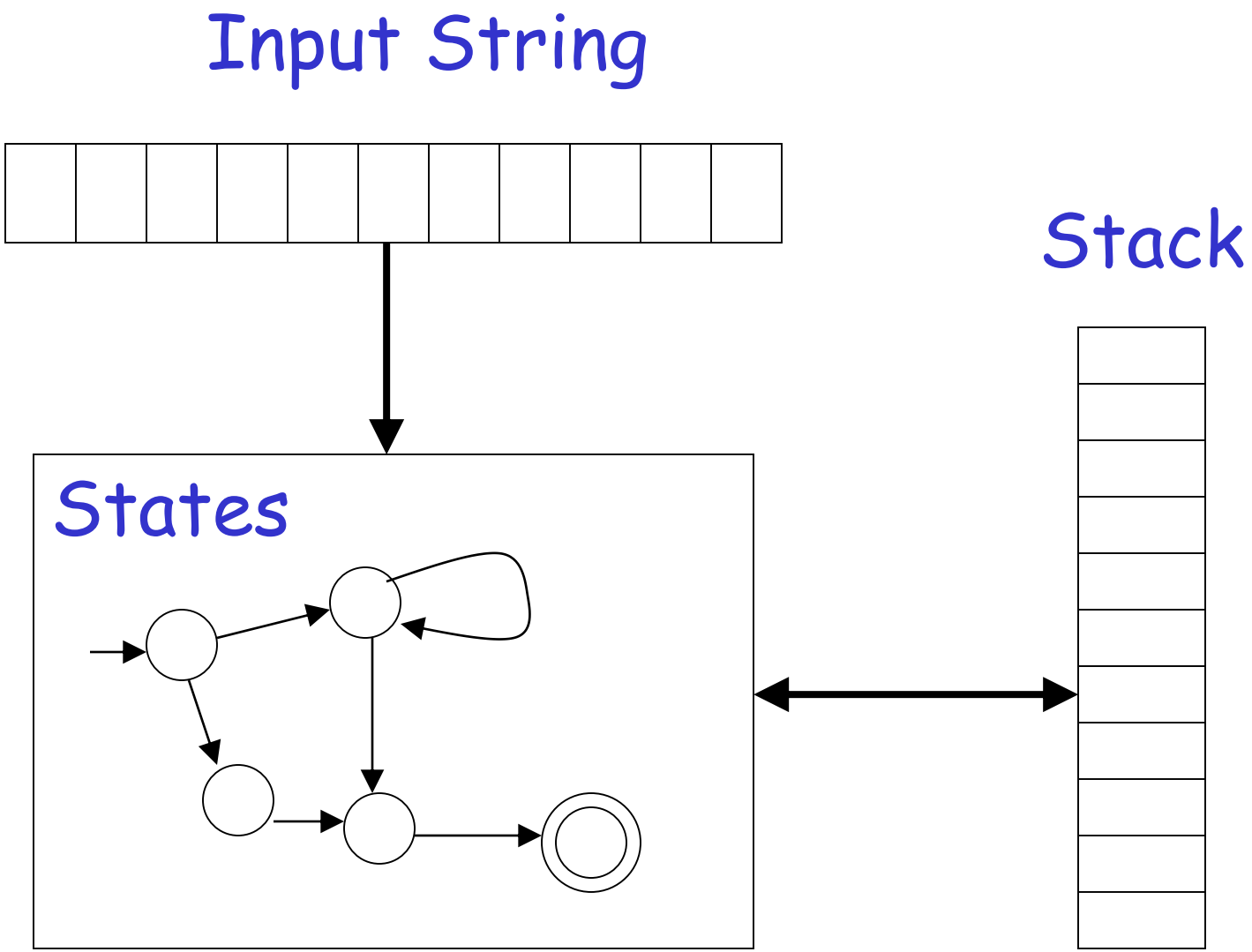
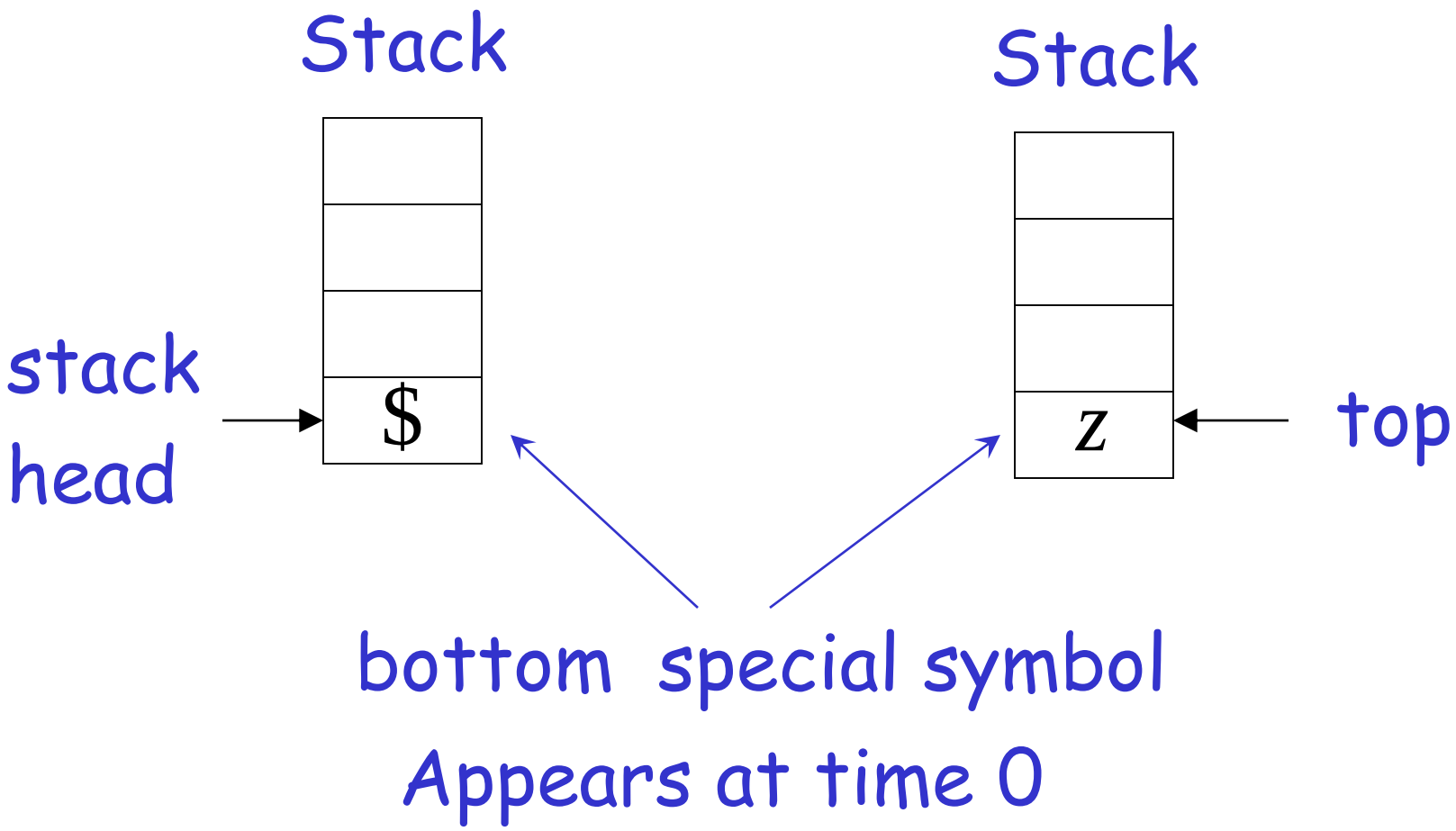
Pushdown Automata
PDAs

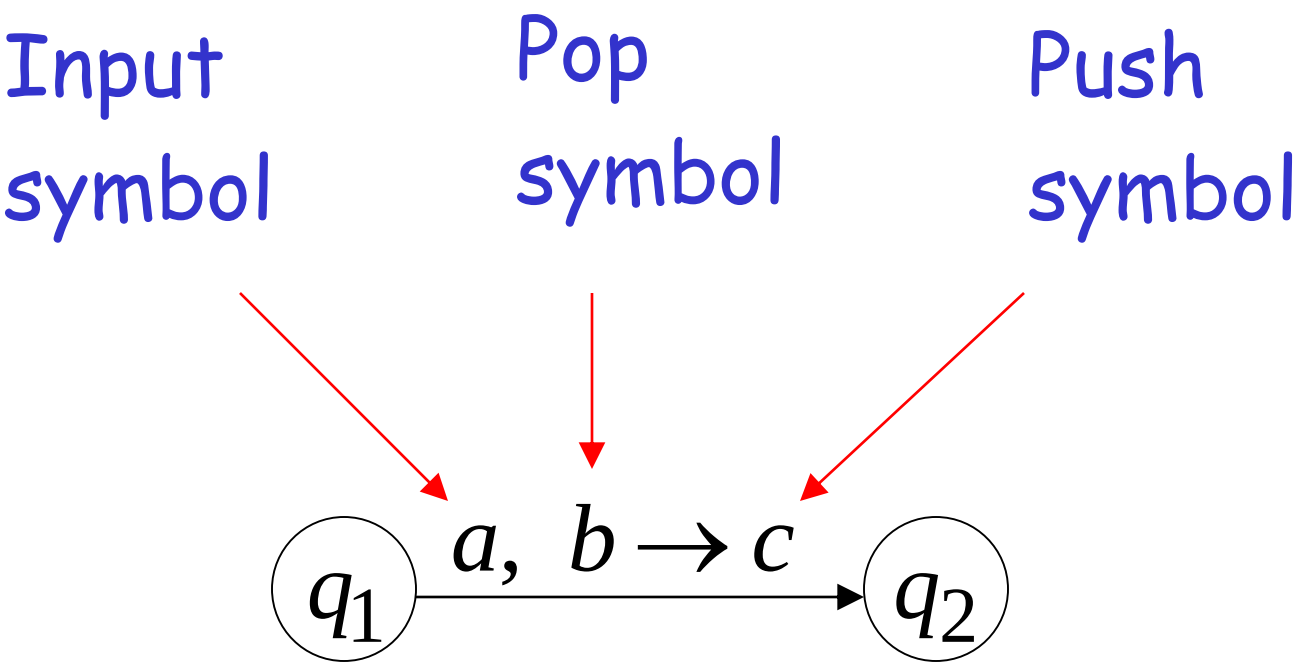
Pushdown Automaton -- PDA

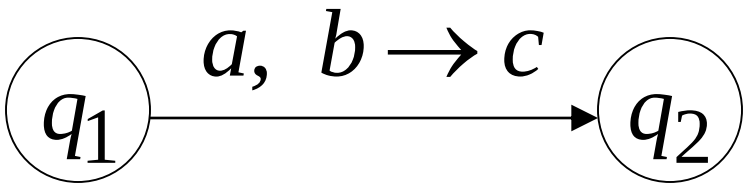


Initial Stack Symbol

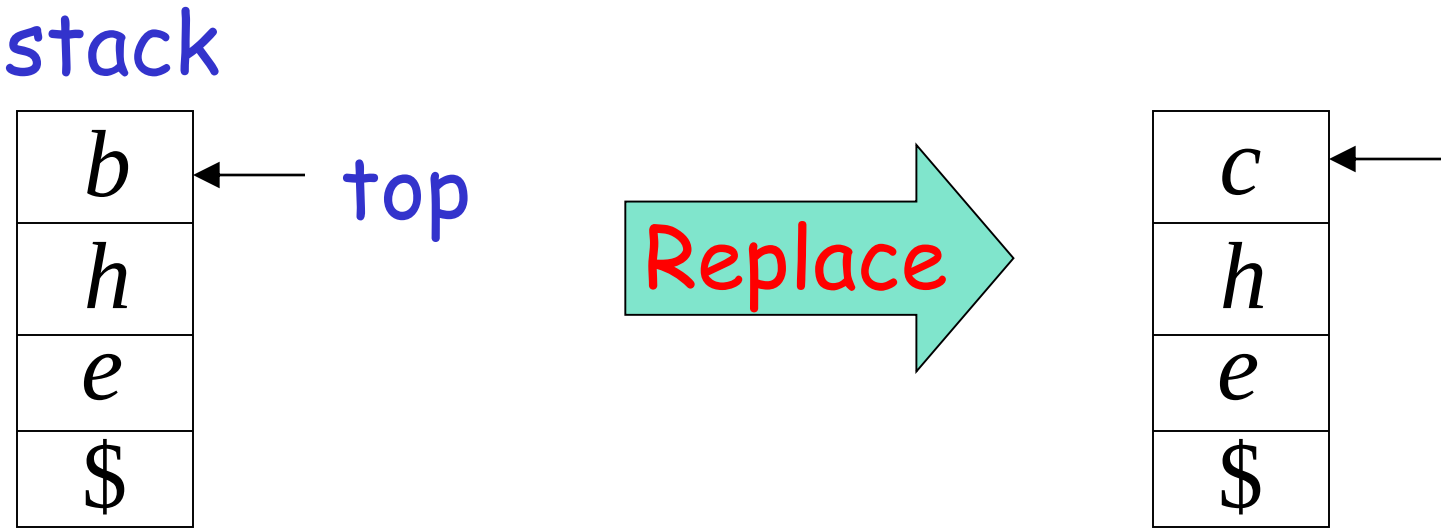


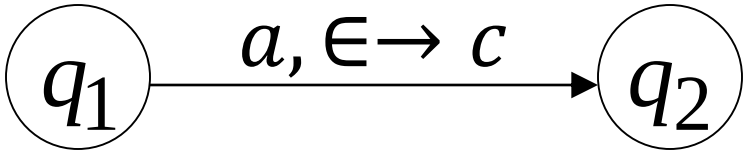
The States



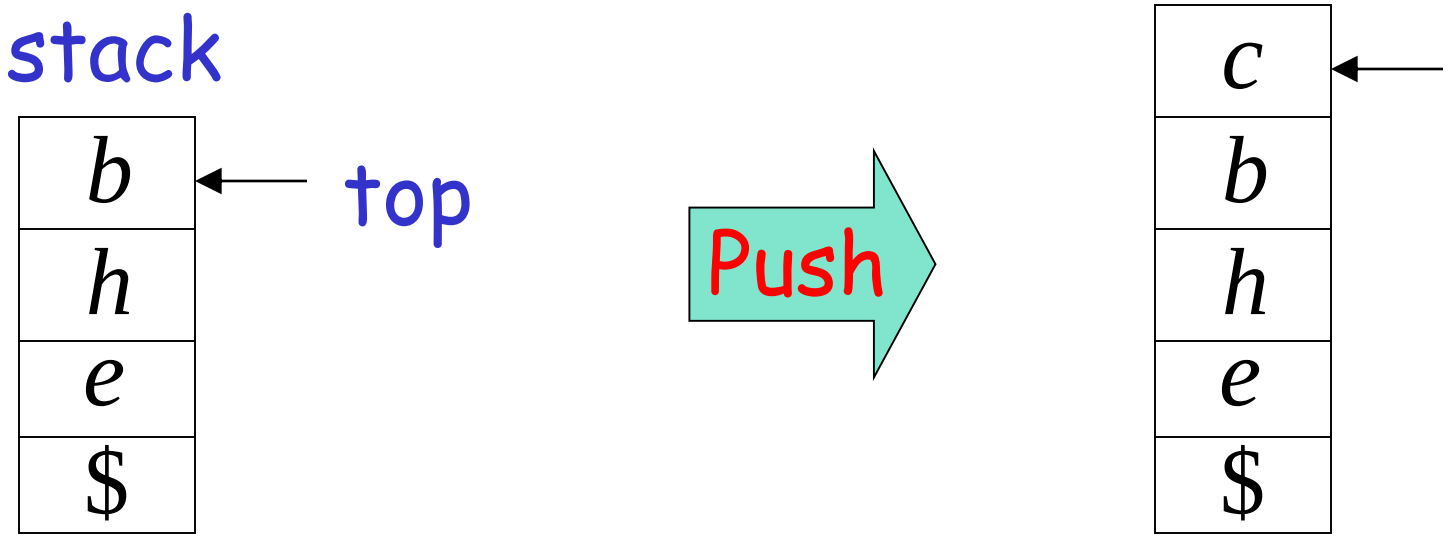

$a,\ b \to c$
q_1
q_2

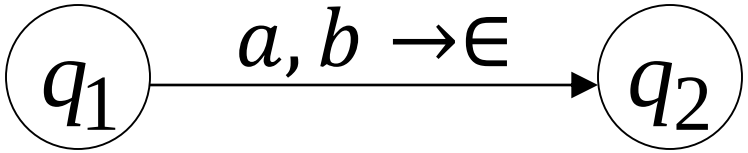

input
...
a
...
...
a
...


stack
b
h
e
$
top
Replace
c
h
e
$

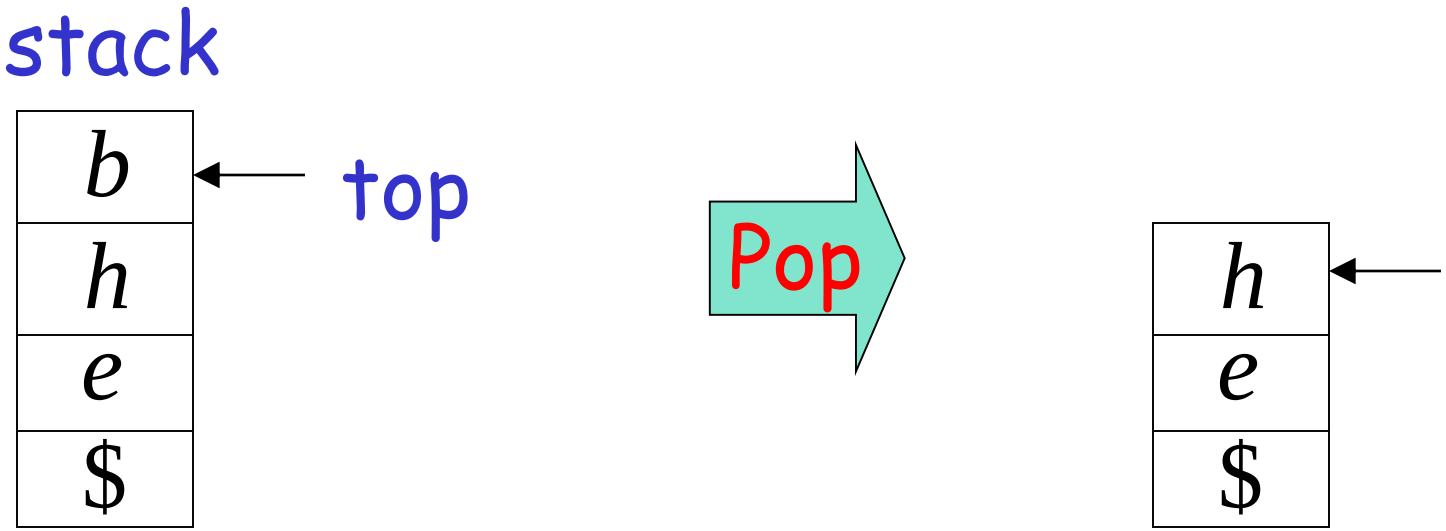

q_1
$a, \in \rightarrow c$
q_2

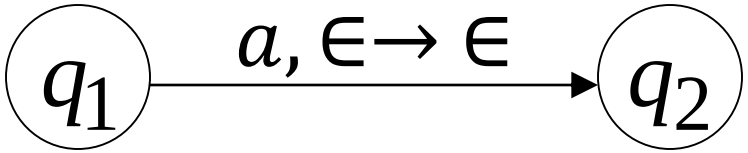

input
. . .
a
. . .
. . .
a
. . .


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

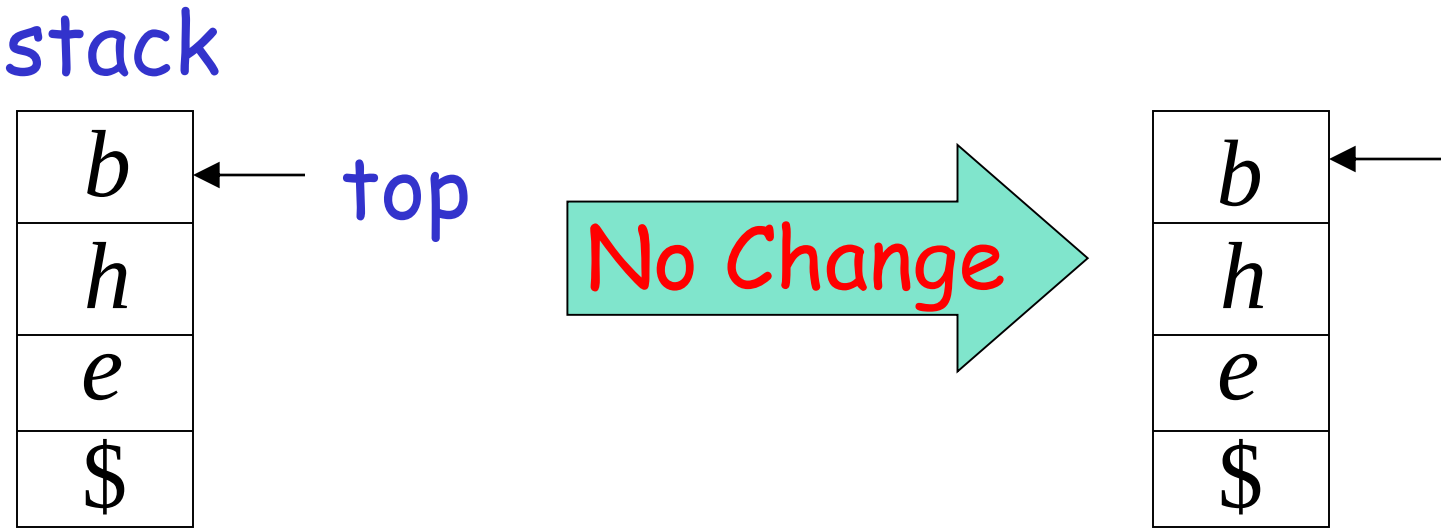

q_1
$a, b \rightarrow \in$
q_2


input
. . .
a
. . .
. . .
a
. . .

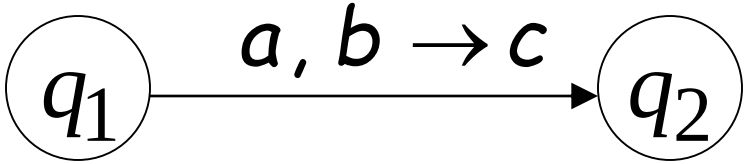

stack
b
top
h
e
$
Pop
h
e
$

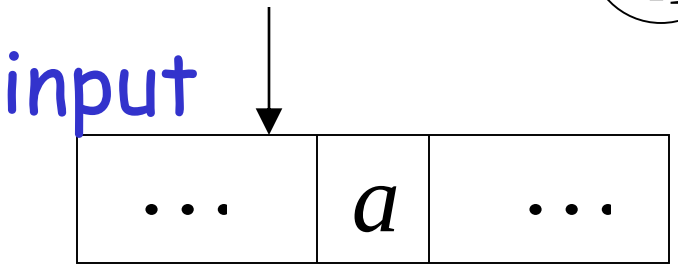

q_1
$a, \in \rightarrow \in$
q_2


input
. . .
a
. . .
. . .
a
. . .

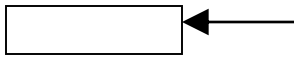

stack
b
h
e
$
top
No Change
b
h
e
$

Pop from Empty Stack





stack



Pop

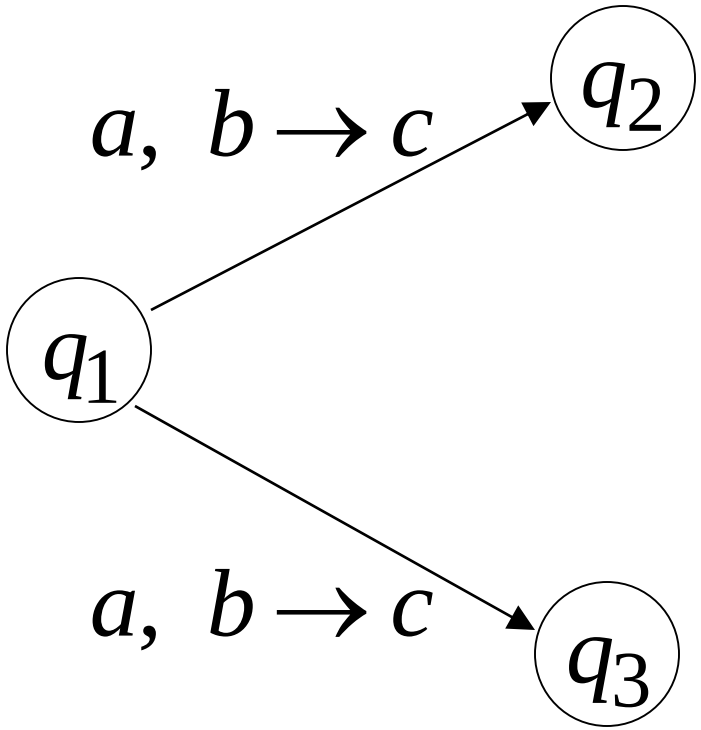
Automaton halts!

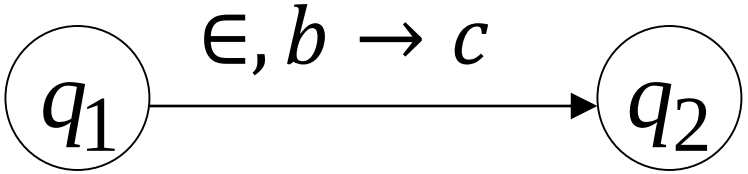
If the automaton attempts to pop from empty stack then it halts and rejects input

Non-Determinism

PDAs are non-deterministic

Allowed non-deterministic transitions

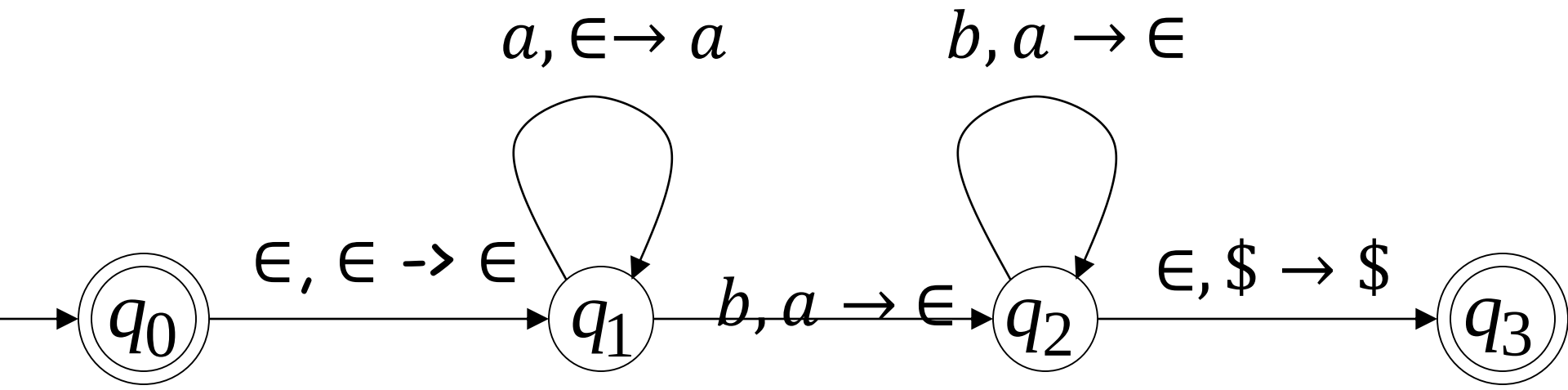




$\in$- transition

Example PDA

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



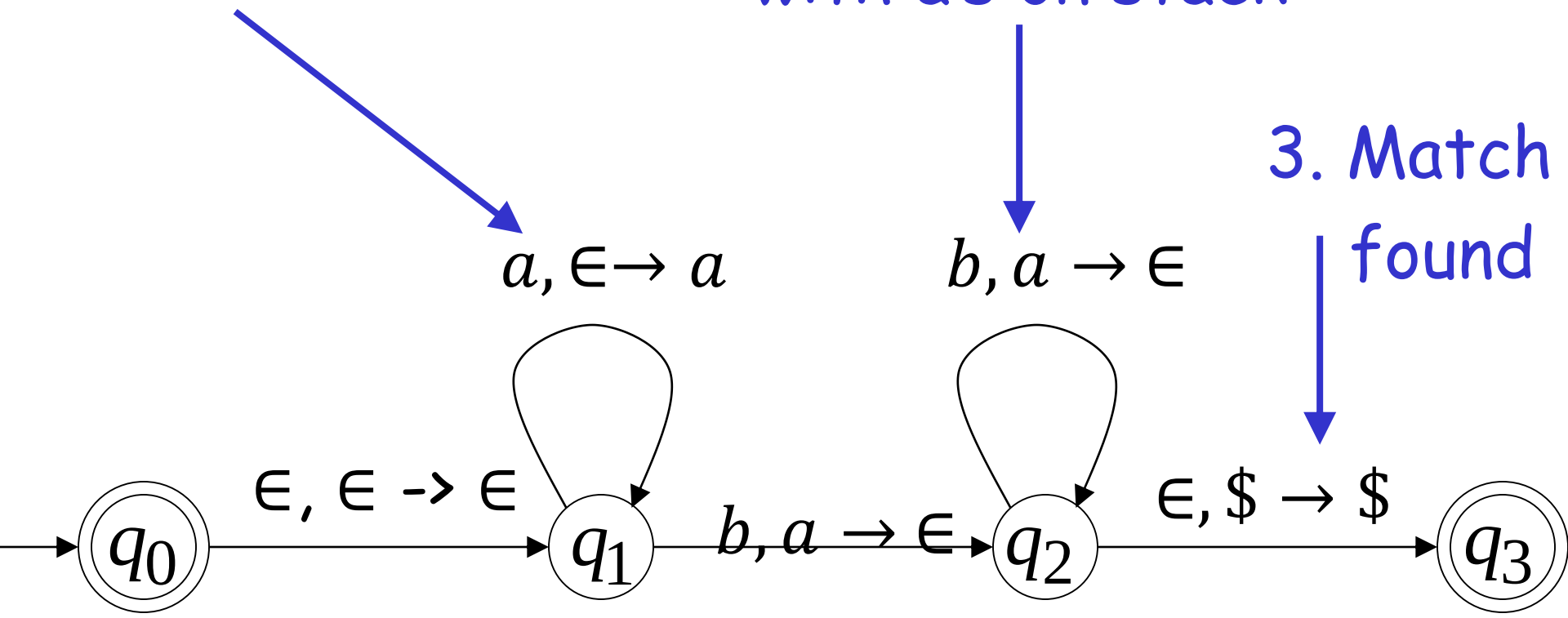
$$L(M) = \{a^n b^n : n \geq 0\}$$

Basic Idea:

1. Push the a's on the stack

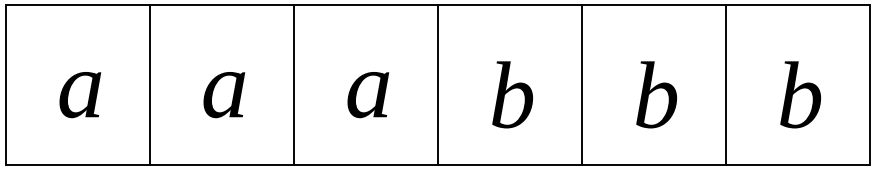
2. Match the b's on input with a's on stack

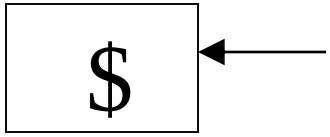
3. Match found



Execution Example: Time 0

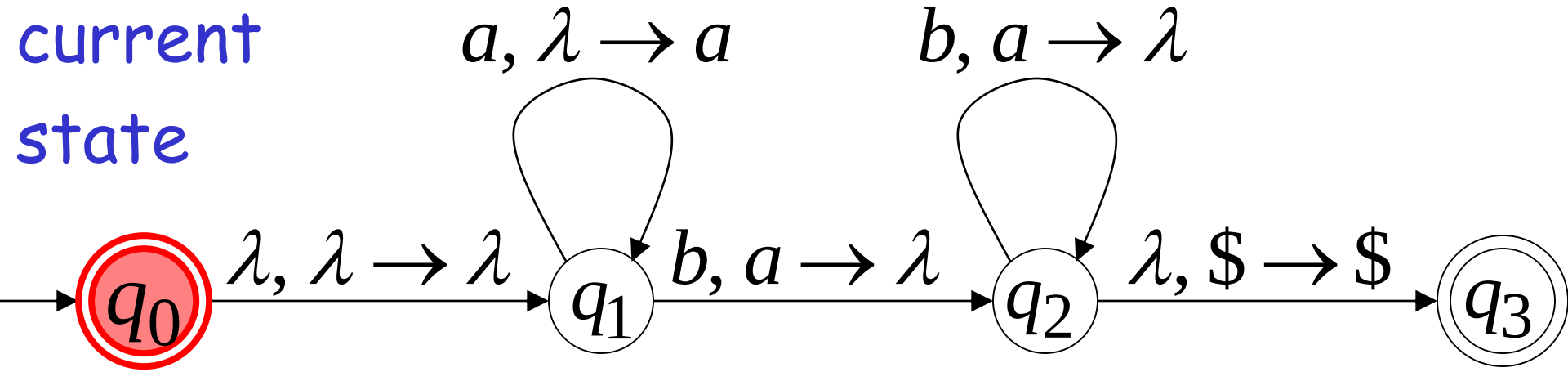
Input





Stack

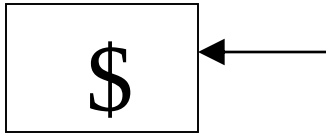
current state



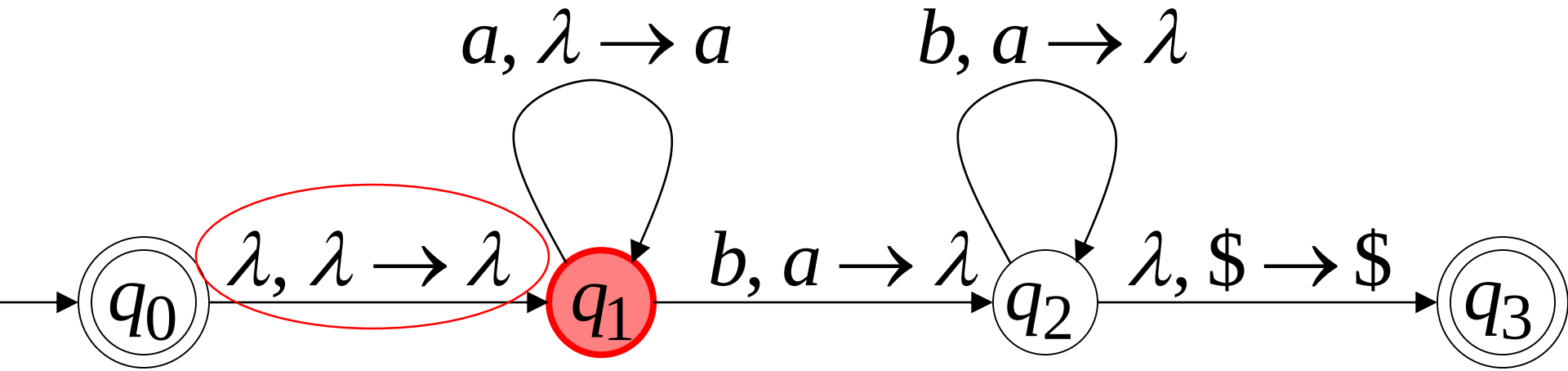
Time 1

Input

a	a	a	b	b	b

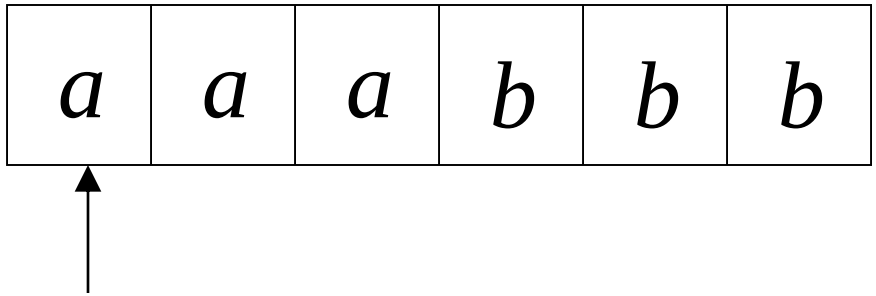


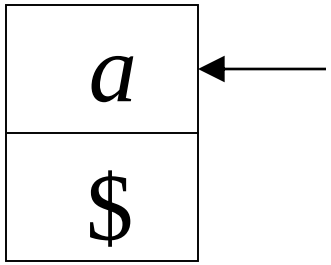
Stack



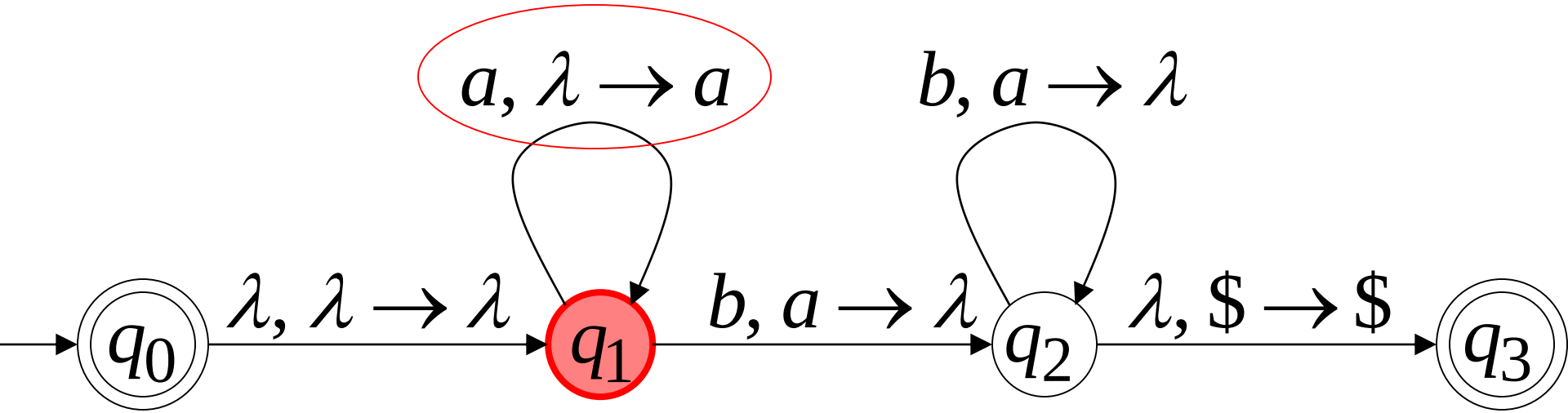
Time 2

Input



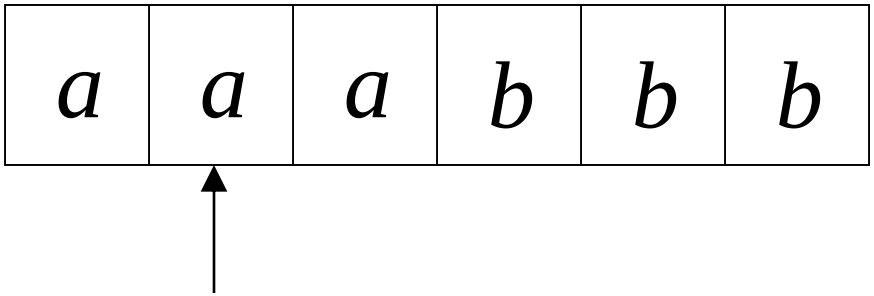


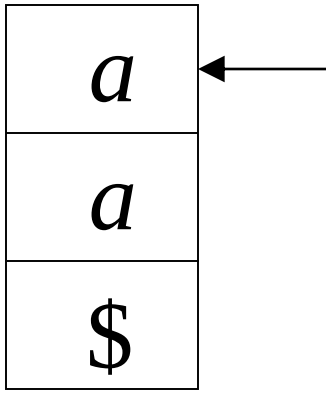
Stack



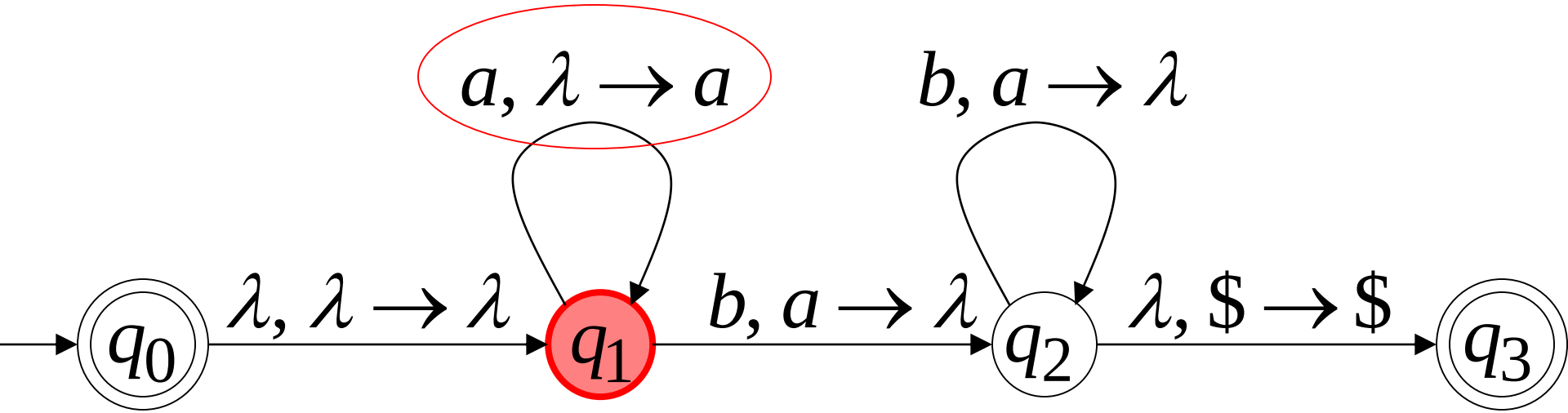
Time 3

Input





Stack



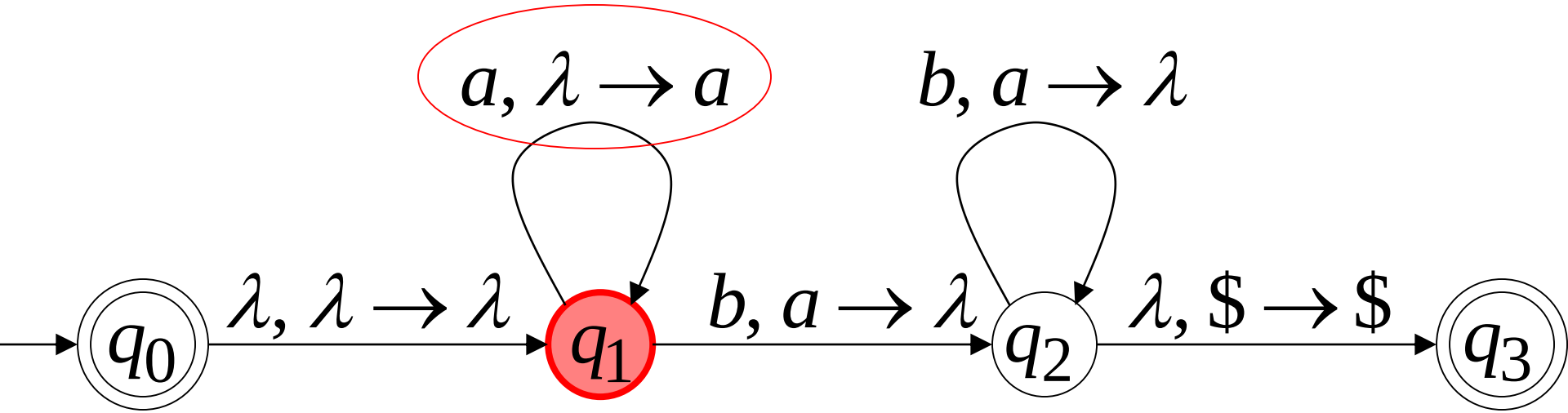
Time 4

Input

a	a	a	b	b	b

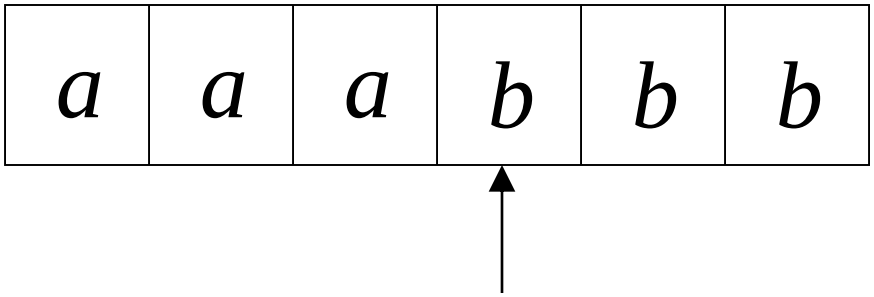
a
a
a
$\$$

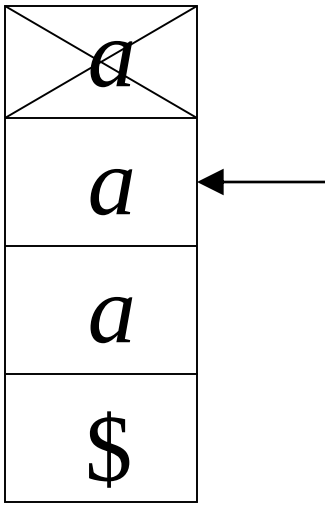
Stack



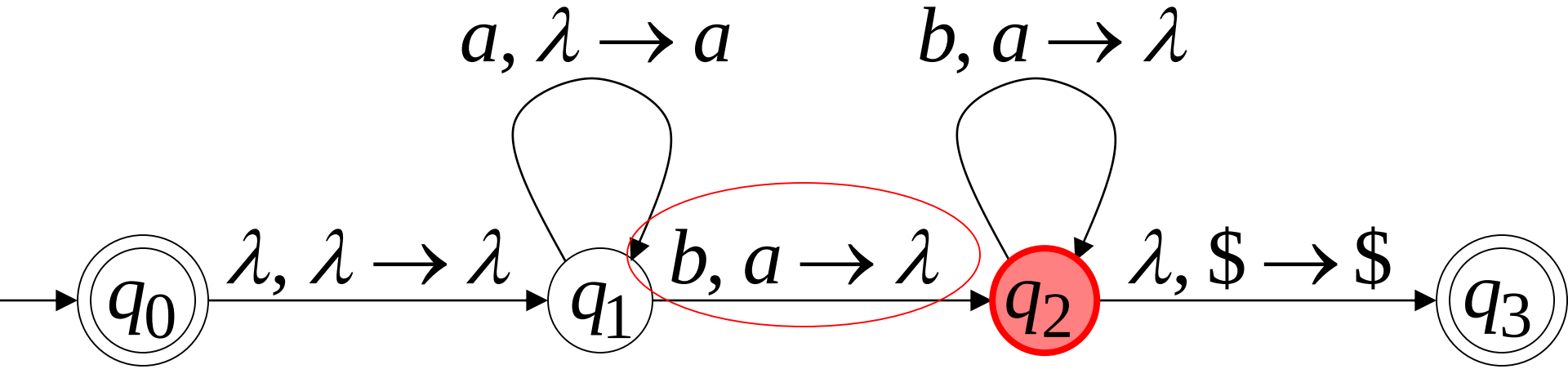
Time 5

Input



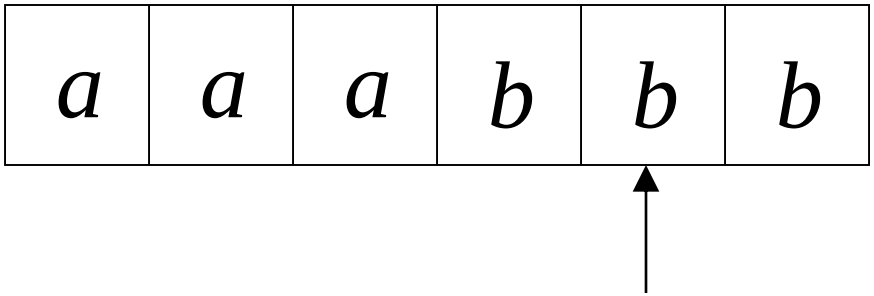


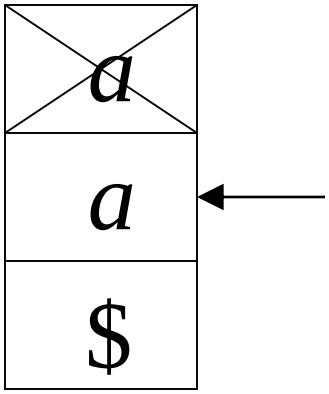
Stack



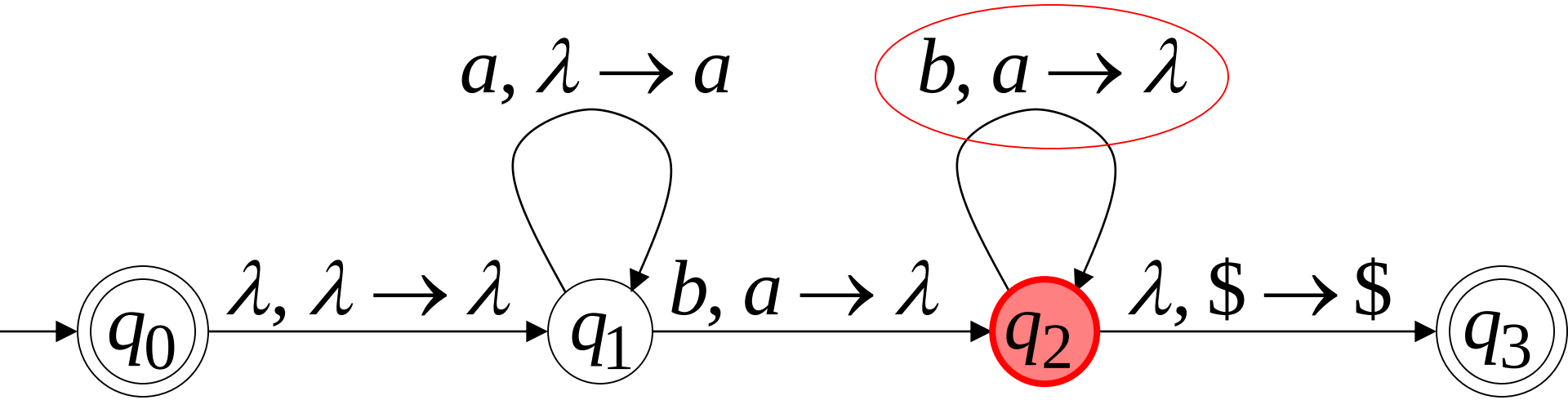
Time 6

Input



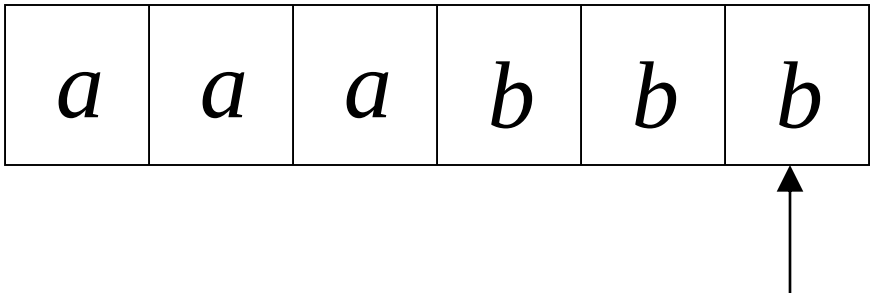


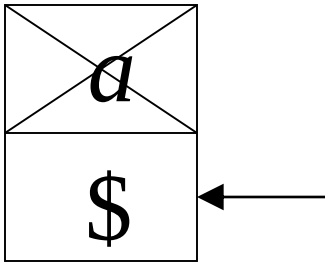
Stack



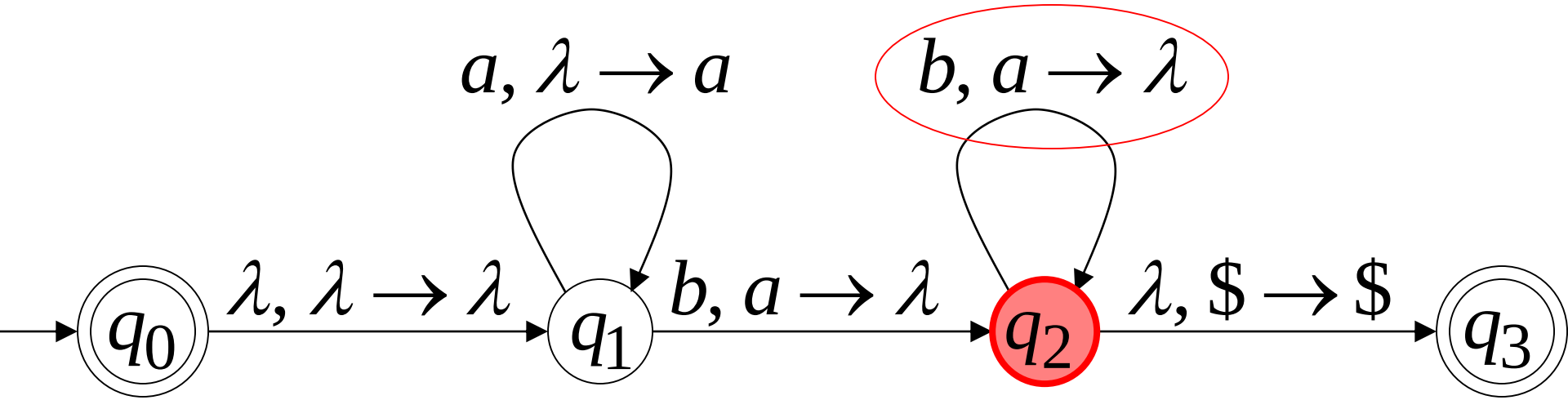
Time 7

Input



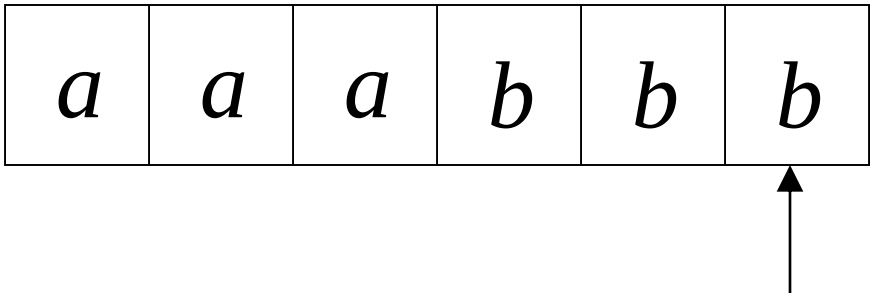


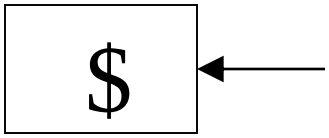
Stack



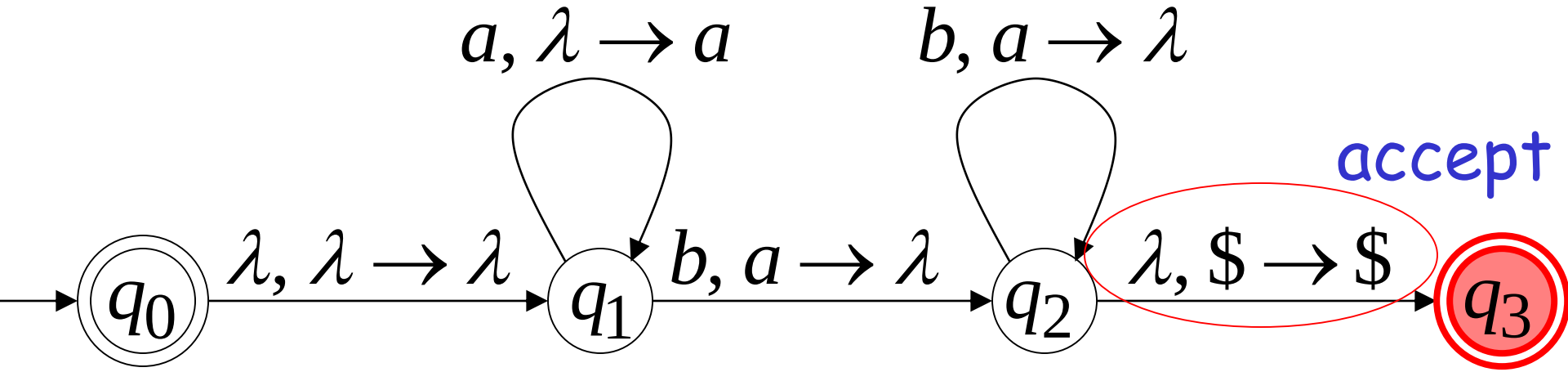
Time 8

Input





Stack



accept

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

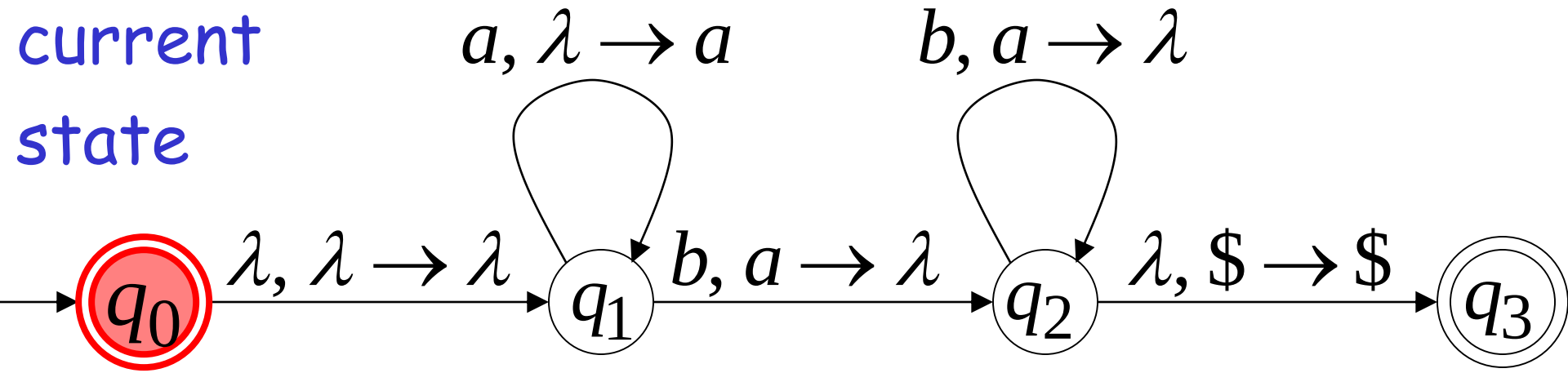
Input

a	a	b

$

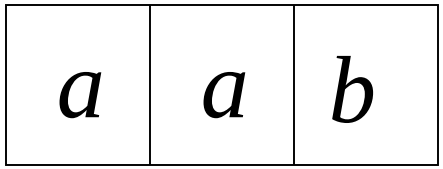
Stack

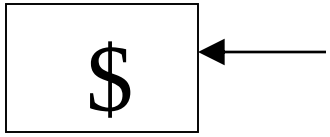
current state



Rejection Example: Time 1

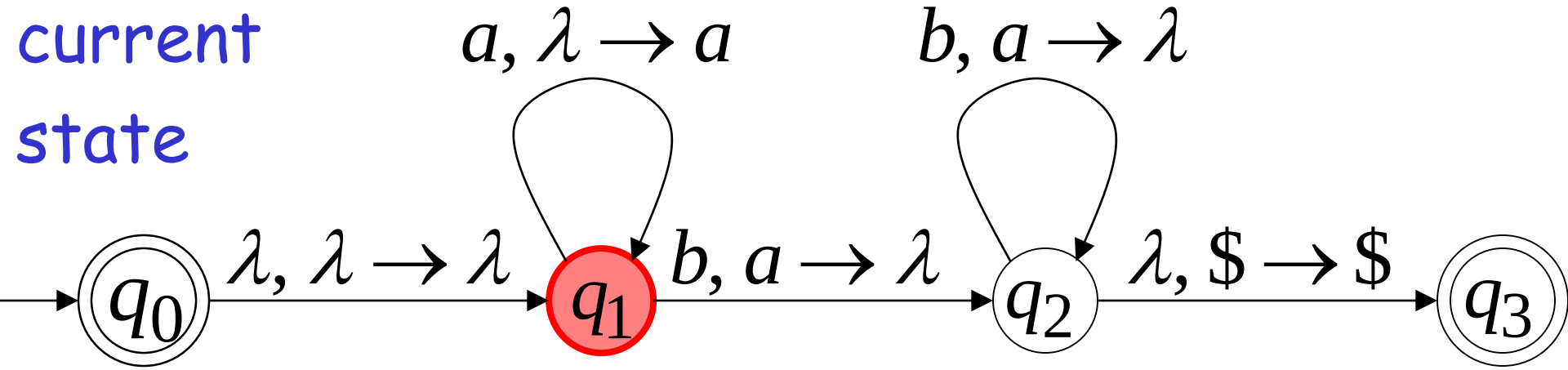
Input





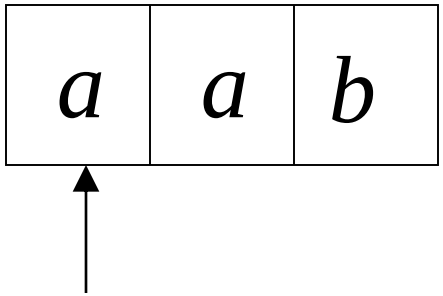
Stack

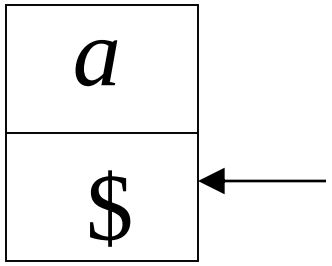
current state



Rejection Example: Time 2

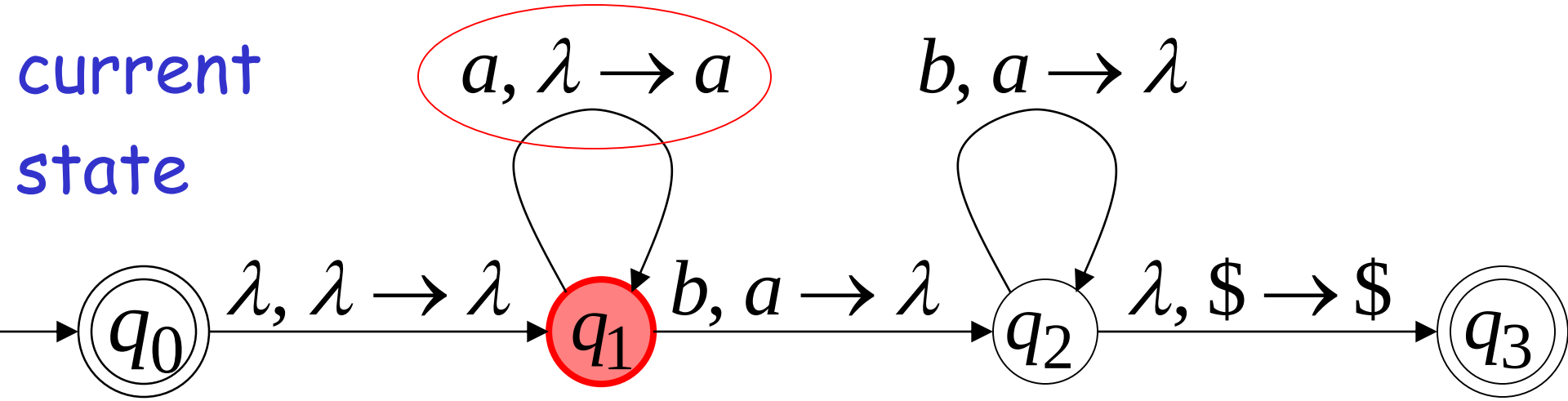
Input





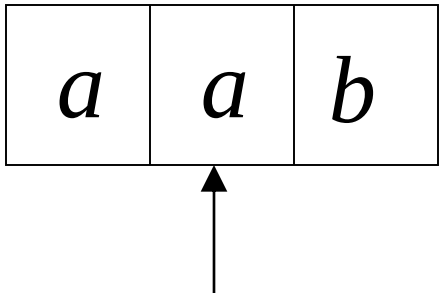
Stack

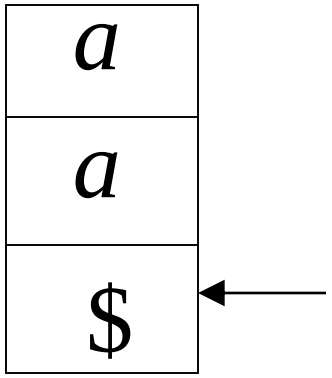
current state



Rejection Example: Time 3

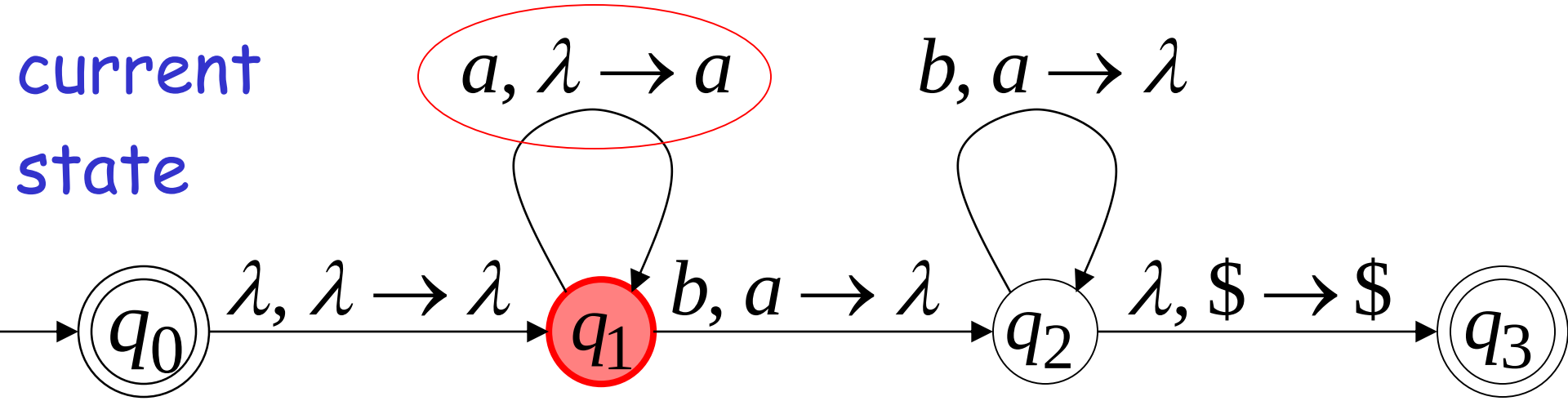
Input





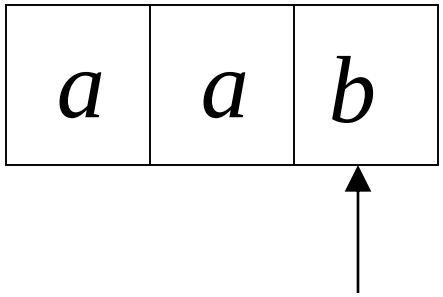
Stack

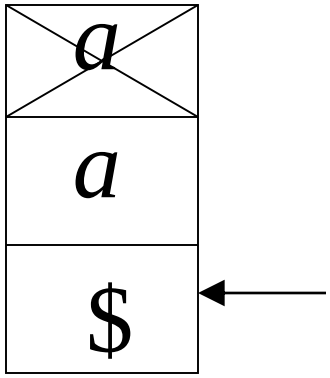
current state



Rejection Example: Time 4

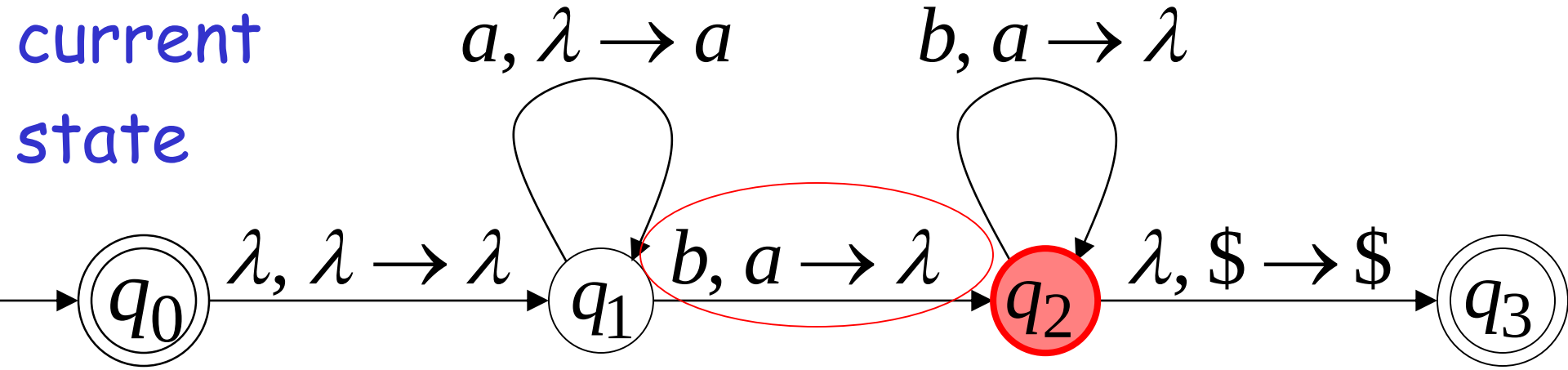
Input





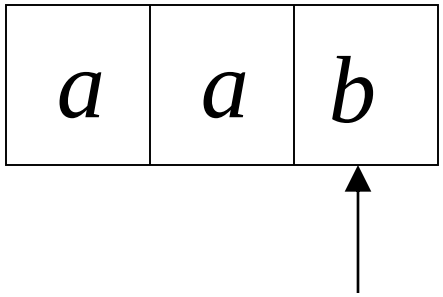
Stack

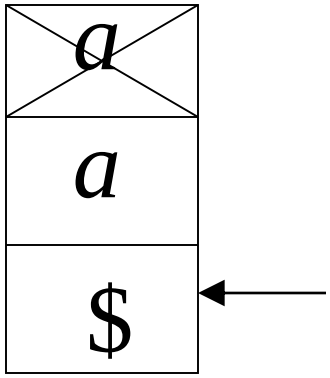
current state



Rejection Example: Time 4

Input

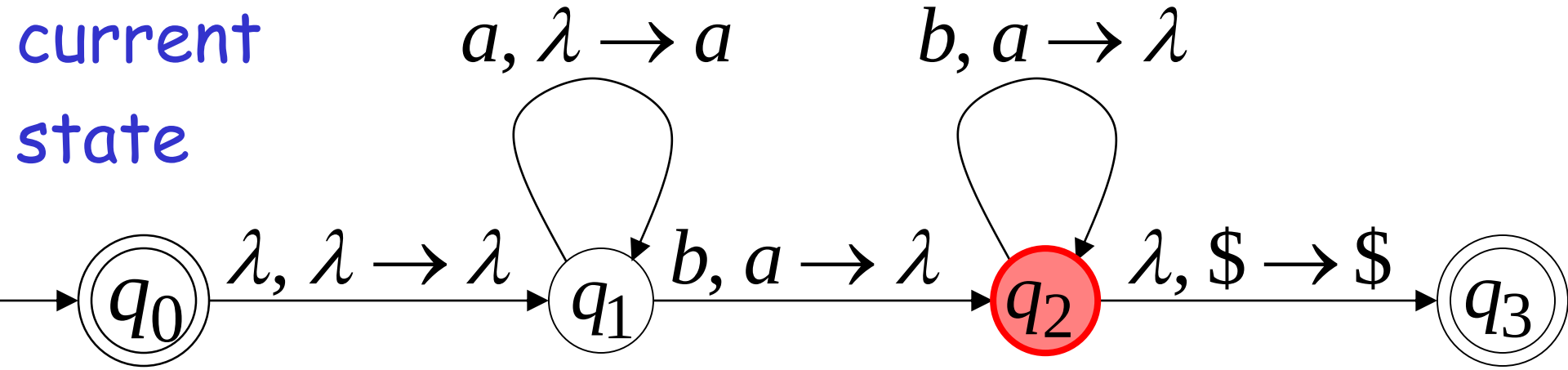




Stack

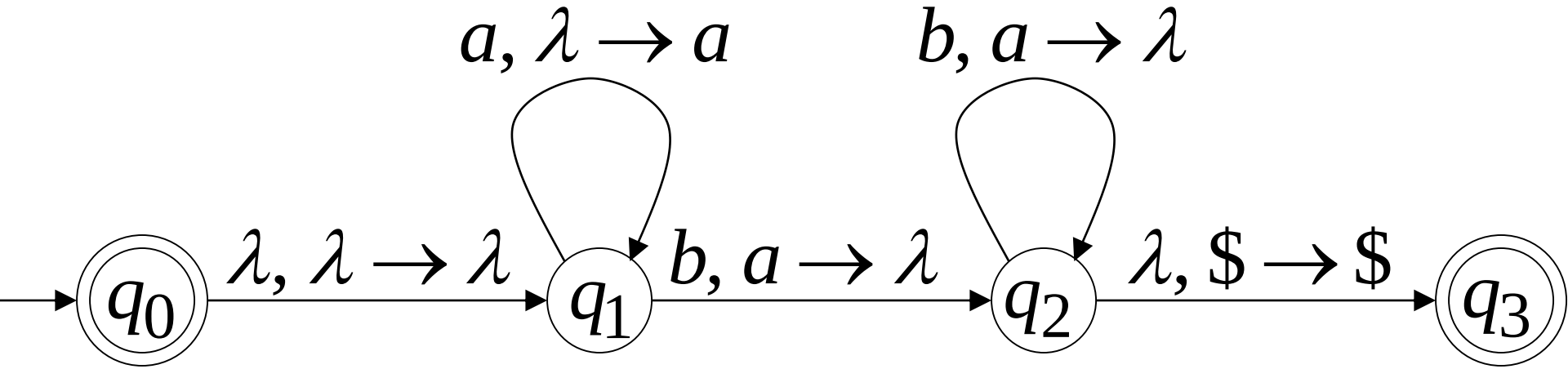
reject

current state



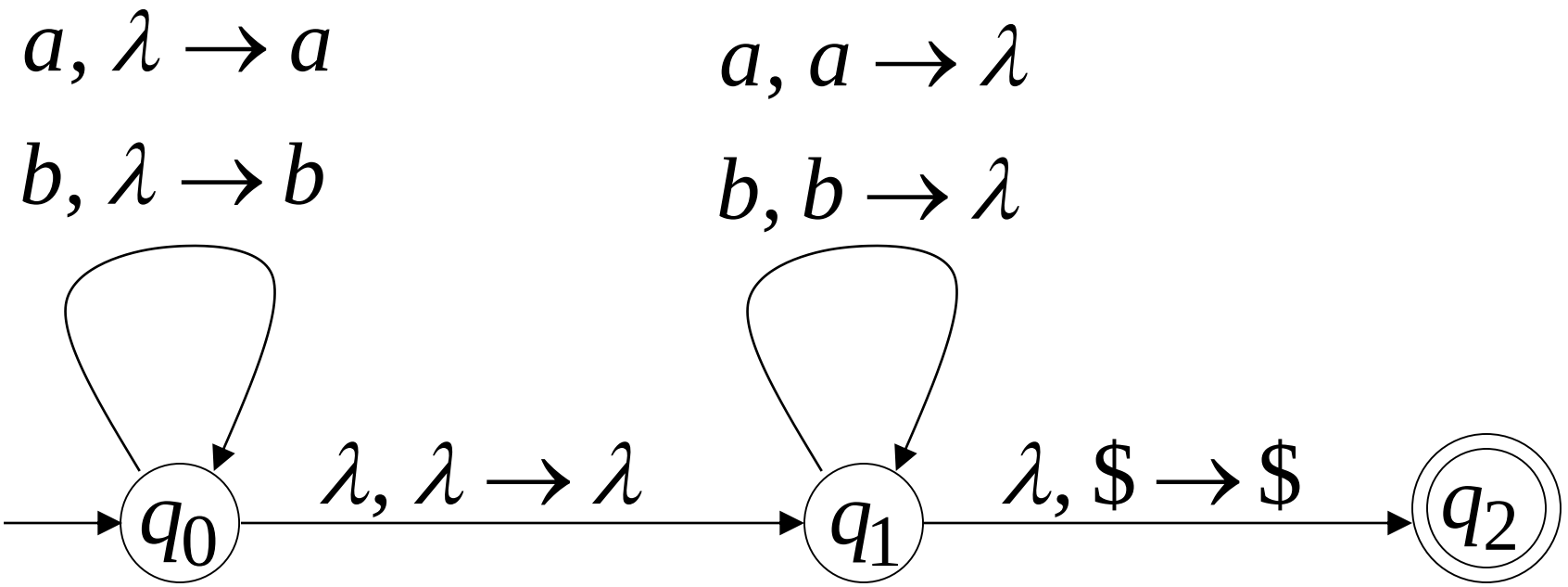
There is no accepting computation for aab

The string aab is rejected by the PDA



Another PDA example

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



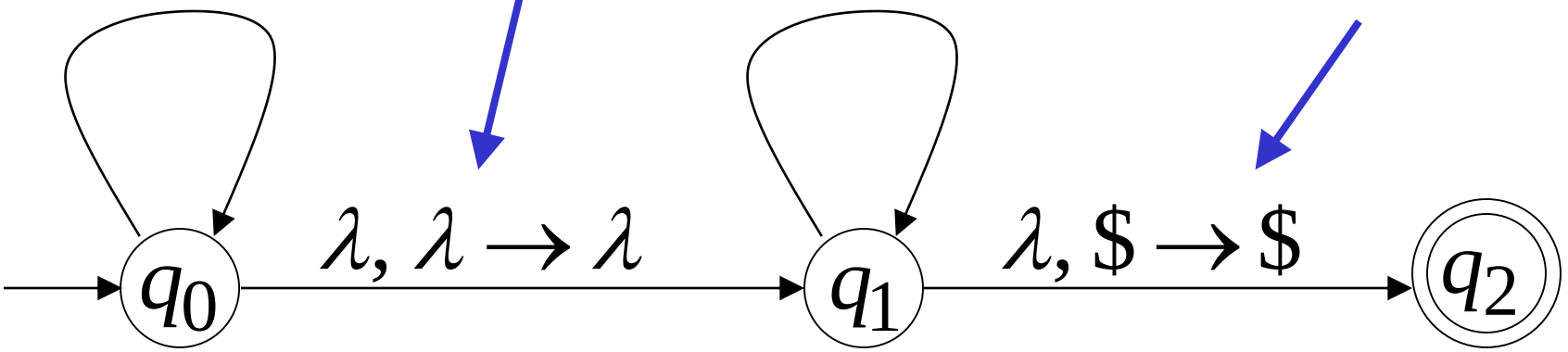
Basic Idea: $L(M) = \{vv^R : v \in \{a,b\}^*\}$

1. Push v on stack

2. Guess middle of input

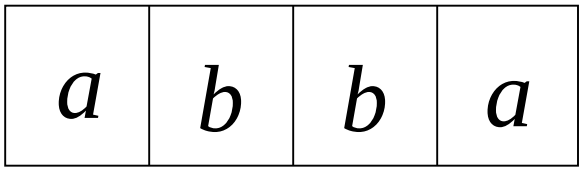
3. Match v^R on input with v on stack

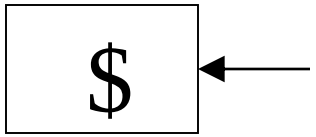
4. Match found



Execution Example: Time 0

Input





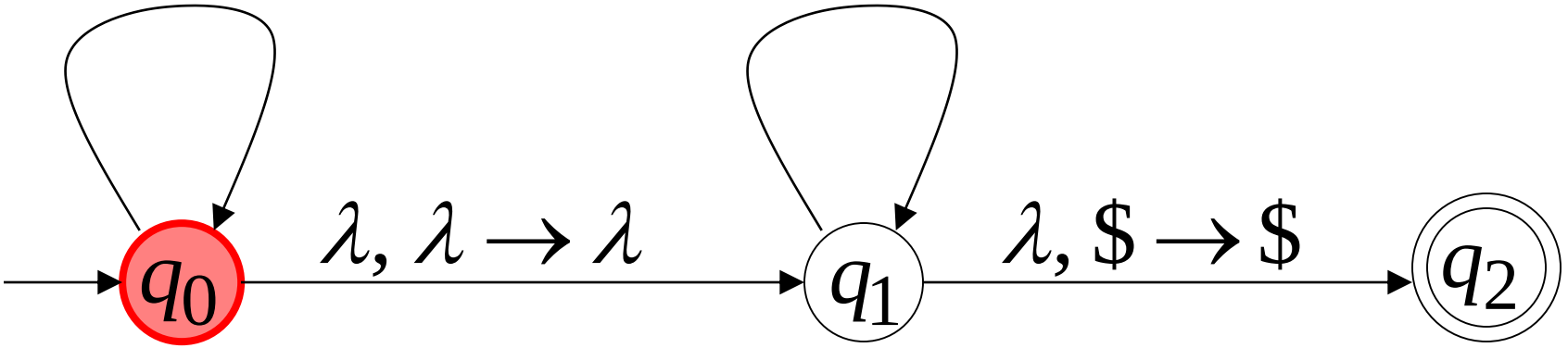
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

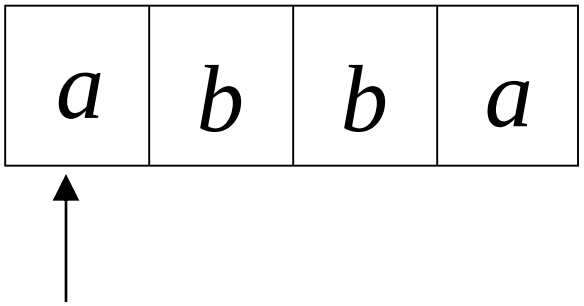
$a, a \rightarrow \lambda$

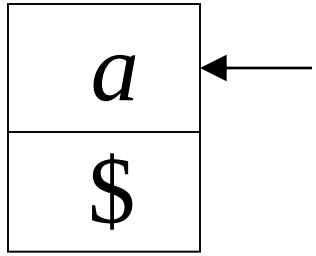
$b, b \rightarrow \lambda$



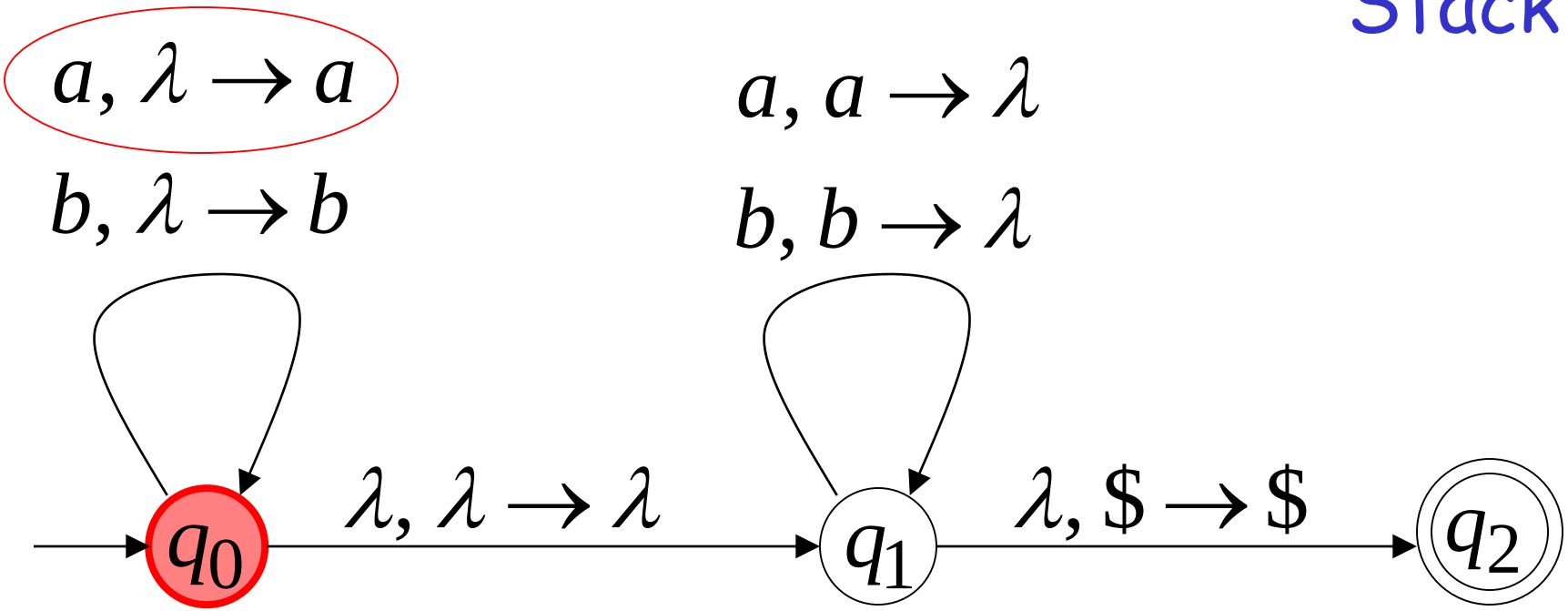
Time 1

Input



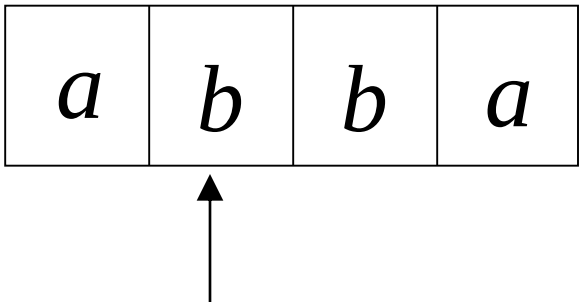


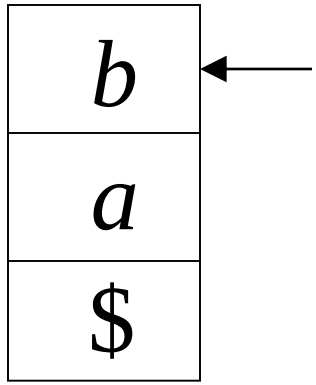
Stack



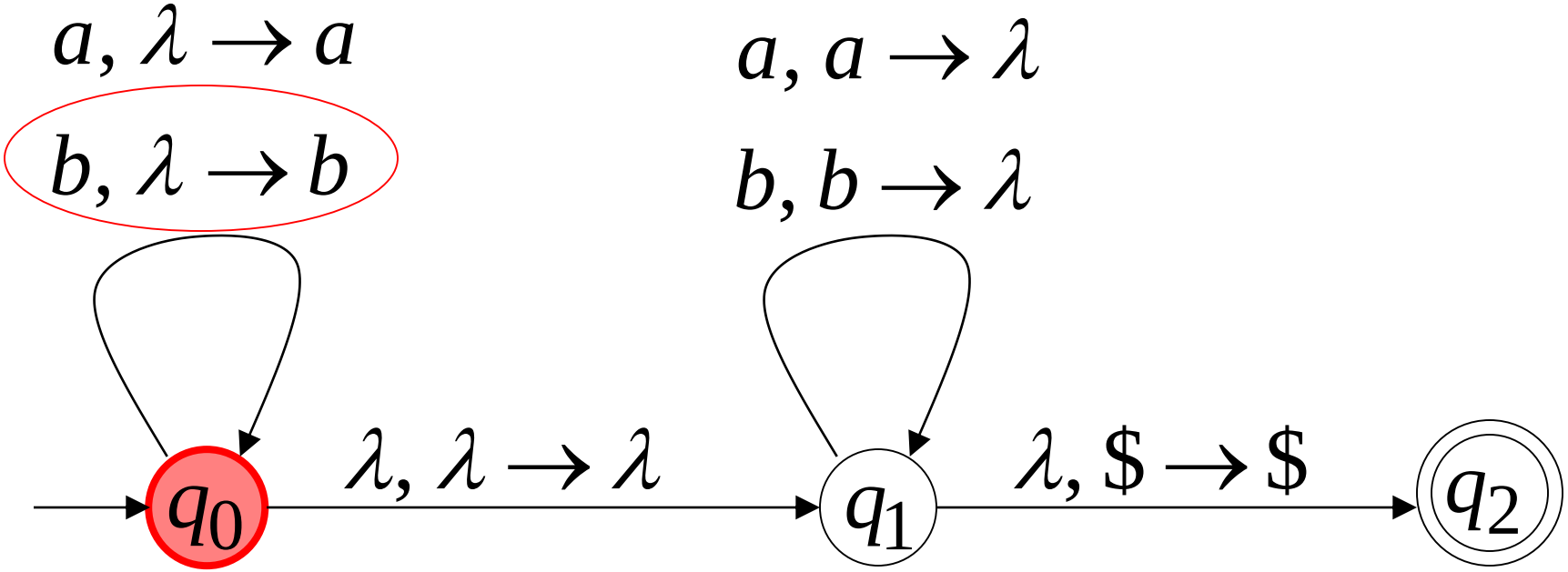
Time 2

Input



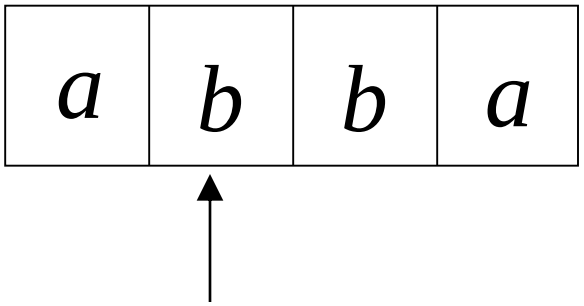


Stack

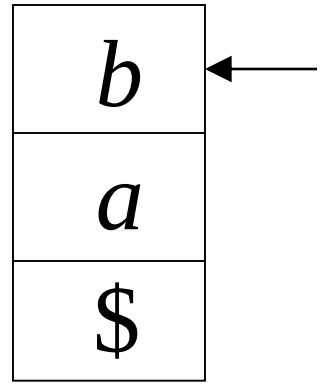


Time 3

Input



Guess the middle of string



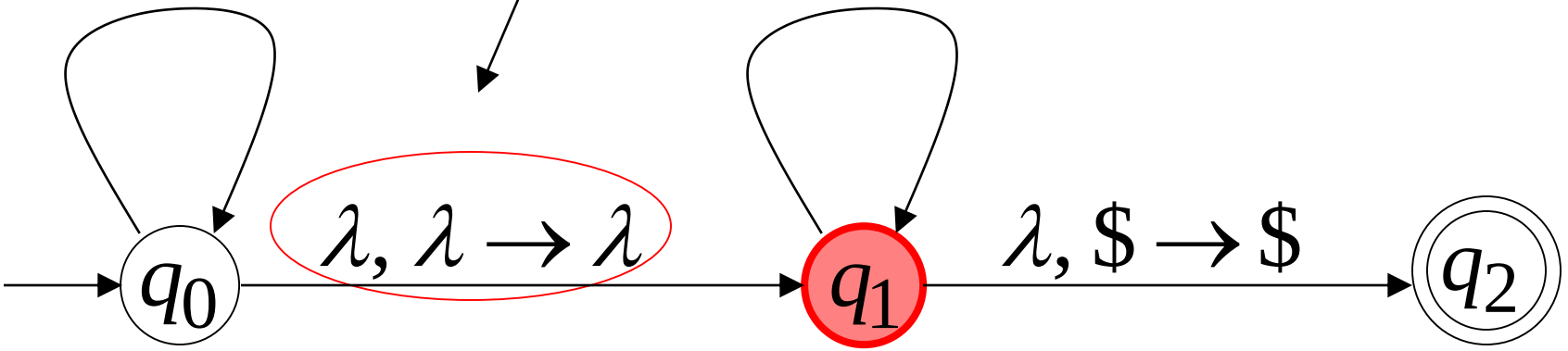
Stack

$a, \lambda \to a$

$b, \lambda \to b$

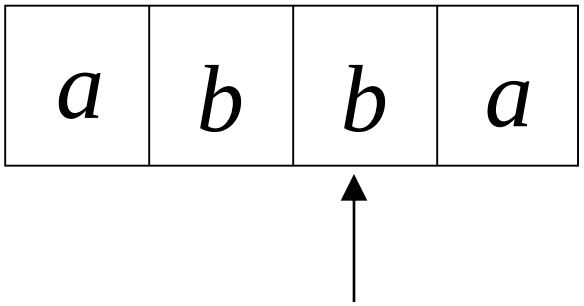
$a, a \to \lambda$

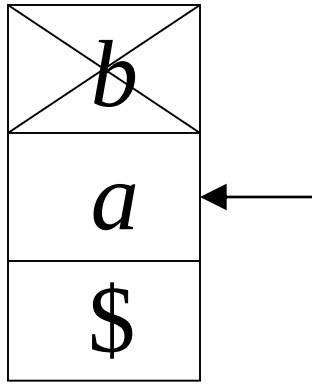
$b, b \to \lambda$



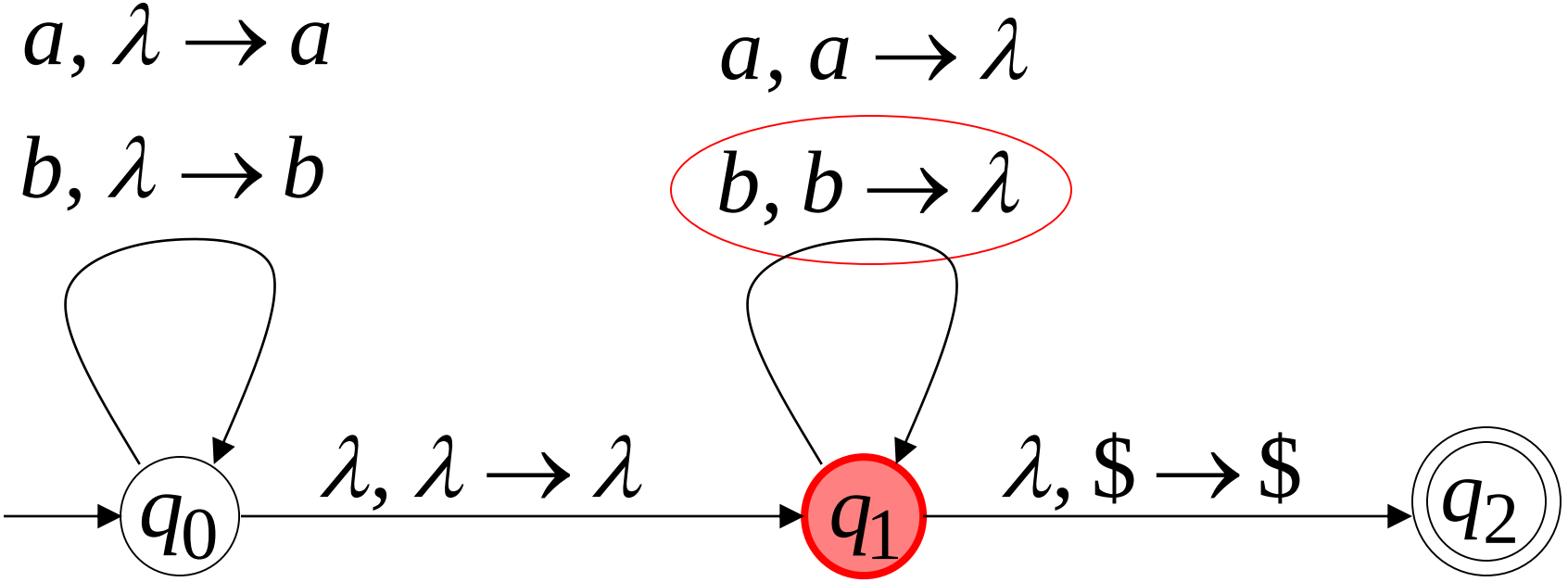
Time 4

Input



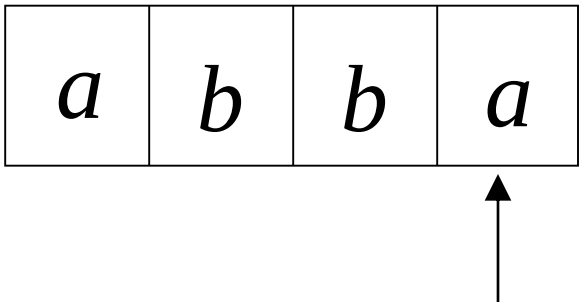


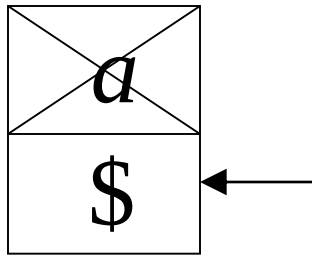
Stack



Time 5

Input





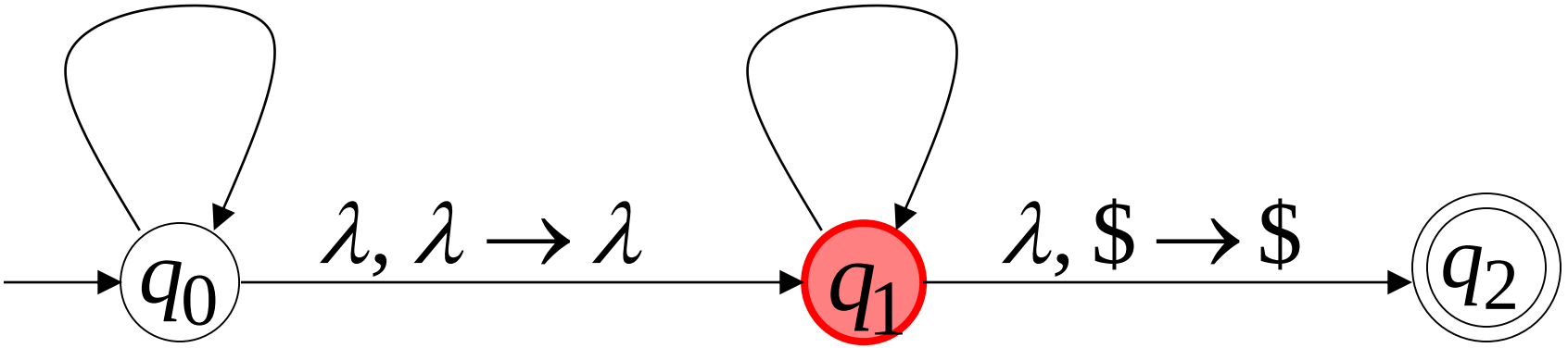
Stack

$a, \lambda \to a$

$b, \lambda \to b$

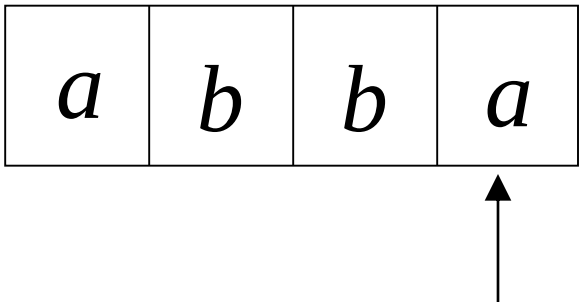
$a, a \to \lambda$

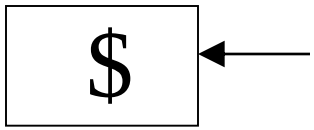
$b, b \to \lambda$



Time 6

Input





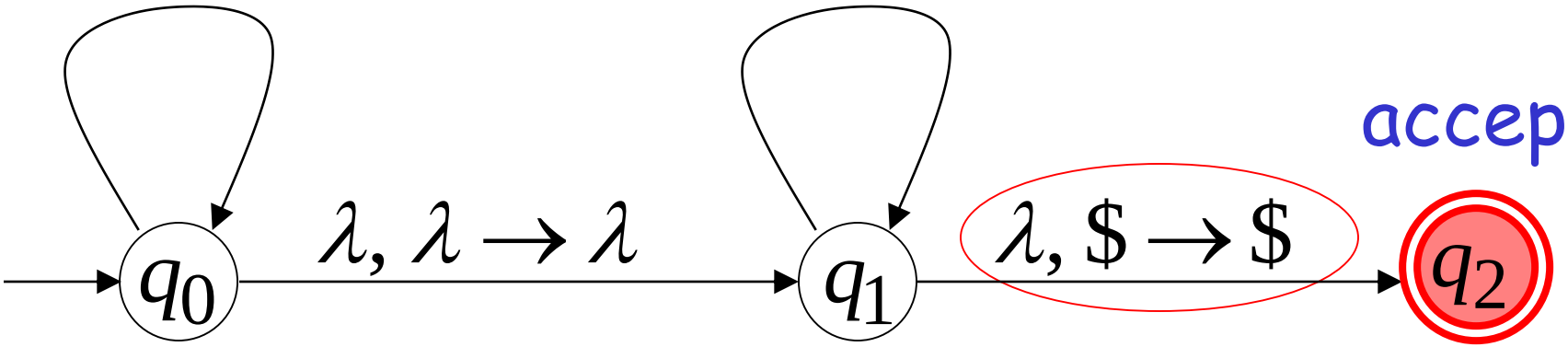
Stack

$a, \lambda \to a$

$b, \lambda \to b$

$a, a \to \lambda$

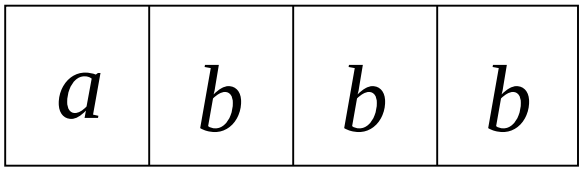
$b, b \to \lambda$

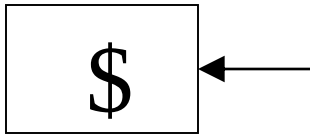


accept

Rejection Example: Time 0

Input





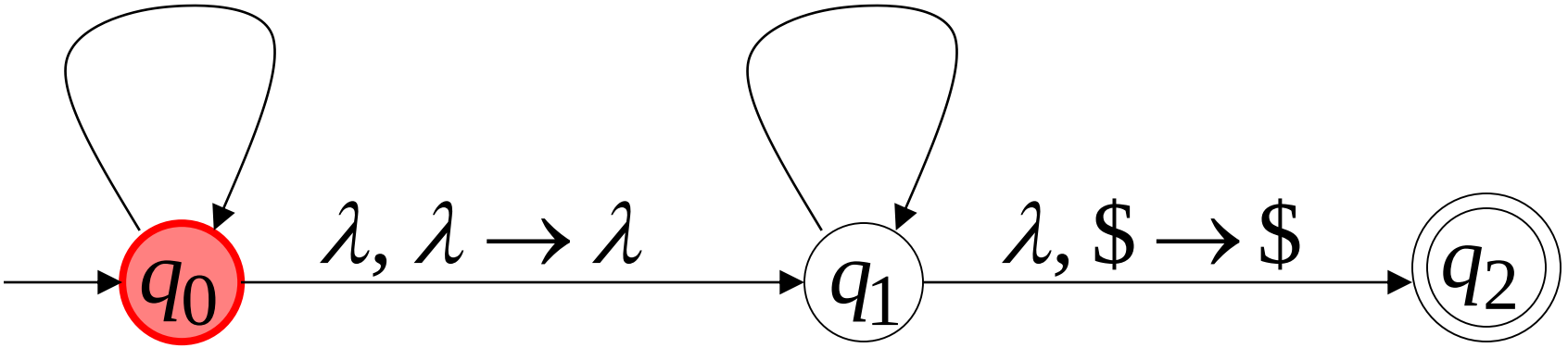
Stack

$a, \lambda \to a$

$b, \lambda \to b$

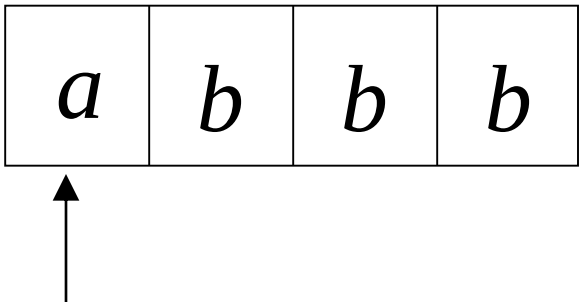
$a, a \to \lambda$

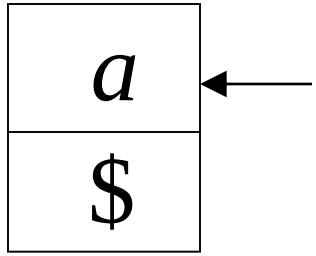
$b, b \to \lambda$



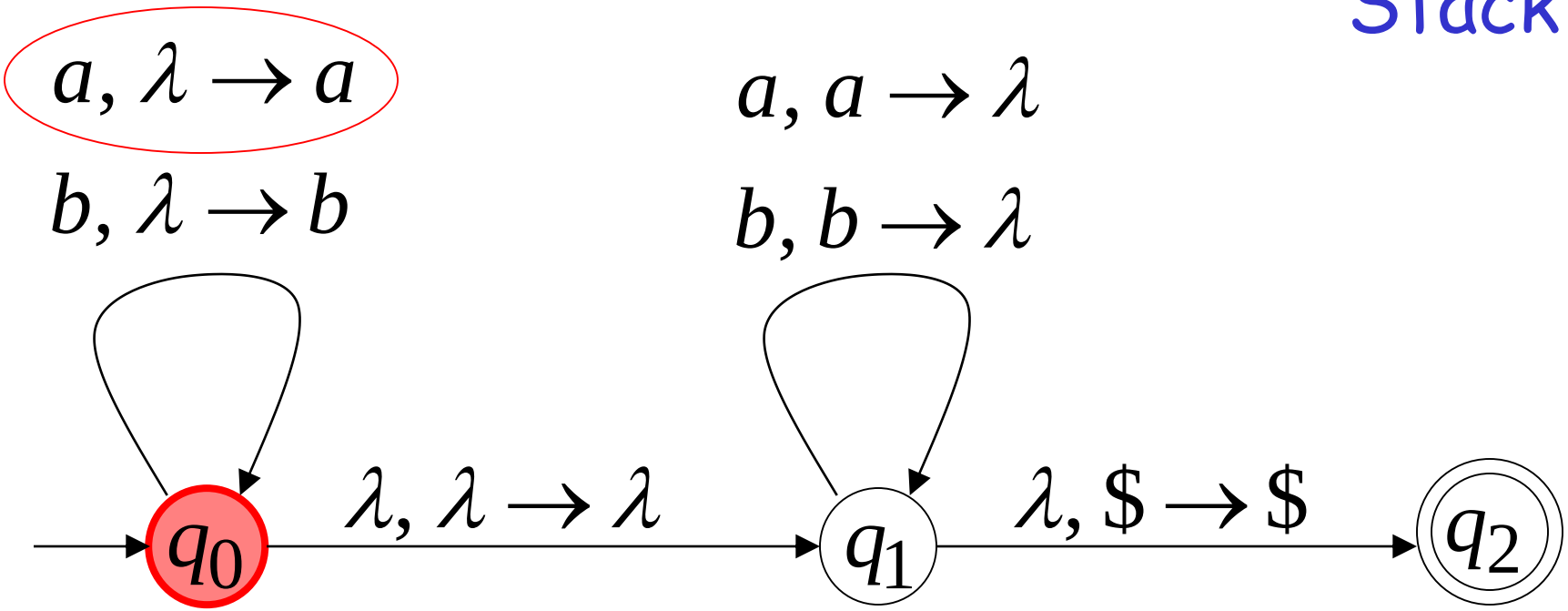
Time 1

Input



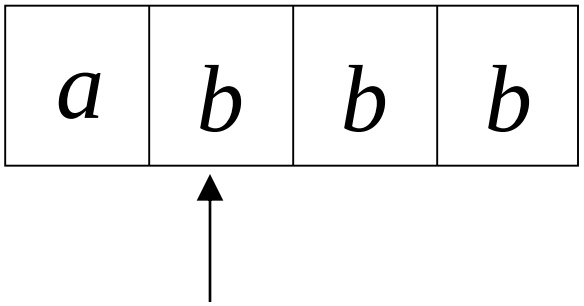


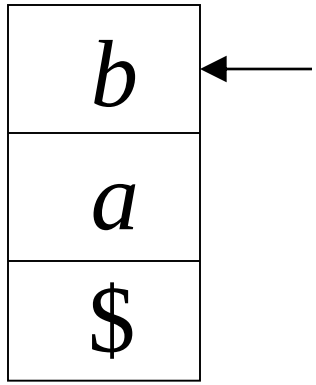
Stack



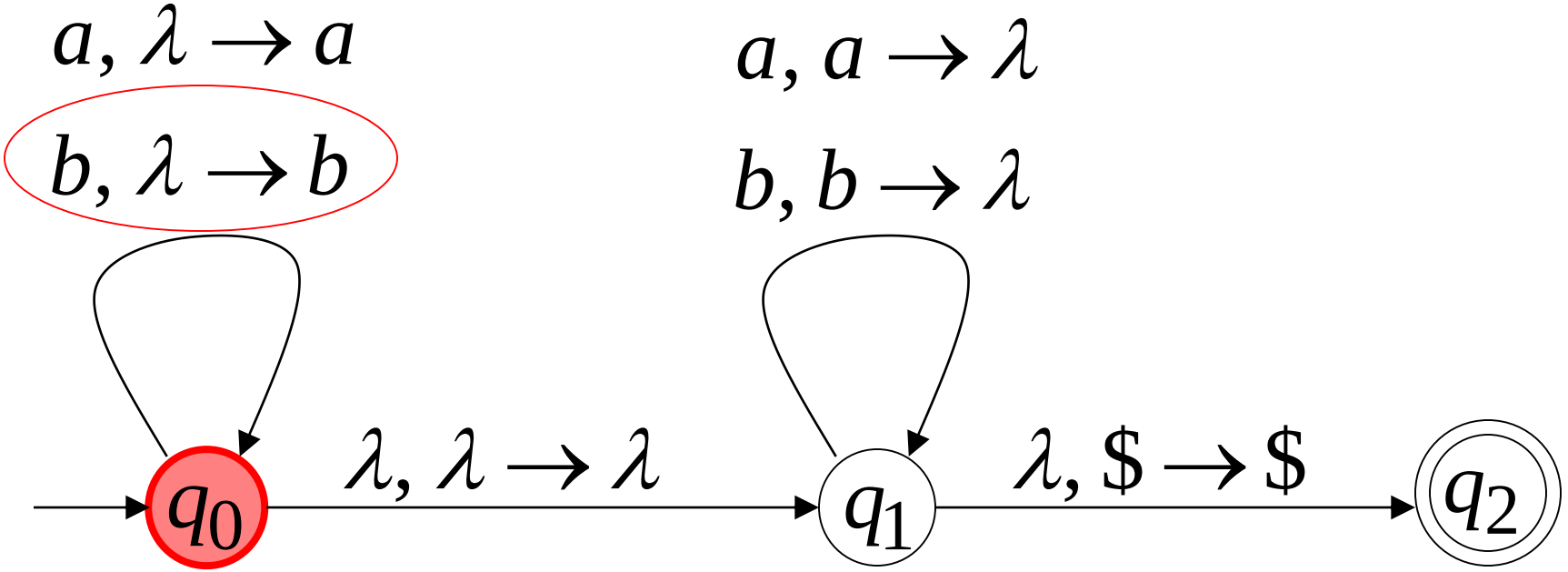
Time 2

Input



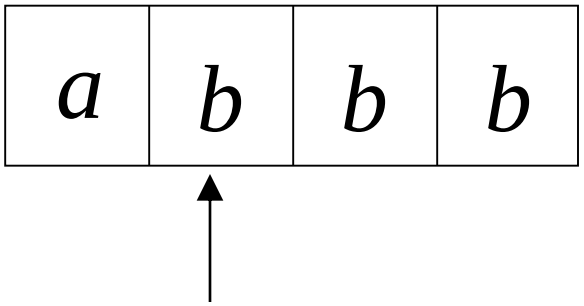


Stack

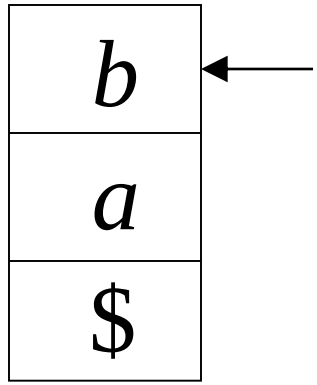


Time 3

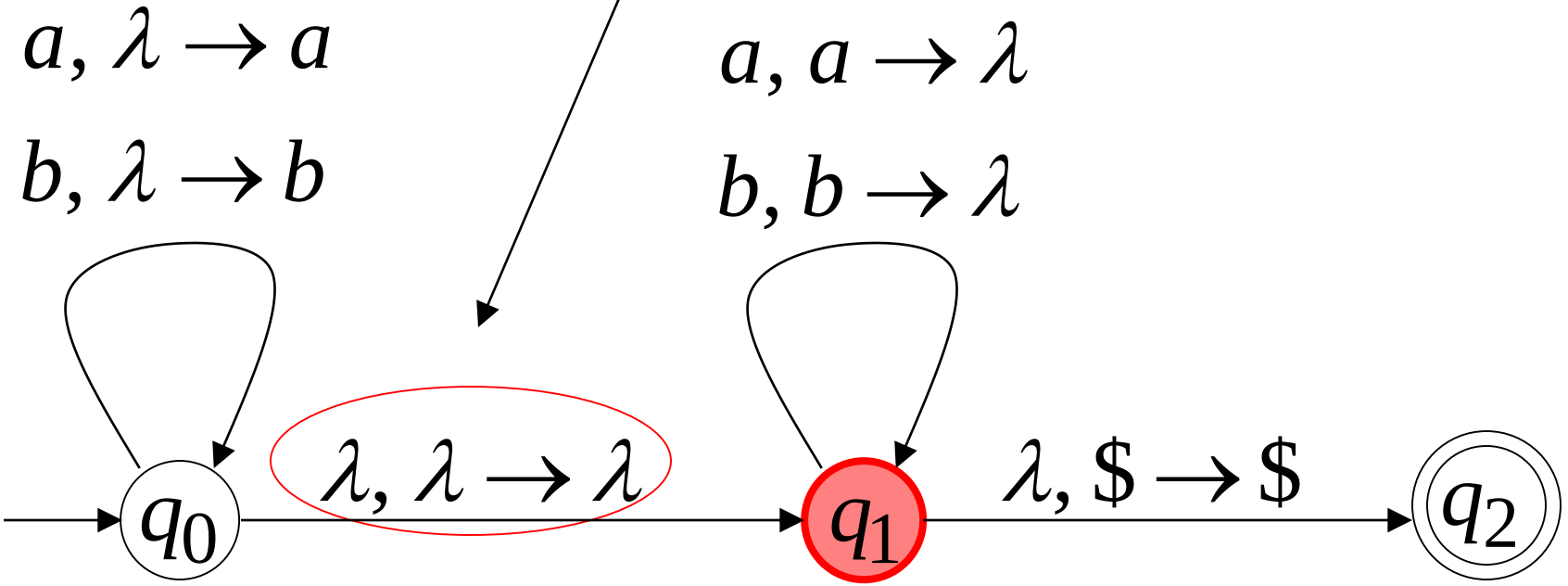
Input



Guess the middle of string

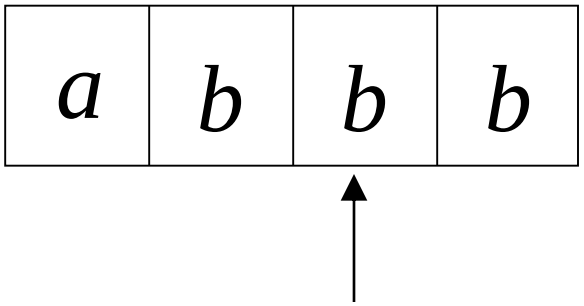


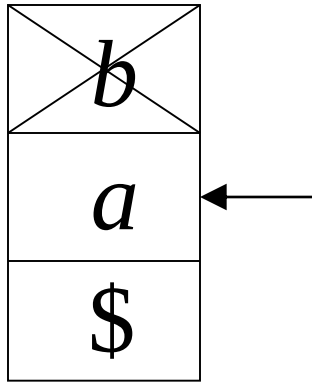
Stack



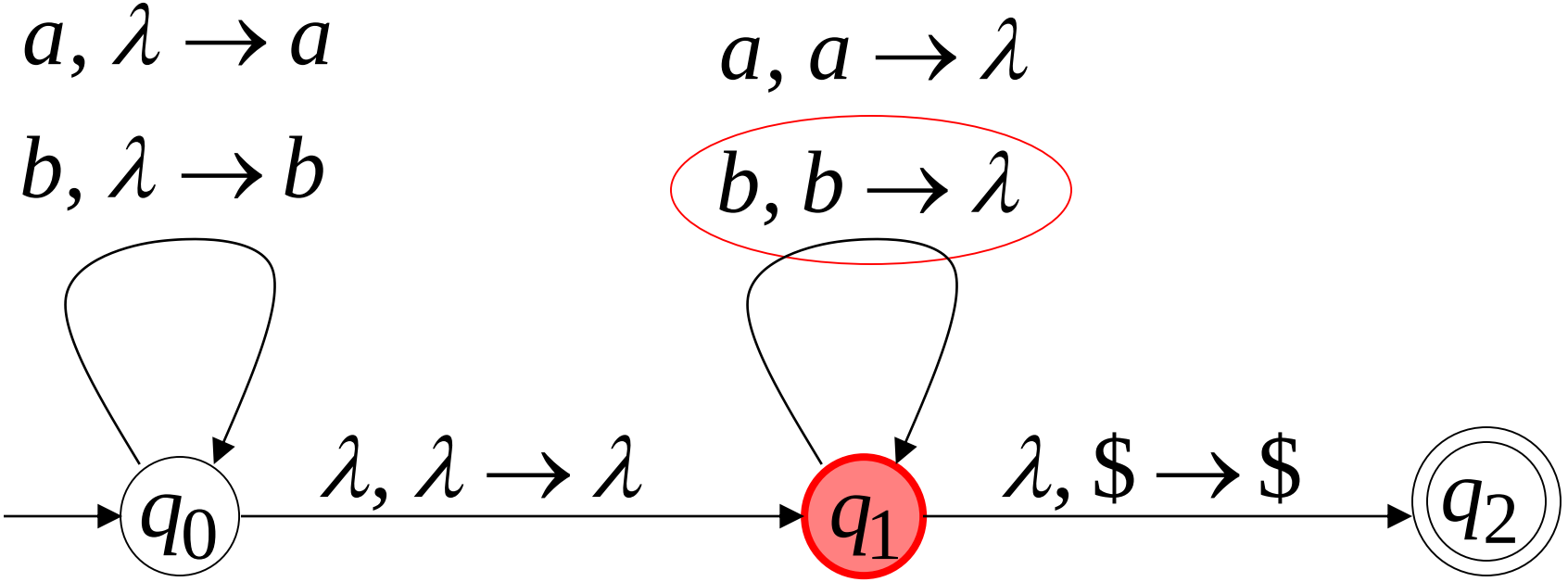
Time 4

Input





Stack



Time 5

There is no possible transition.

Input

a	b	b	b

Input is not consumed

a
$

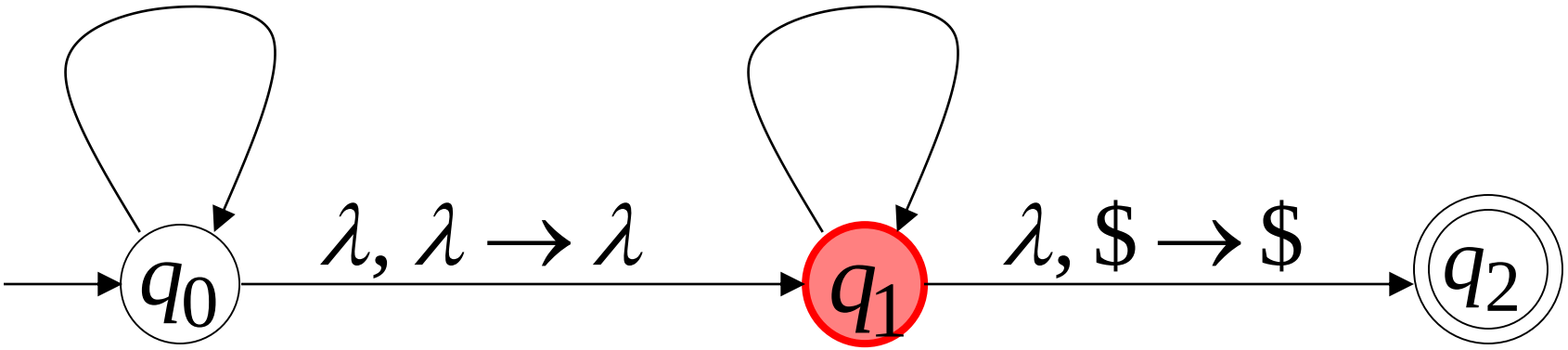
Stack

$a, \lambda \to a$

$b, \lambda \to b$

$a, a \to \lambda$

$b, b \to \lambda$

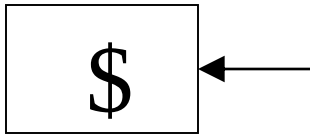


Another computation on same string:

Input

Time 0

a	b	b	b



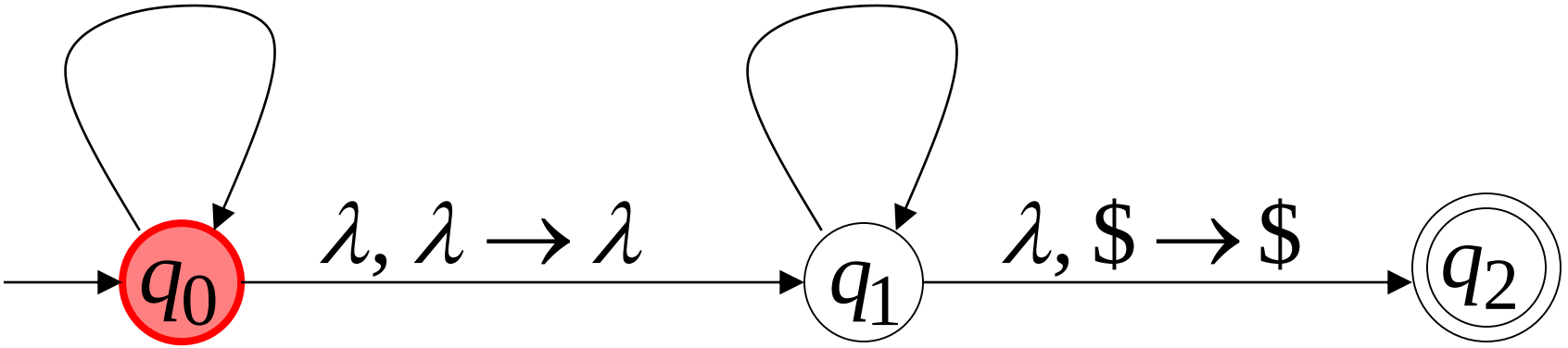
Stack

$a, \lambda \to a$

$b, \lambda \to b$

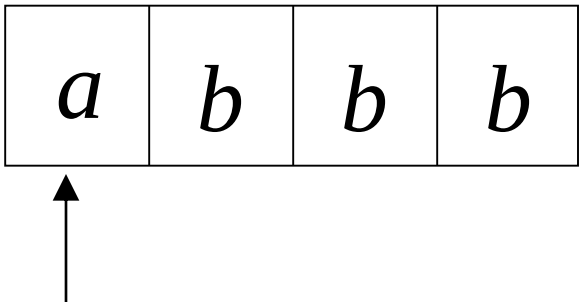
$a, a \to \lambda$

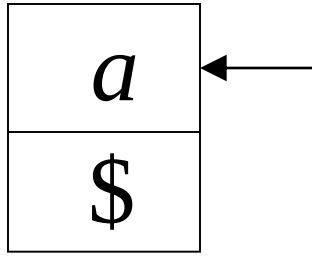
$b, b \to \lambda$



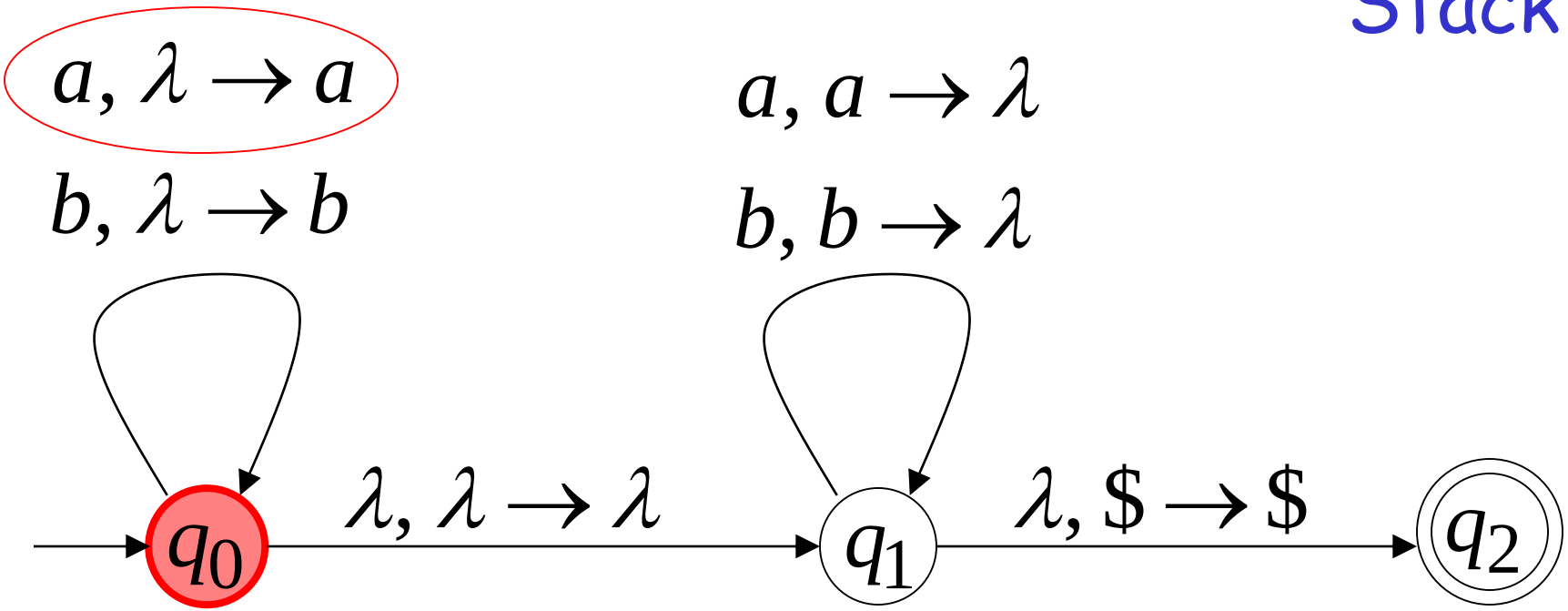
Time 1

Input



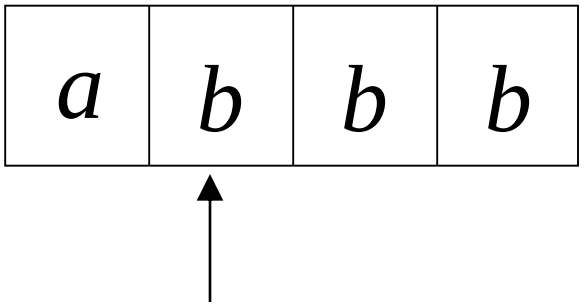


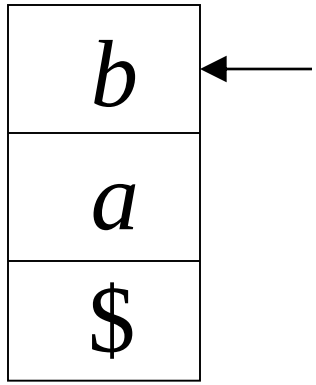
Stack



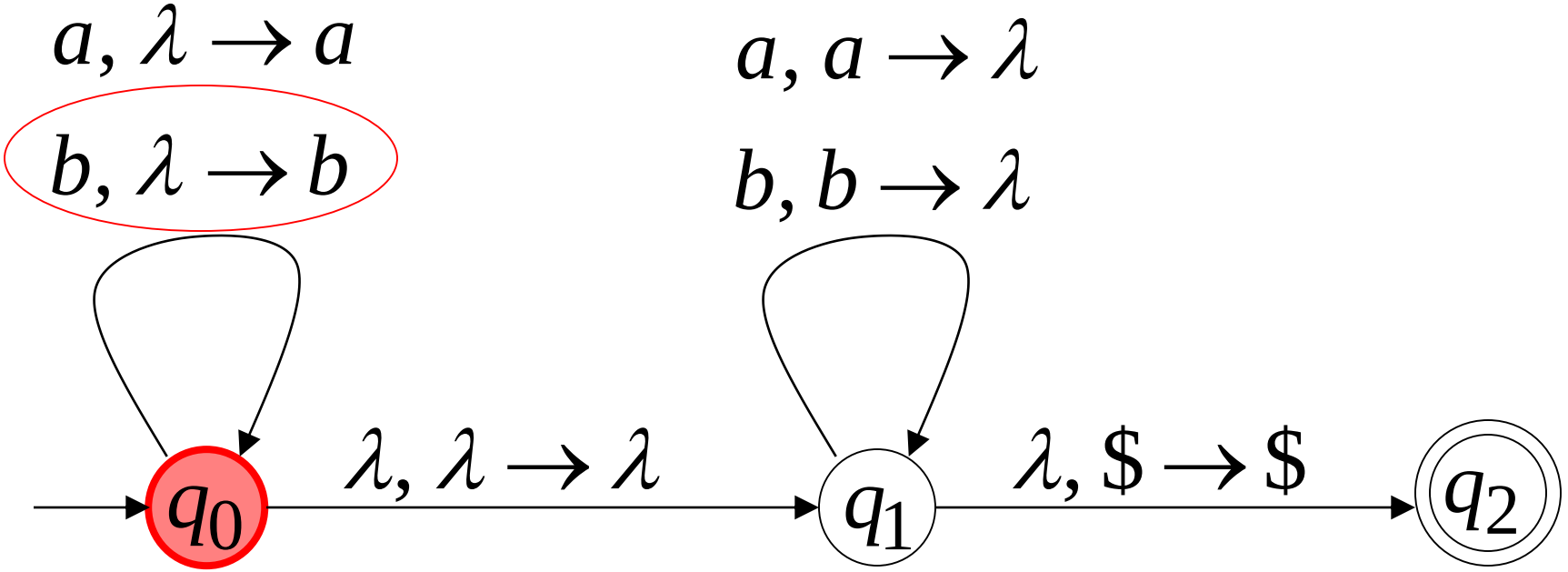
Time 2

Input



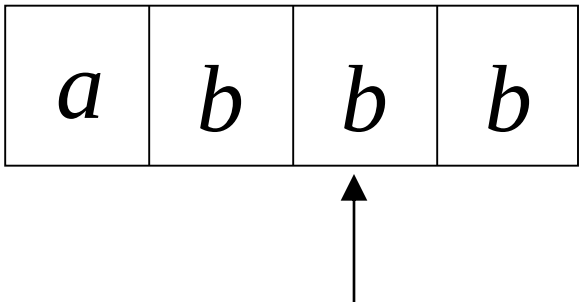


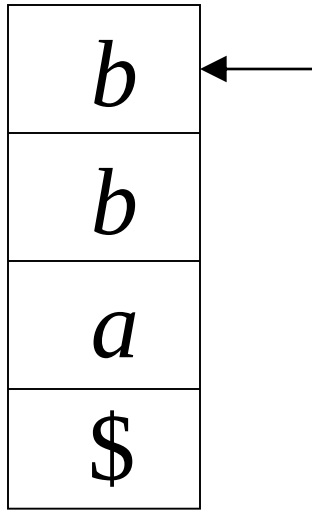
Stack



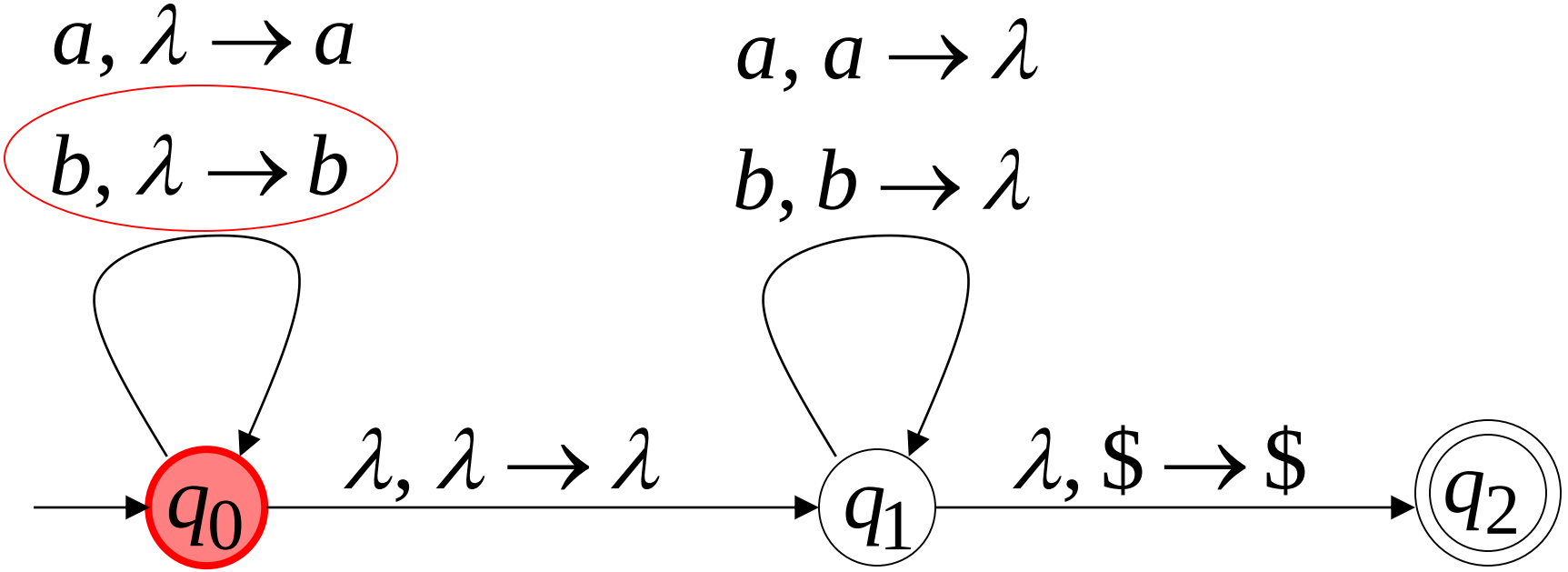
Time 3

Input





Stack



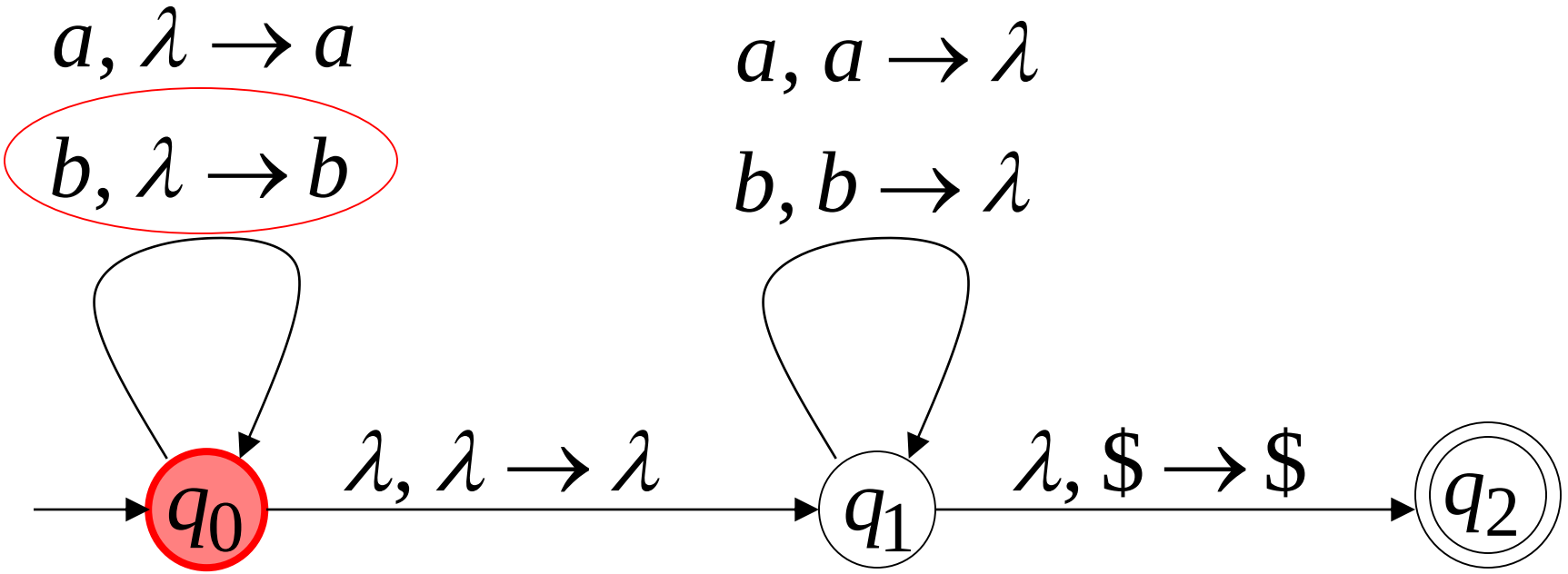
Time 4

Input

a	b	b	b

Stack

b
b
b
a
\$



Time 5

Input

a	b	b	b

No accept state is reached

b
b
b
a
$\$$

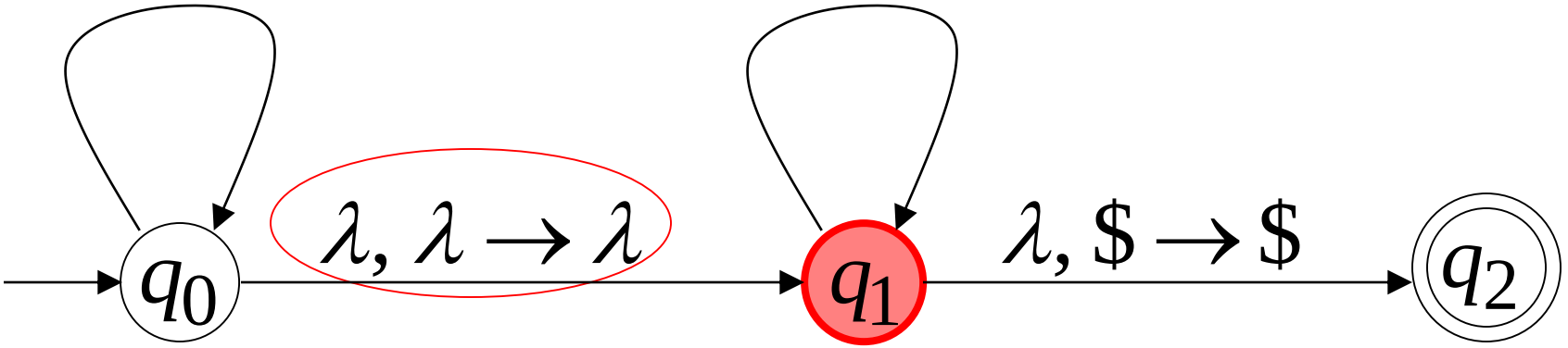
Stack

$a, \lambda \to a$

$b, \lambda \to b$

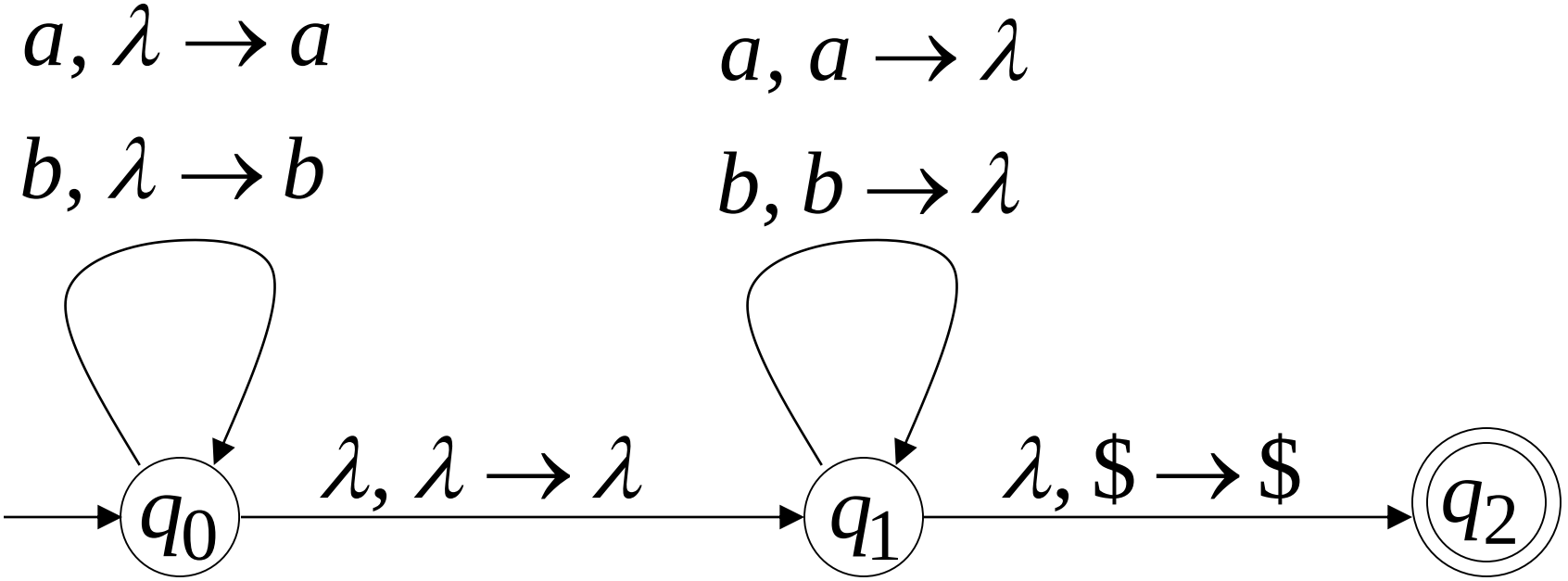
$a, a \to \lambda$

$b, b \to \lambda$



There is no computation that accepts string $abbb$

$$abbb \notin L(M)$$



Pushing & Popping Strings