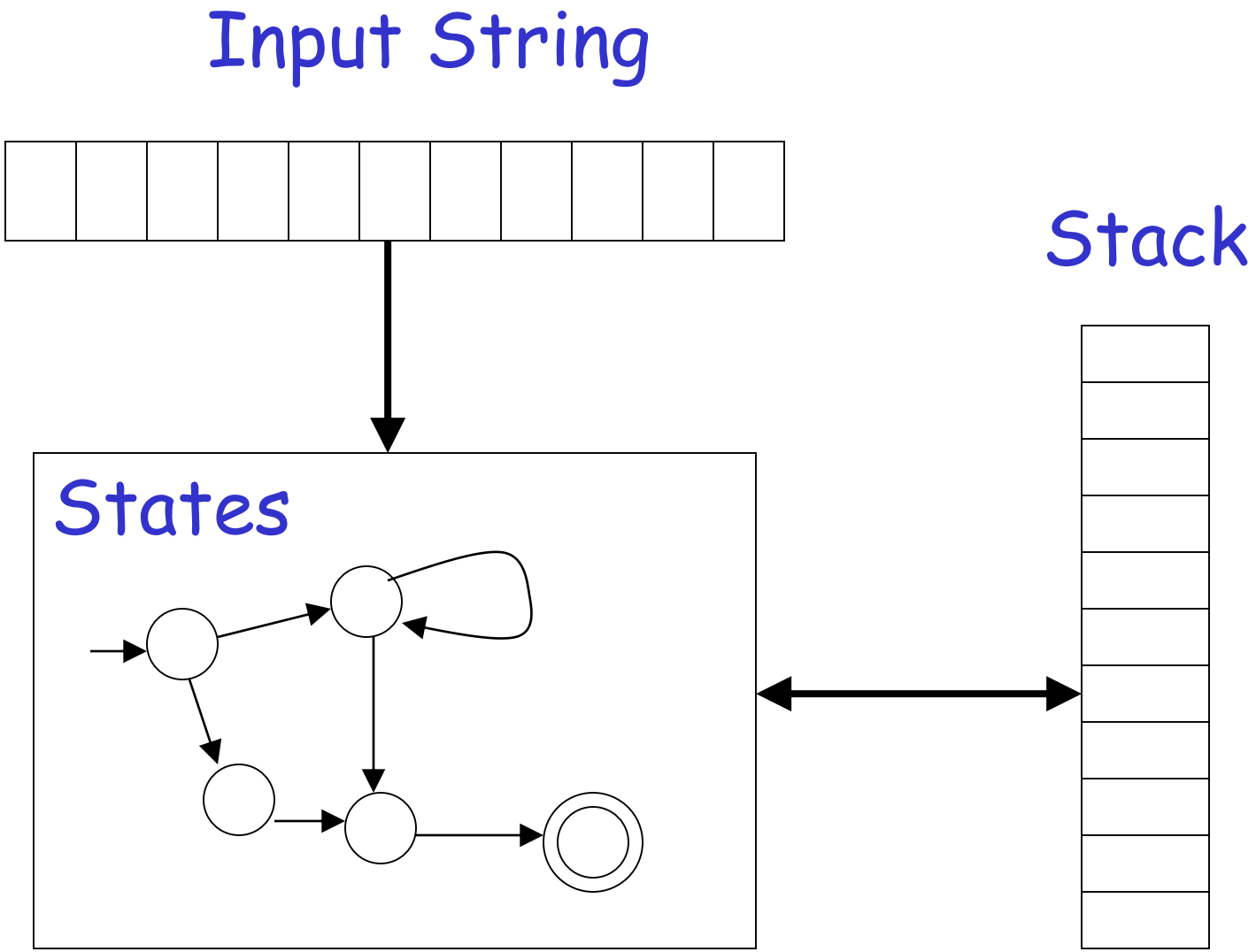
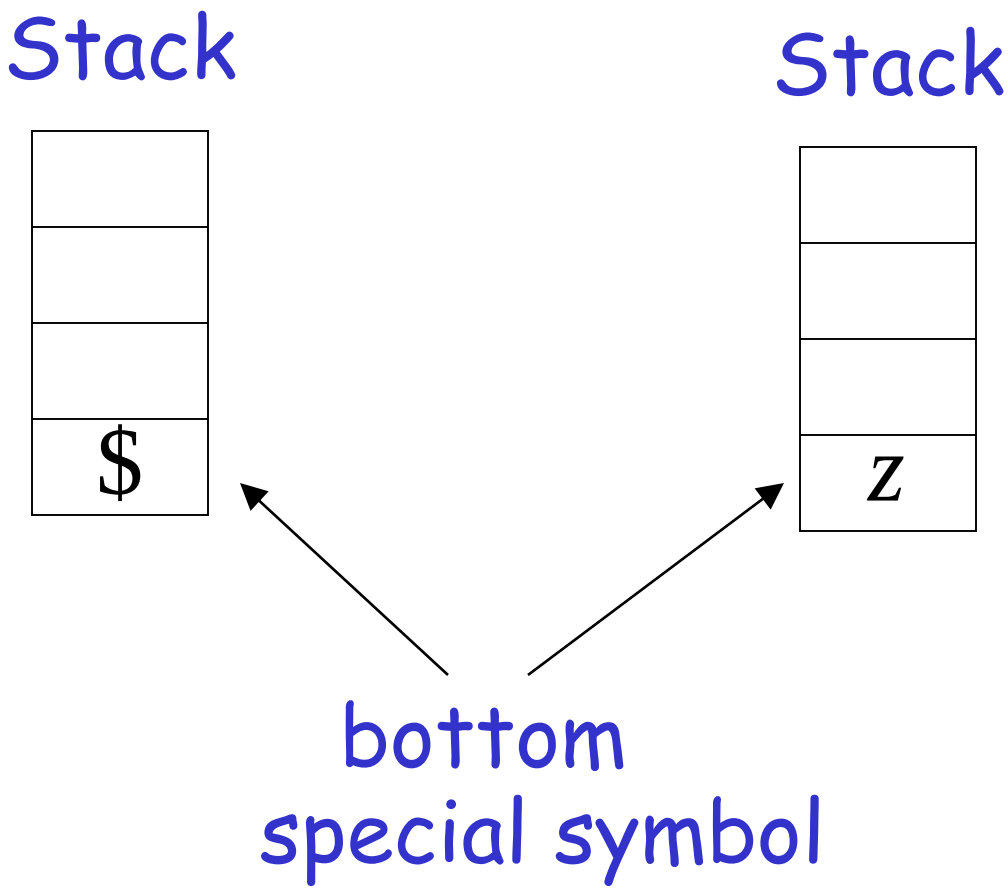
Pushdown Automata
PDAs

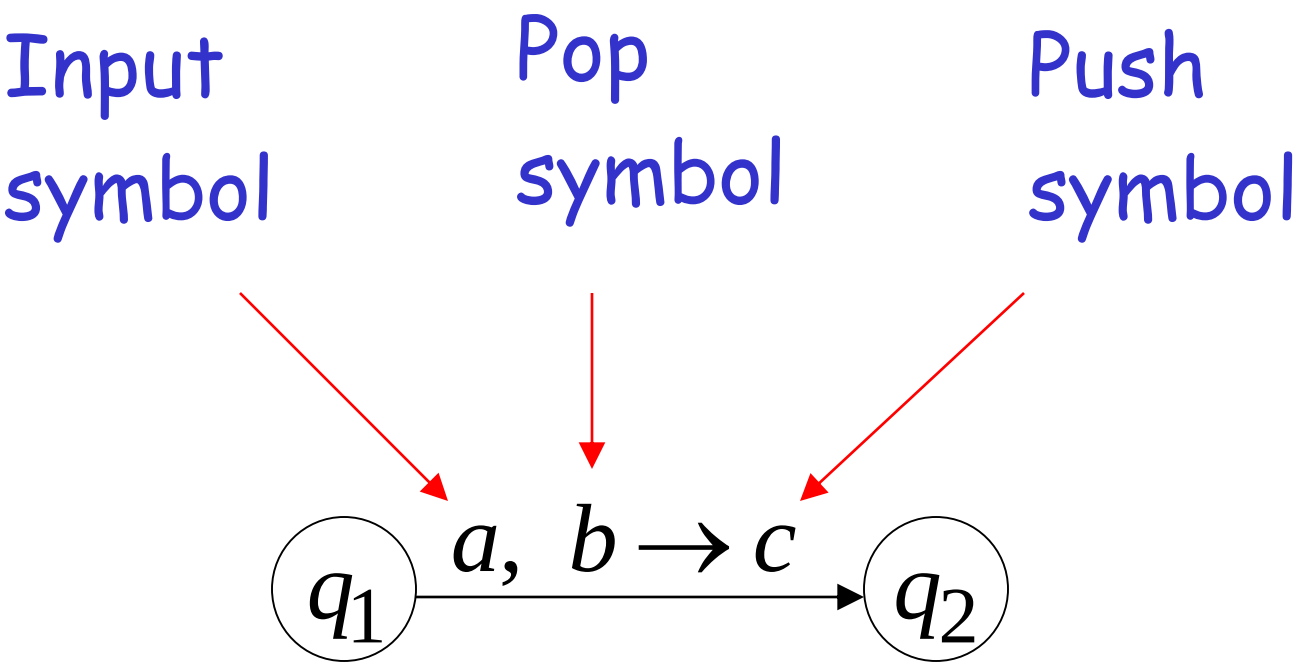
Pushdown Automaton -- PDA

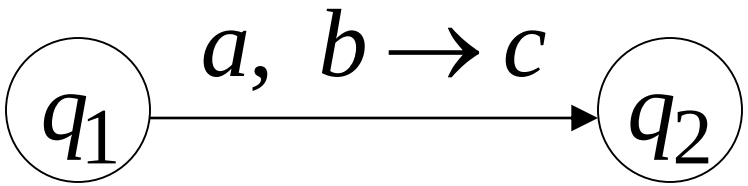


Initial Stack Symbol

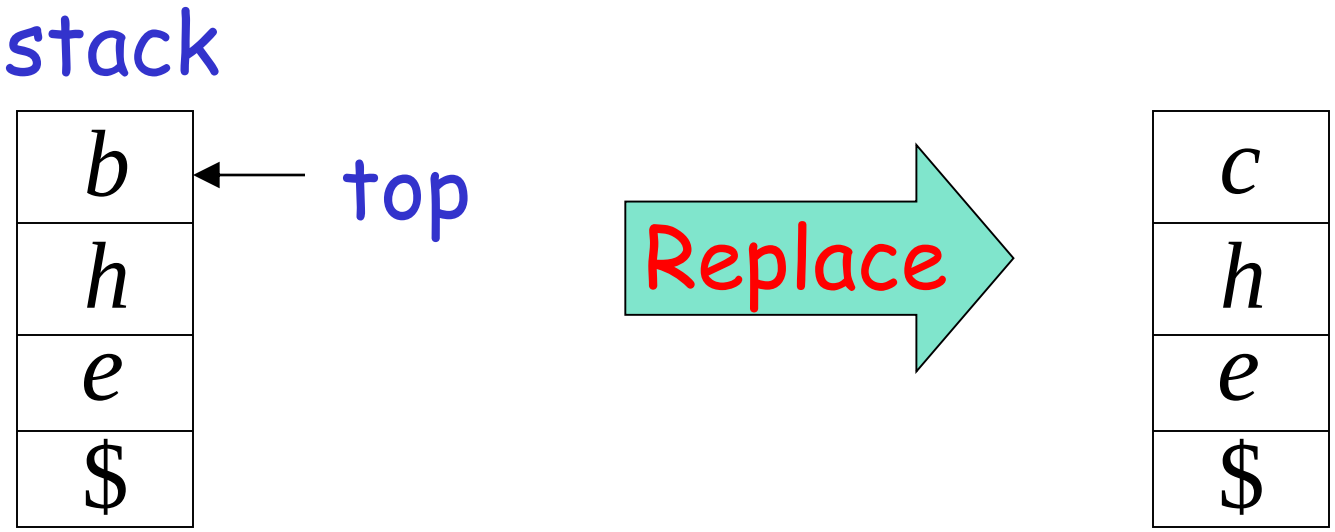


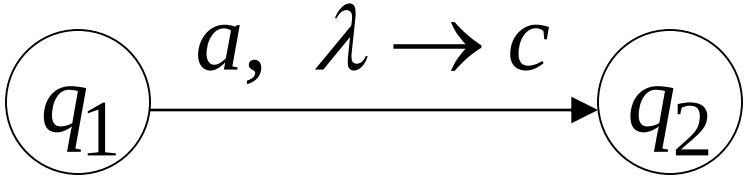
The States



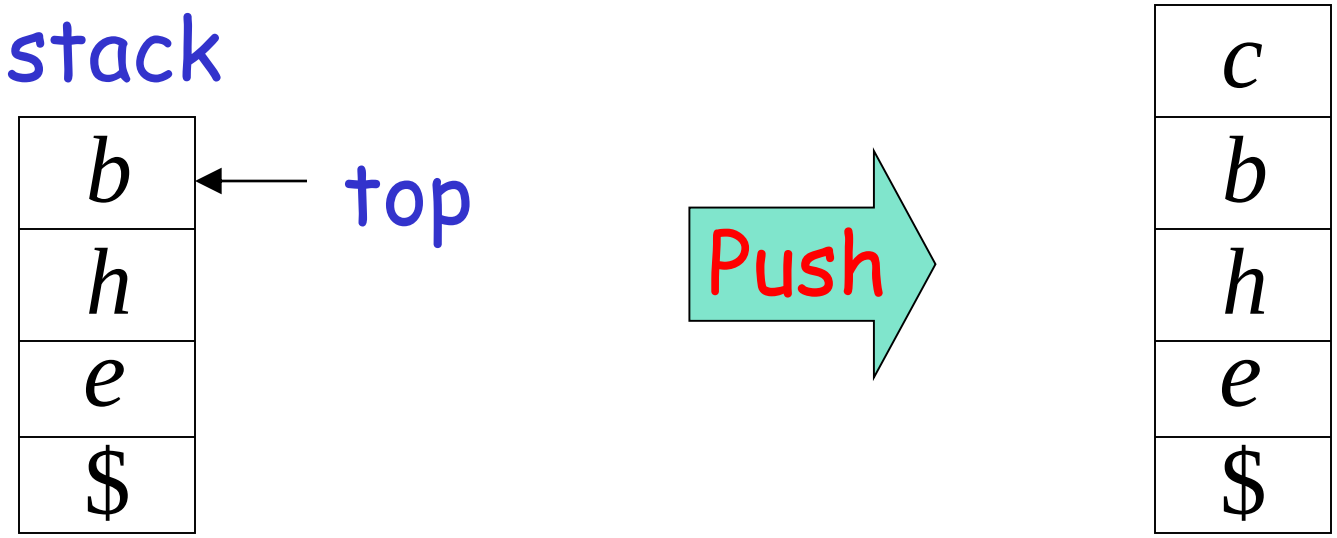

$a,\ b \to c$
q_1
q_2

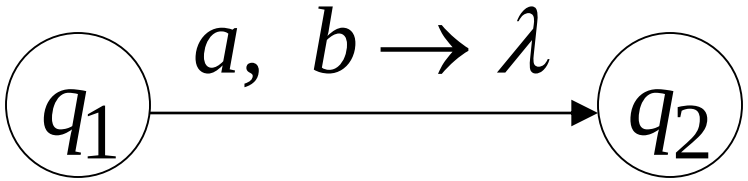

input
…
a
…
…
a
…


stack
b
top
h
e
$
Replace
c
h
e
$

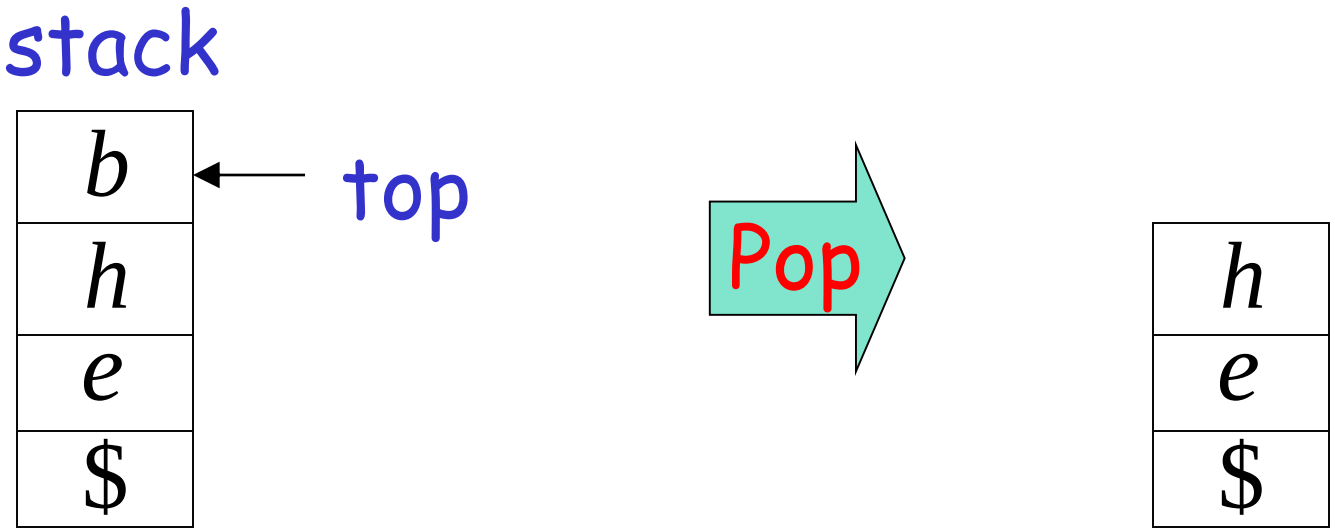

$a,\ \lambda \rightarrow c$
q_1
q_2

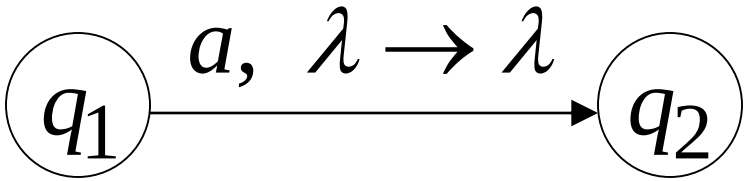

input
…
a
…
…
a
…


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


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


input
…
a
…
…
a
…

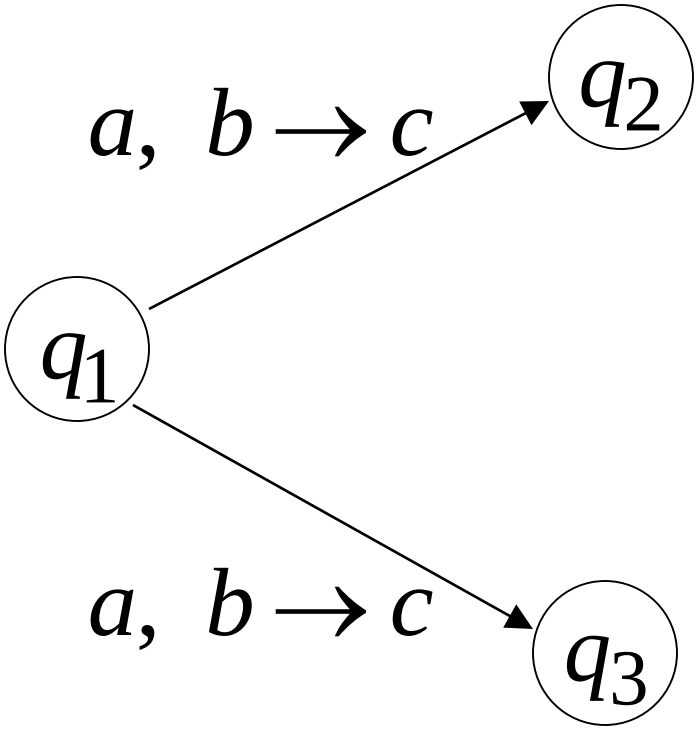

stack
b
top
h
e
$
Pop
h
e
$

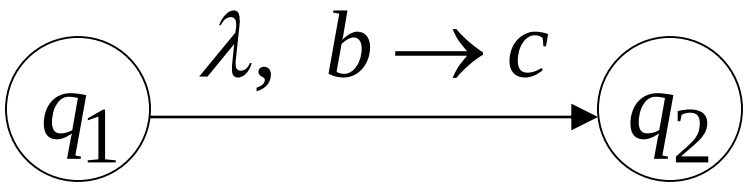

$a,\ \lambda \to \lambda$
q_1
q_2


input
…
a
…
…
a
…


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

Non-Determinism





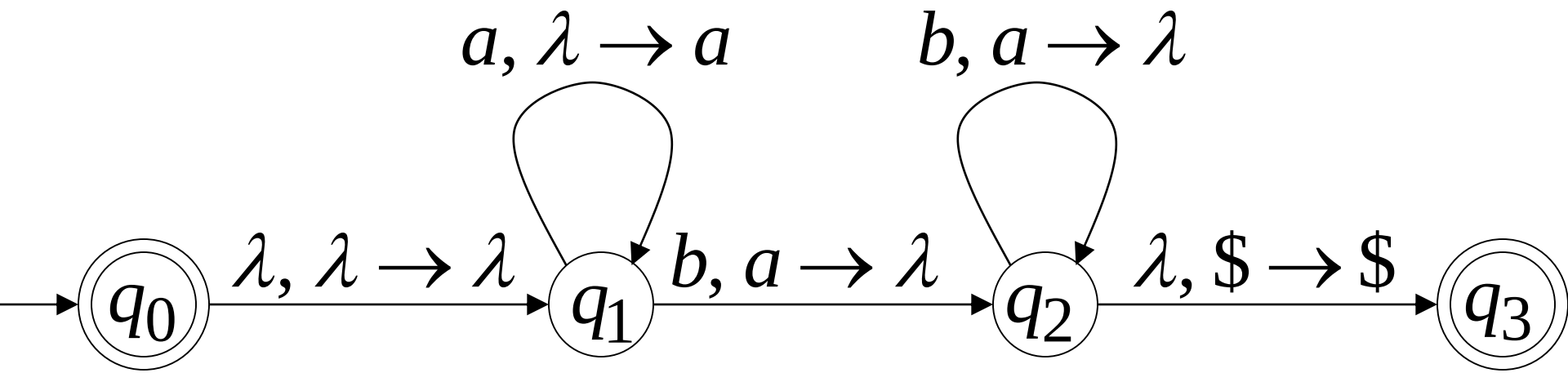
λ – transition

These are allowed transitions in a Non-deterministic PDA (NPDA)

NPDA: Non-Deterministic PDA

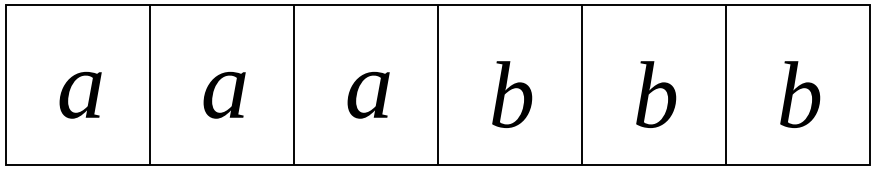
Example:

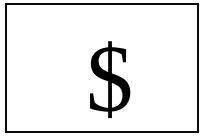
This is not a valid PDA. It is suppose to be a^n b^n. See LInz 6th, Section 7.1, Example 7.2 & 7.3, pages 184-185.



Execution Example: Time 0

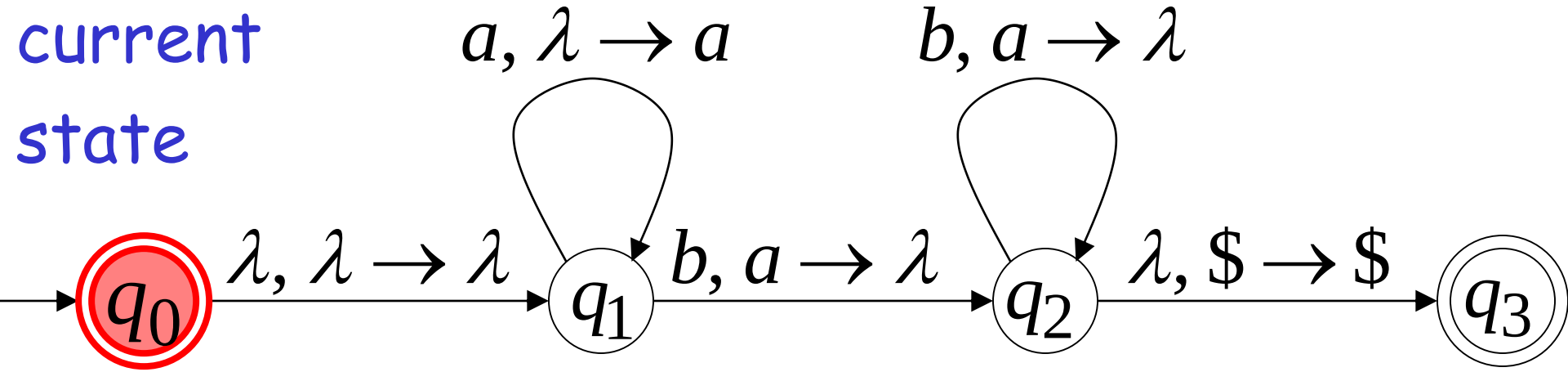
Input





Stack

current state



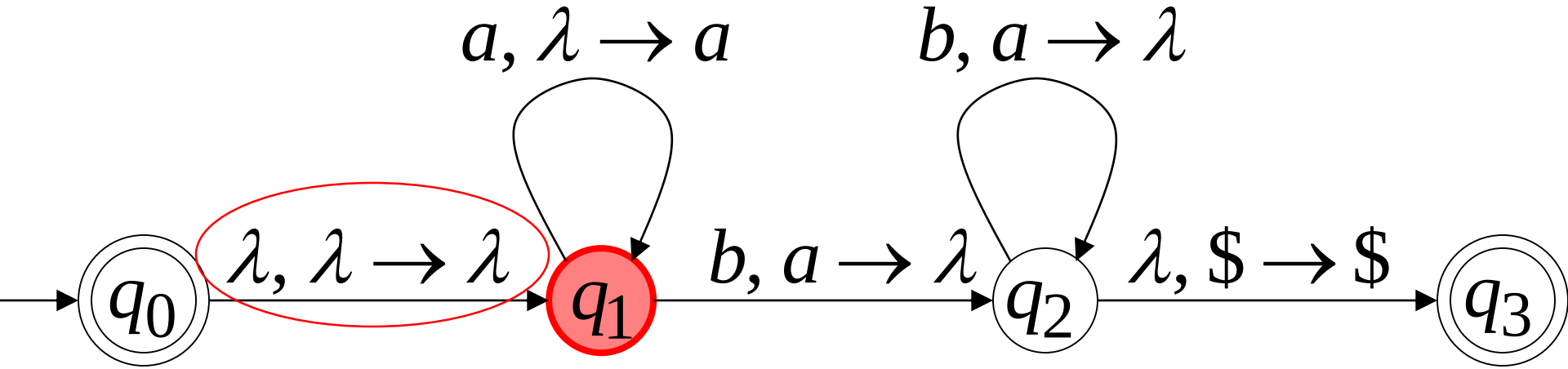
Time 1

Input

a	a	a	b	b	b

$

Stack



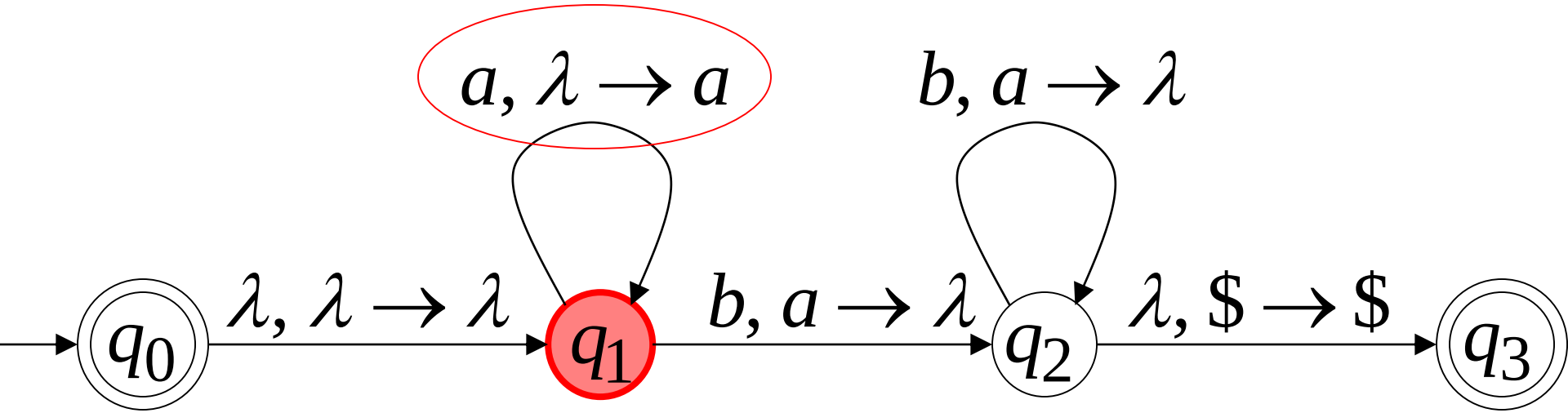
Time 2

Input

a	a	a	b	b	b

a
\$

Stack



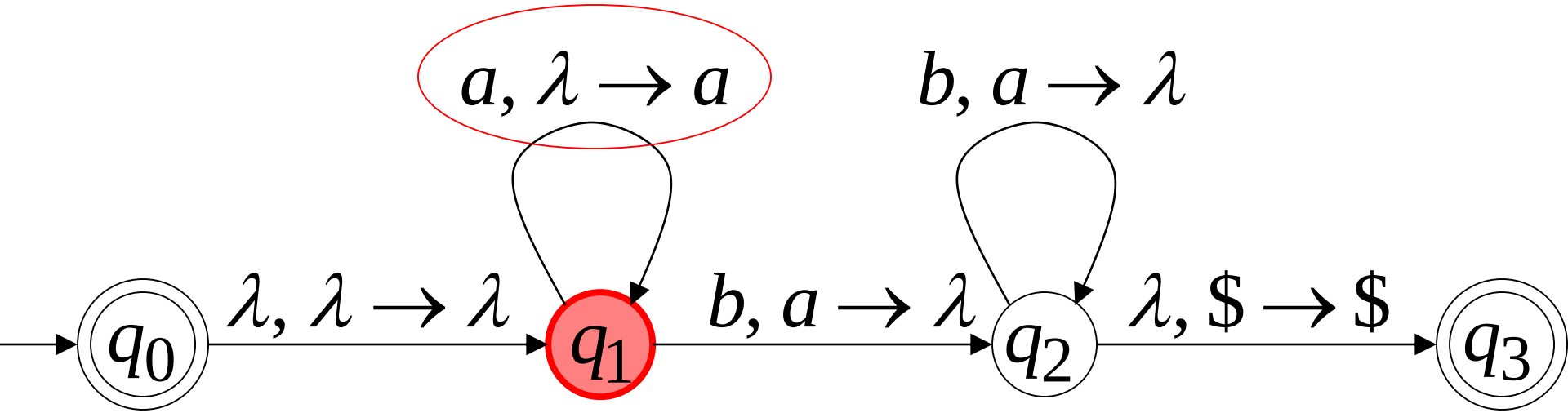
Time 3

Input

a	a	a	b	b	b

a
a
$

Stack



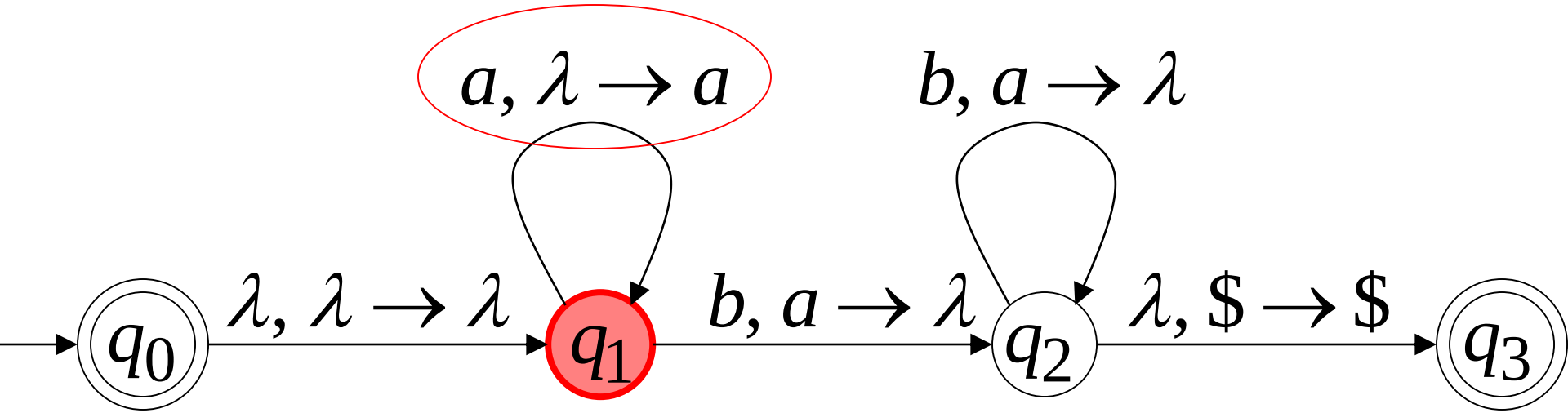
Time 4

Input

a	a	a	b	b	b

a
a
a
$\$$

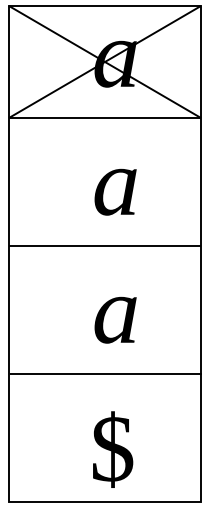
Stack



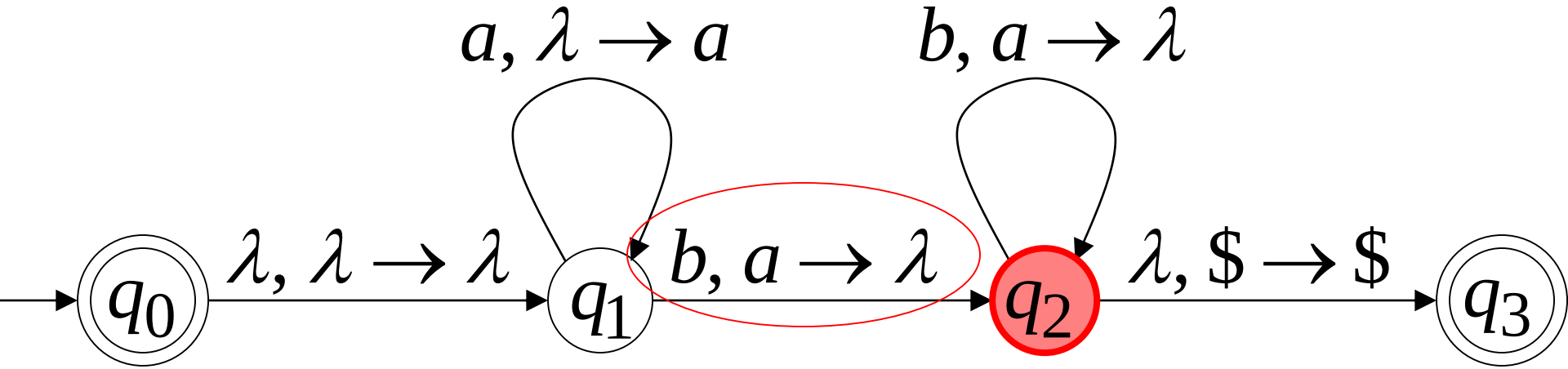
Time 5

Input

a	a	a	b	b	b

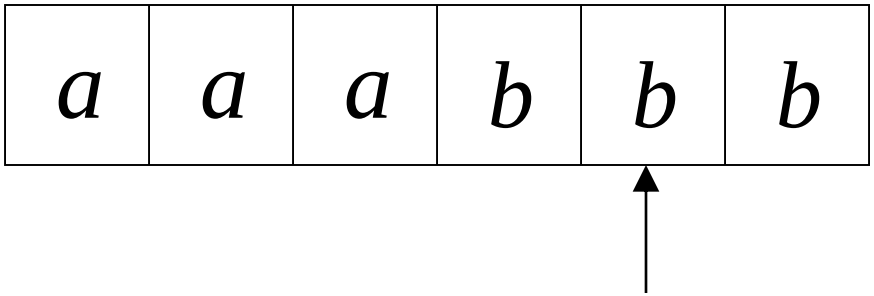


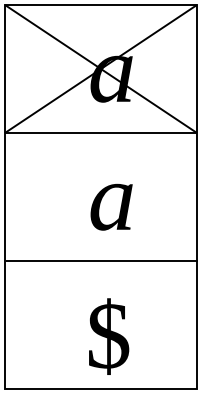
Stack



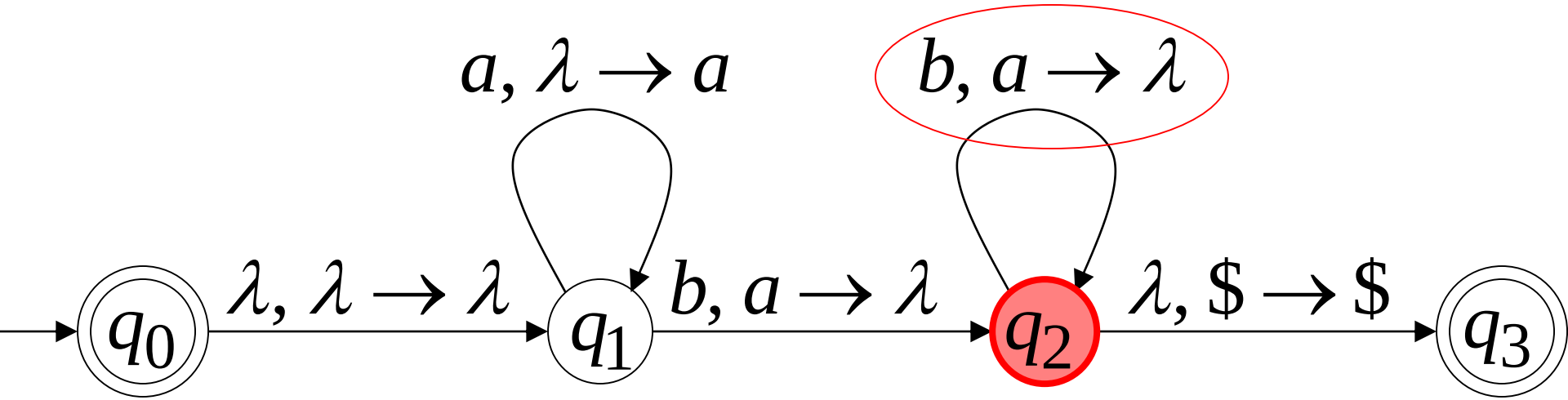
Time 6

Input



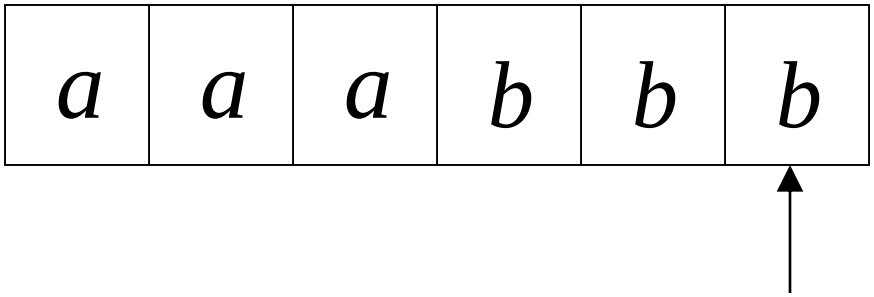


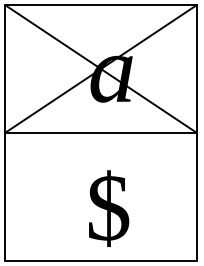
Stack



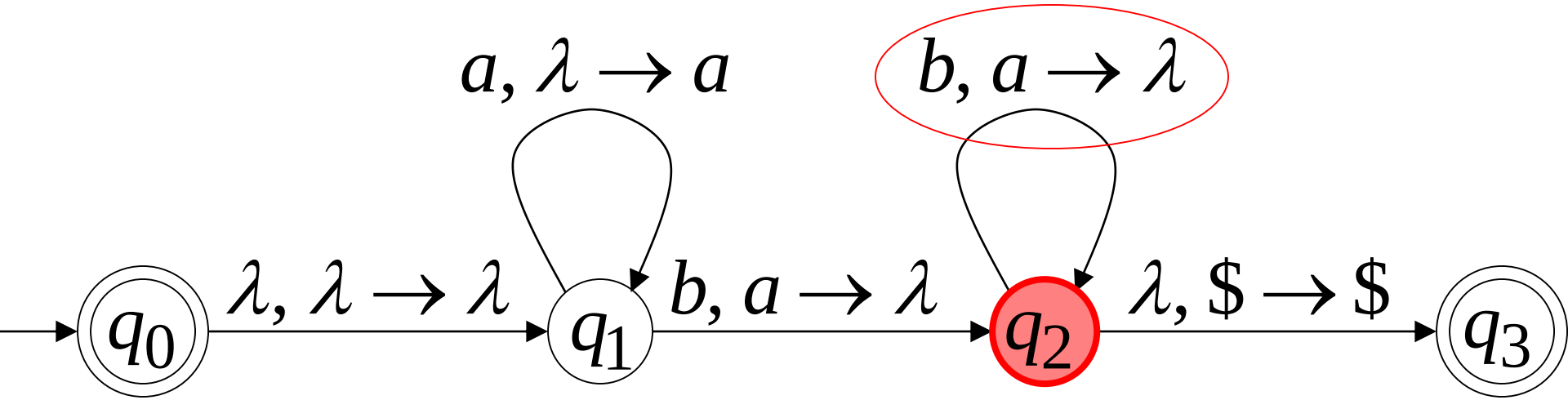
Time 7

Input



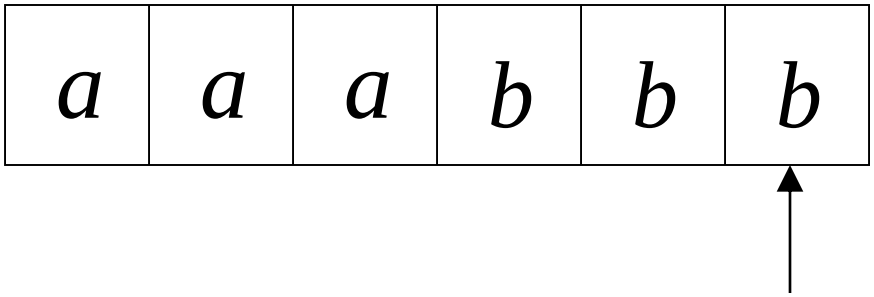


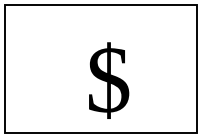
Stack



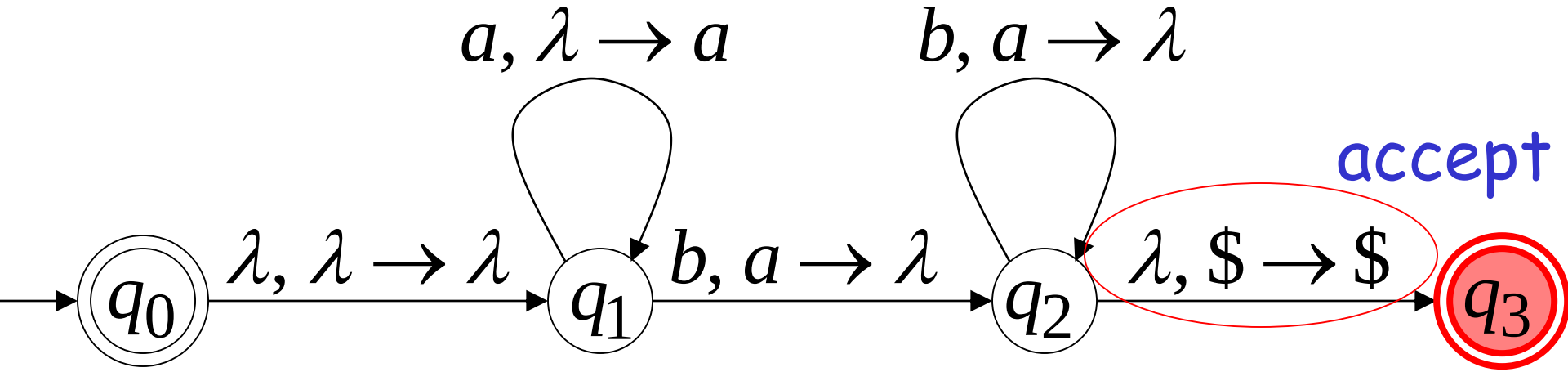
Time 8

Input





Stack



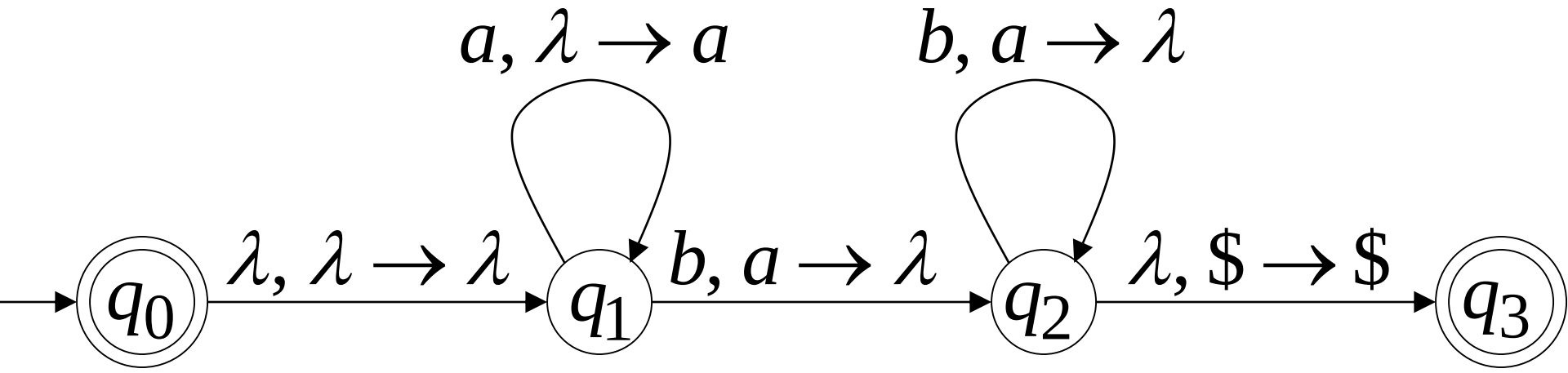
accept

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

- All the input is consumed
- The last state is a final state

At the end of the computation,
we do not care about the stack contents

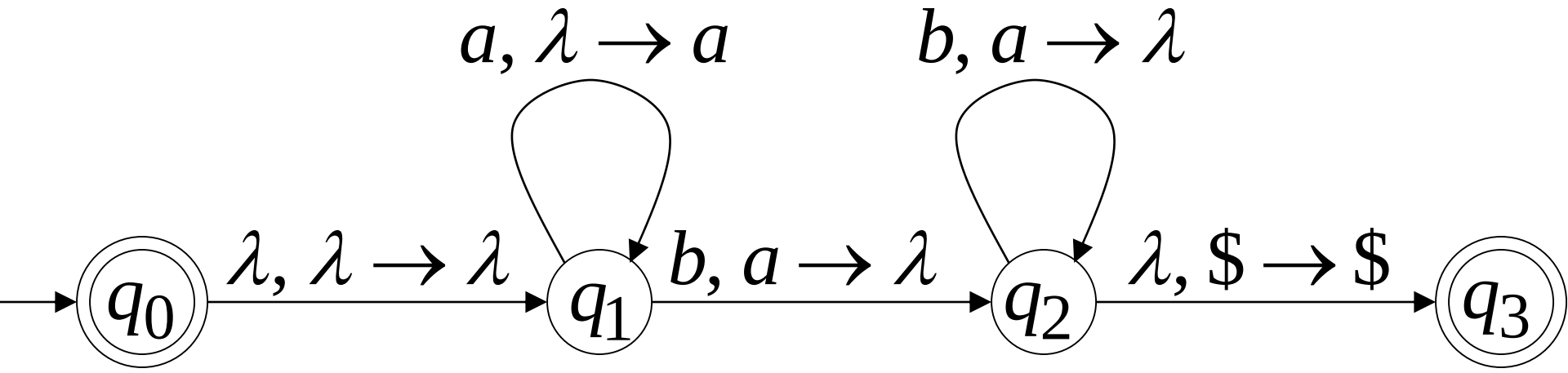
The input string $aaabbb$
is accepted by the NPDA:



In general,

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

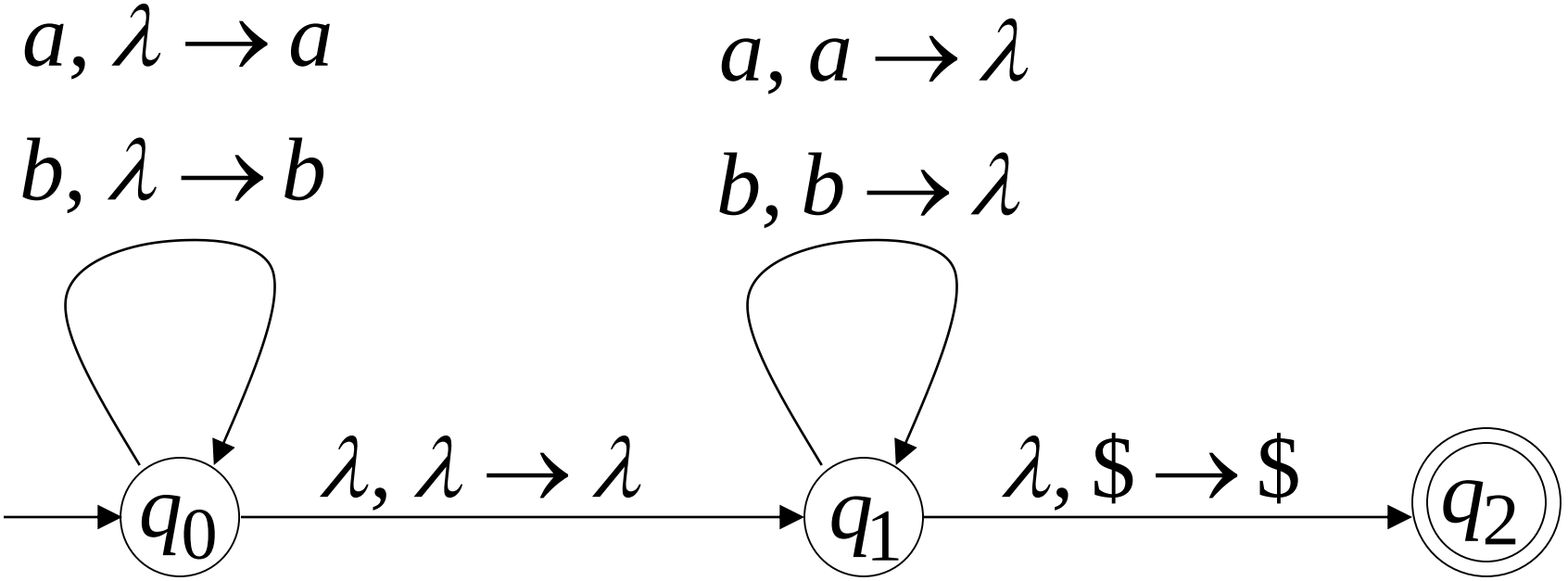
is the language accepted by the NPDA:



Another NPDA example

NPDA M

$$L(M) = \{ww^R\}$$



Execution Example: Time 0

Input

a	b	b	a

↑

$

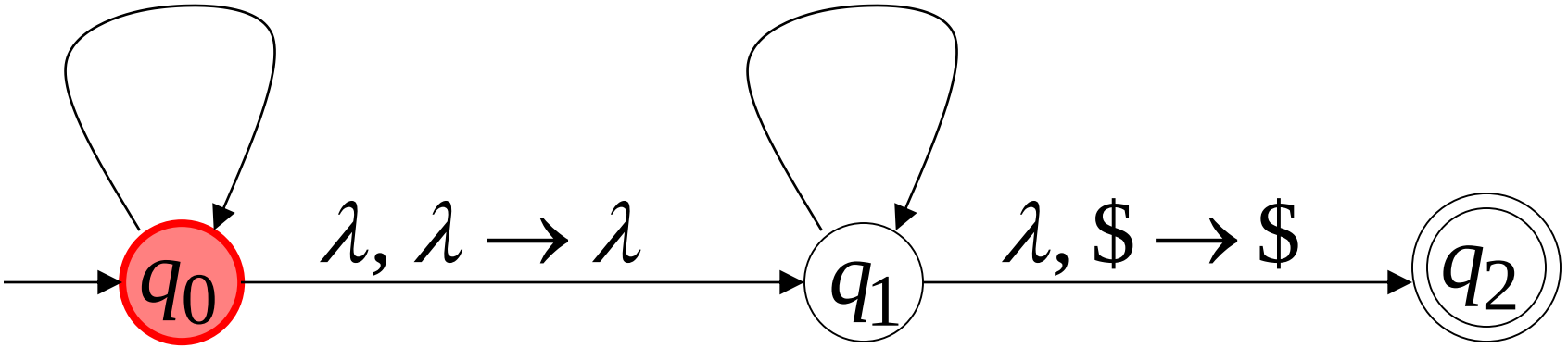
Stack

$a, \lambda \to a$

$b, \lambda \to b$

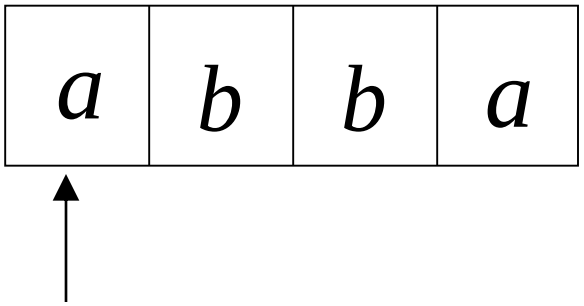
$a, a \to \lambda$

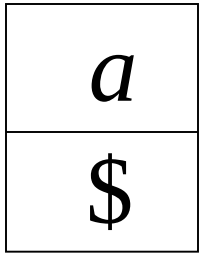
$b, b \to \lambda$



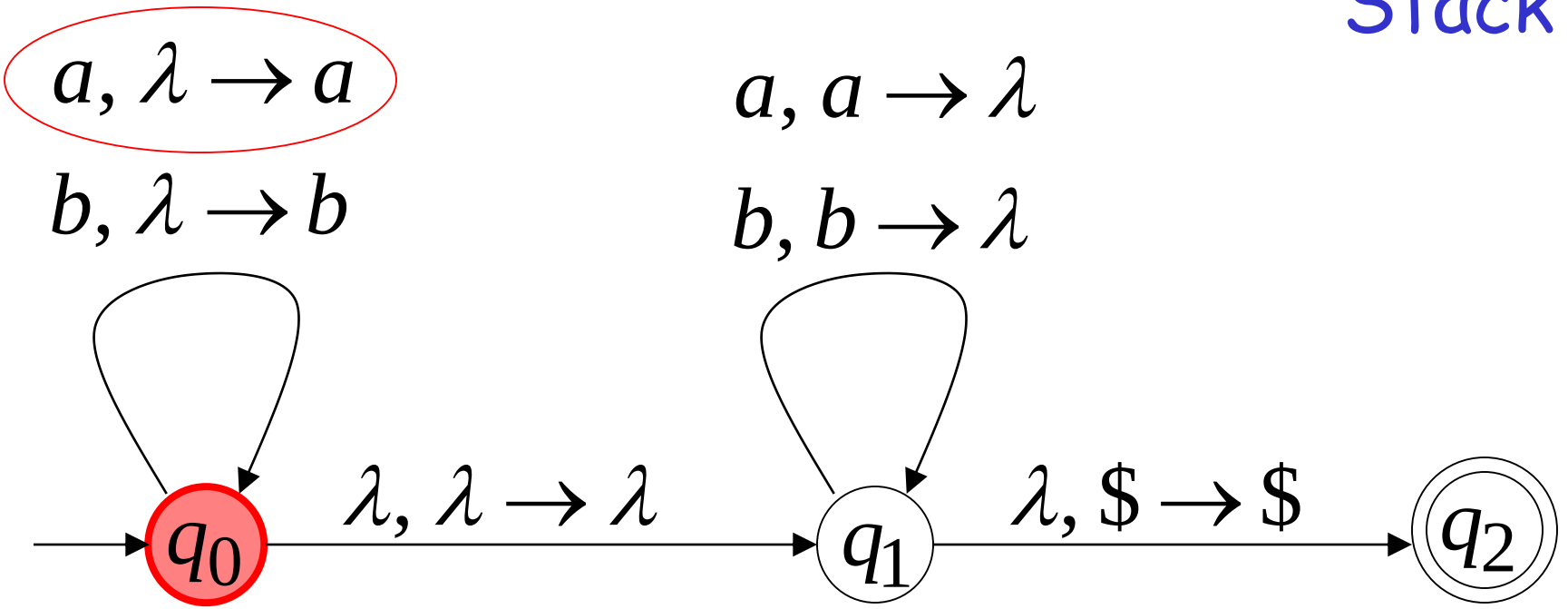
Time 1

Input



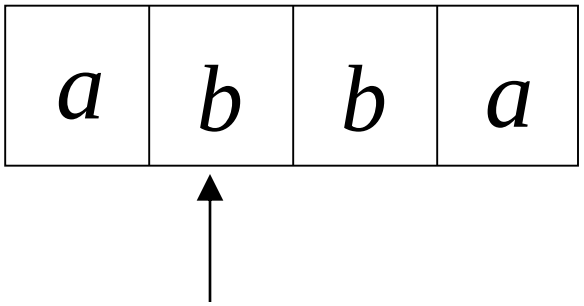


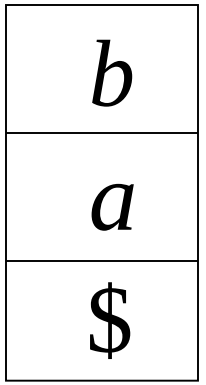
Stack



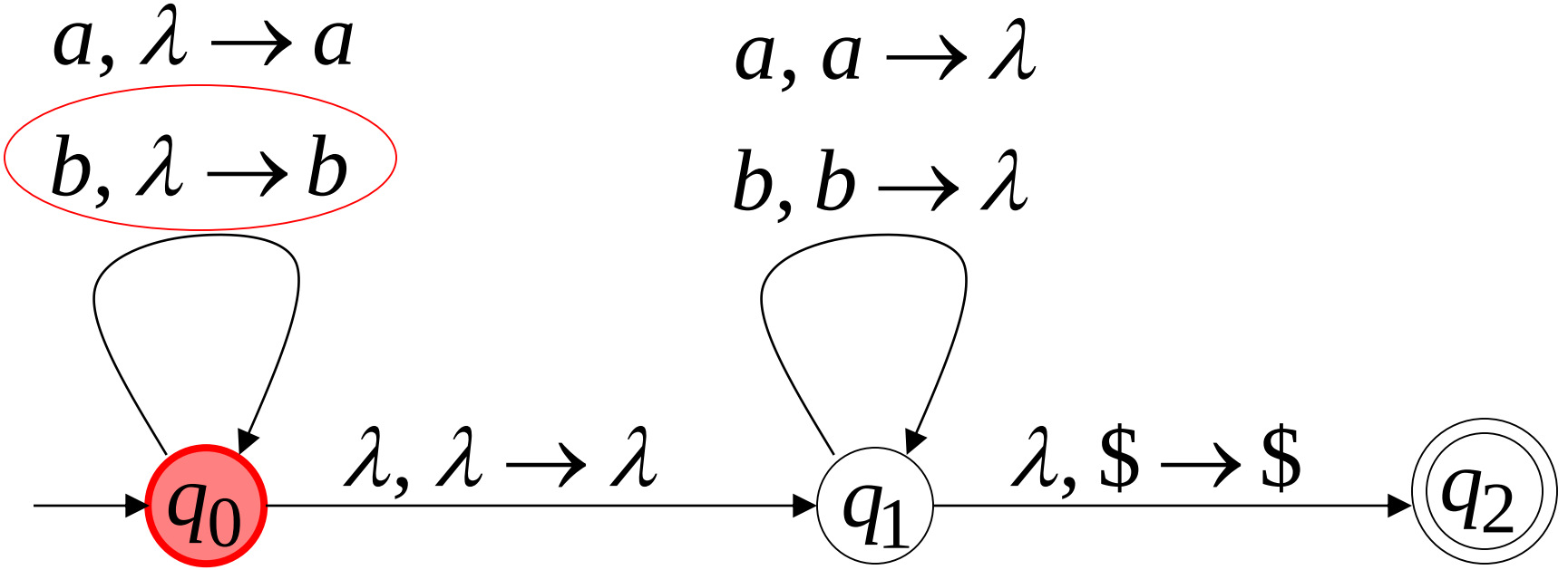
Time 2

Input





Stack



Time 3

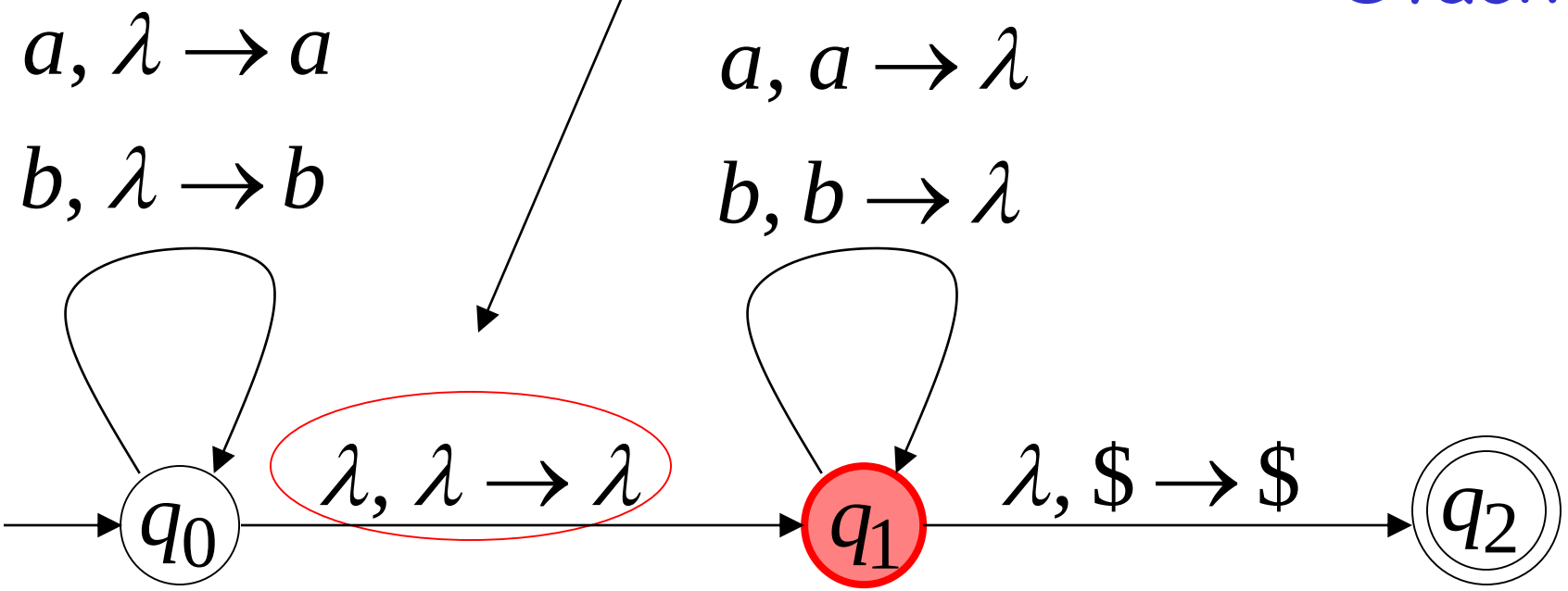
Input

a	b	b	a

Guess the middle of string

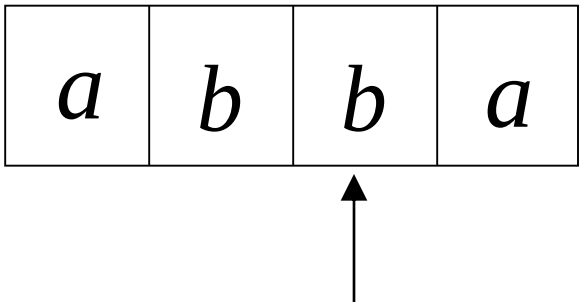
b
a
\$

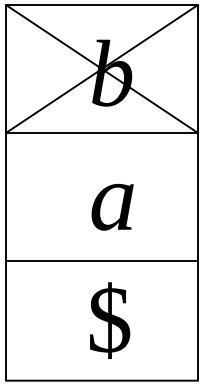
Stack



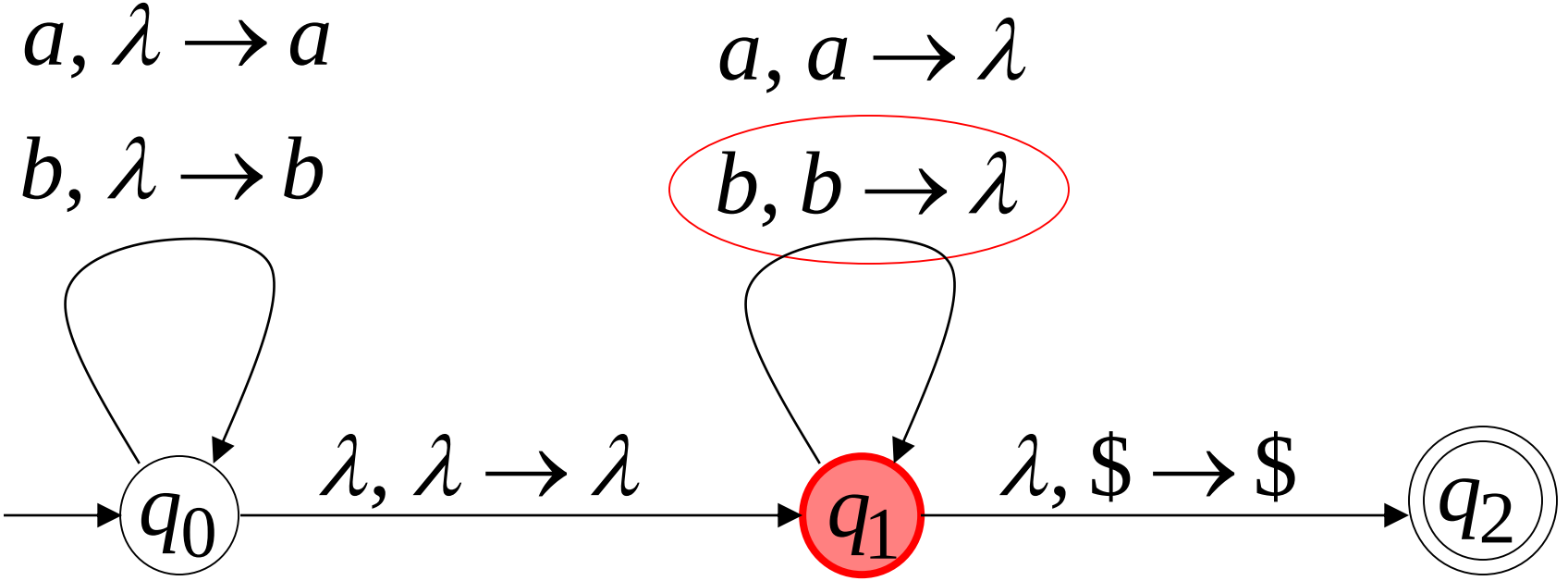
Time 4

Input



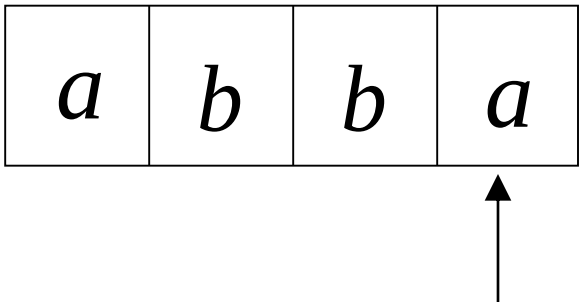


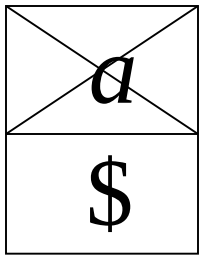
Stack



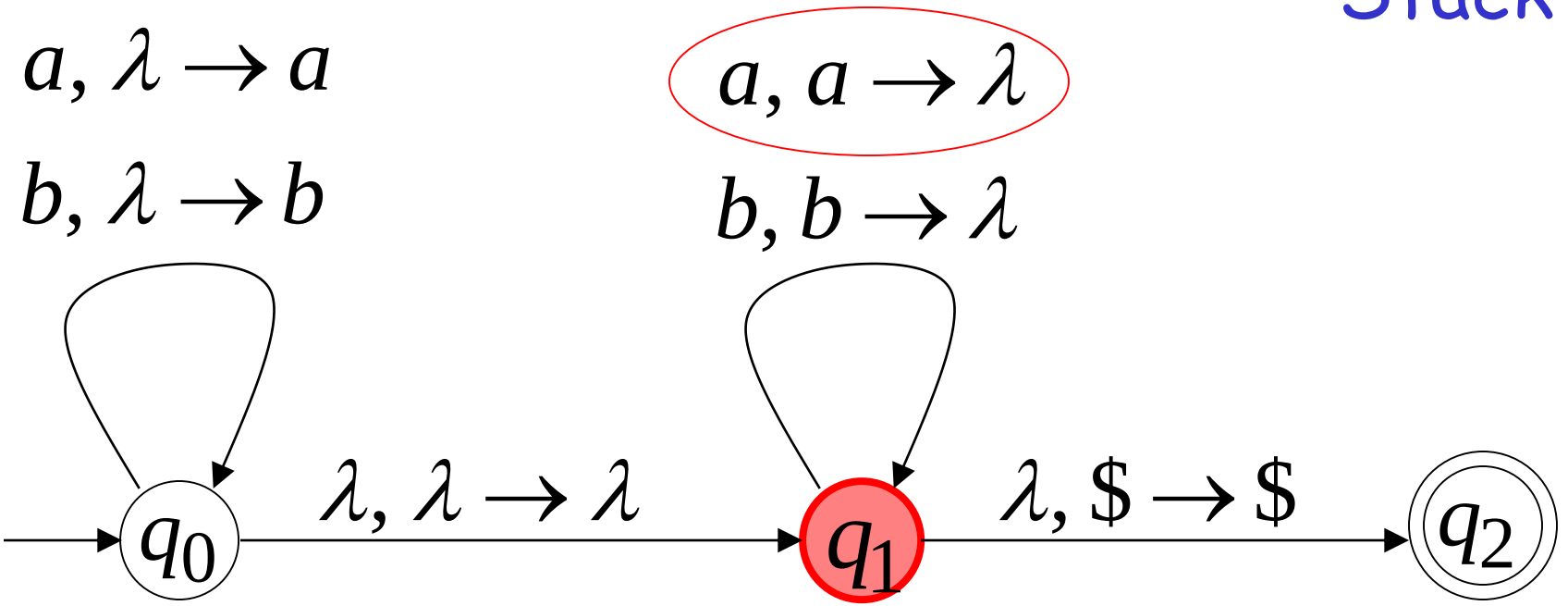
Time 5

Input





Stack



Time 6

Input

a	b	b	a

$\$$

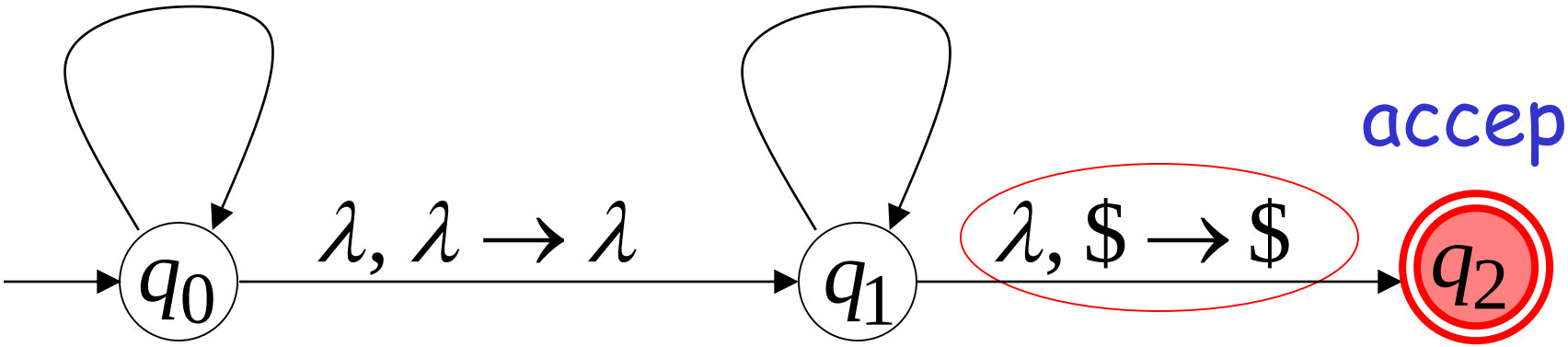
Stack

$a, \lambda \to a$

$b, \lambda \to b$

$a, a \to \lambda$

$b, b \to \lambda$



accept

Rejection Example:

Time 0

Input

a	b	b	b

$\$$

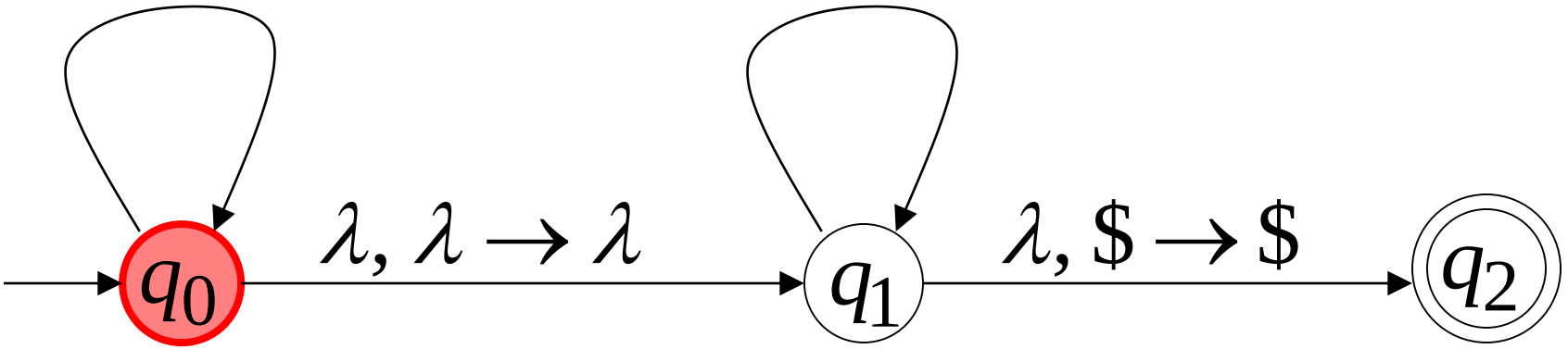
Stack

$a, \lambda \to a$

$b, \lambda \to b$

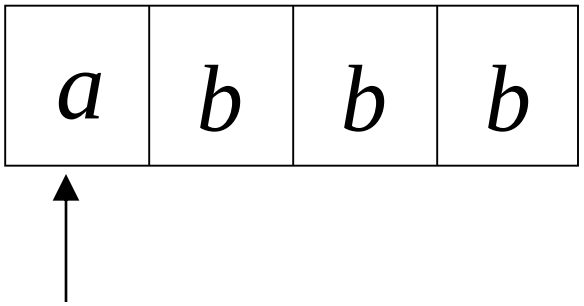
$a, a \to \lambda$

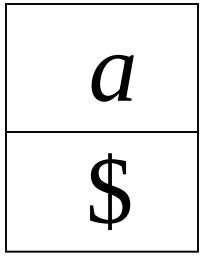
$b, b \to \lambda$



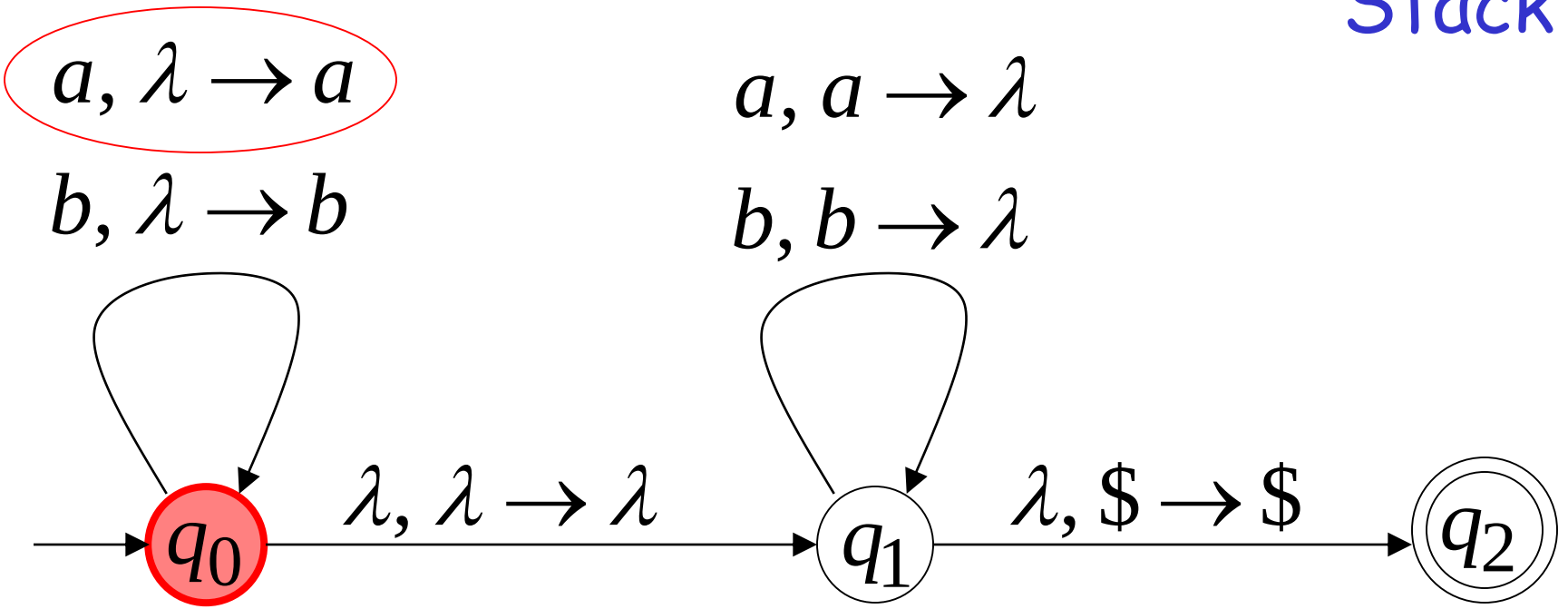
Time 1

Input





Stack



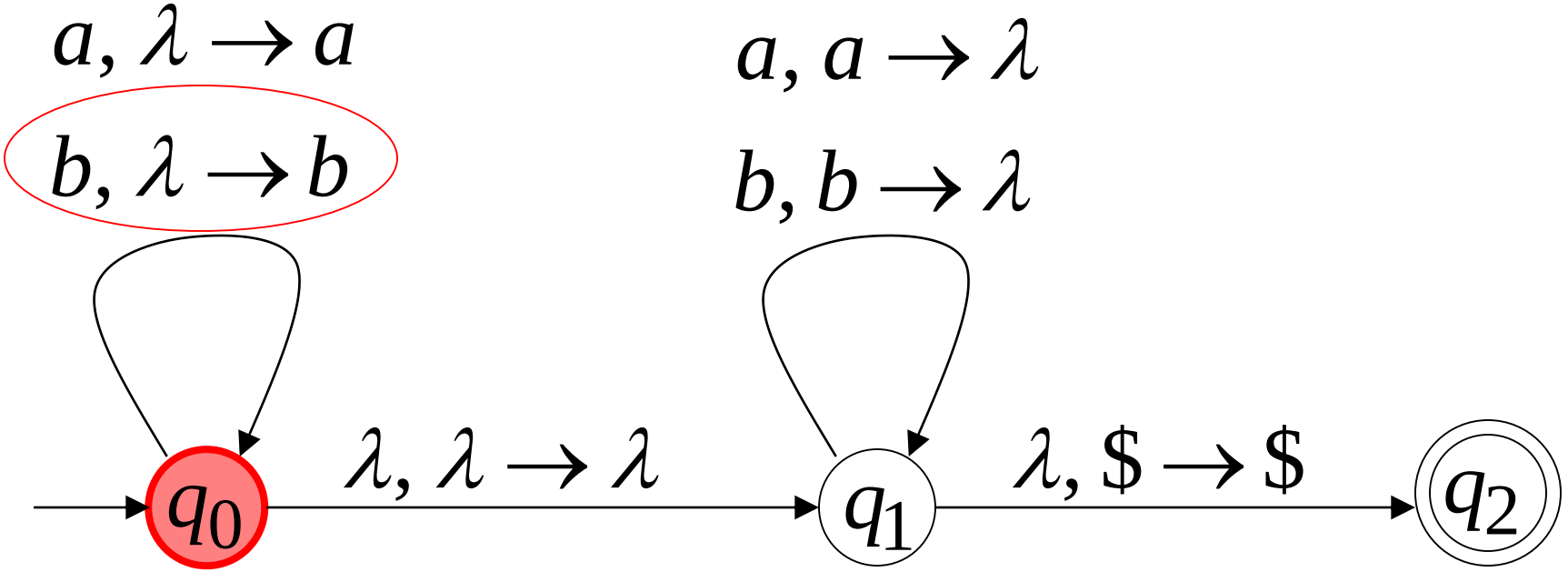
Time 2

Input

a	b	b	b

b
a
\$

Stack



Time 3

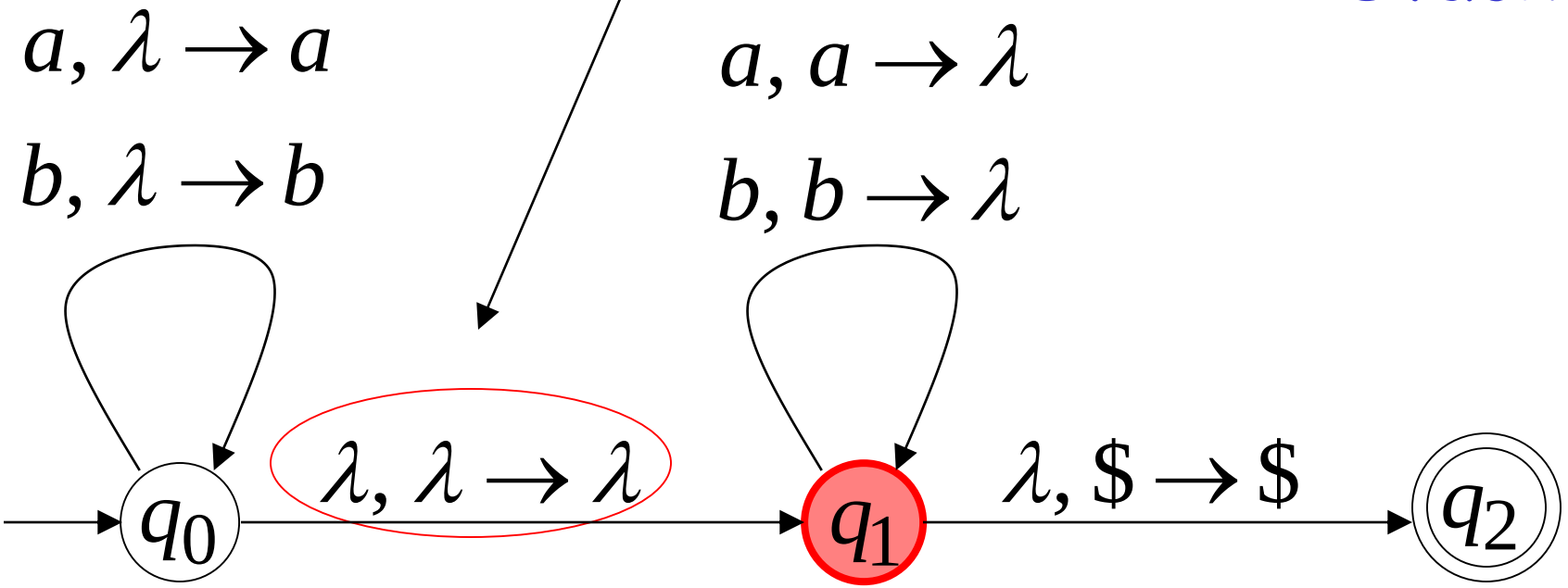
Input

a	b	b	b

Guess the middle of string

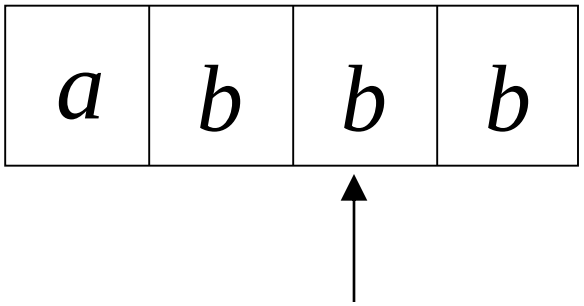
b
a
\$

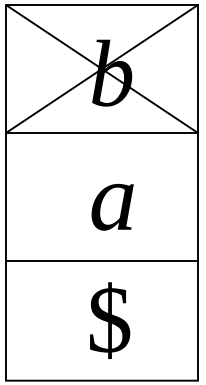
Stack



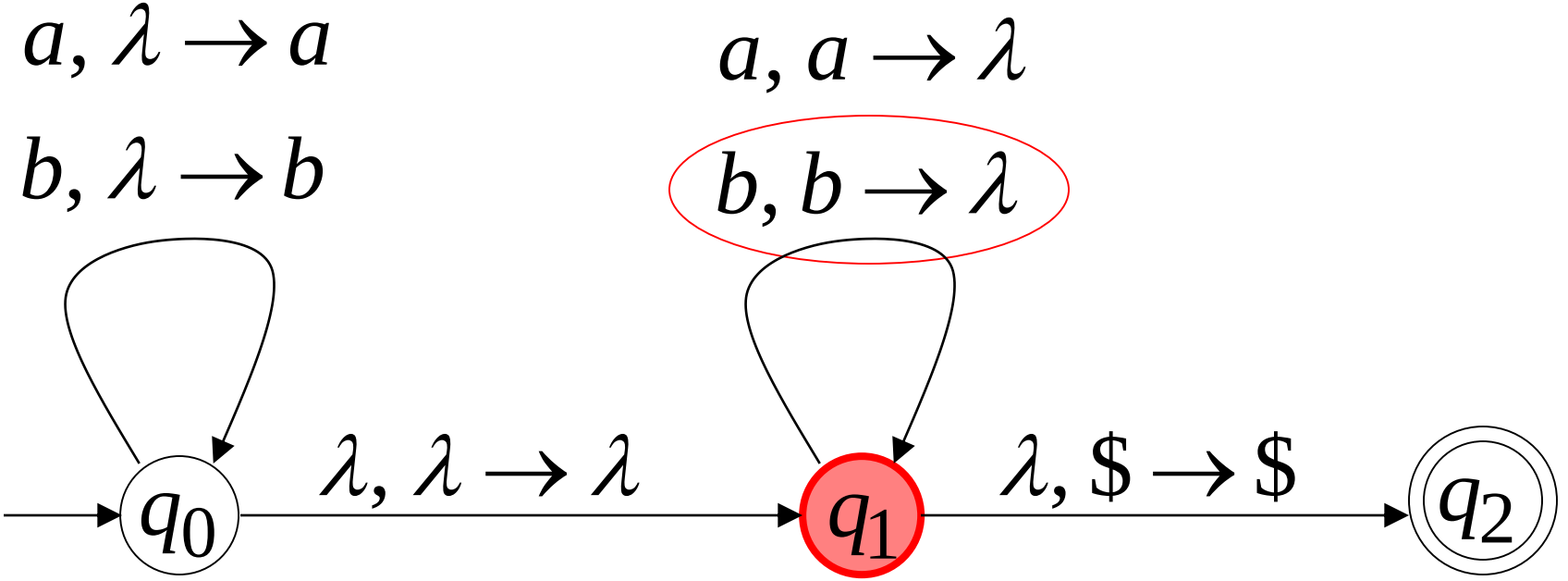
Time 4

Input





Stack



Time 5

There is no possible transition.

Input

a	b	b	b

Input is not consumed

a
$

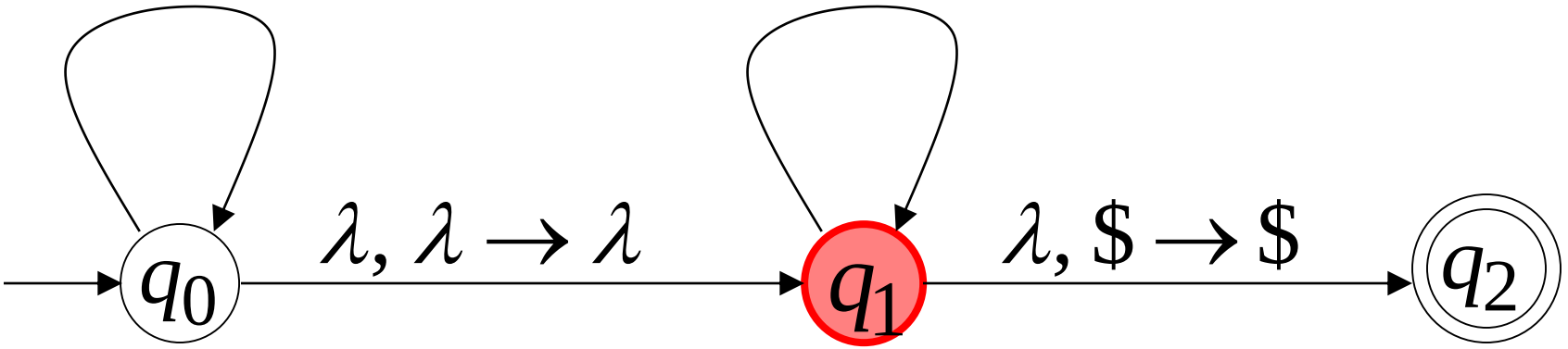
Stack

$a, \lambda \to a$

$b, \lambda \to b$

$a, a \to \lambda$

$b, b \to \lambda$



Another computation on same string:

Input

Time 0

a	b	b	b

$\$$

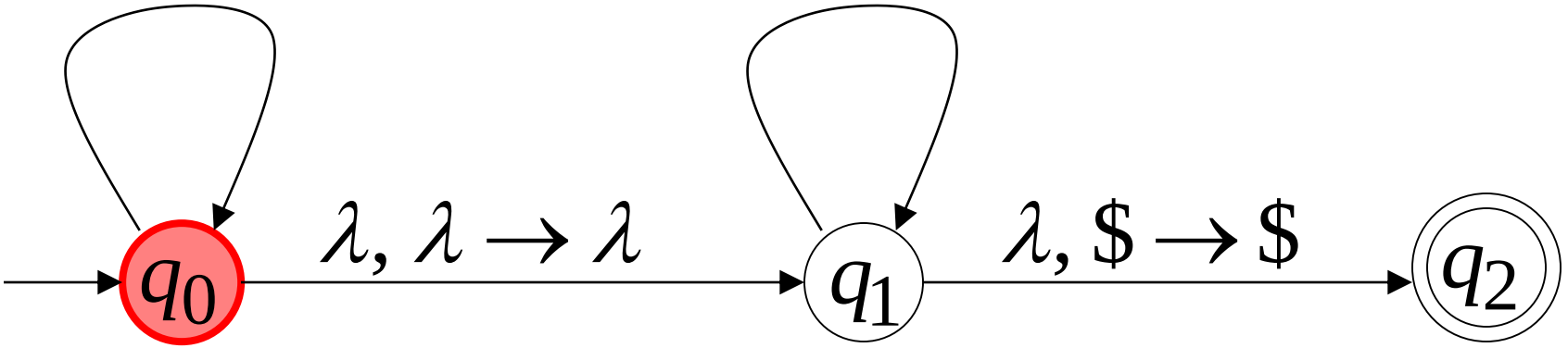
Stack

$a, \lambda \to a$

$b, \lambda \to b$

$a, a \to \lambda$

$b, b \to \lambda$



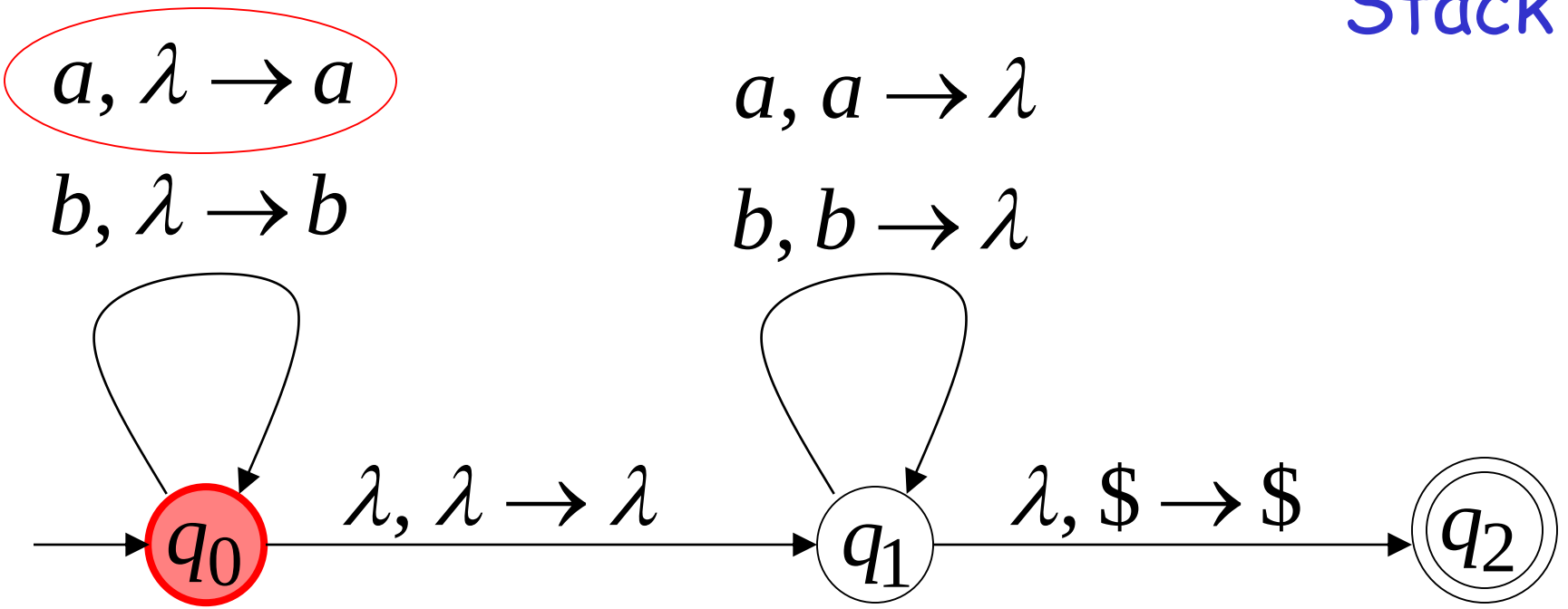
Time 1

Input

a	b	b	b

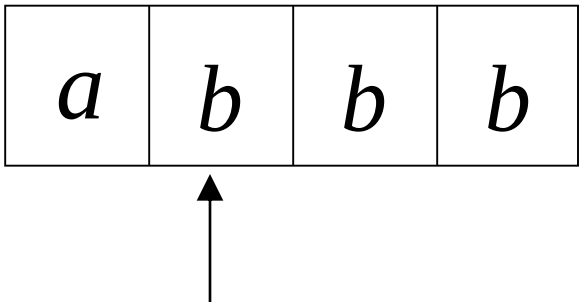
a
$

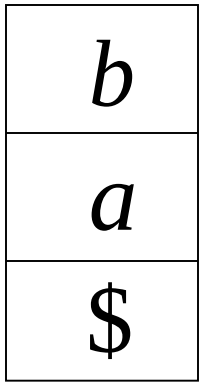
Stack



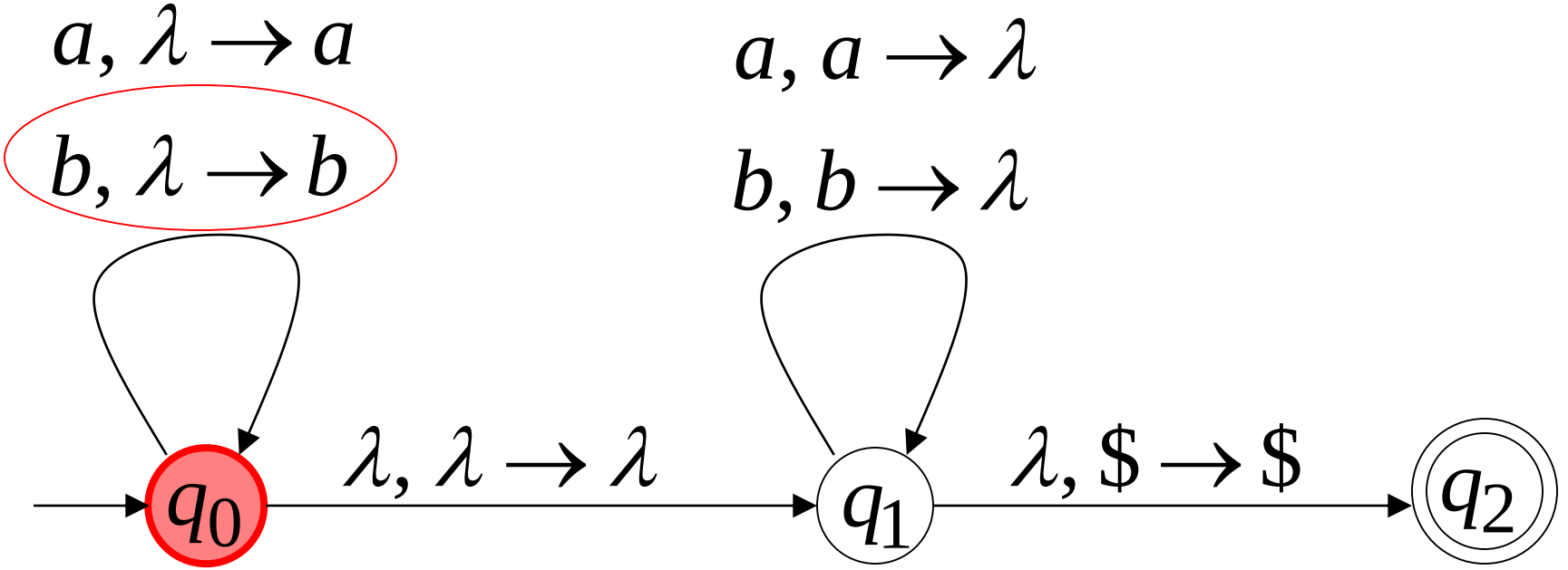
Time 2

Input





Stack



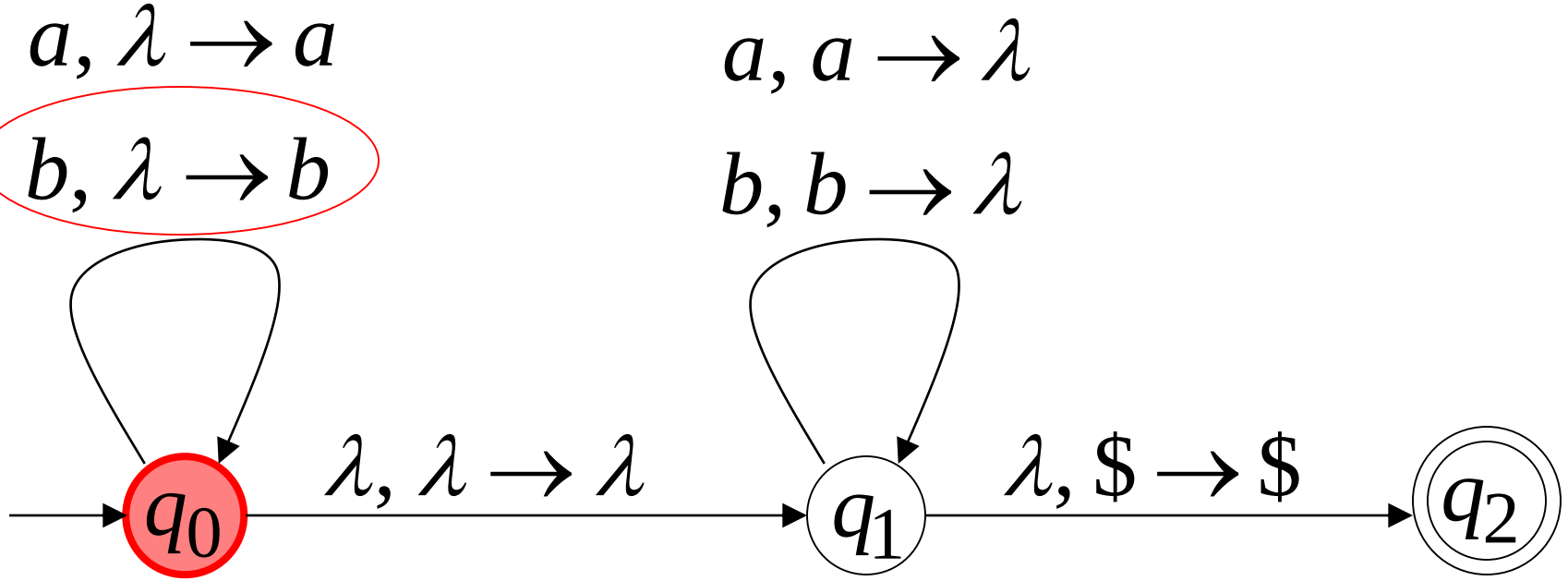
Time 3

Input

a	b	b	b

b
b
a
$\$$

Stack



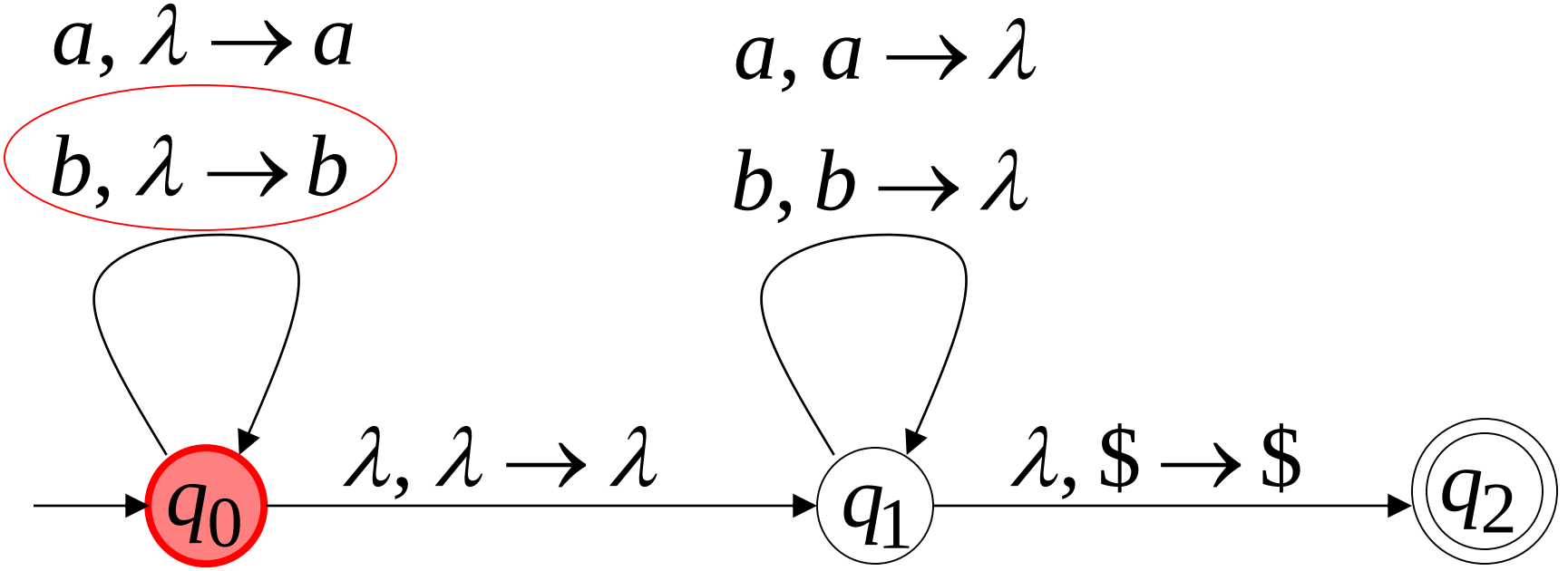
Time 4

Input

a	b	b	b

b
b
b
a
\$

Stack



Time 5

Input

a	b	b	b

No final state is reached

b
b
b
a
\$

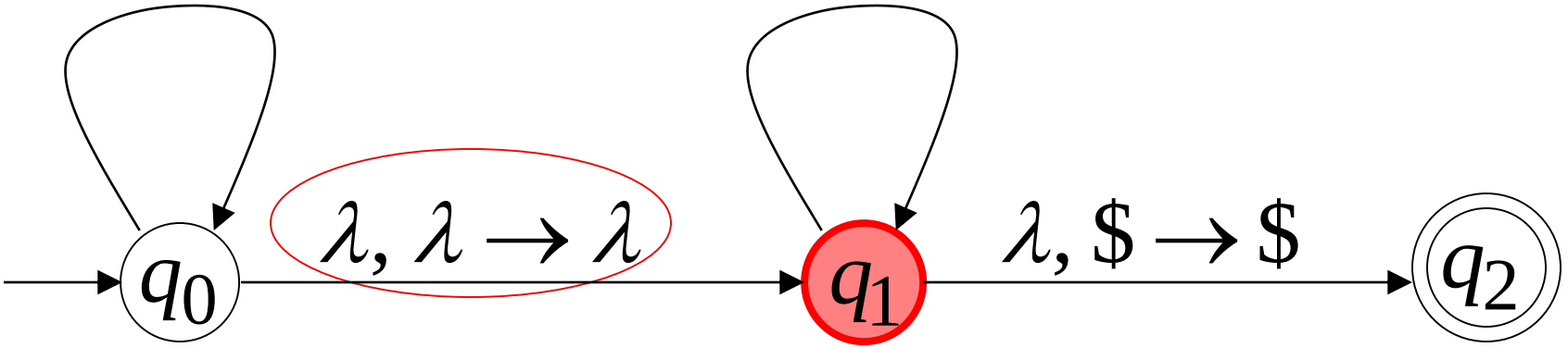
Stack

$a, \lambda \to a$

$b, \lambda \to b$

$a, a \to \lambda$

$b, b \to \lambda$



There is no computation that accepts string $abbb$

$$abbb \notin L(M)$$