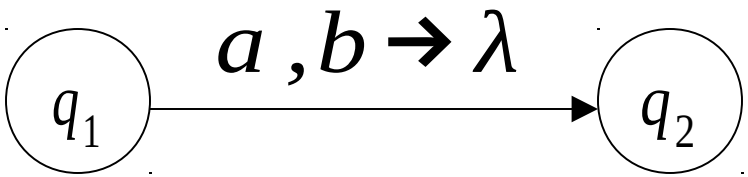

input
- - -
a
- - -
- - -
a
- - -


stack
b
h
e
$
top
Replace
c
h
e
$

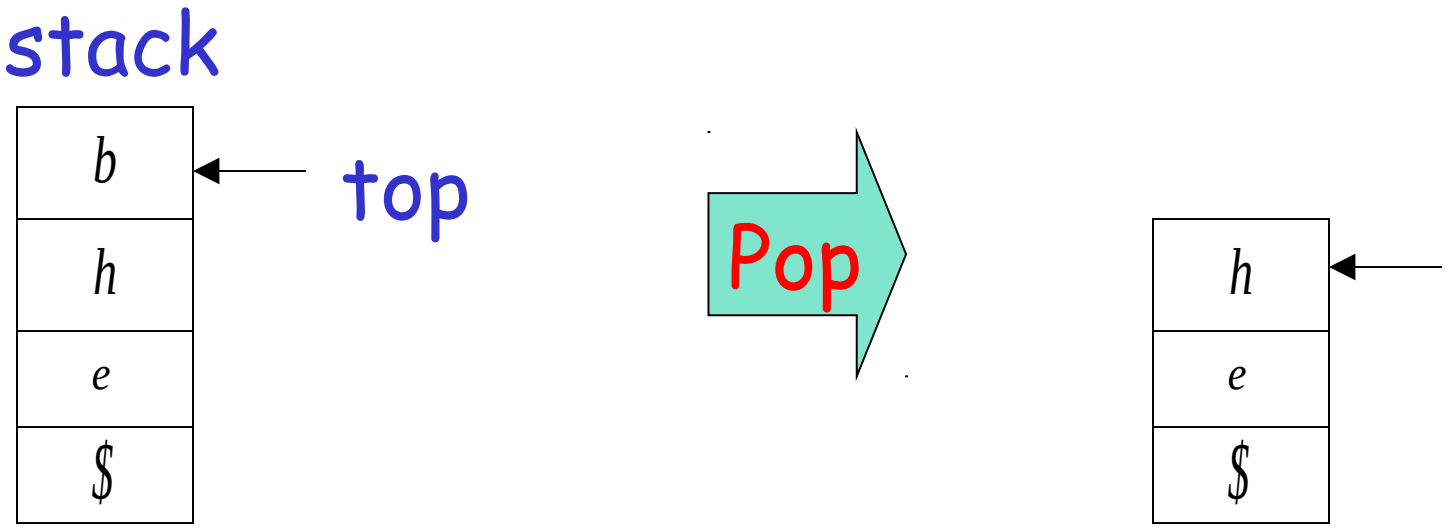

$a, \lambda \rightarrow c$
q_1
q_2

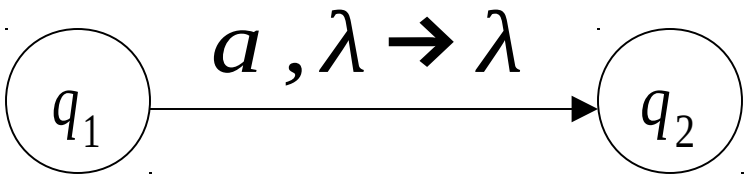

input
- - -
a
- - -
- - -
a
- - -


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

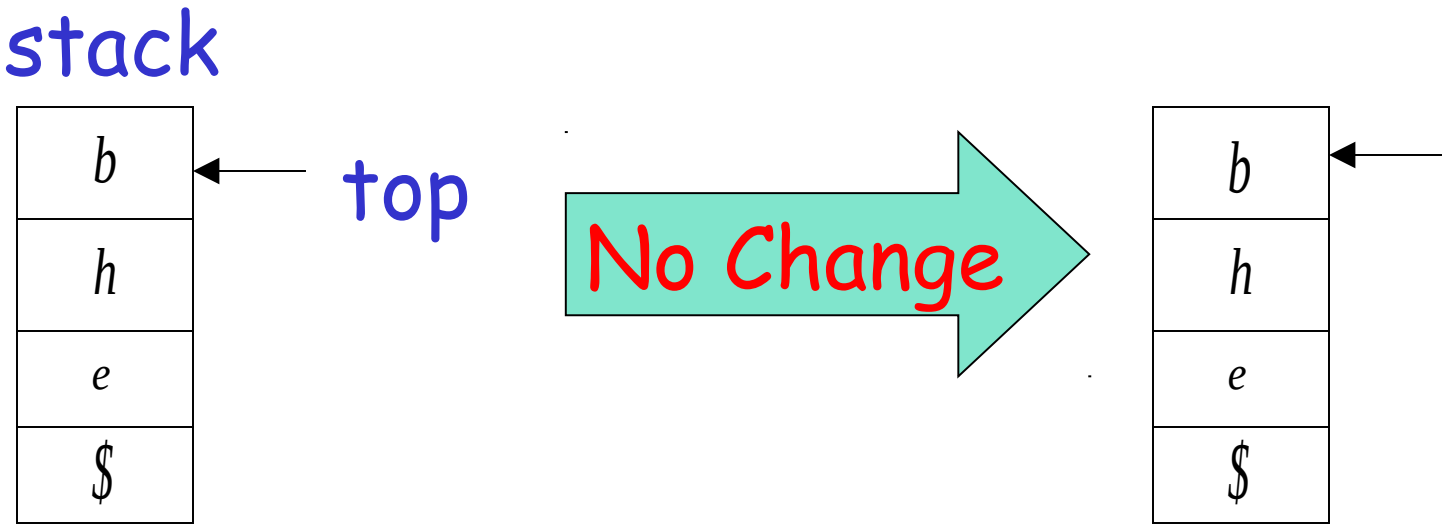

a ,b → λ
q1
q2


input
a
a


stack
b
top
h
e
$
Pop
h
e
$


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

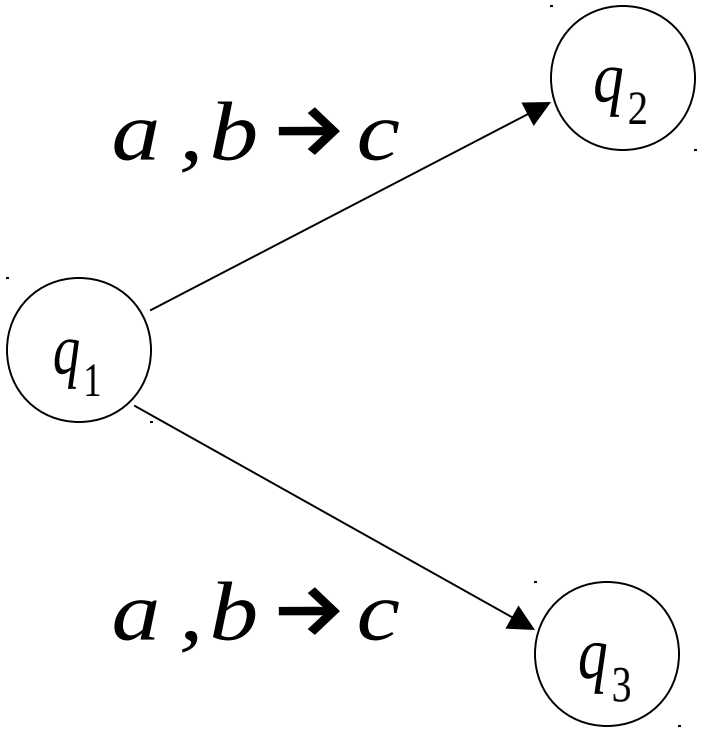

input
- - -
a
- - -
- - -
a
- - -

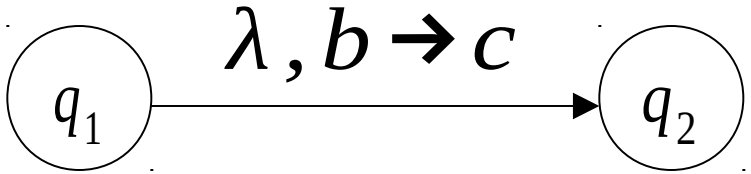

stack
b
h
e
$
top
No Change
b
h
e
$

Non-Determinism

PDAs are non-deterministic

Allowed non-deterministic transitions



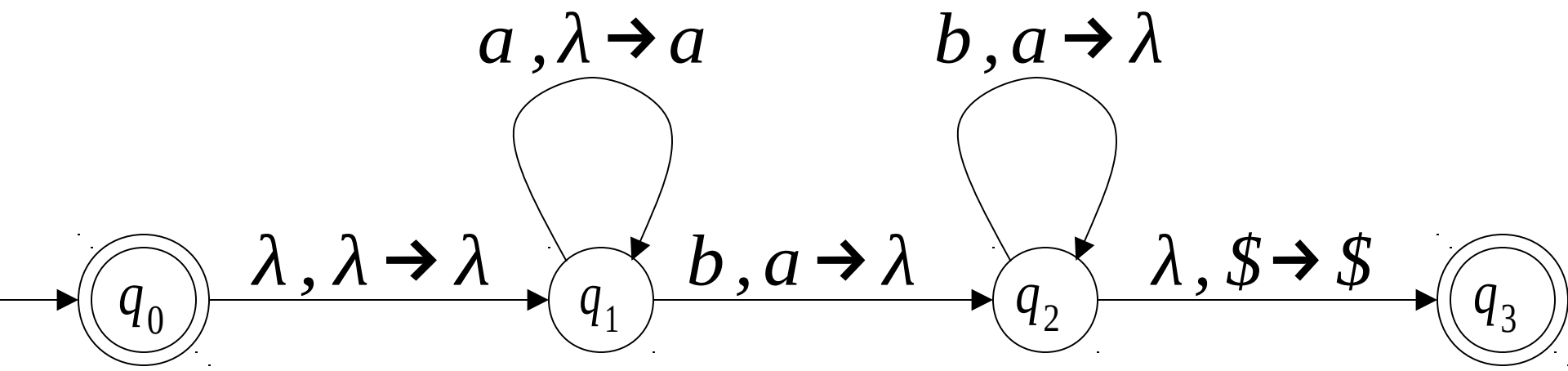


$\lambda -$ transition

Example PDA

PDA M :

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



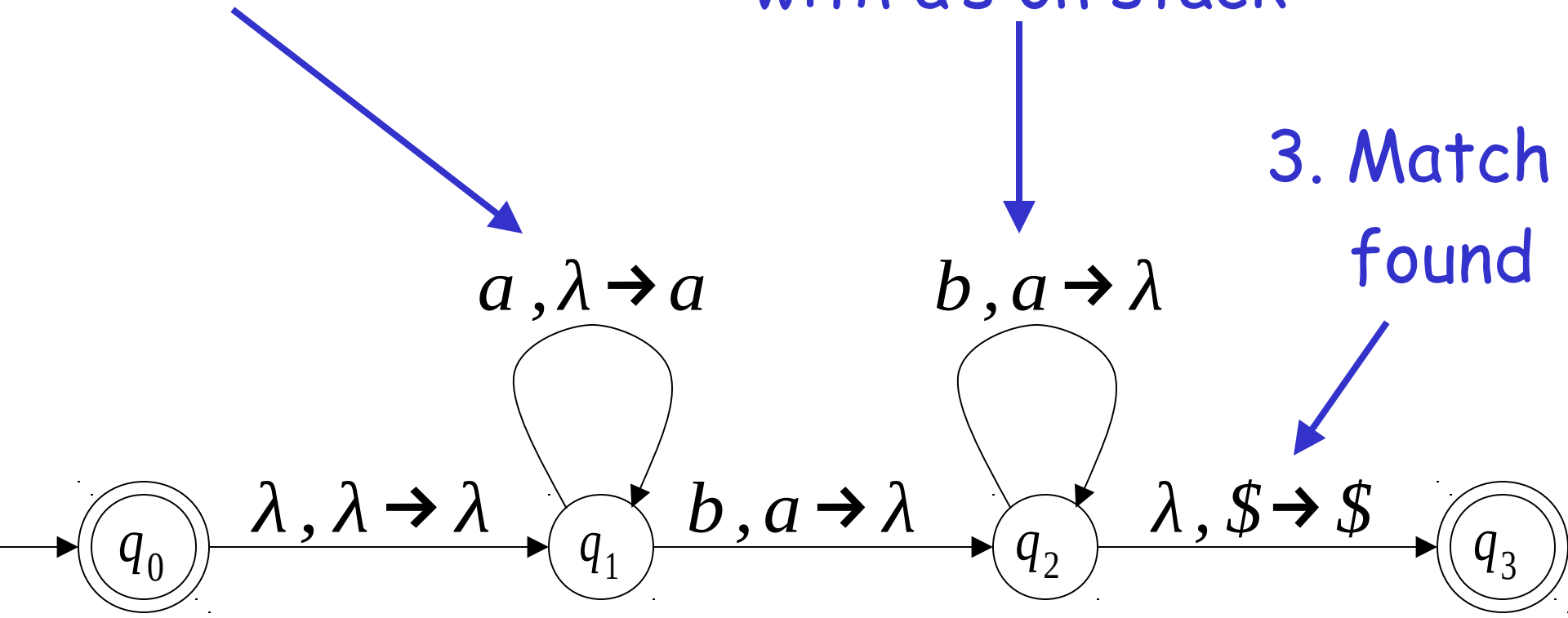
$$L(M)=\{a^n b^n : n \geq 0\}$$

Basic Idea:

1. Push the a's on the stack

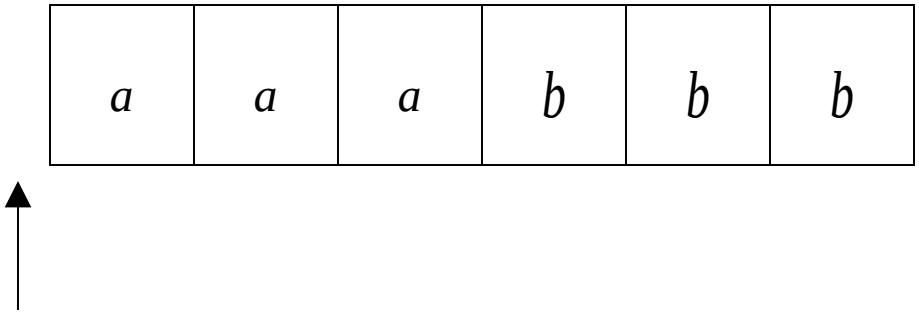
2. Match the b's on input with a's on stack

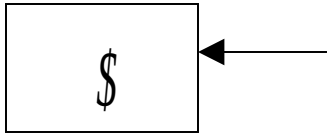
3. Match found



Execution Example: Time 0

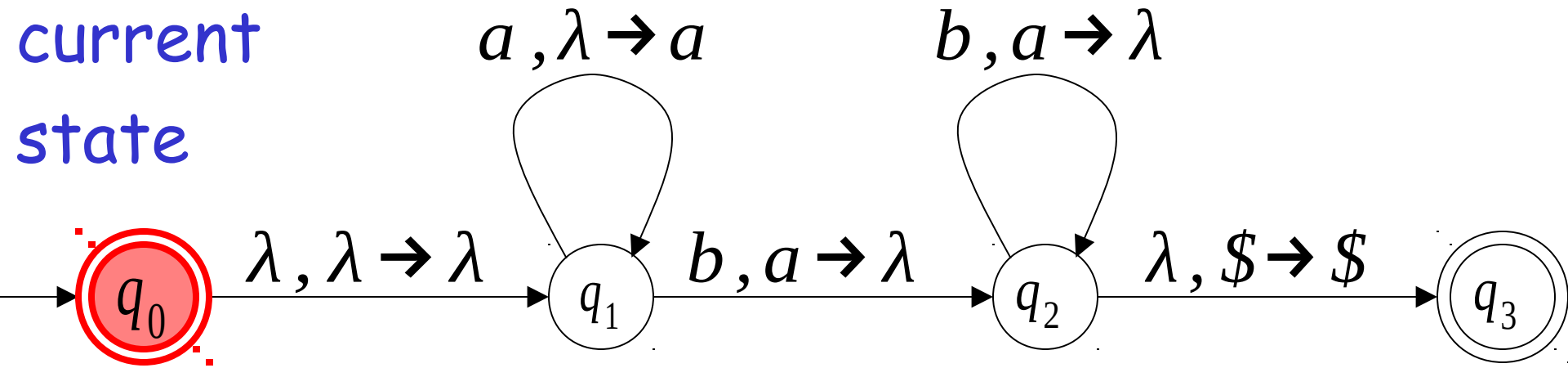
Input





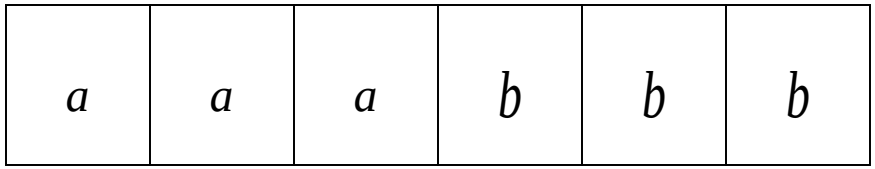
Stack

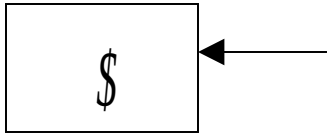
current state



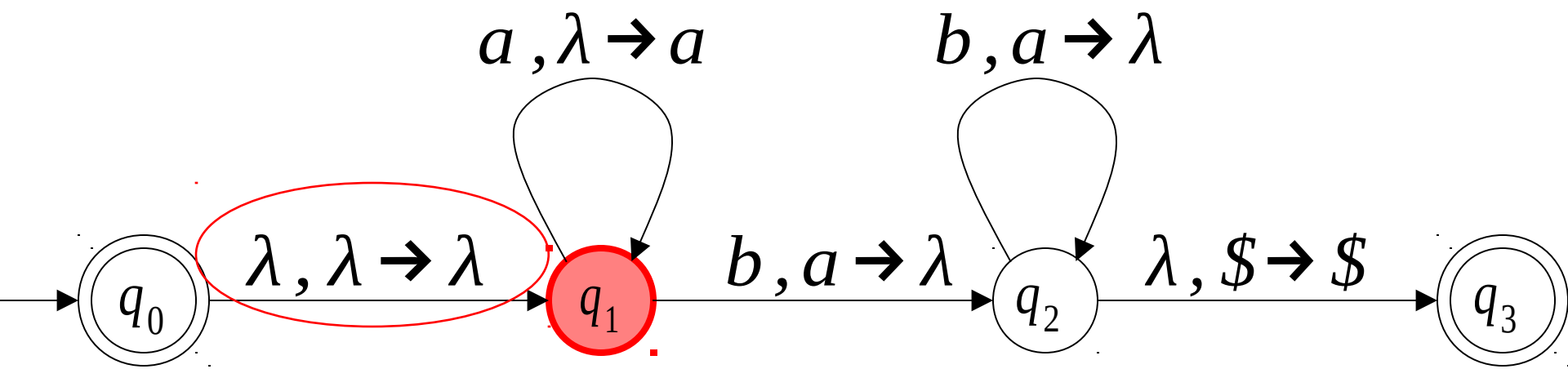
Time 1

Input



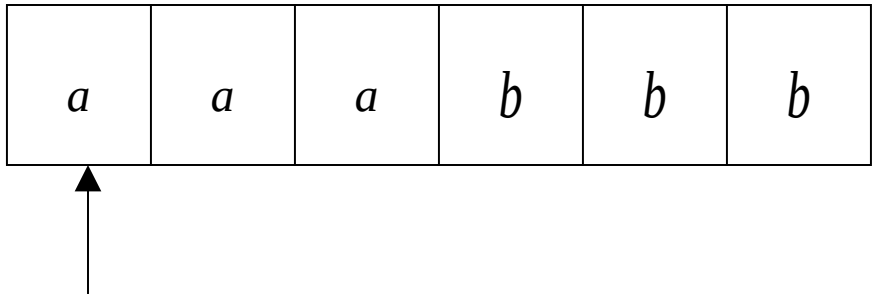


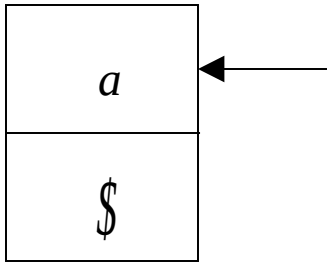
Stack



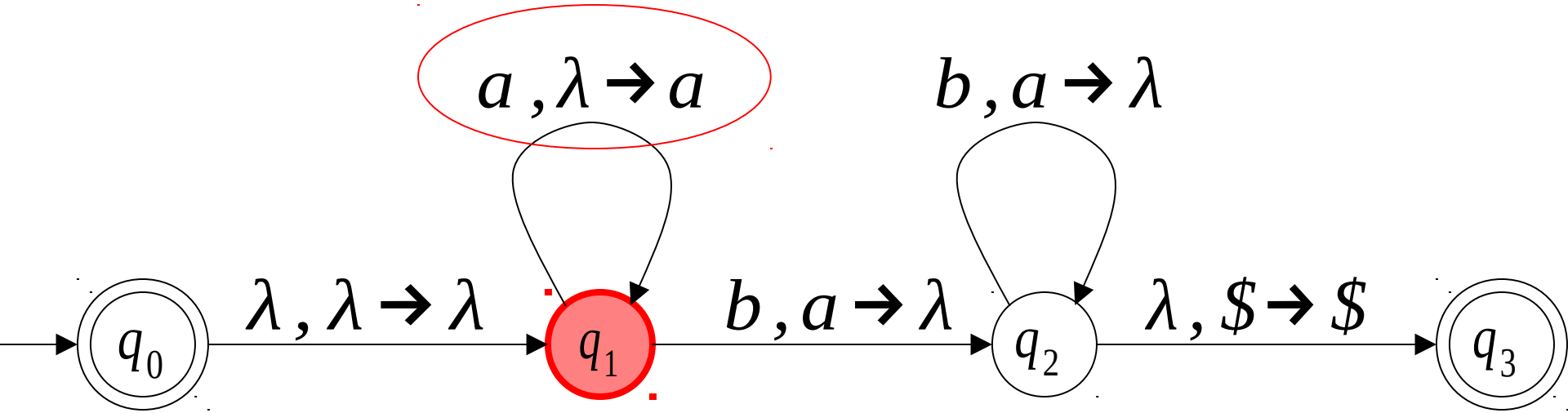
Time 2

Input



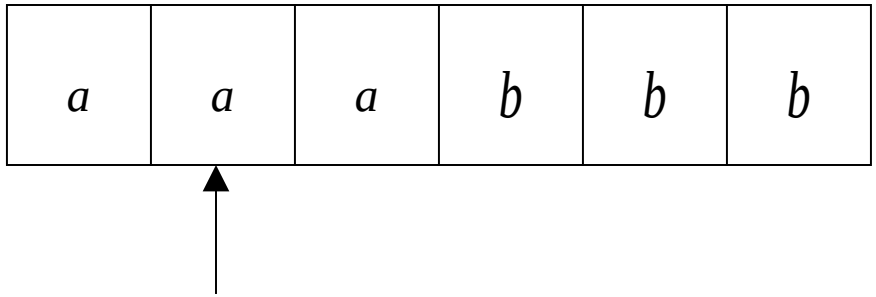


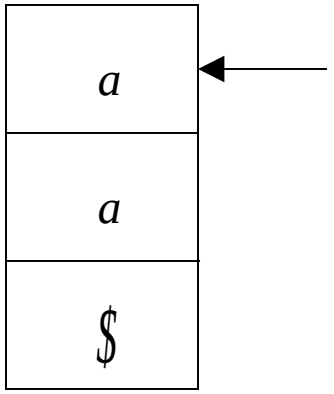
Stack



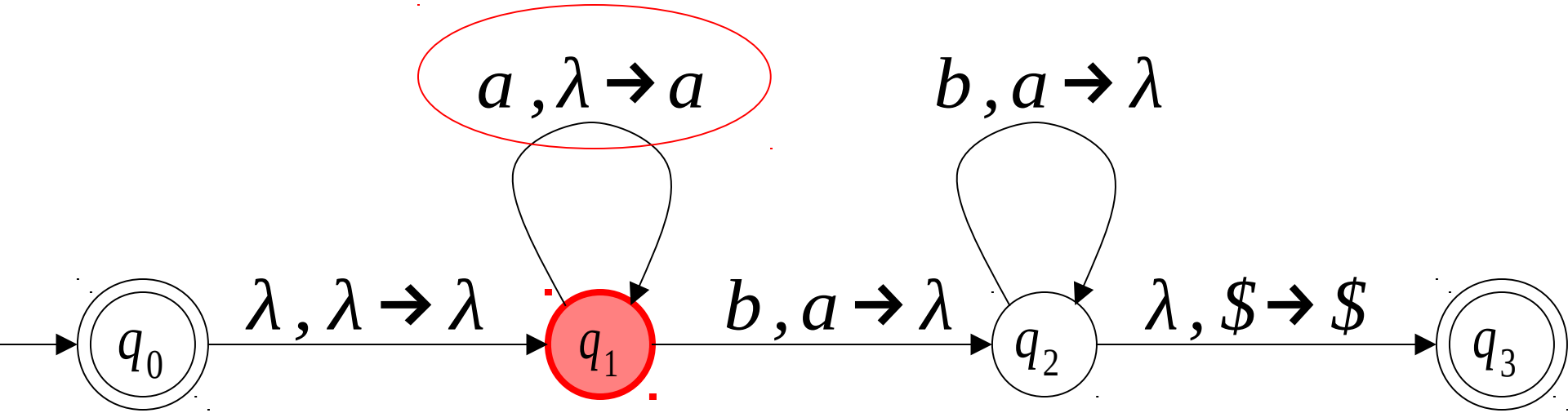
Time 3

Input



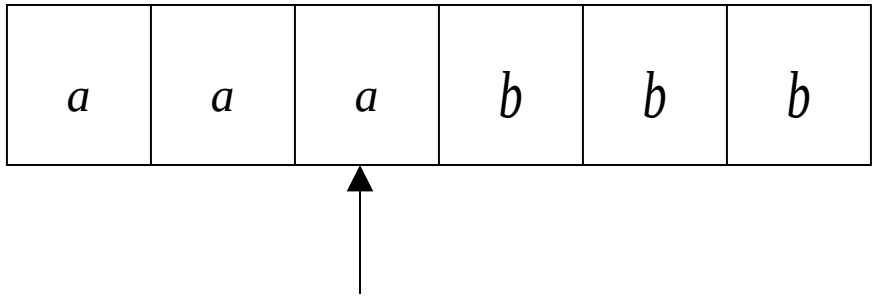


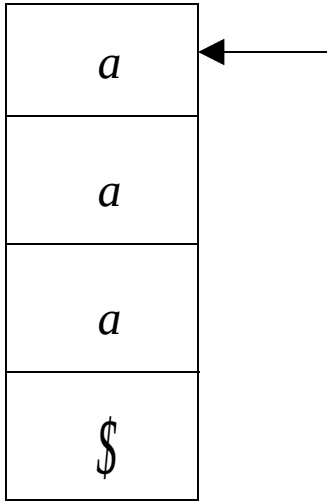
Stack



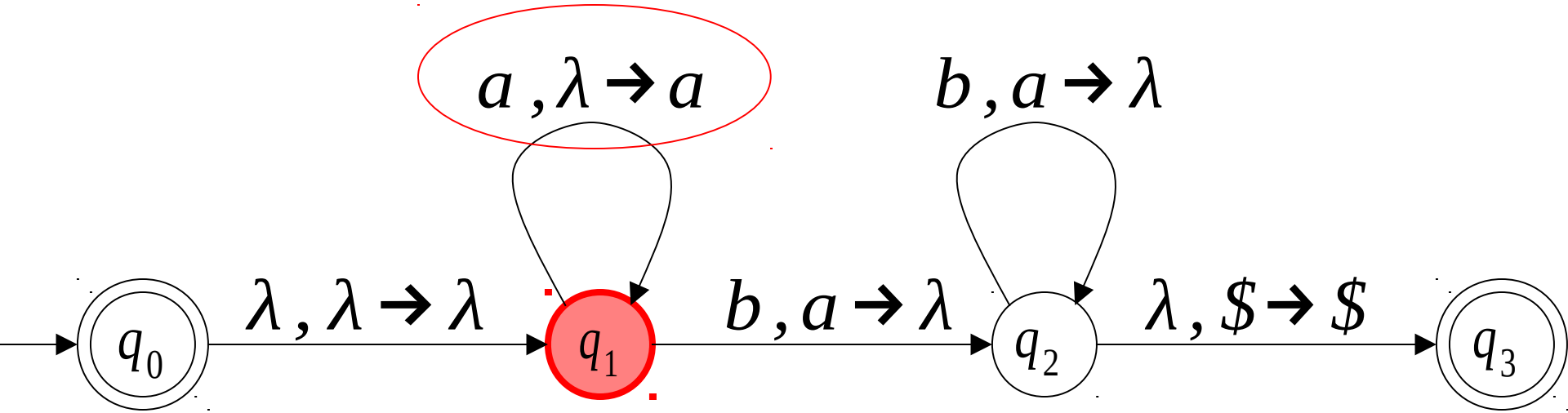
Time 4

Input



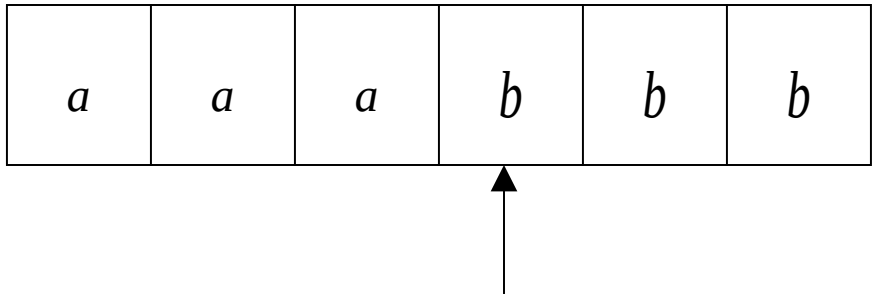


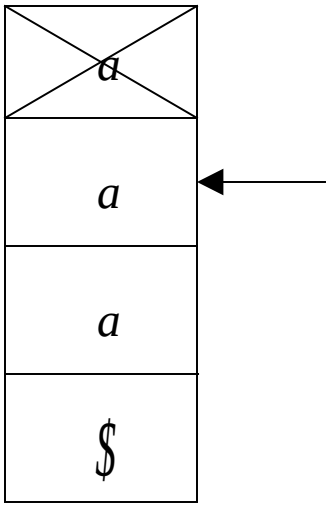
Stack



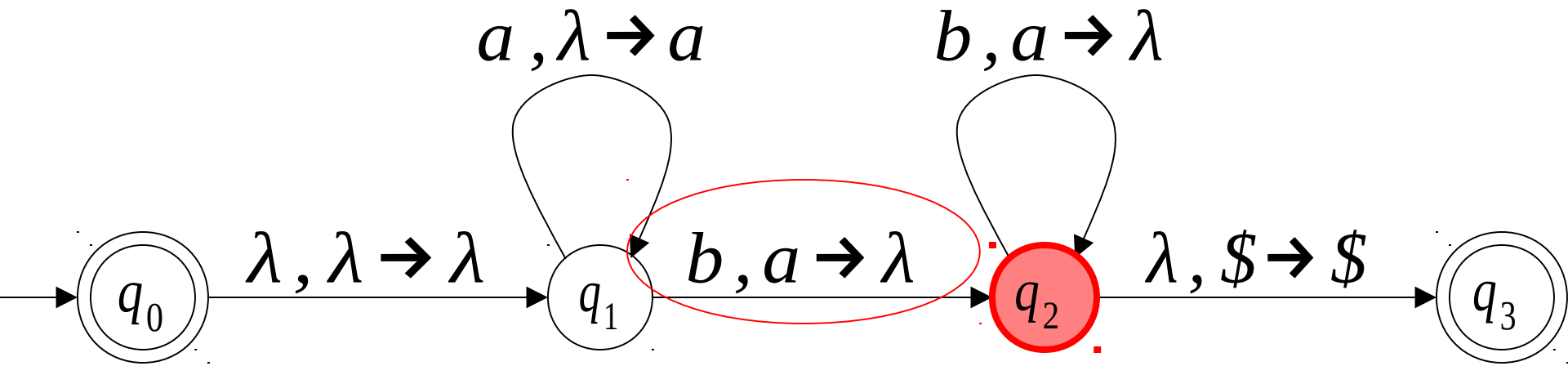
Time 5

Input



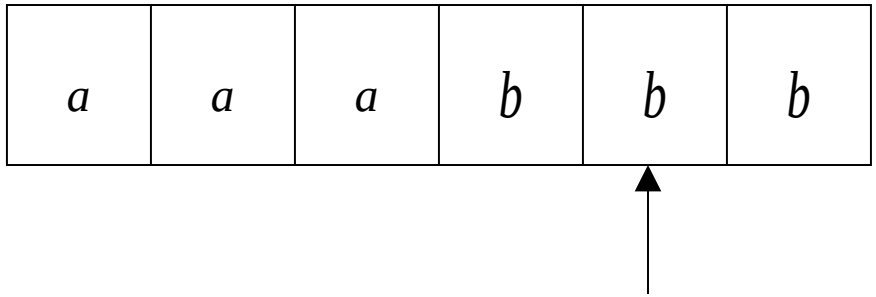


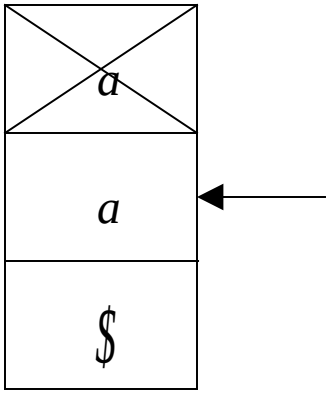
Stack



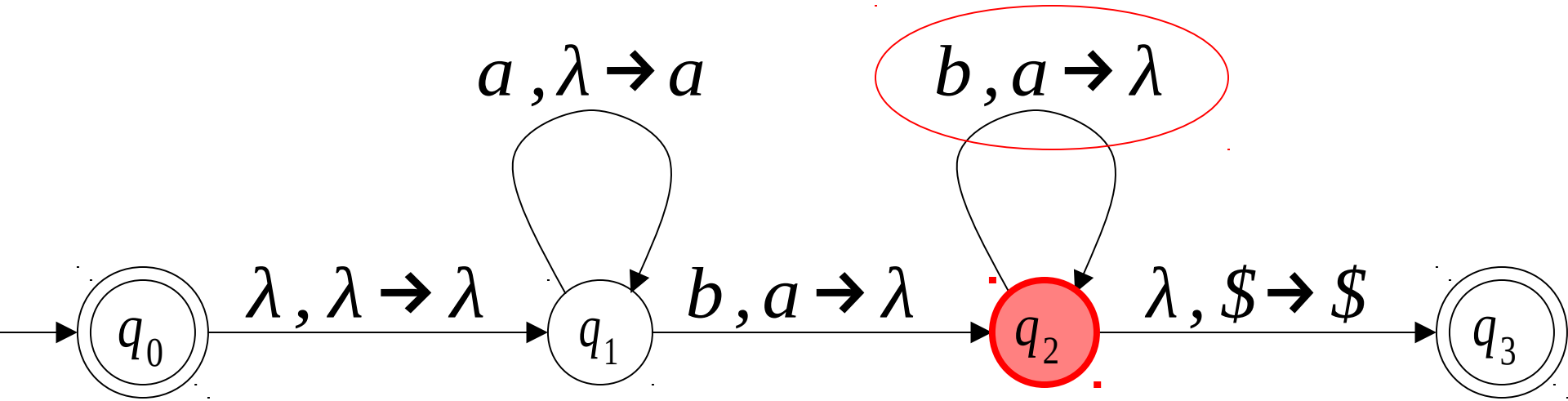
Time 6

Input



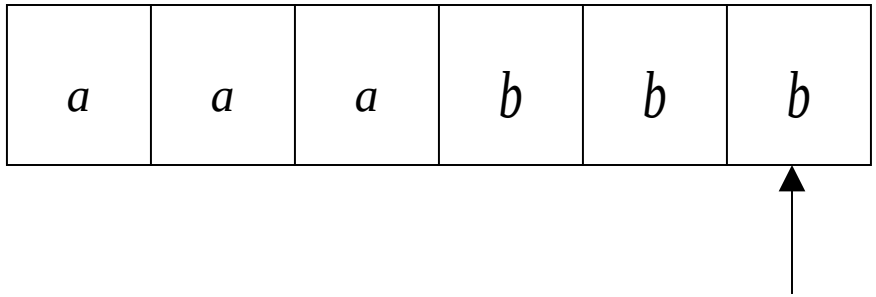


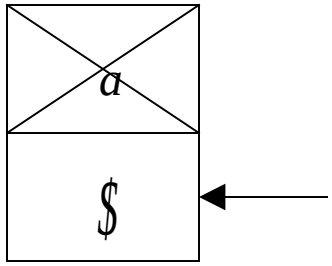
Stack



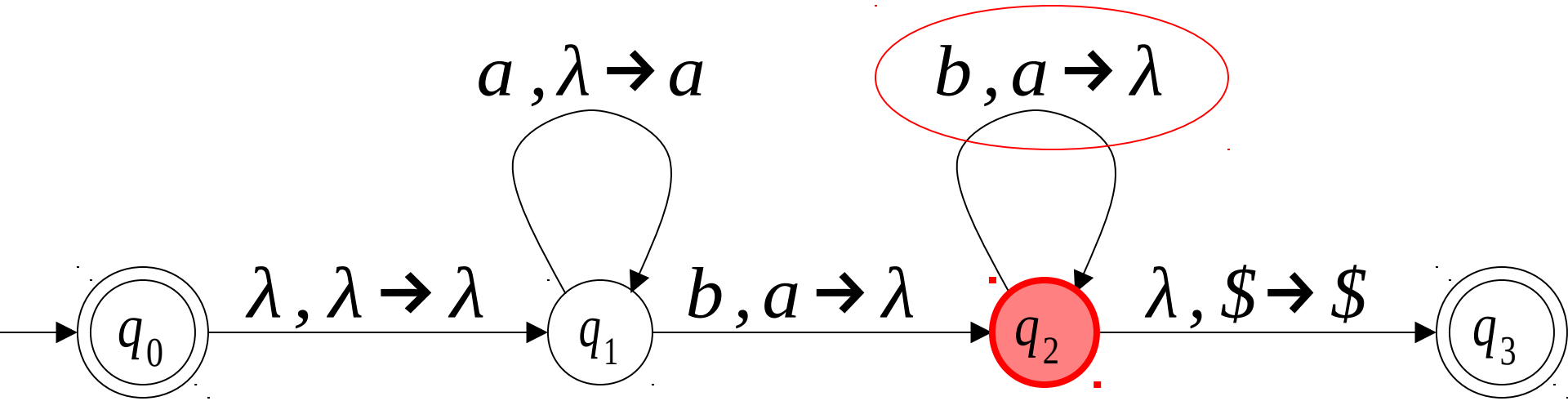
Time 7

Input



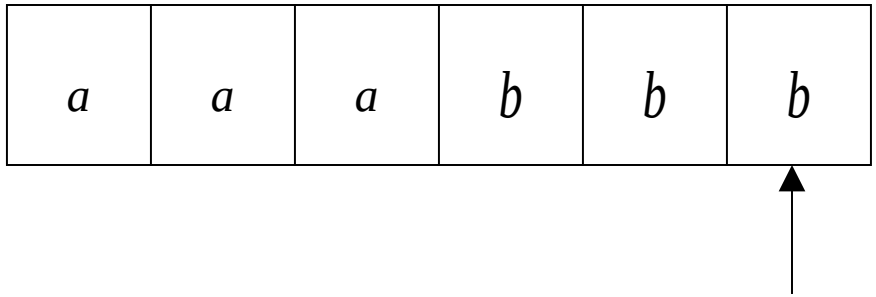


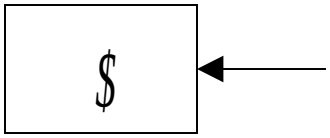
Stack



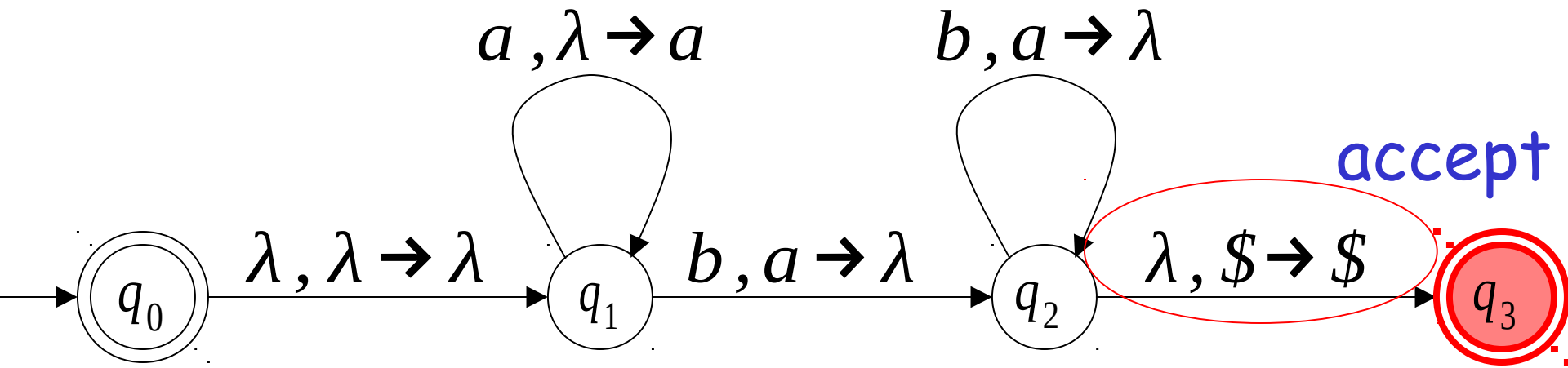
Time 8

Input





Stack



A string is accepted if there is a computation such that:

All the input is consumed

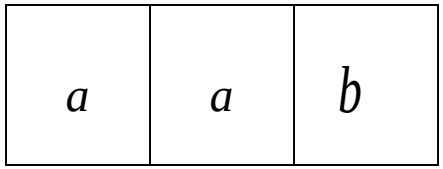
AND

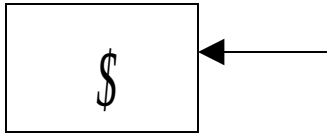
The last state is an accepting state

we do not care about the stack contents at the end of the accepting computation

Rejection Example: Time 0

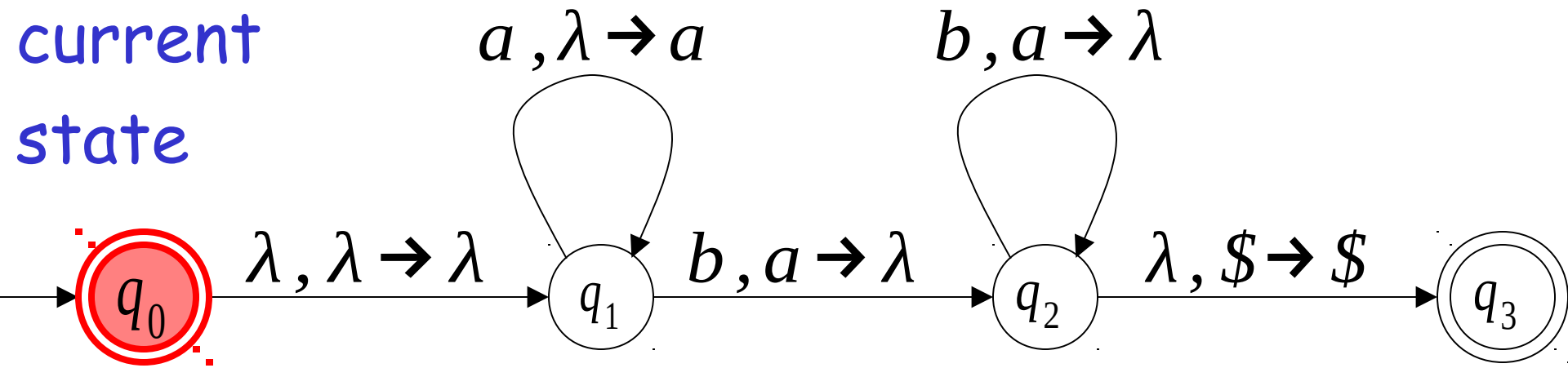
Input





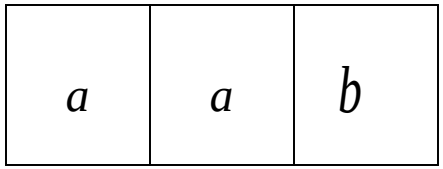
Stack

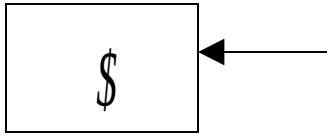
current state



Rejection Example: Time 1

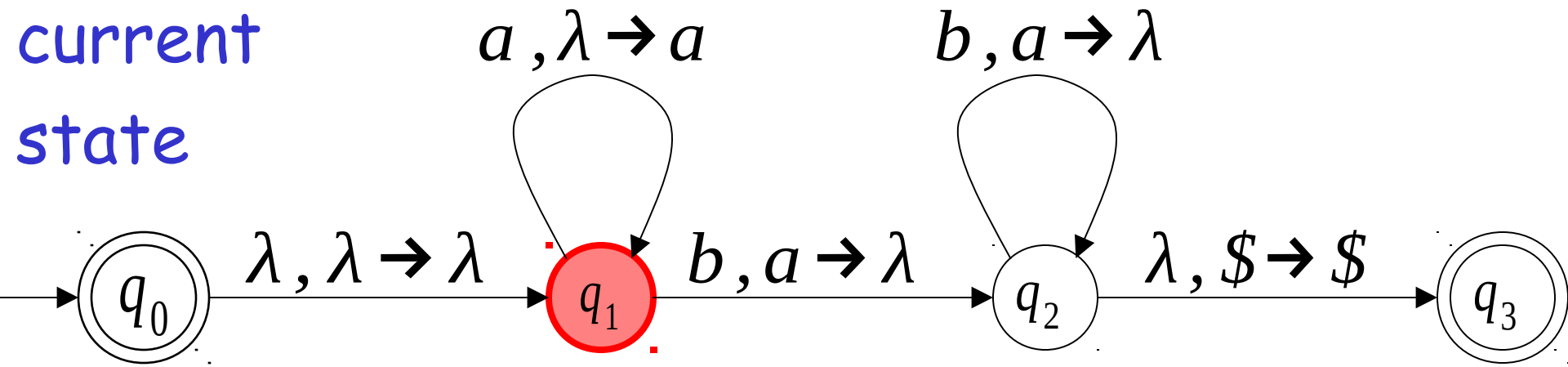
Input





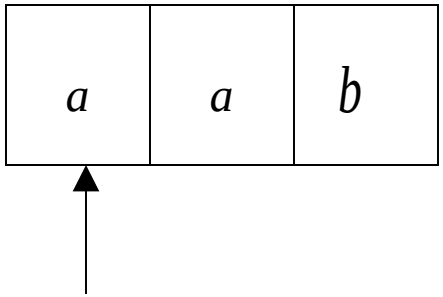
Stack

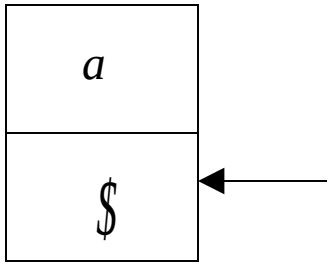
current state



Rejection Example: Time 2

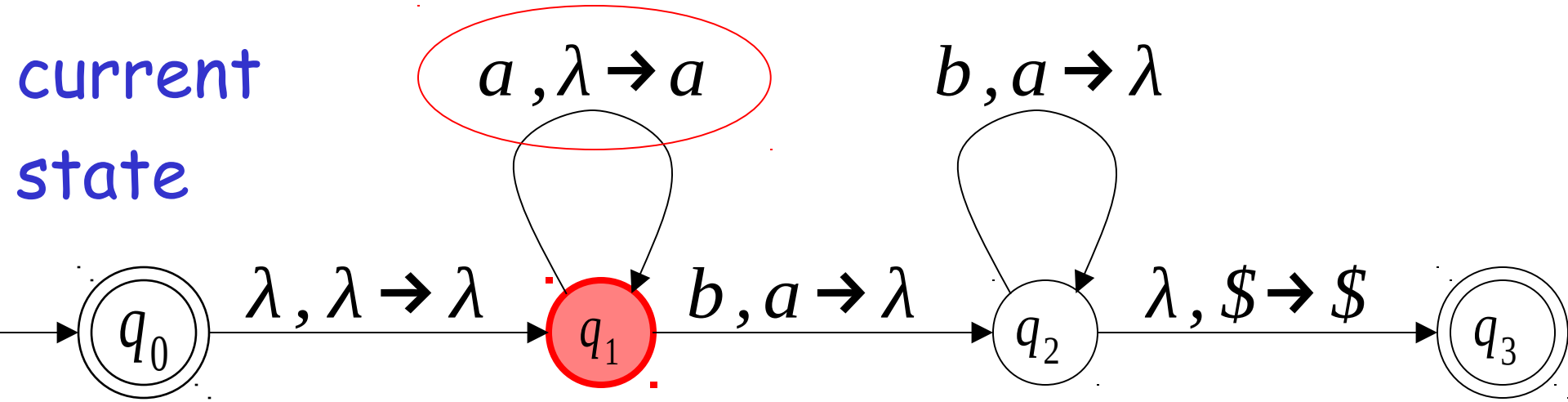
Input





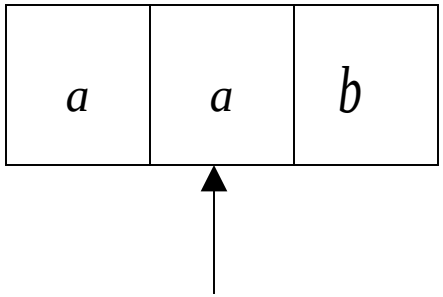
Stack

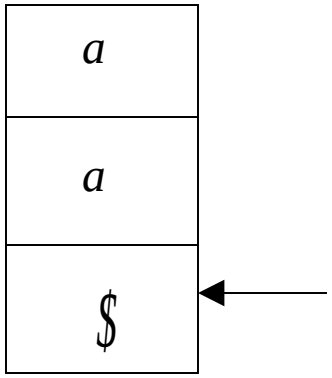
current state



Rejection Example: Time 3

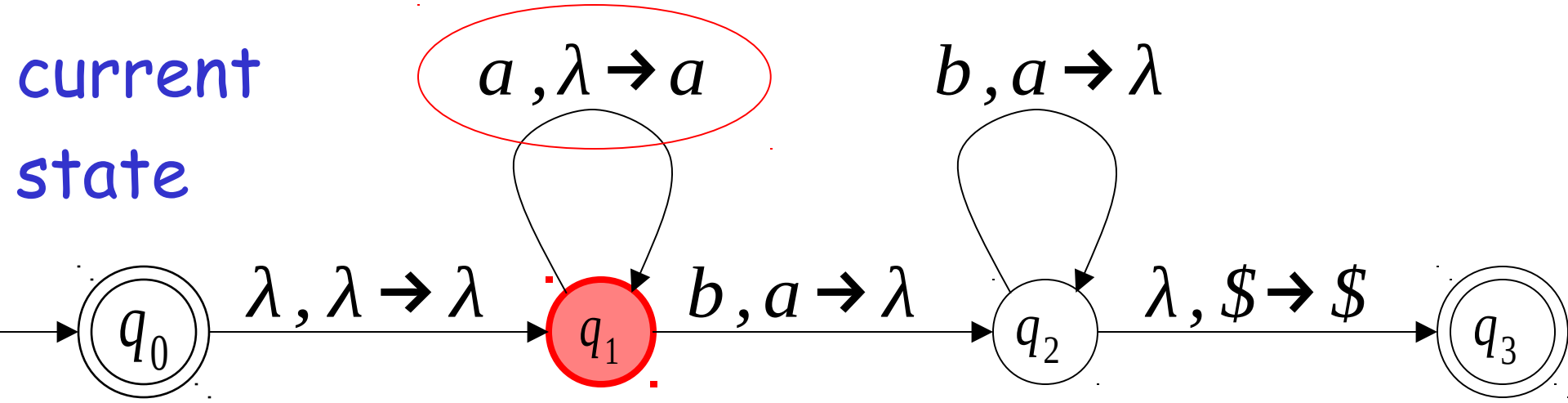
Input





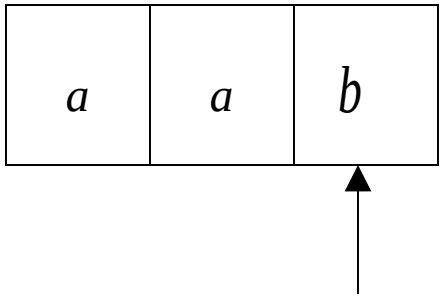
Stack

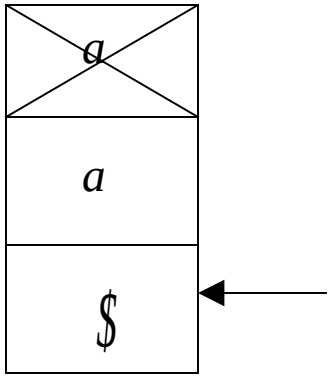
current state



Rejection Example: Time 4

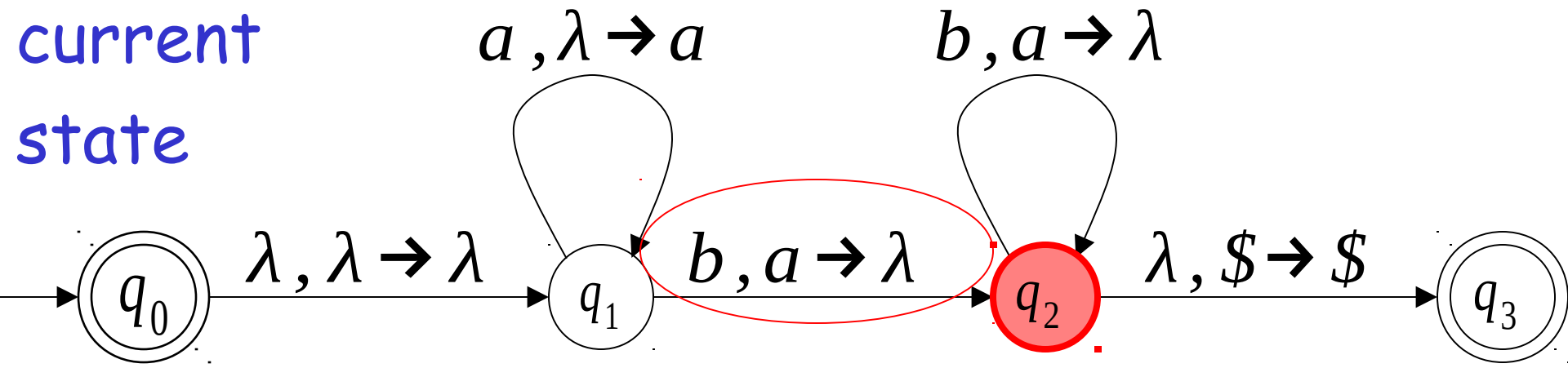
Input





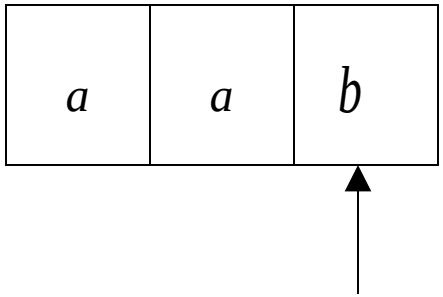
Stack

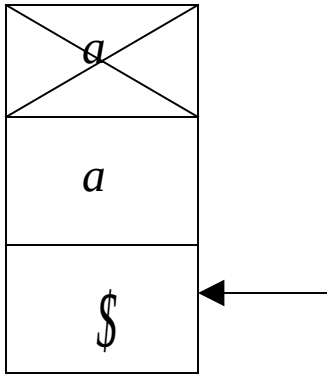
current state



Rejection Example: Time 4

Input

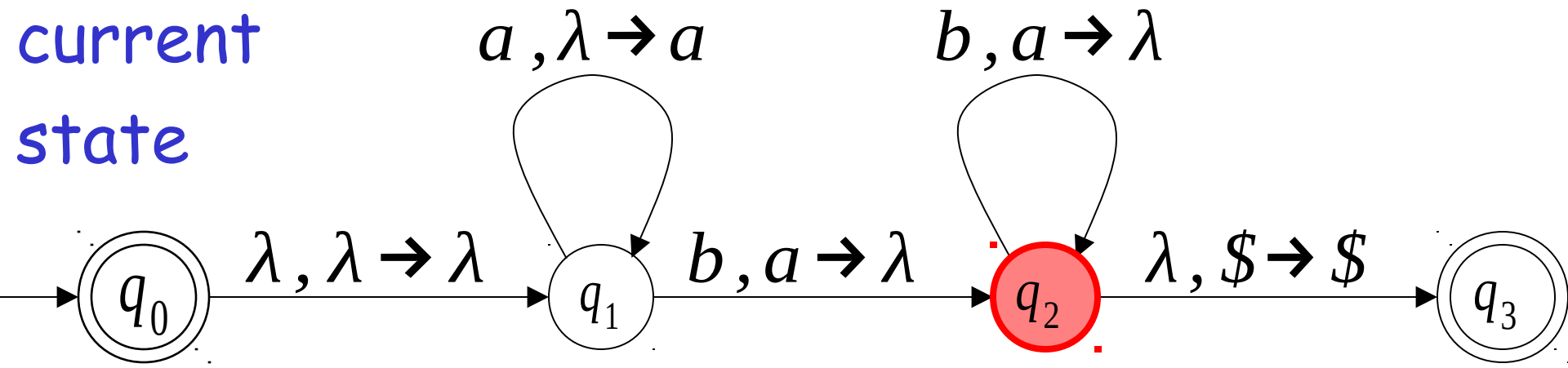




Stack

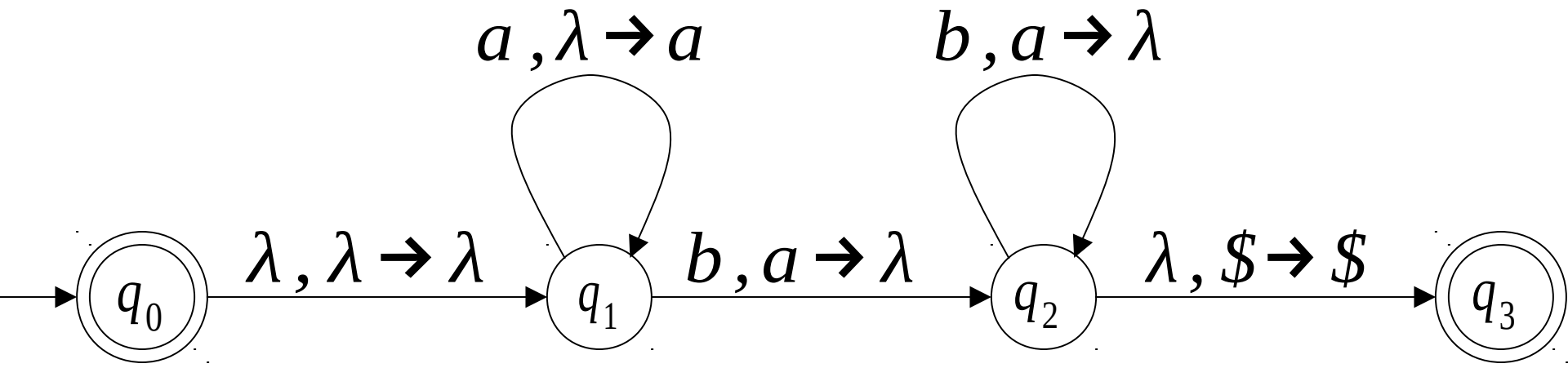
reject

current state



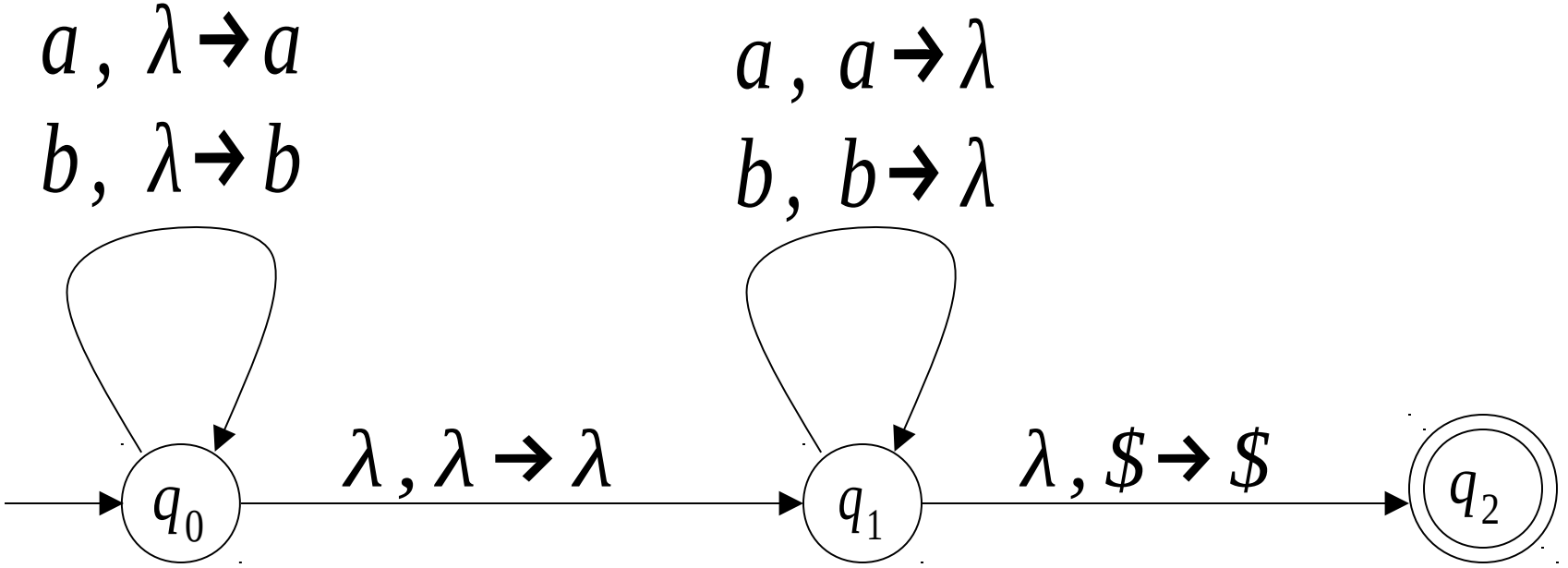
There is no accepting computation for aab

The string aab is rejected by the PDA



Another PDA example

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



Basic Idea:

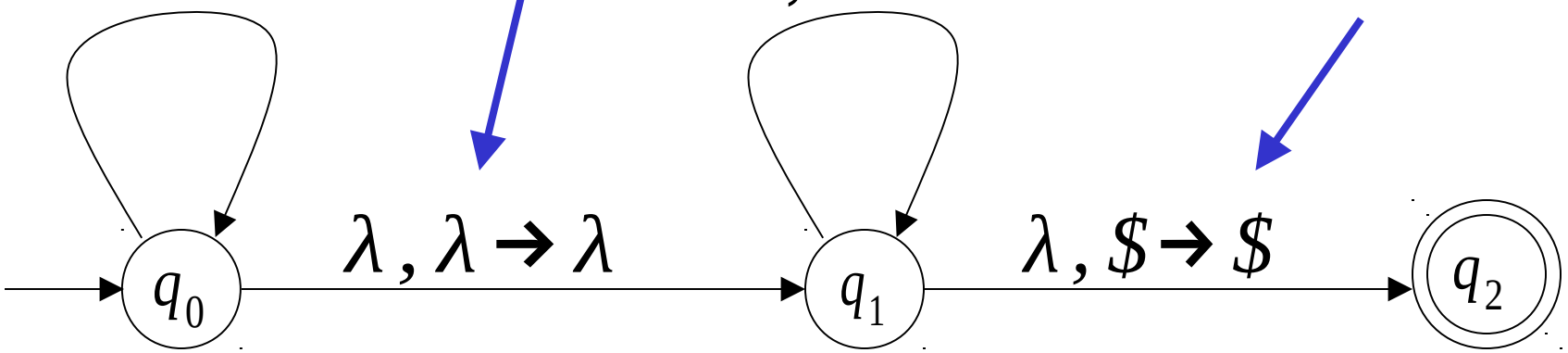
$$L(M)=\{vv^R : v\in\{a,b\}^{¿}\}$$

1. Push v on stack

2. Guess middle of input

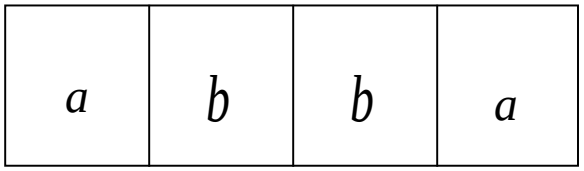
3. Match v^R on input with v on stack

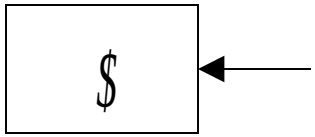
4. Match found



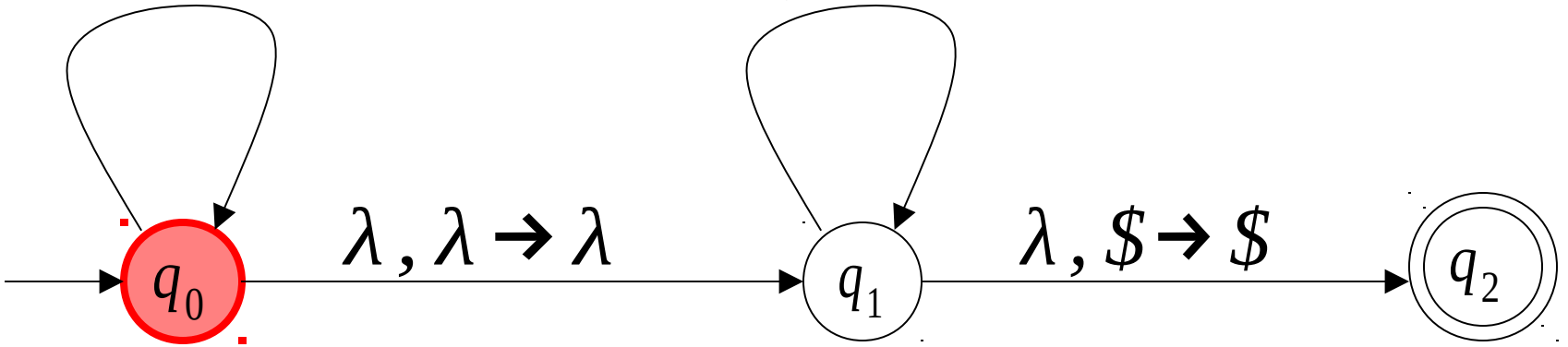
Execution Example: Time 0

Input



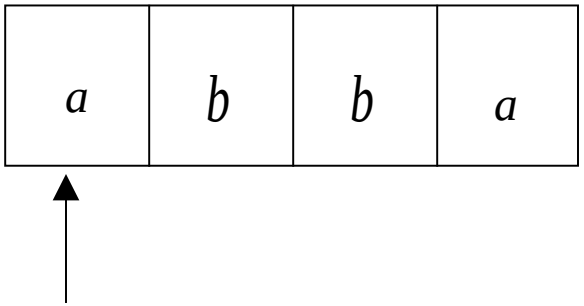


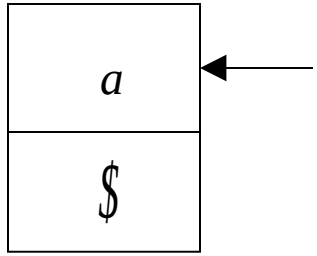
Stack



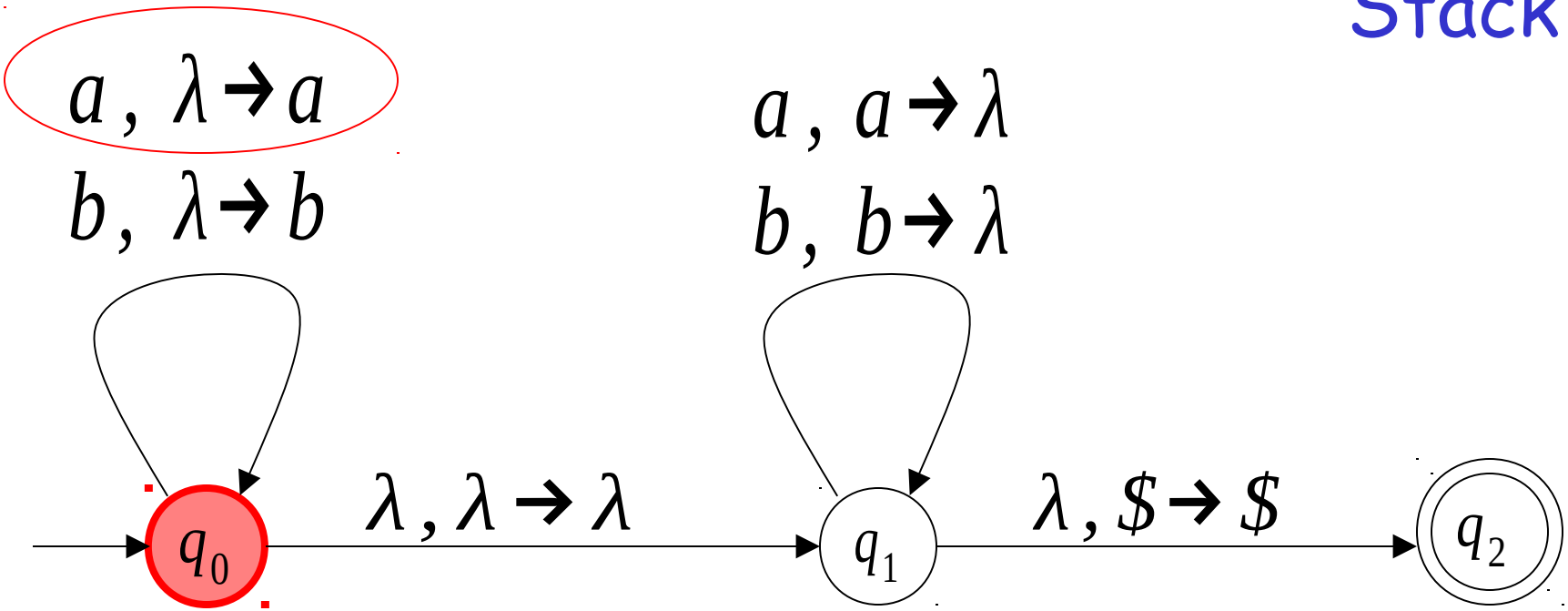
Time 1

Input



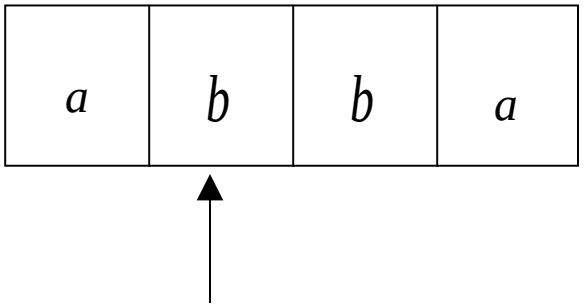


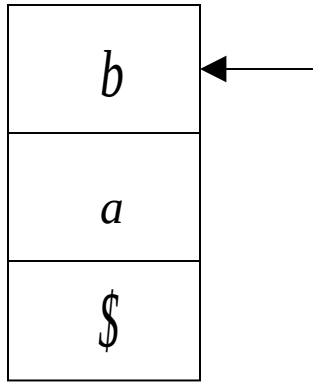
Stack



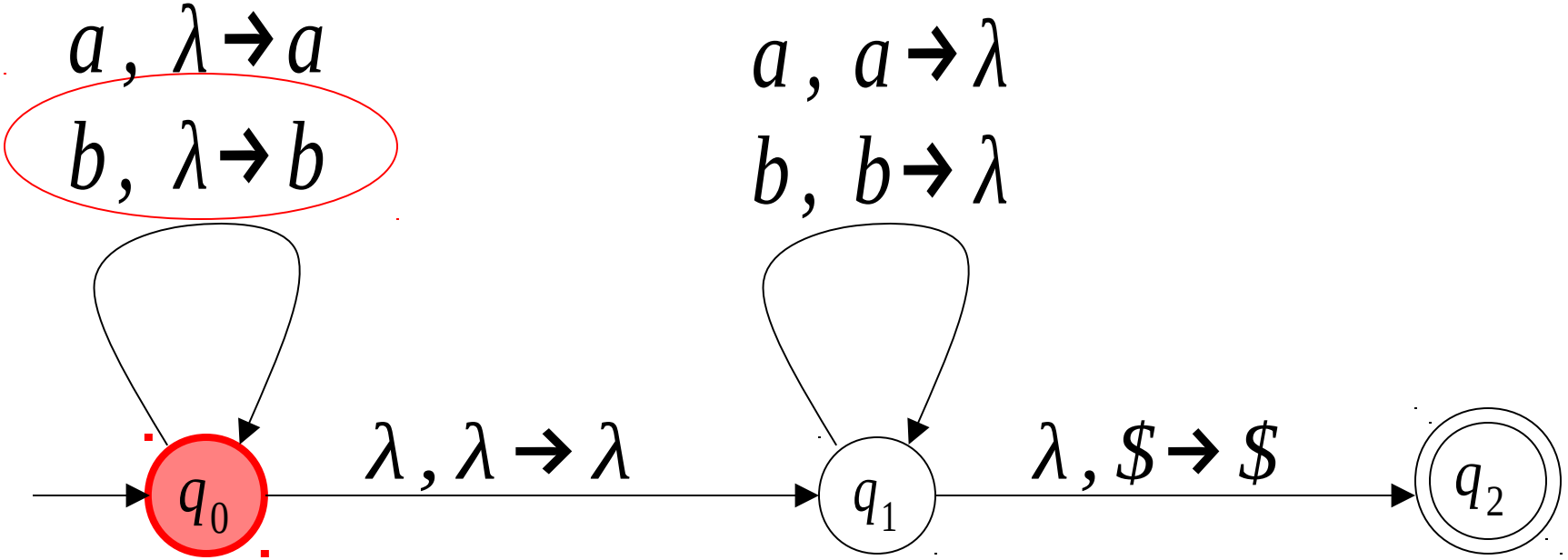
Time 2

Input



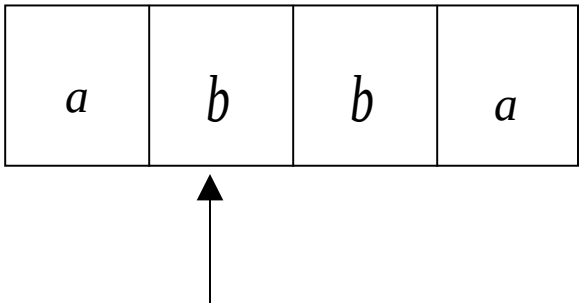


Stack

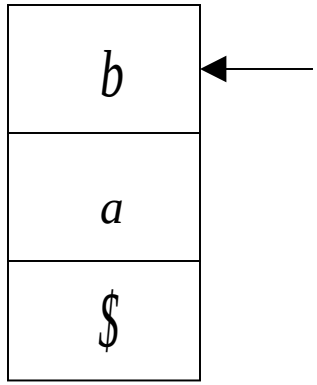


Time 3

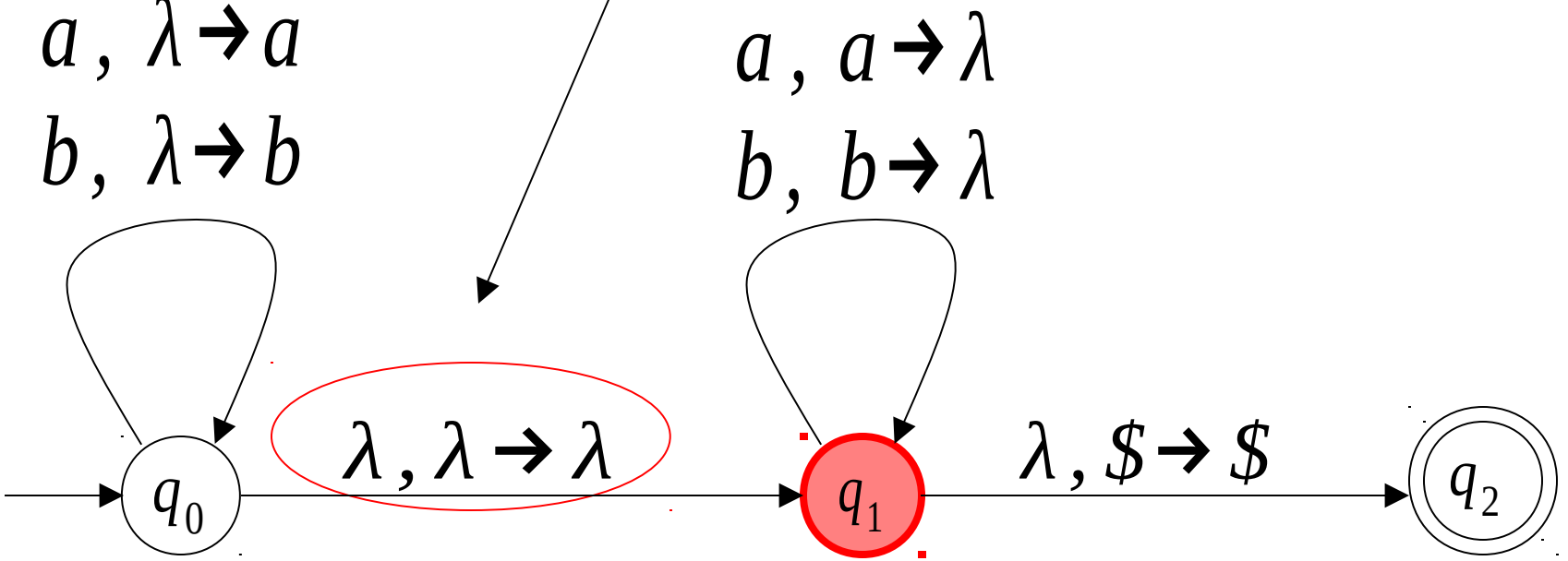
Input



Guess the middle of string

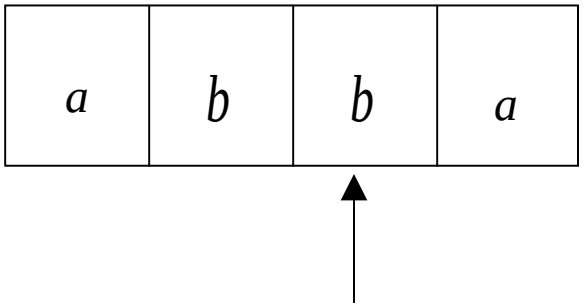


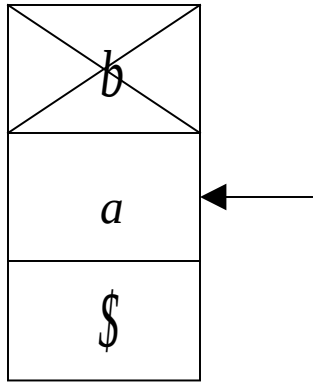
Stack



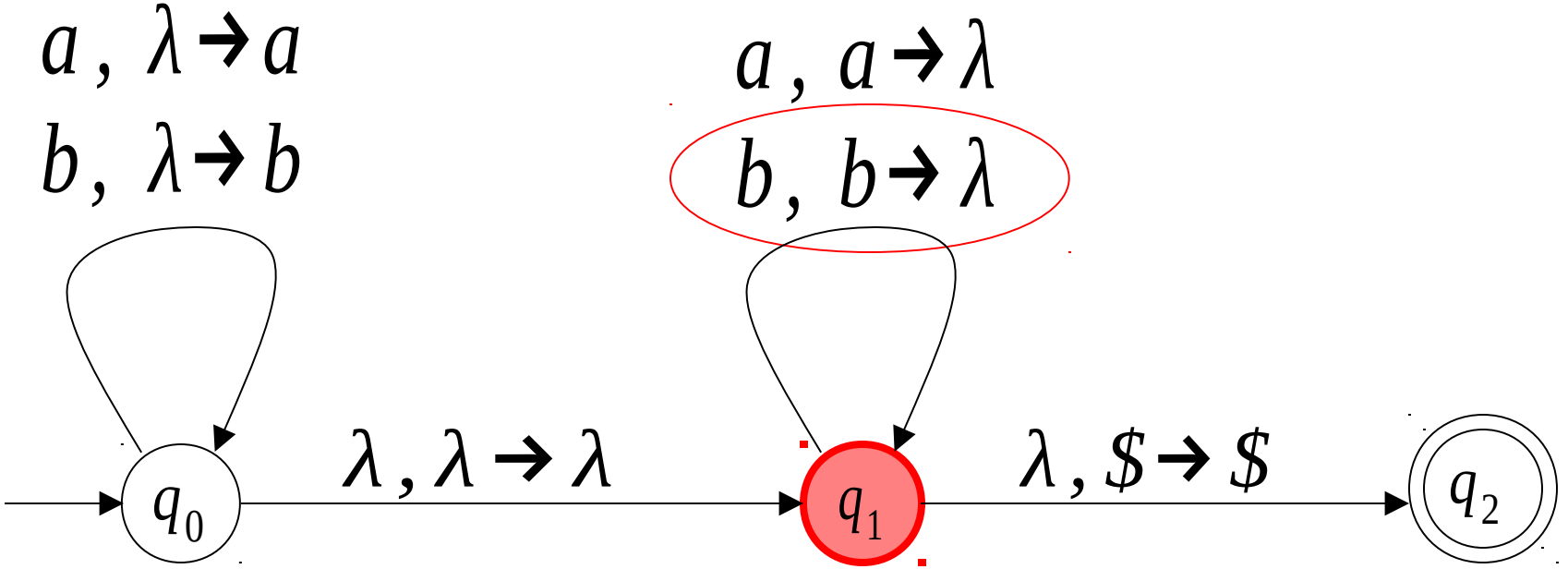
Time 4

Input



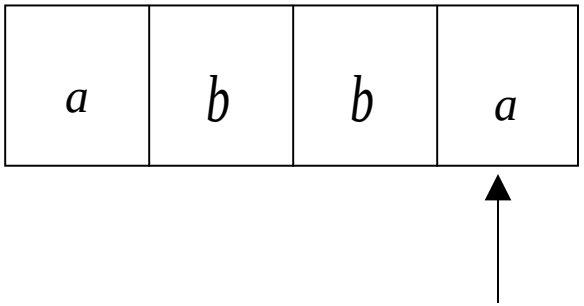


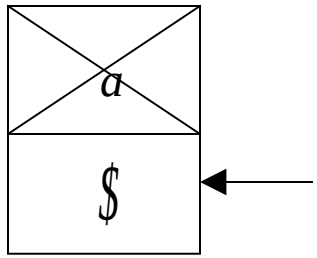
Stack



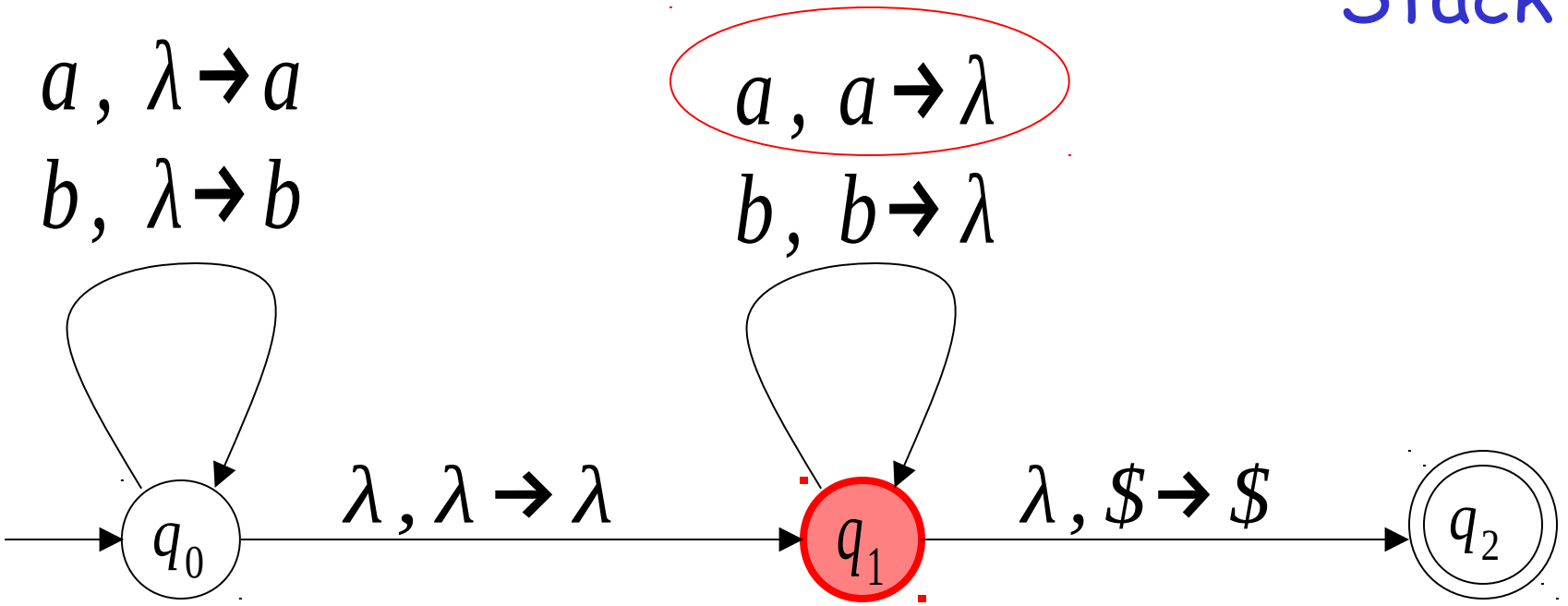
Time 5

Input



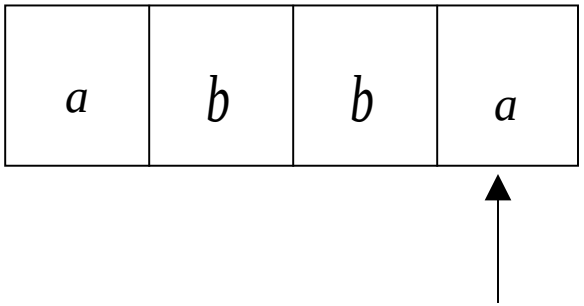


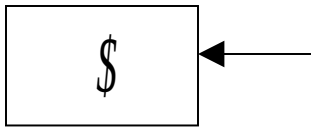
Stack



Time 6

Input





Stack

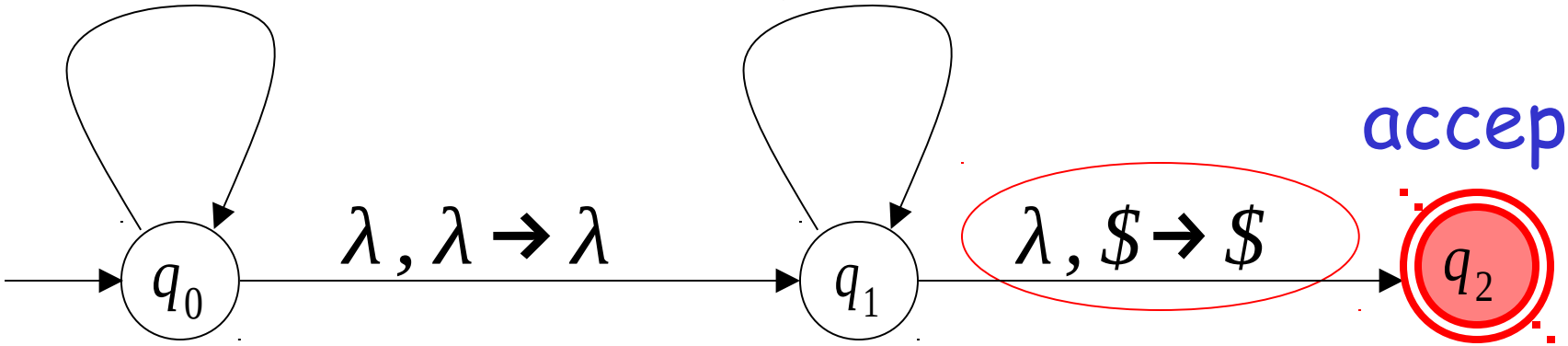
$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

$a, a \rightarrow \lambda$

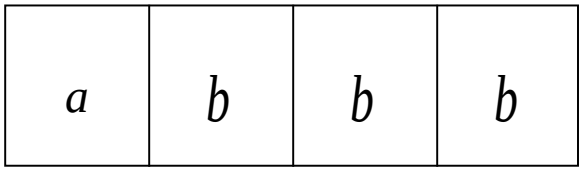
$b, b \rightarrow \lambda$

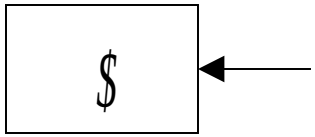
accept



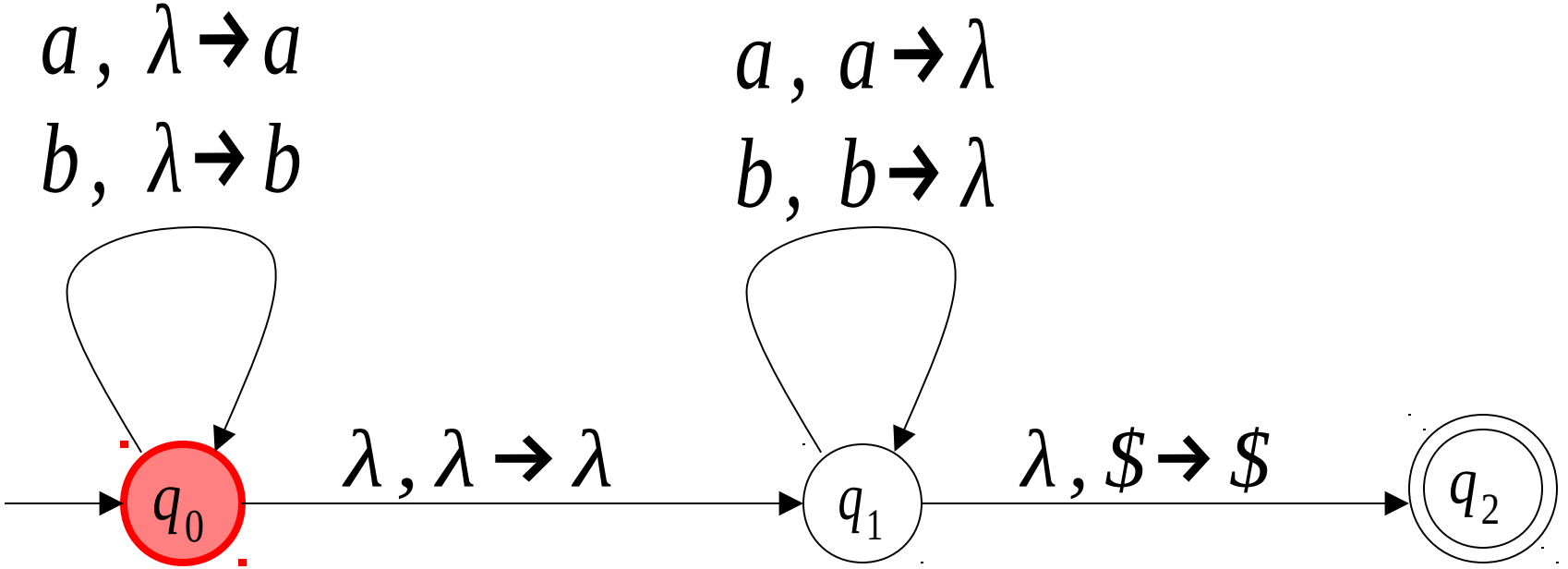
Rejection Example: Time 0

Input



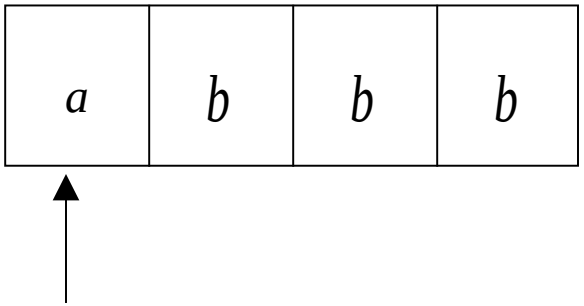


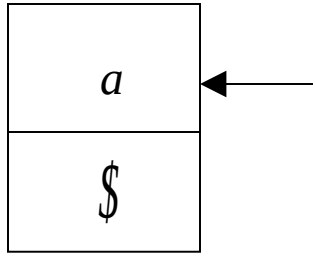
Stack



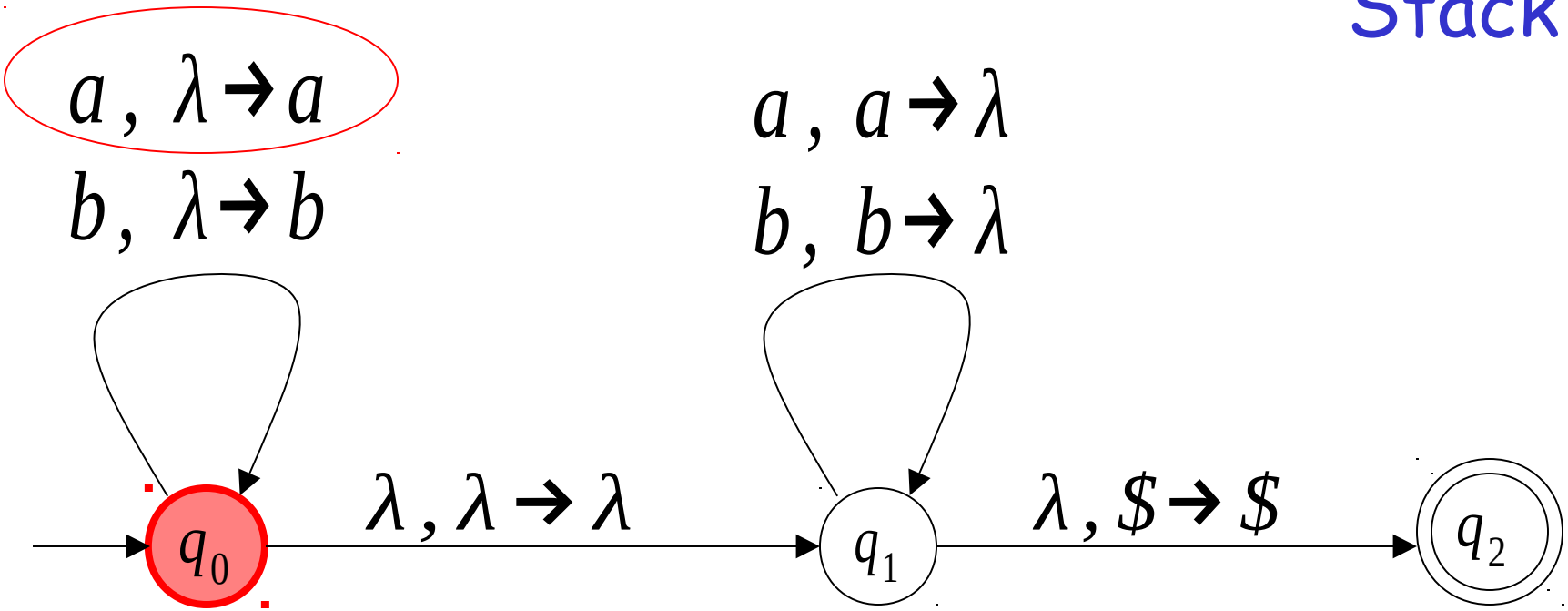
Time 1

Input



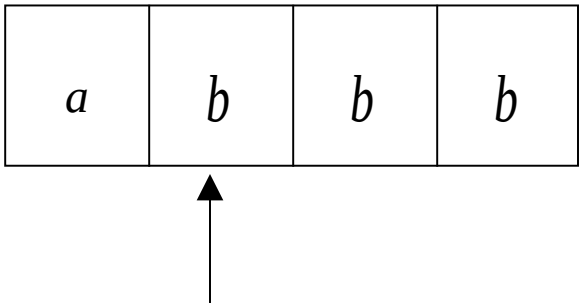


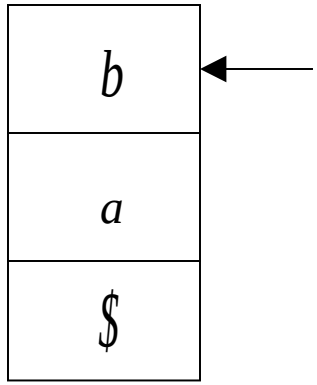
Stack



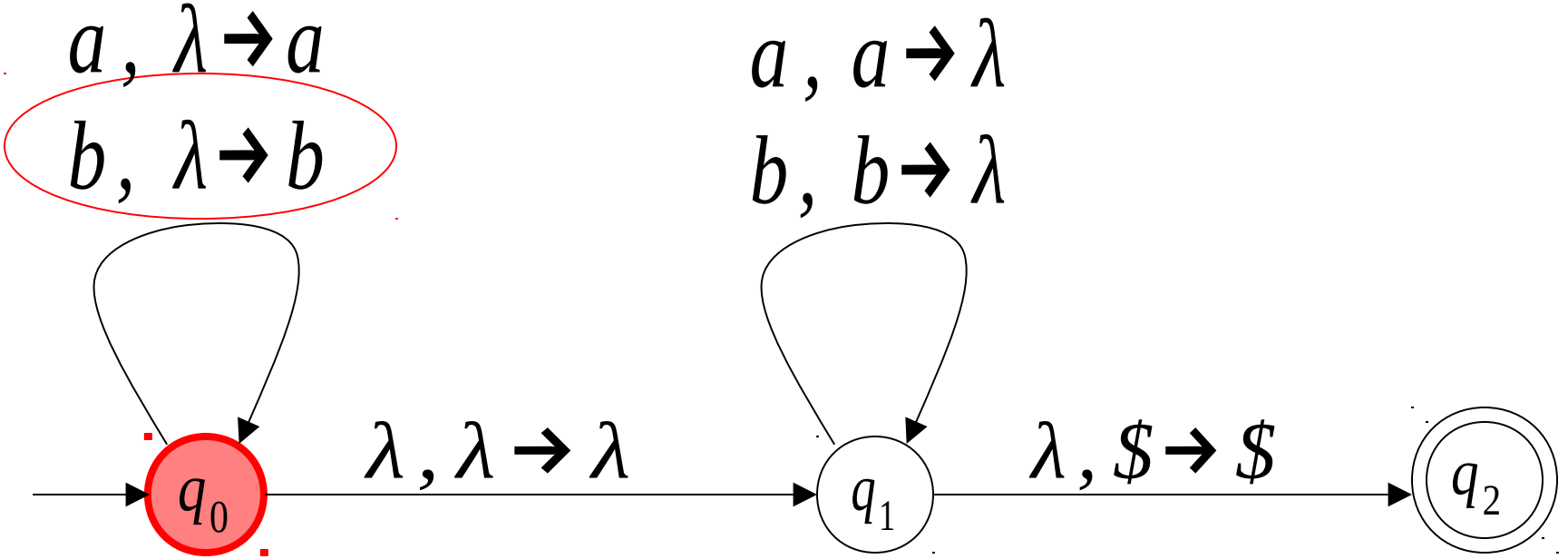
Time 2

Input



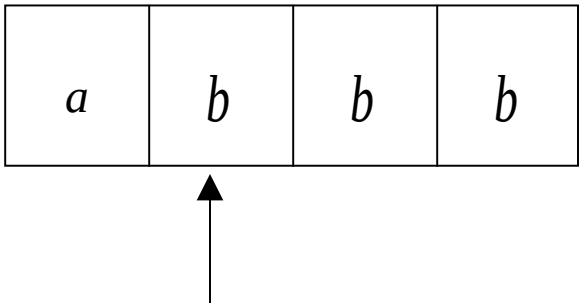


Stack

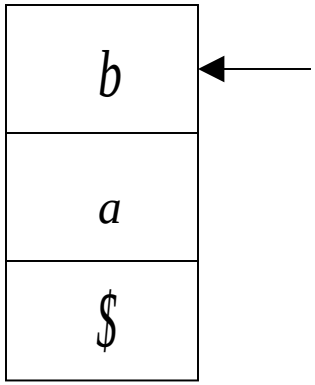


Time 3

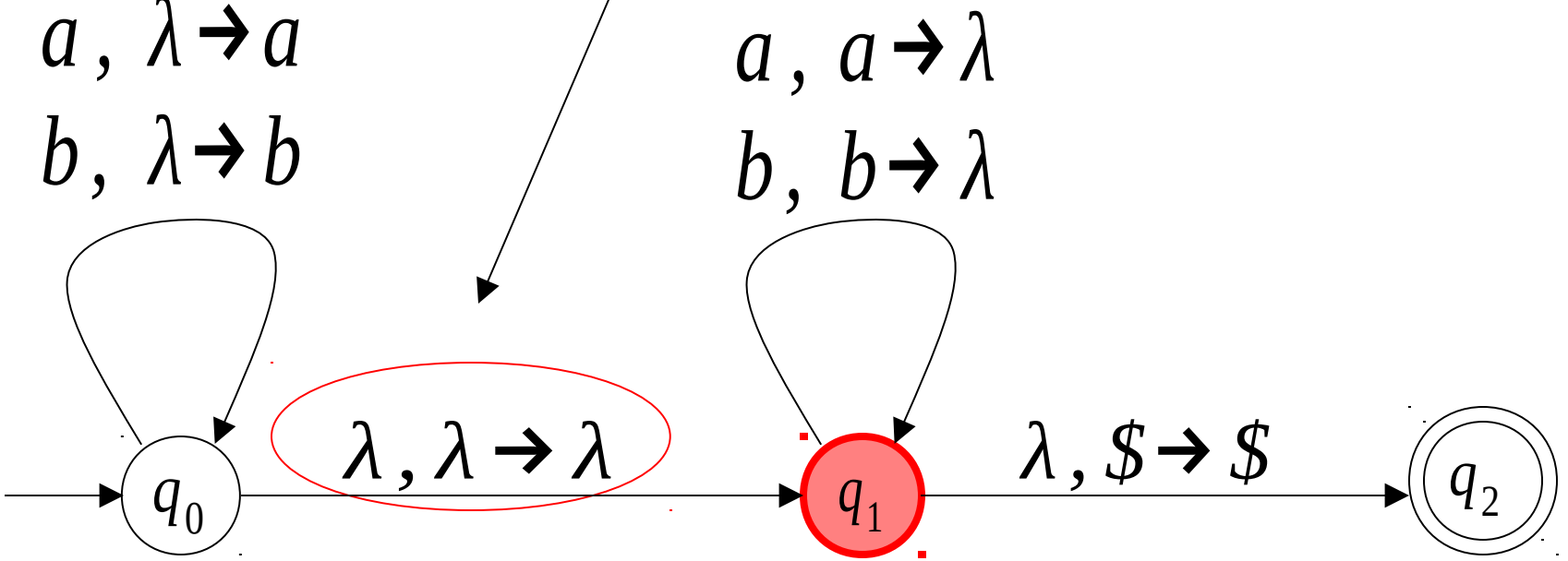
Input



Guess the middle of string

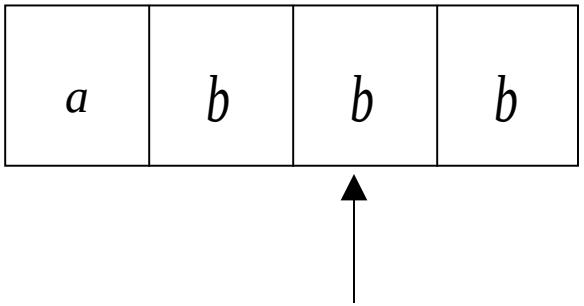


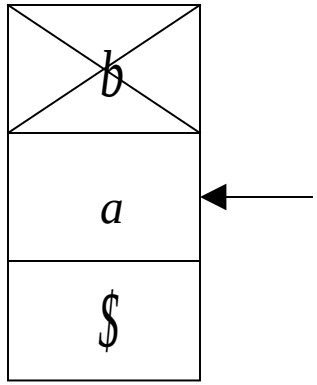
Stack



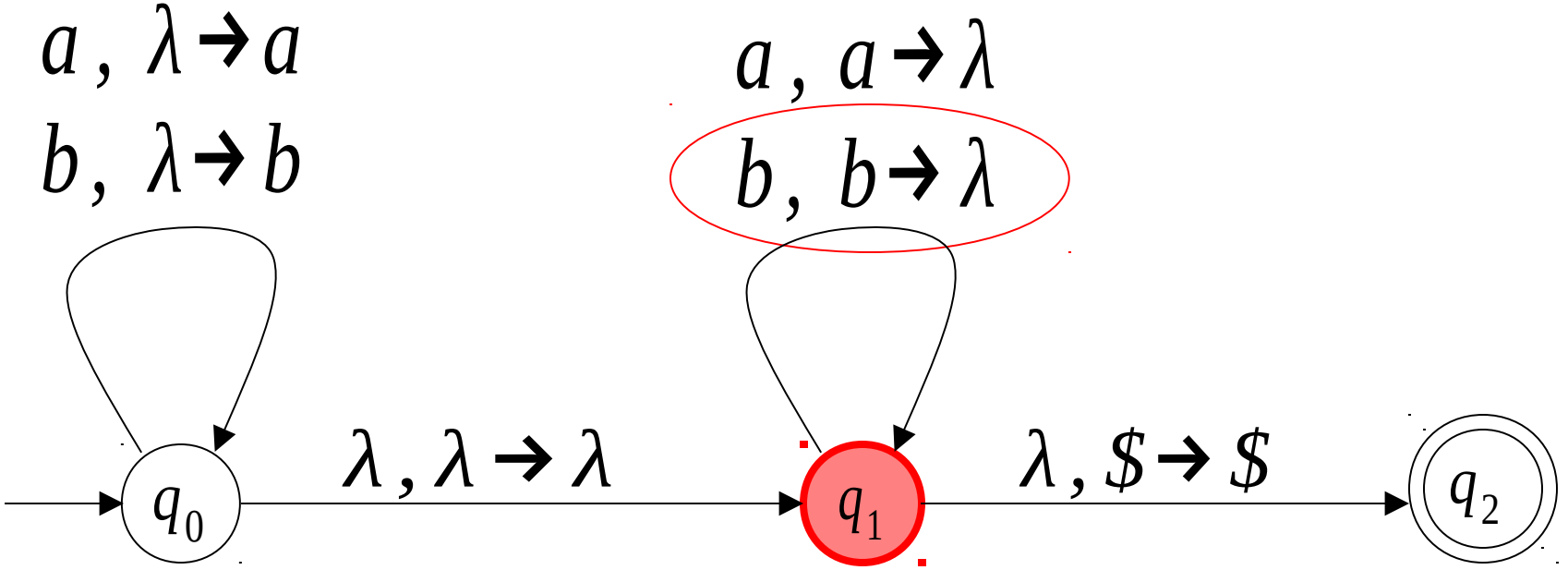
Time 4

Input





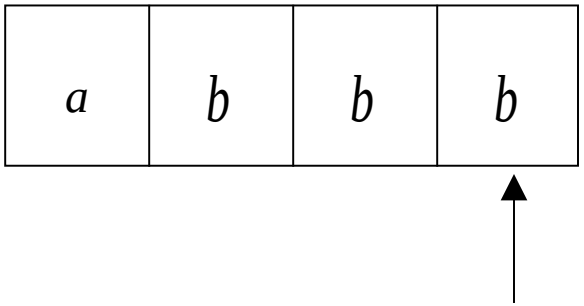
Stack



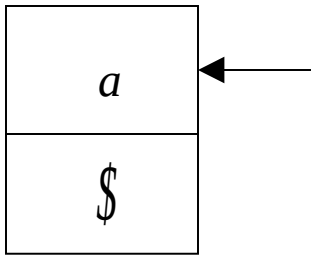
Time 5

There is no possible transition.

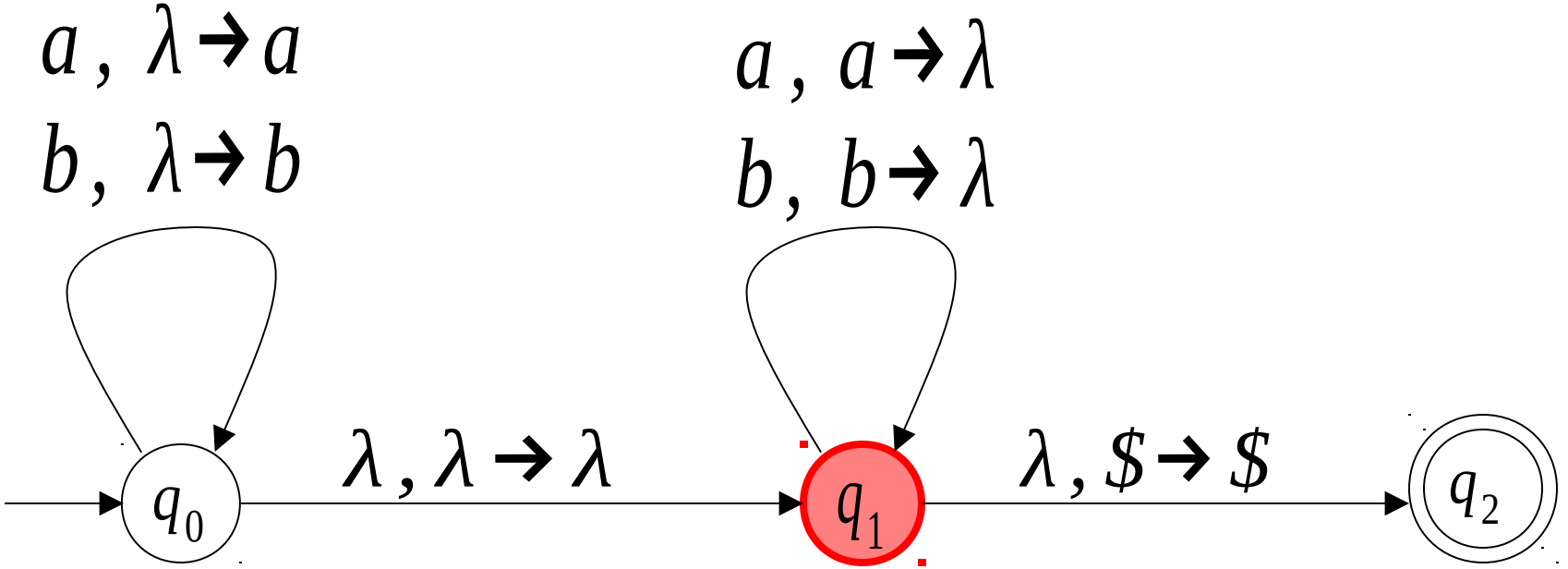
Input



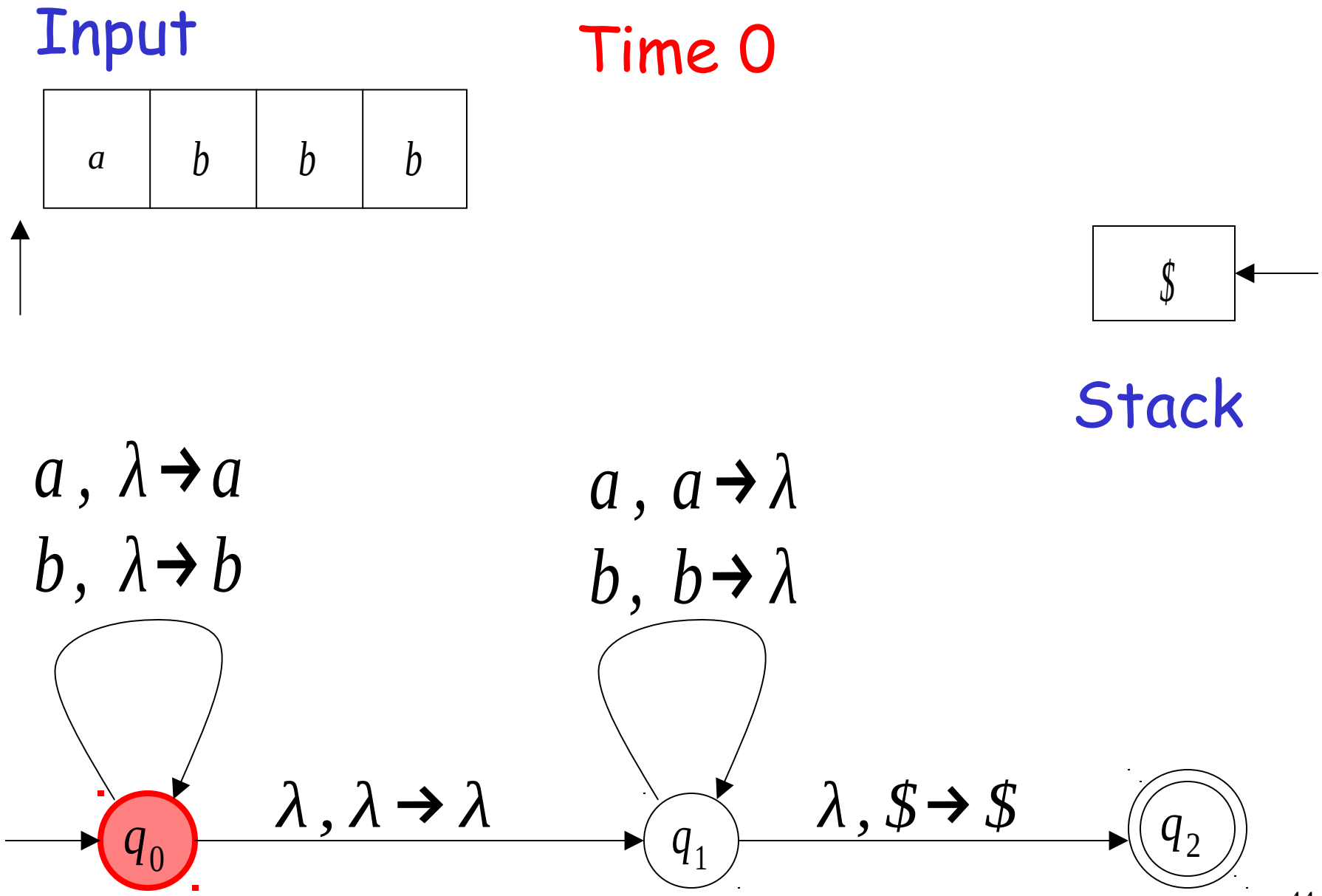
Input is not consumed



Stack

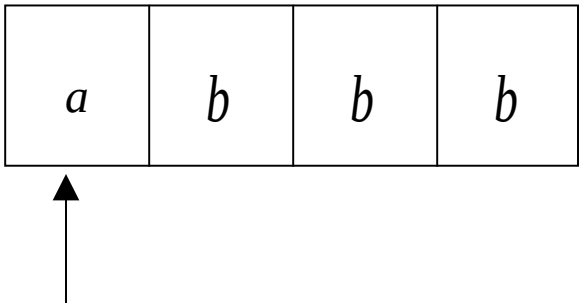


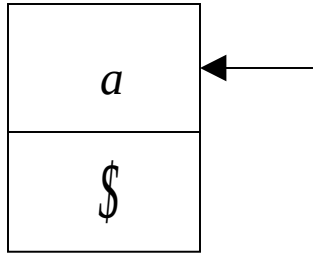
Another computation on same string:



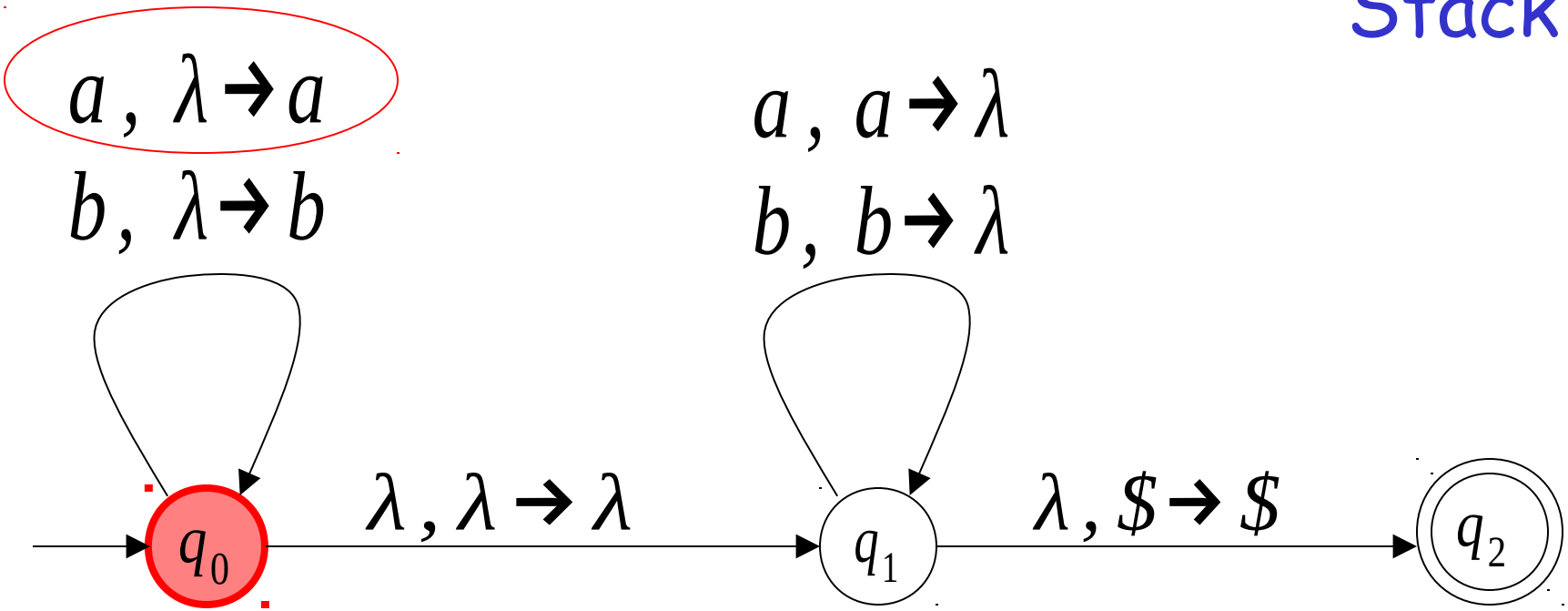
Time 1

Input



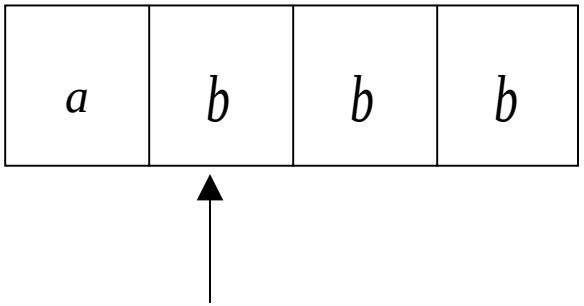


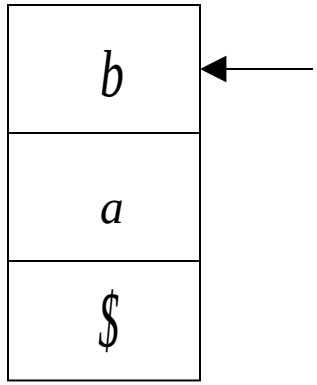
Stack



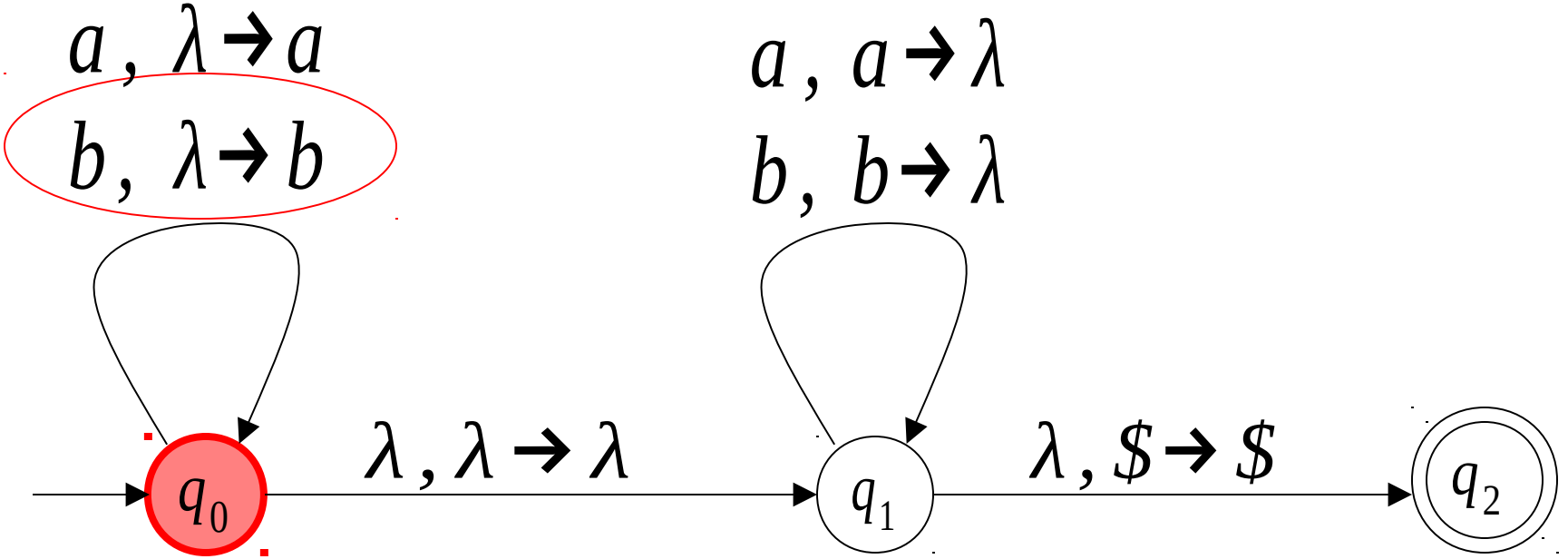
Time 2

Input



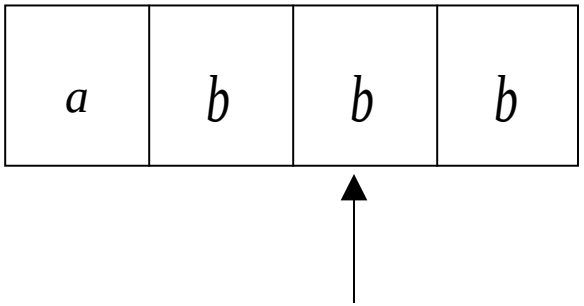


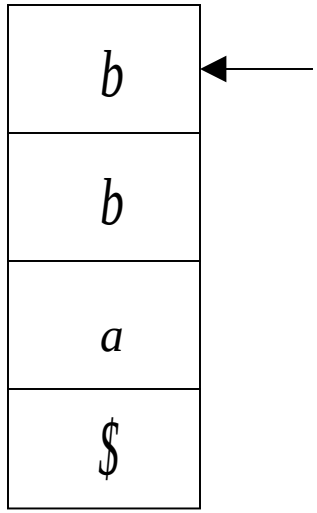
Stack



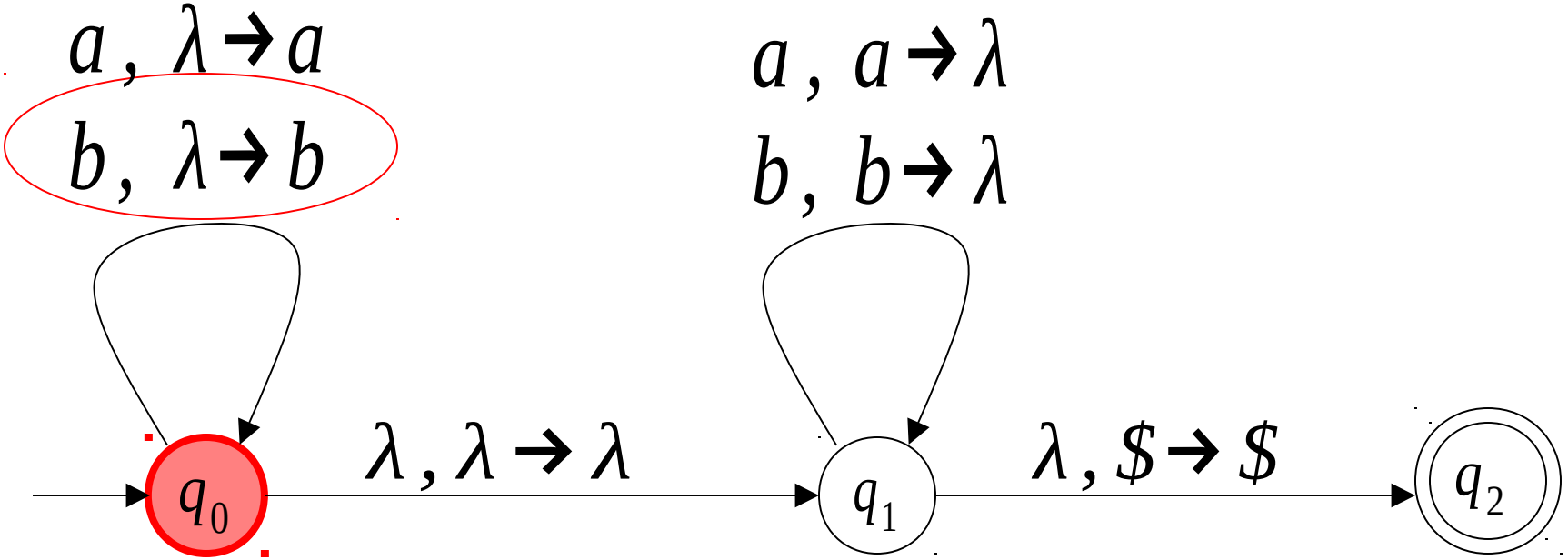
Time 3

Input



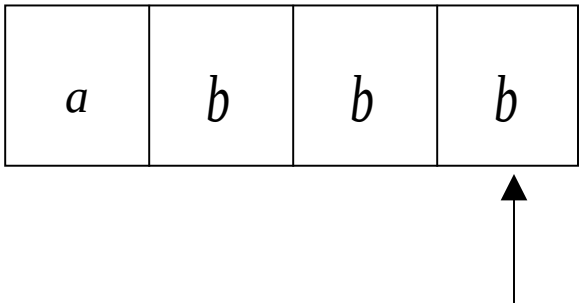


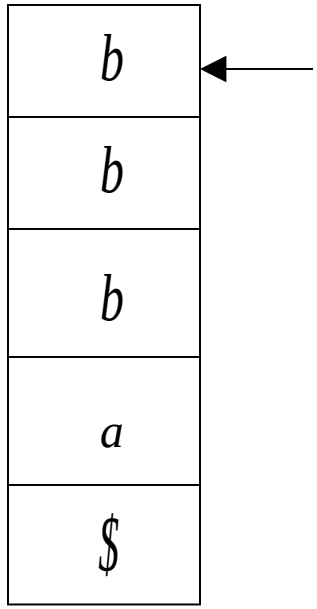
Stack



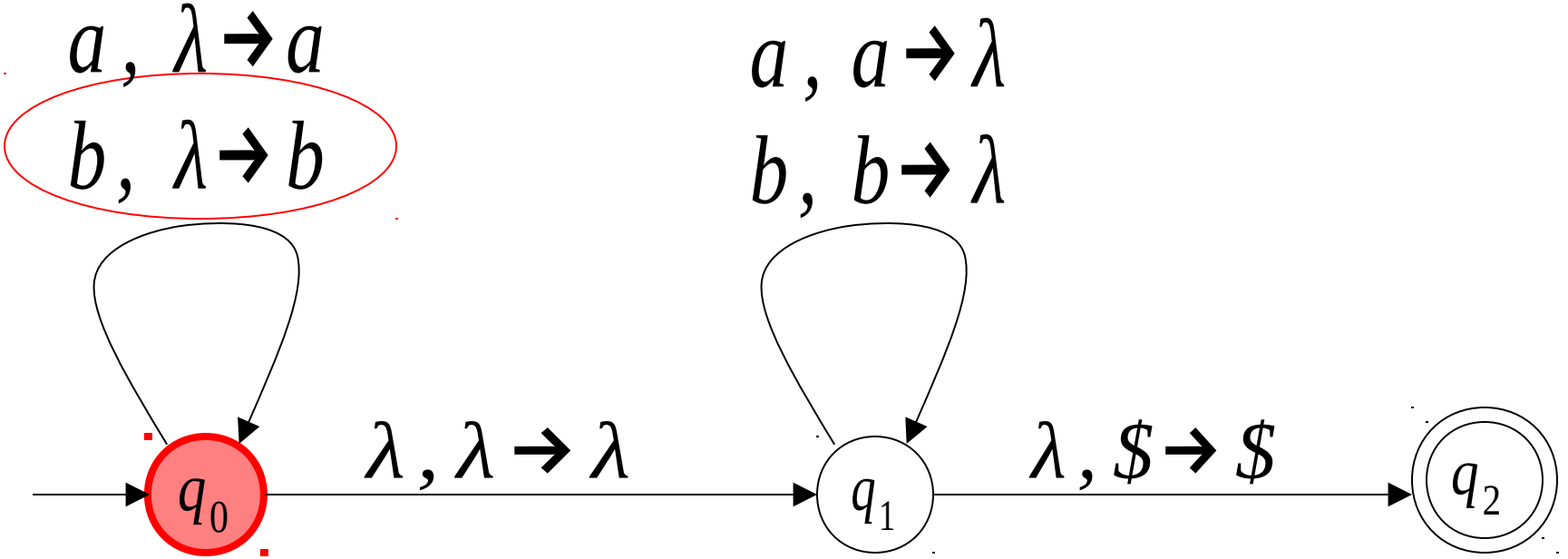
Time 4

Input



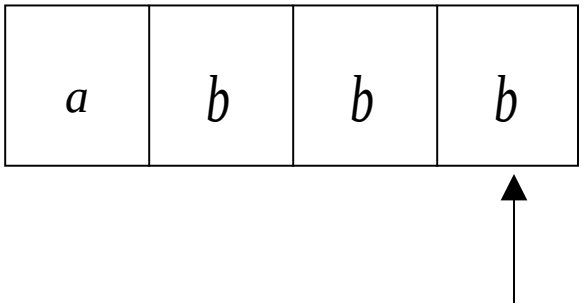


Stack

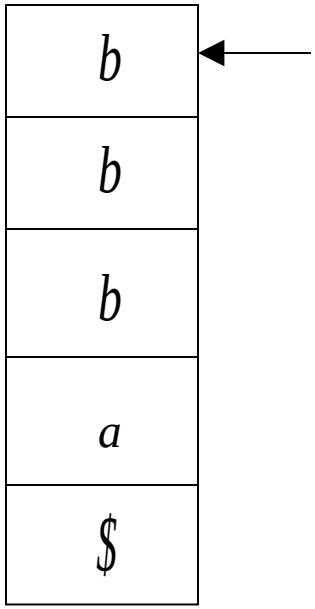


Time 5

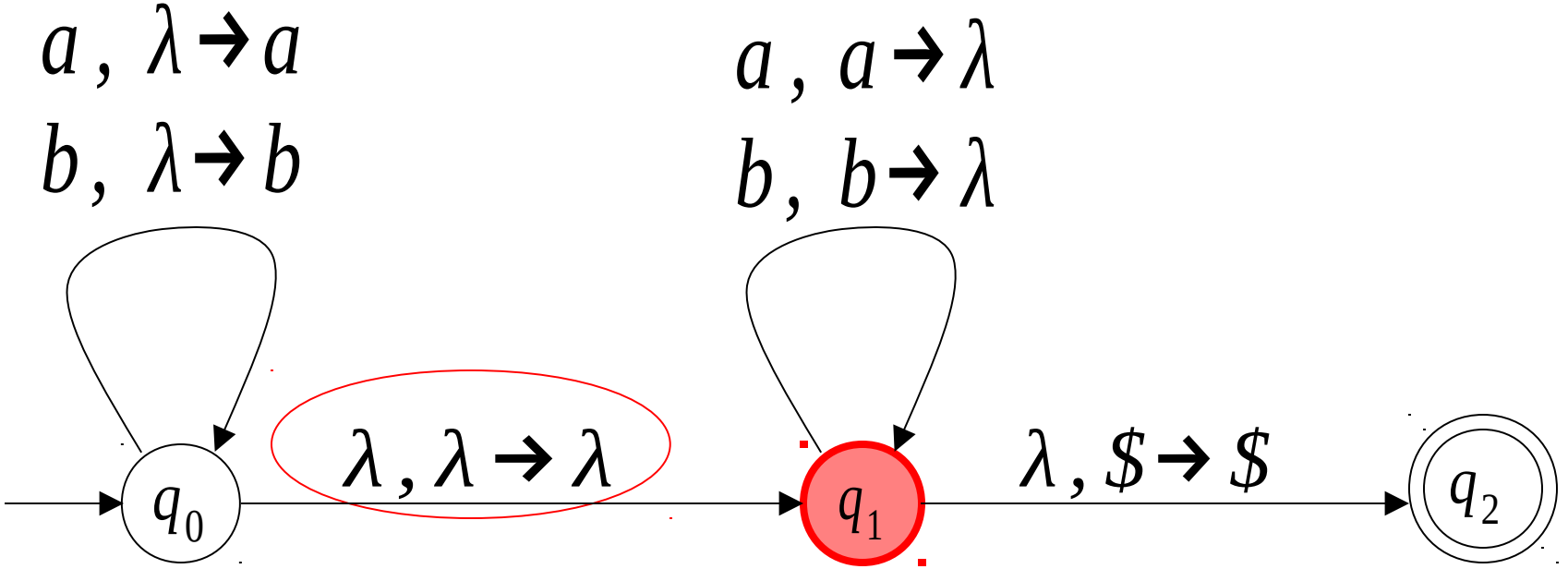
Input



No accept state is reached

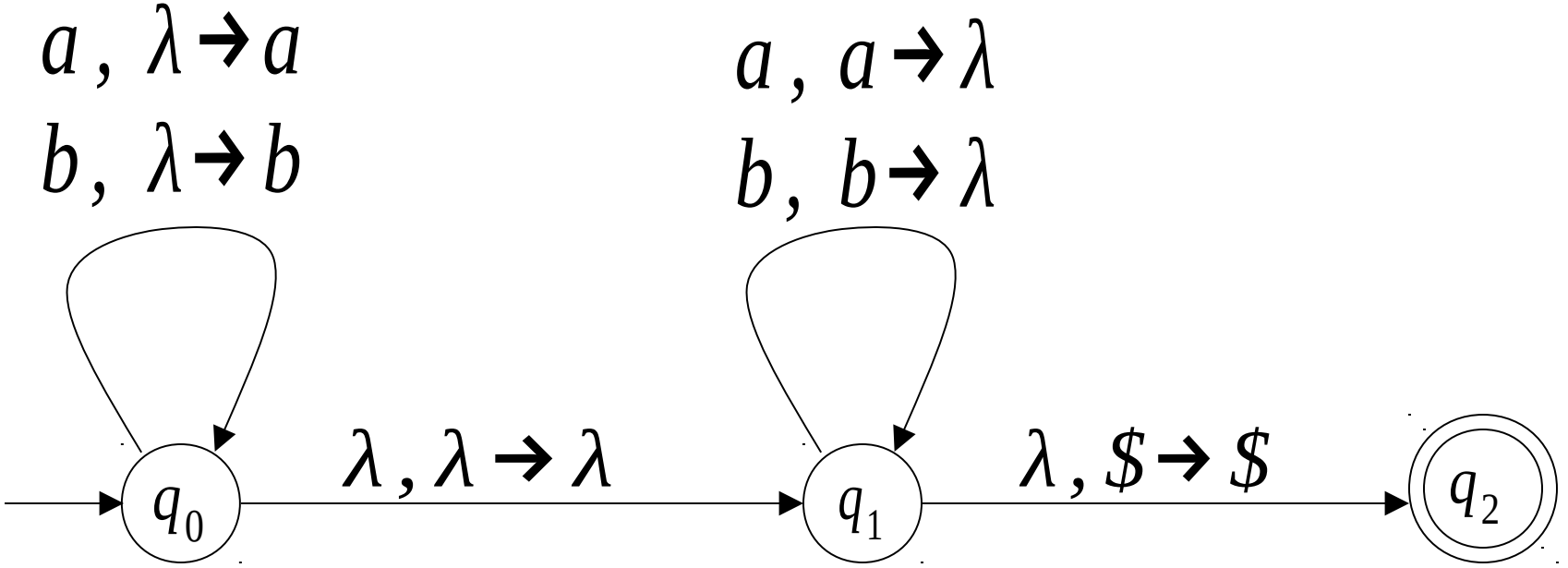


Stack

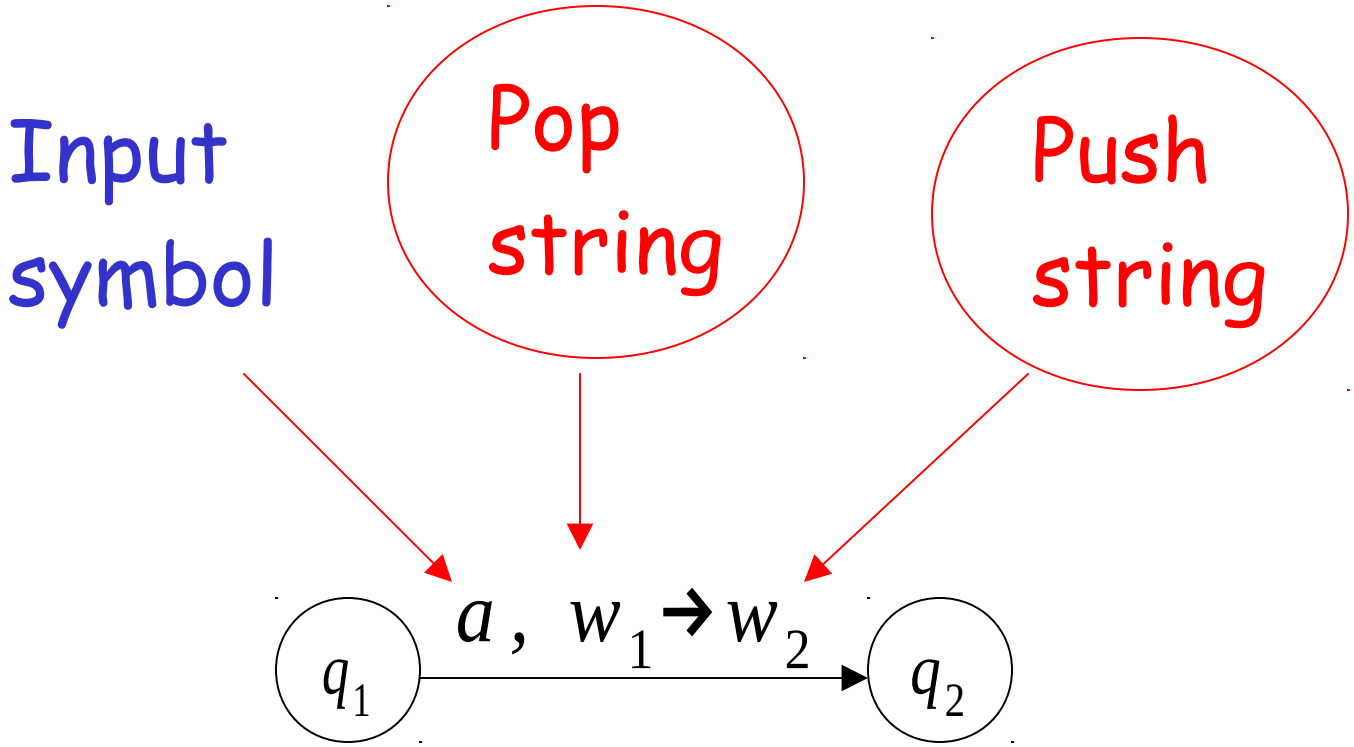


There is no computation that accepts string $abbb$

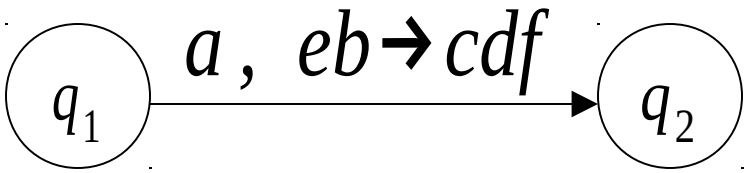
$$abbb \notin L(M)$$



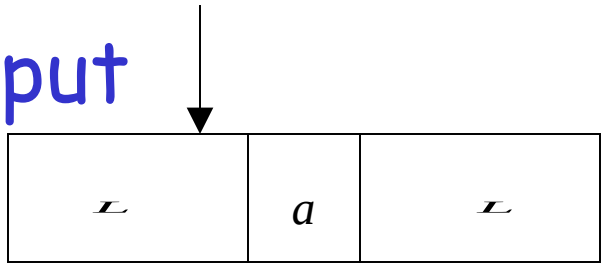
Pushing & Popping Strings

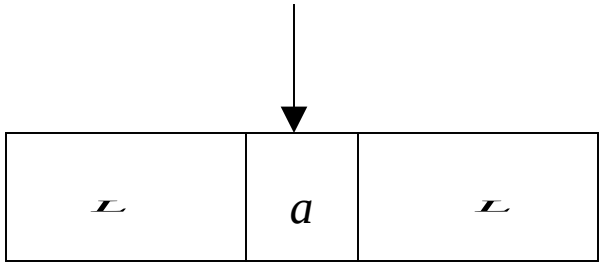


Example:



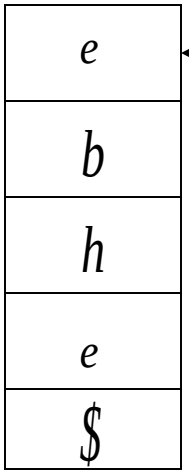
input



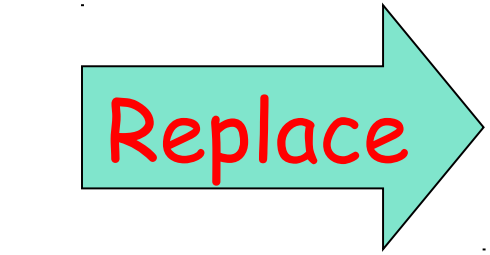


stack

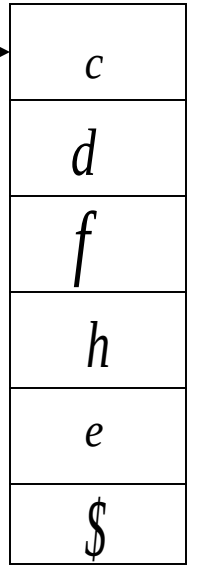
pop string



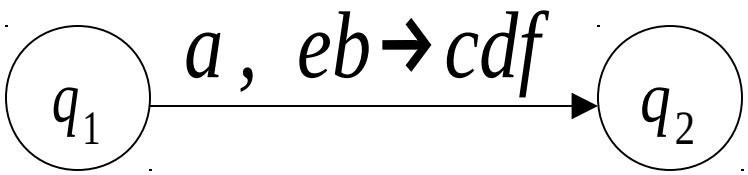
top

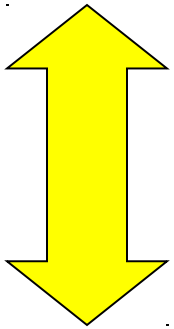


top



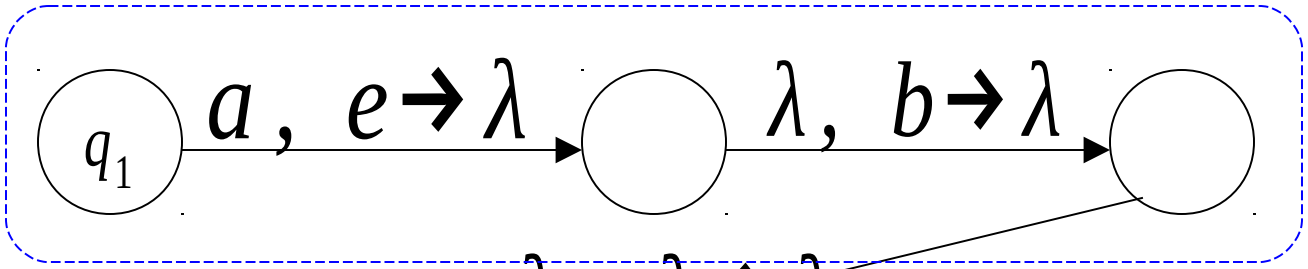
push string

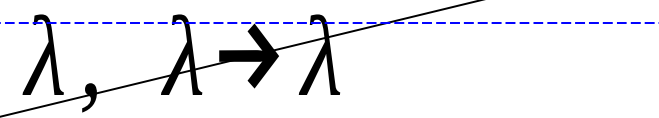




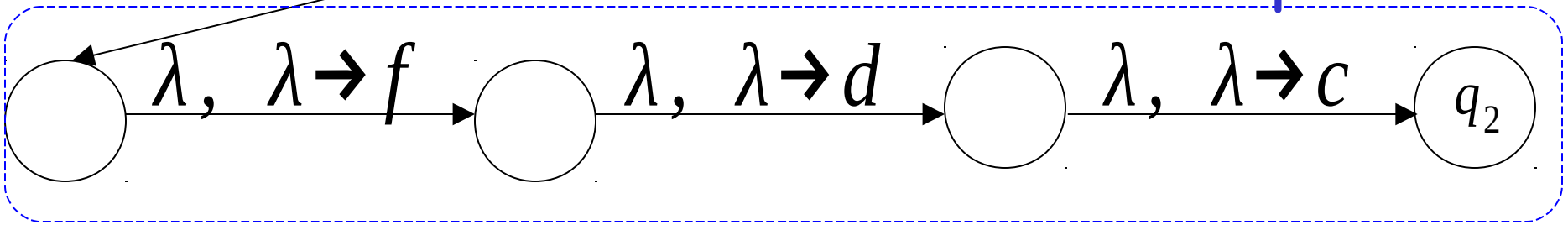
Equivalent transitions

pop





push



Another PDA example

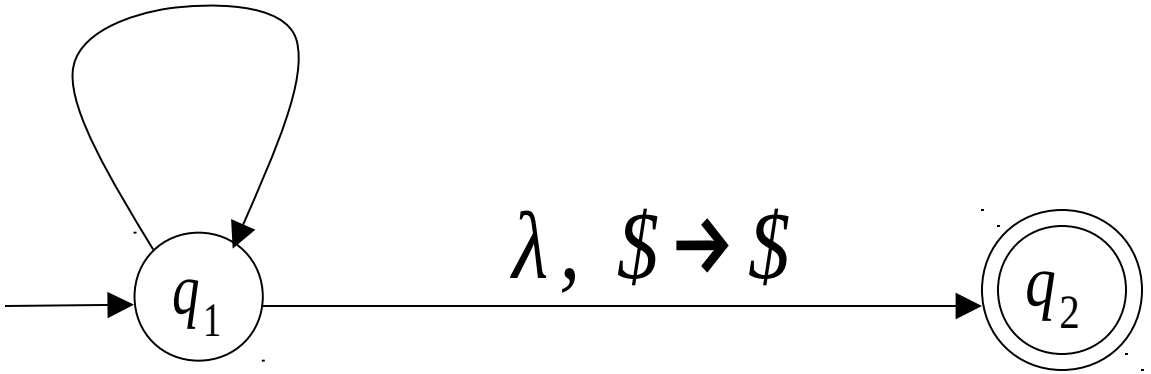
$$L(M)=\{w\in\{a,b\}^{¿}:\ n_a(w)=n_b(w)\}$$

PDA M

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

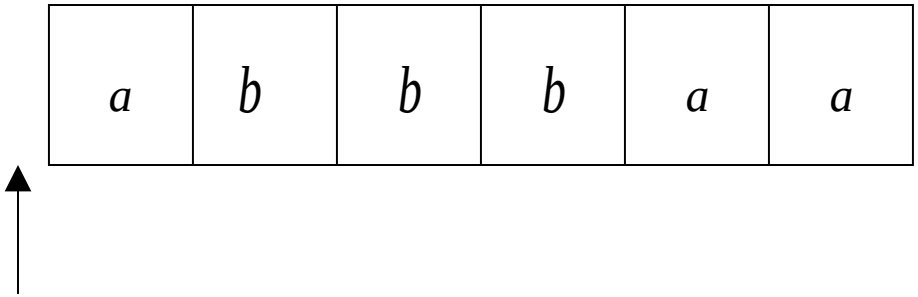
$a,0\rightarrow 00$ $\quad$ $b,1\rightarrow 11$

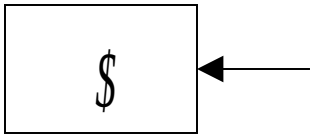
$a,1\rightarrow \lambda$ $\quad$ $b,0\rightarrow \lambda$



Execution Example: Time 0

Input





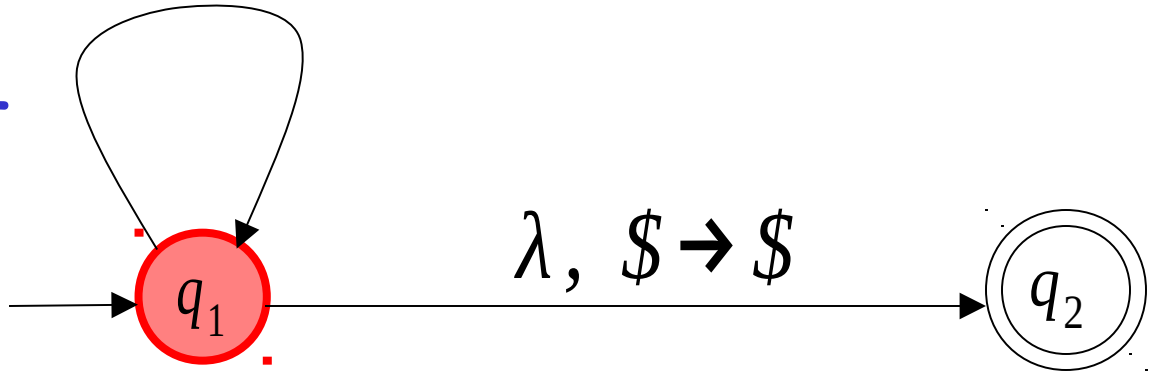
Stack

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

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

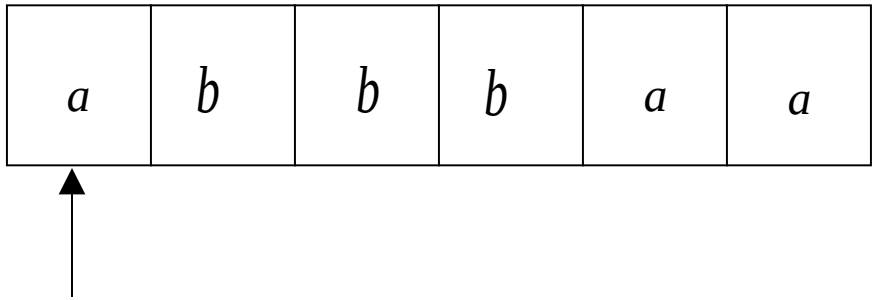
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$

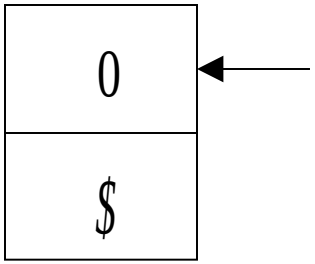
current state



Time 1

Input



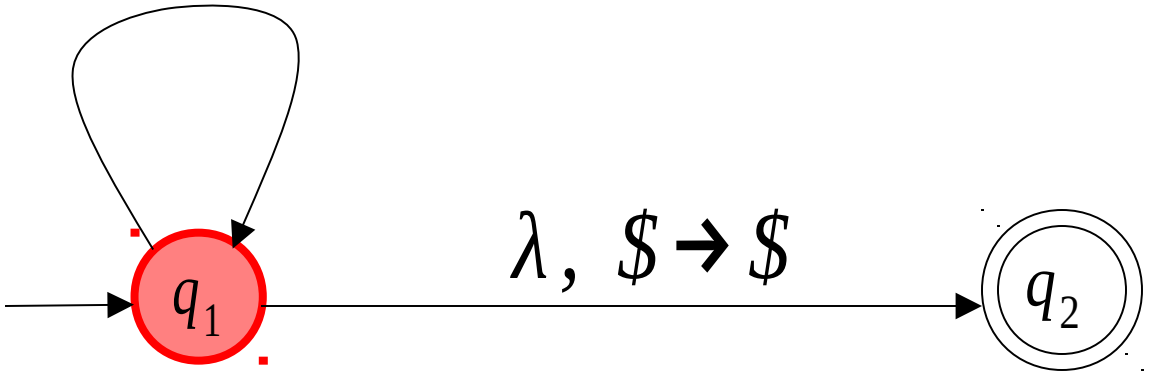


Stack

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

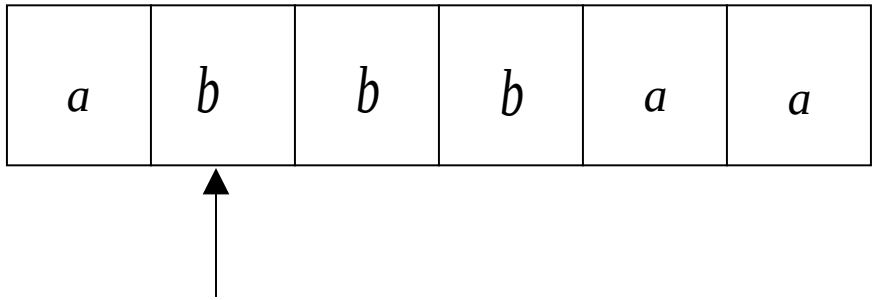
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

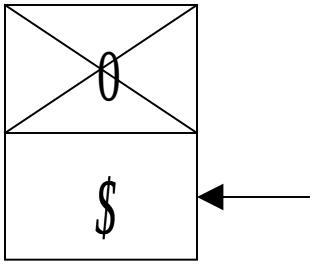
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 3

Input



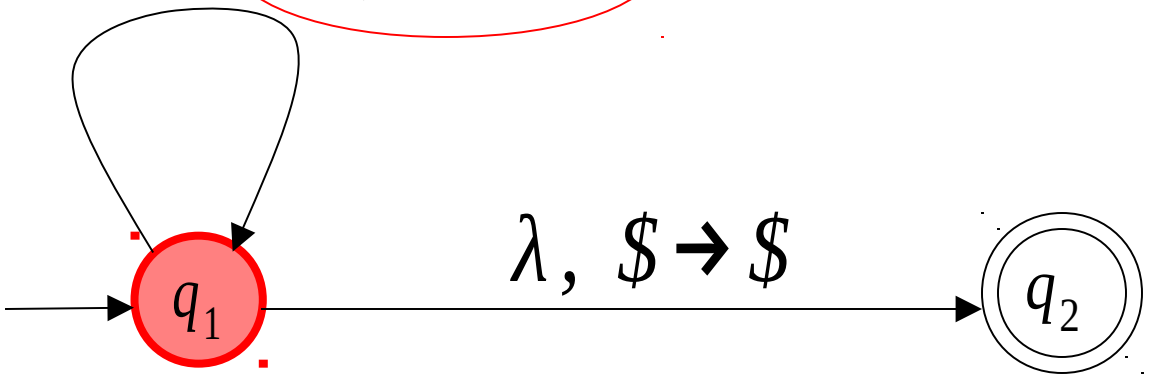


Stack

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

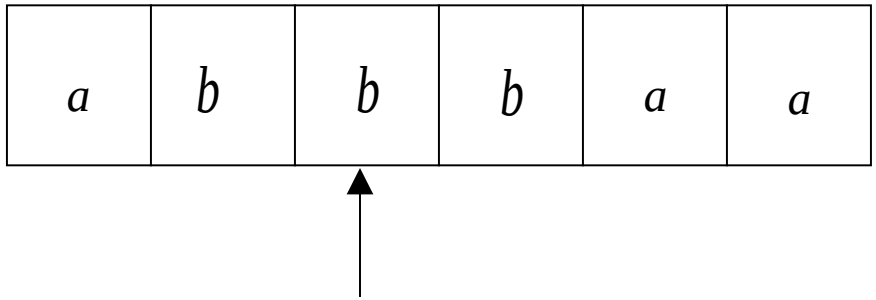
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

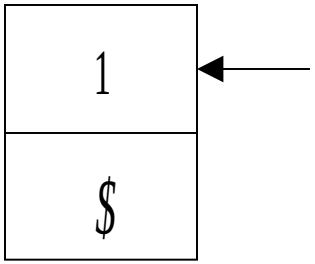
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 4

Input



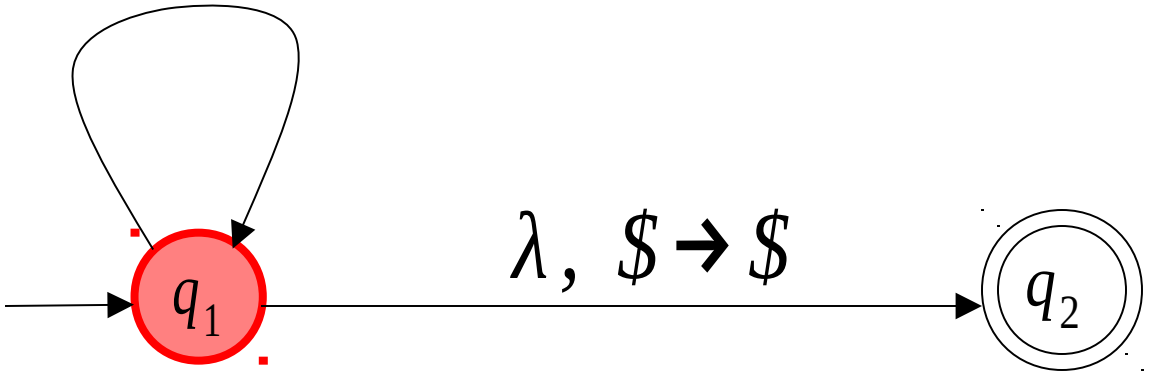


Stack

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

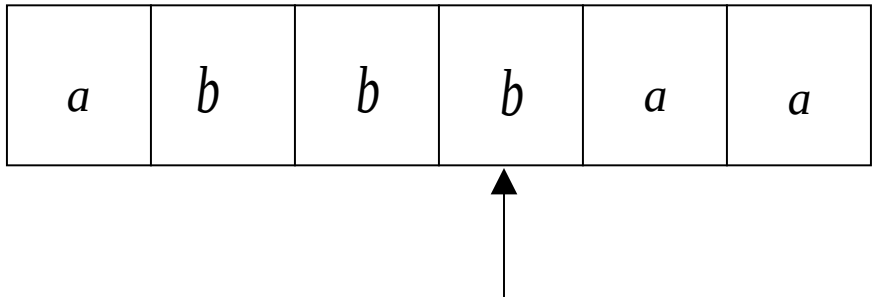
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

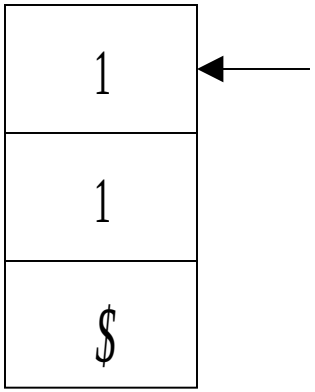
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 5

Input



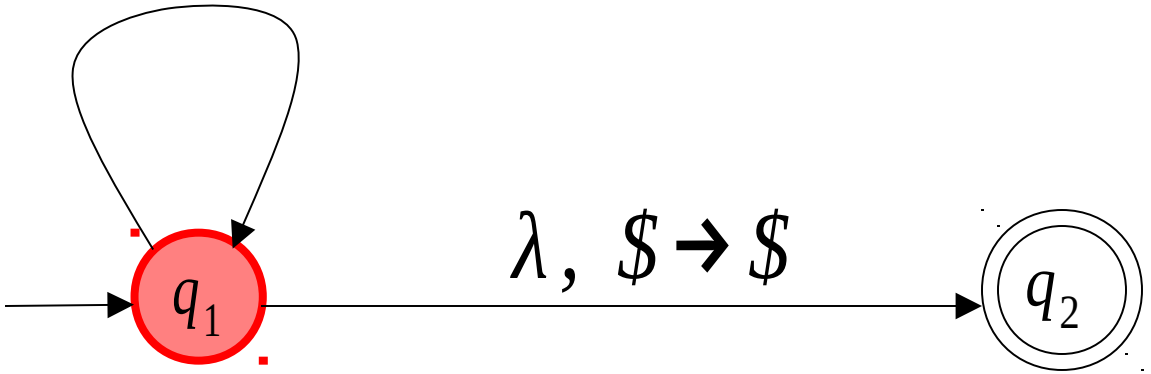


Stack

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

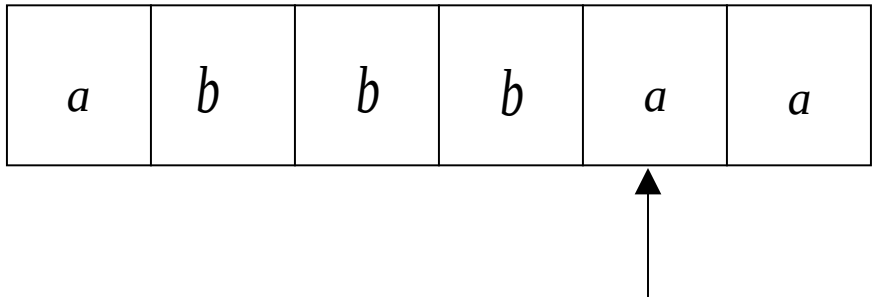
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

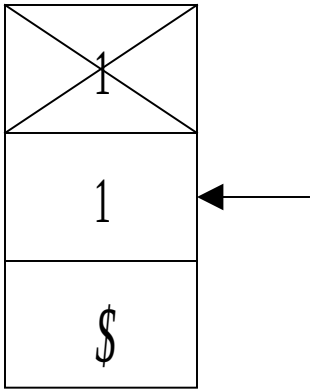
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 6

Input



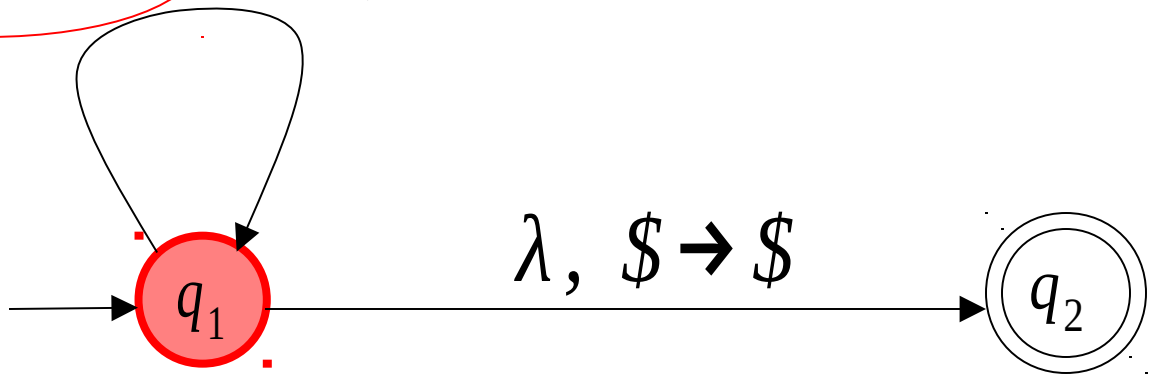


Stack

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

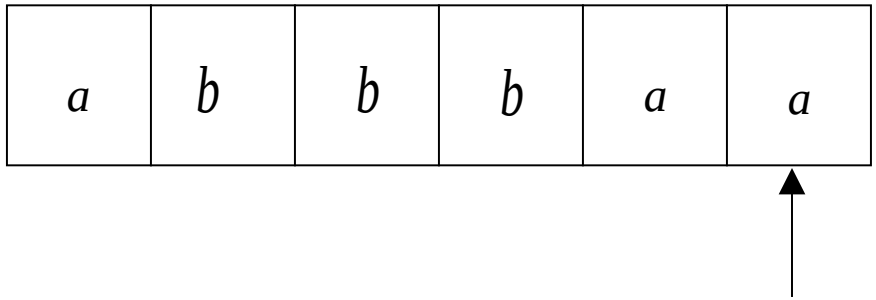
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

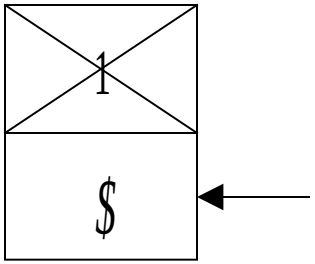
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 7

Input



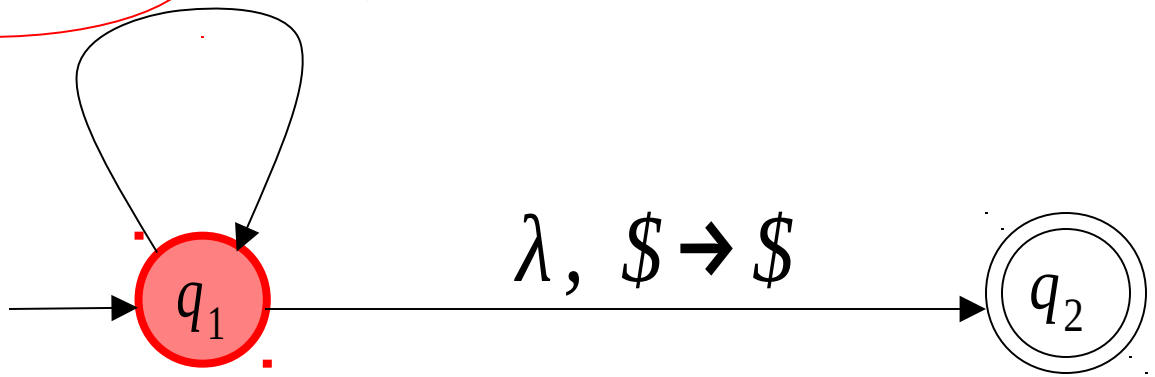


Stack

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

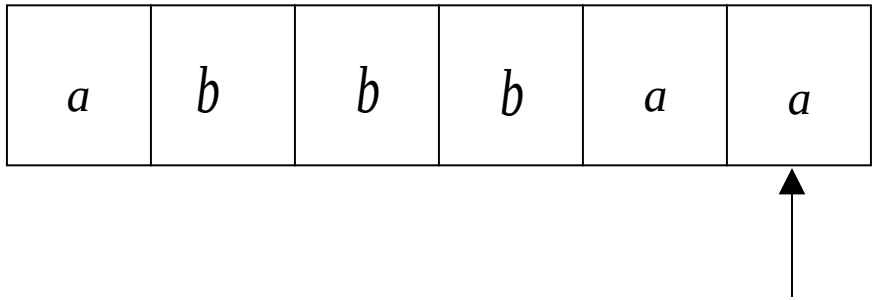
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

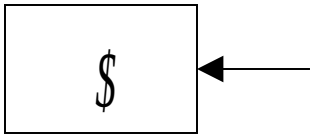
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 8

Input



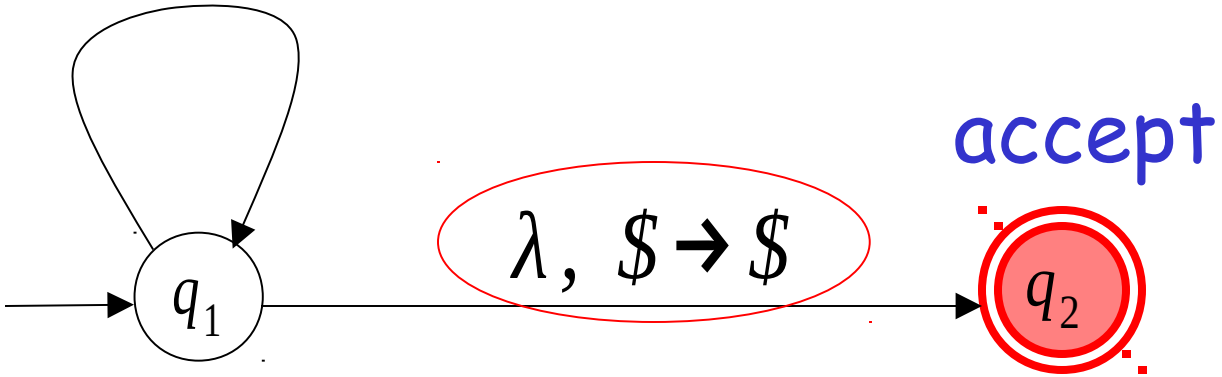


Stack

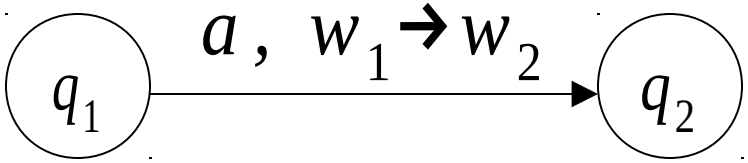
$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

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

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

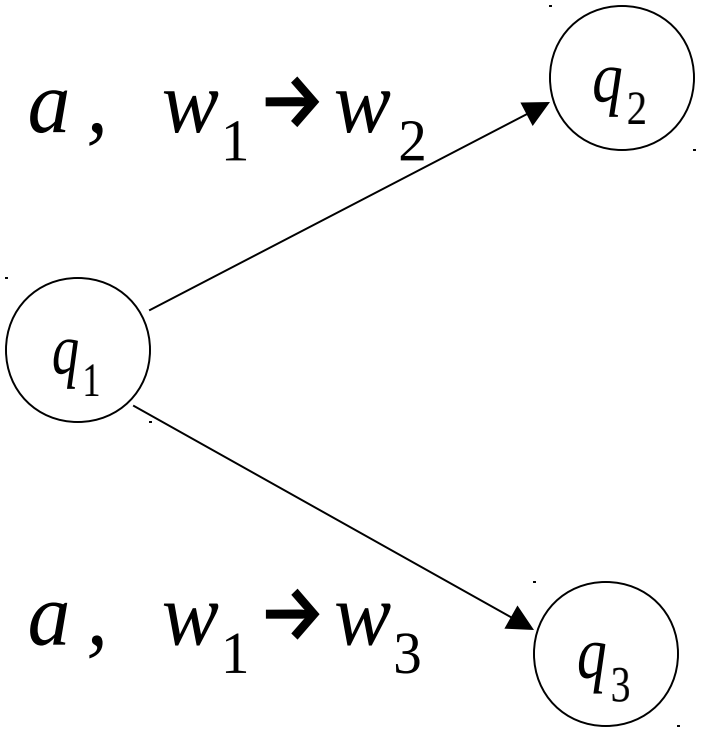


Formalities for PDAs



Transition function:

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

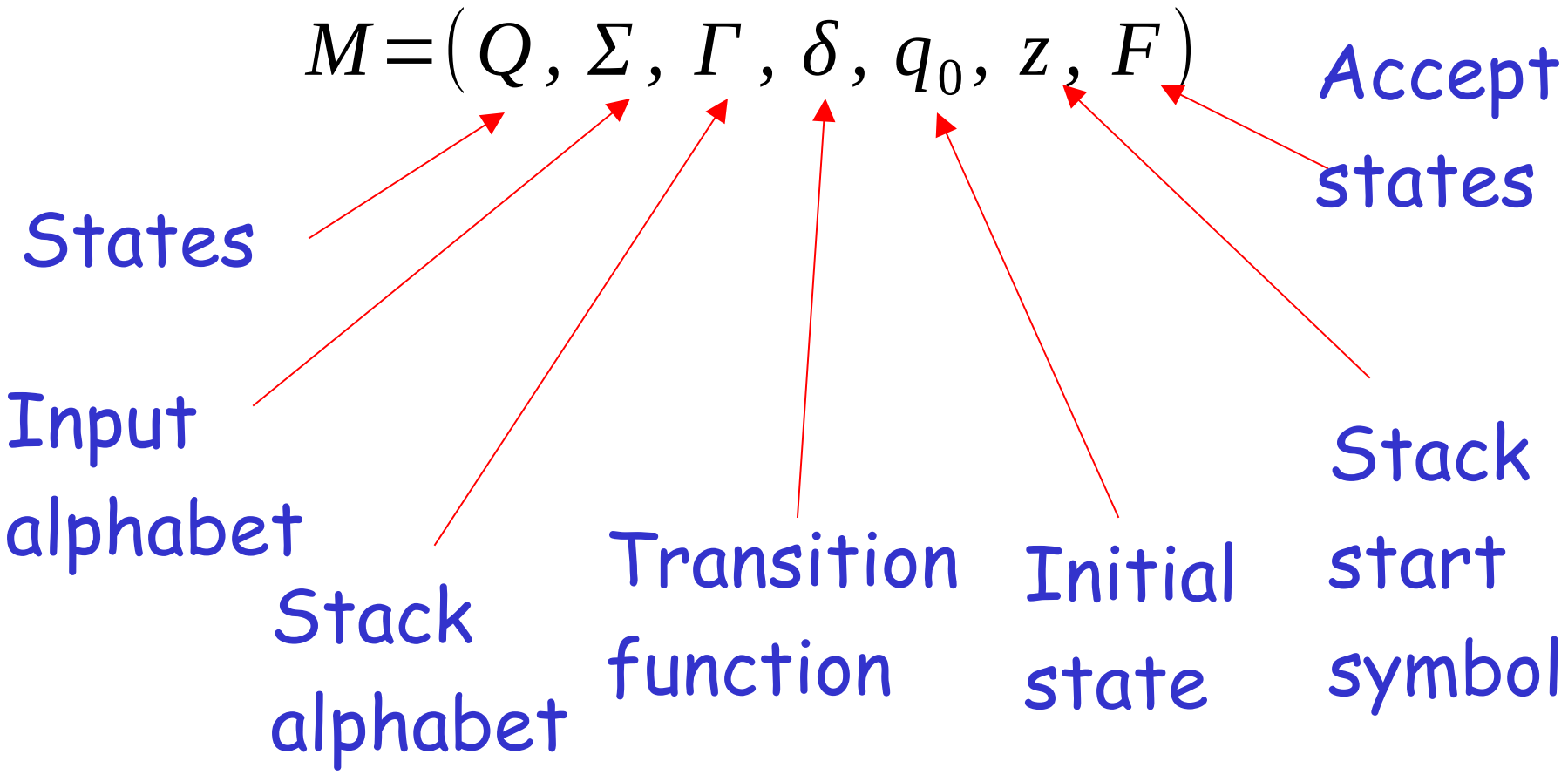


Transition function:

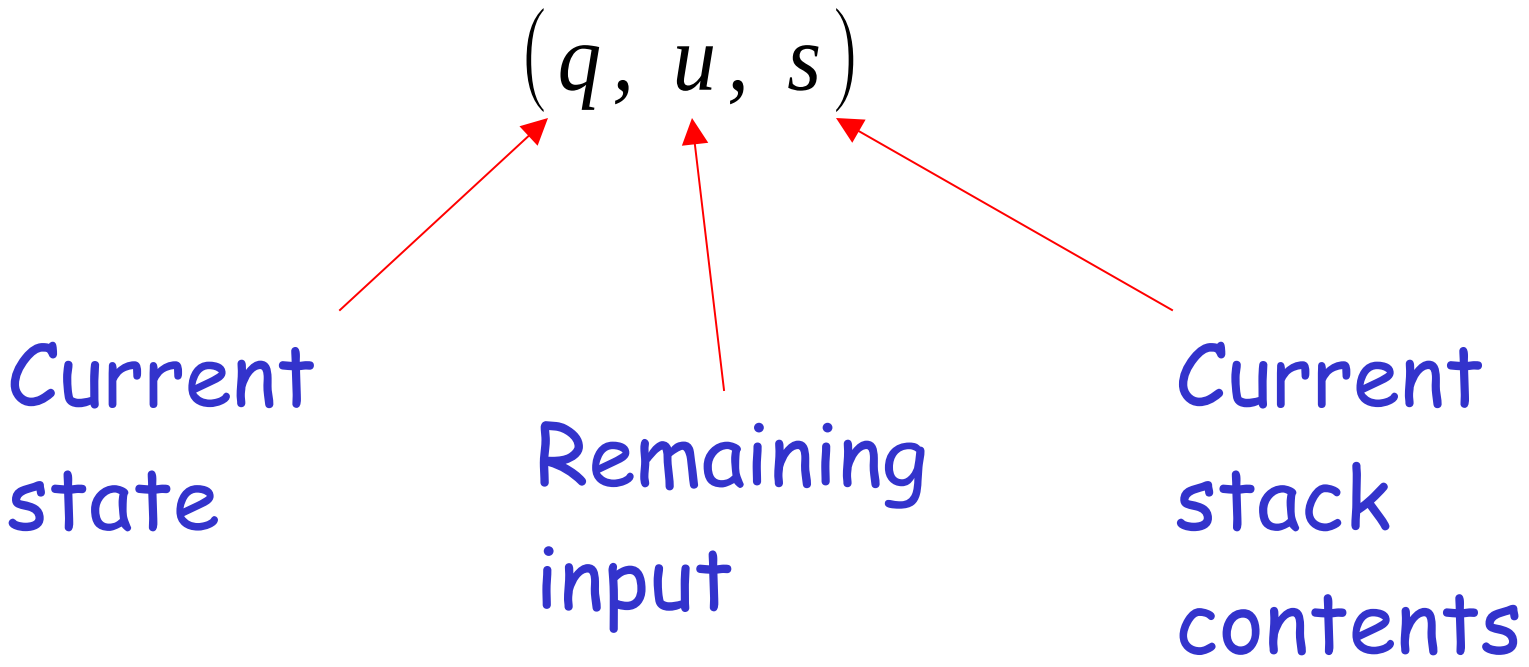
$$\delta(q_1, a, w_1) = \{(q_2, w_2), (q_3, w_3)\}$$

Formal Definition

Pushdown Automaton (PDA)



Instantaneous Description

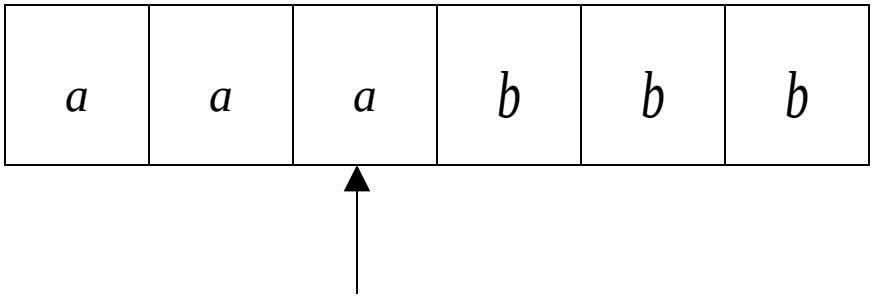


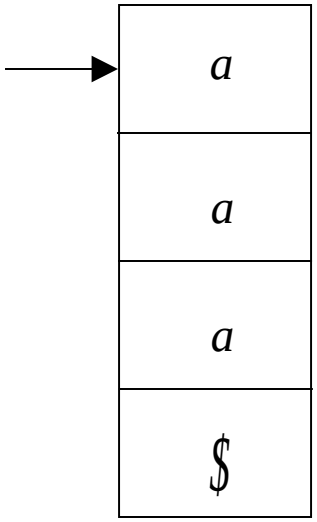
Example: **Instantaneous Description**

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

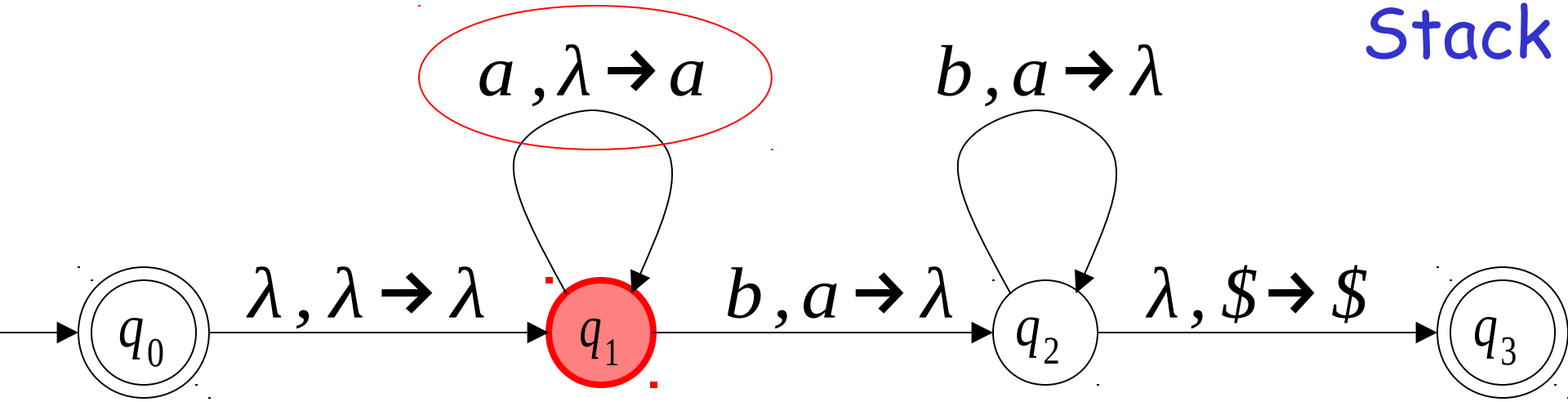
Time 4:

Input





Stack

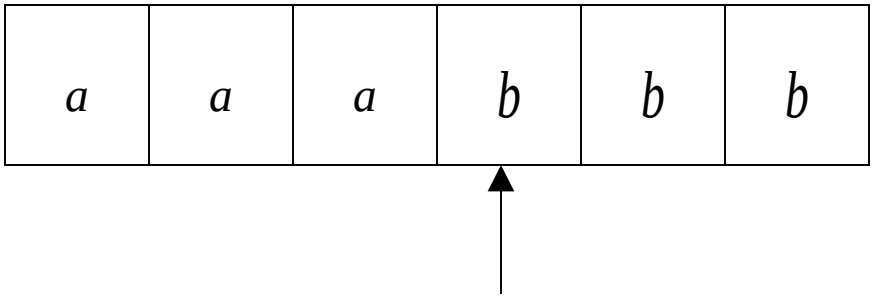


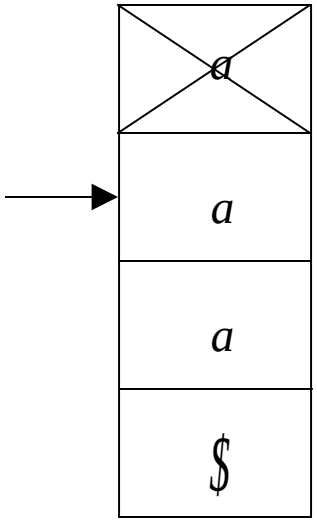
Example: **Instantaneous Description**

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

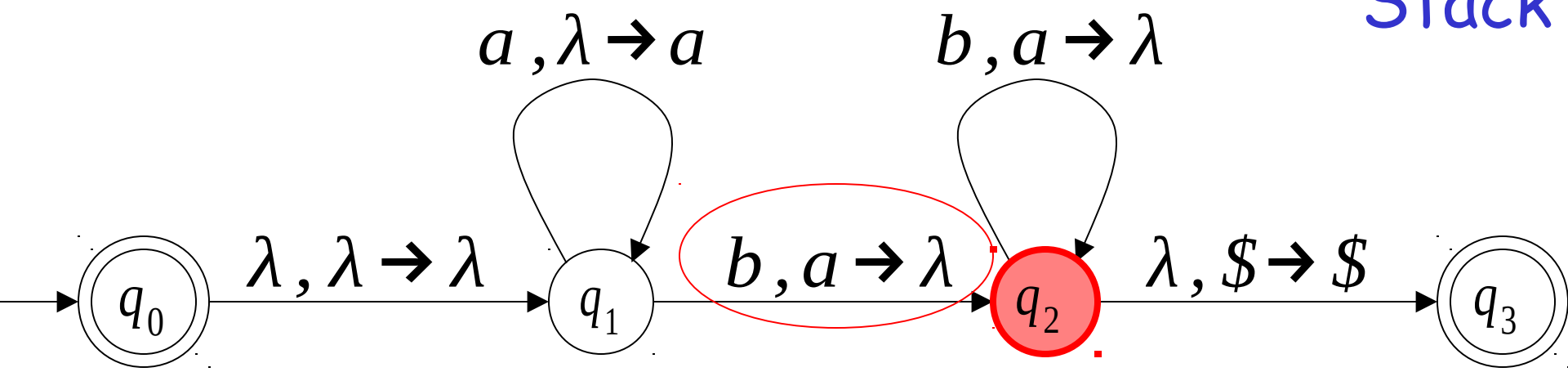
Time 5:

Input





Stack



We write:

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

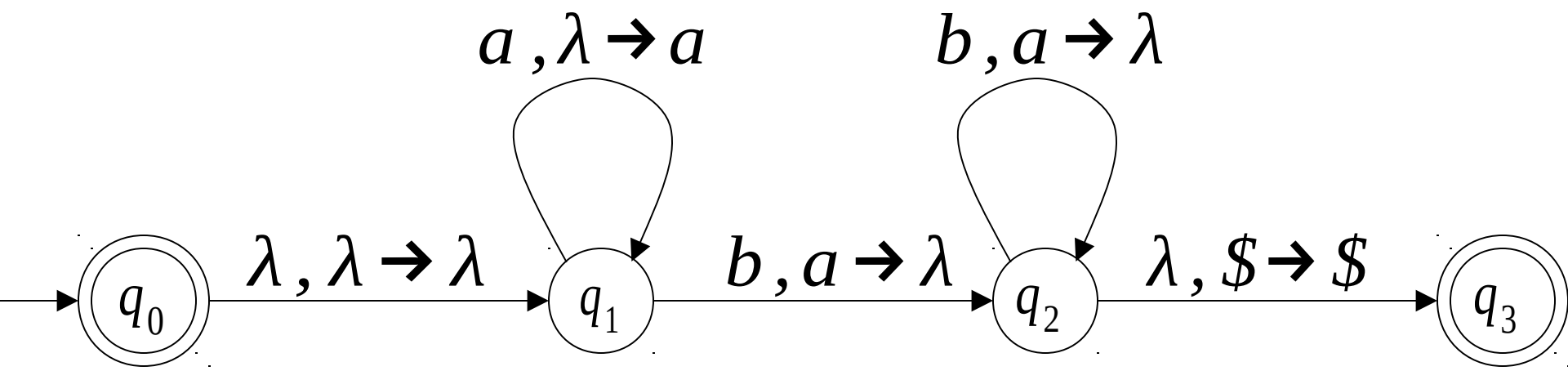
Time 4 Time 5

A computation:

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

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

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



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

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

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

For convenience we write:

$$(q_0, aaabbb, \$) \text{ ¿ } (q_3, \lambda, \$)$$

Language of PDA

Language $L(M)$ accepted by PDA M :

$$L(M)=\{w:\ (q_0,w,z)\ \text{¿}^{*}\ (q_f,\lambda,s)\}$$

Initial state

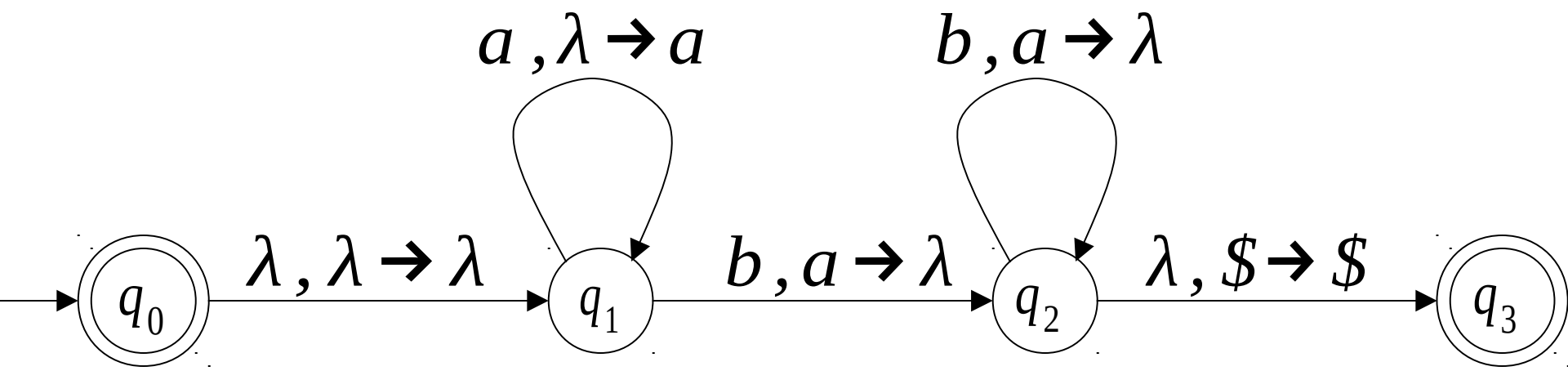
Accept state

Example:

$$(q_0, aaabbb, \$) \text{ ¿ } (q_3, \lambda, \$)$$

$$aaabbb \in L(M)$$

PDA M :

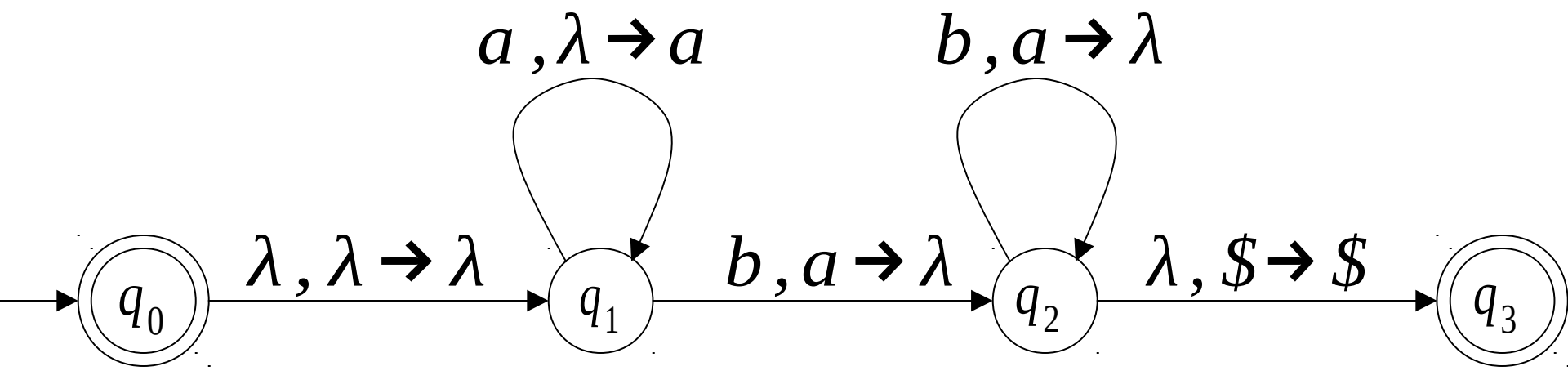


$$(q_0, a^n b^n, \$) \text{ ¿} (q_3, \lambda, \$)$$

$$\Downarrow$$

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

PDA M :



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

PDA M :