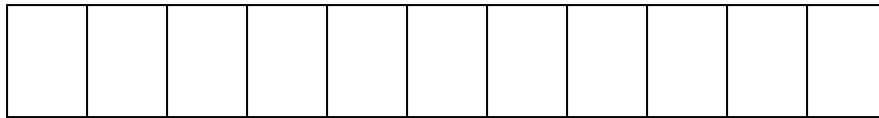


Pushdown Automata

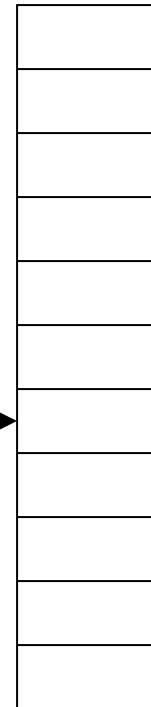
PDA's

Pushdown Automaton -- PDA

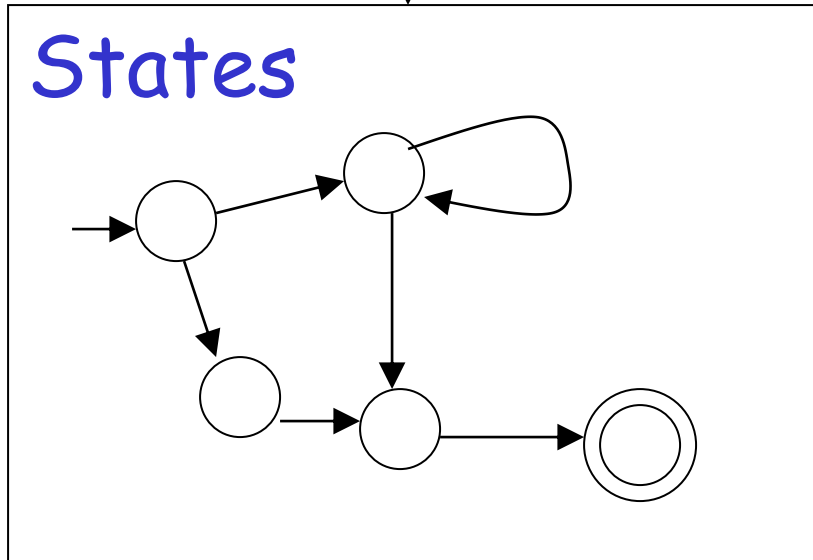
Input String



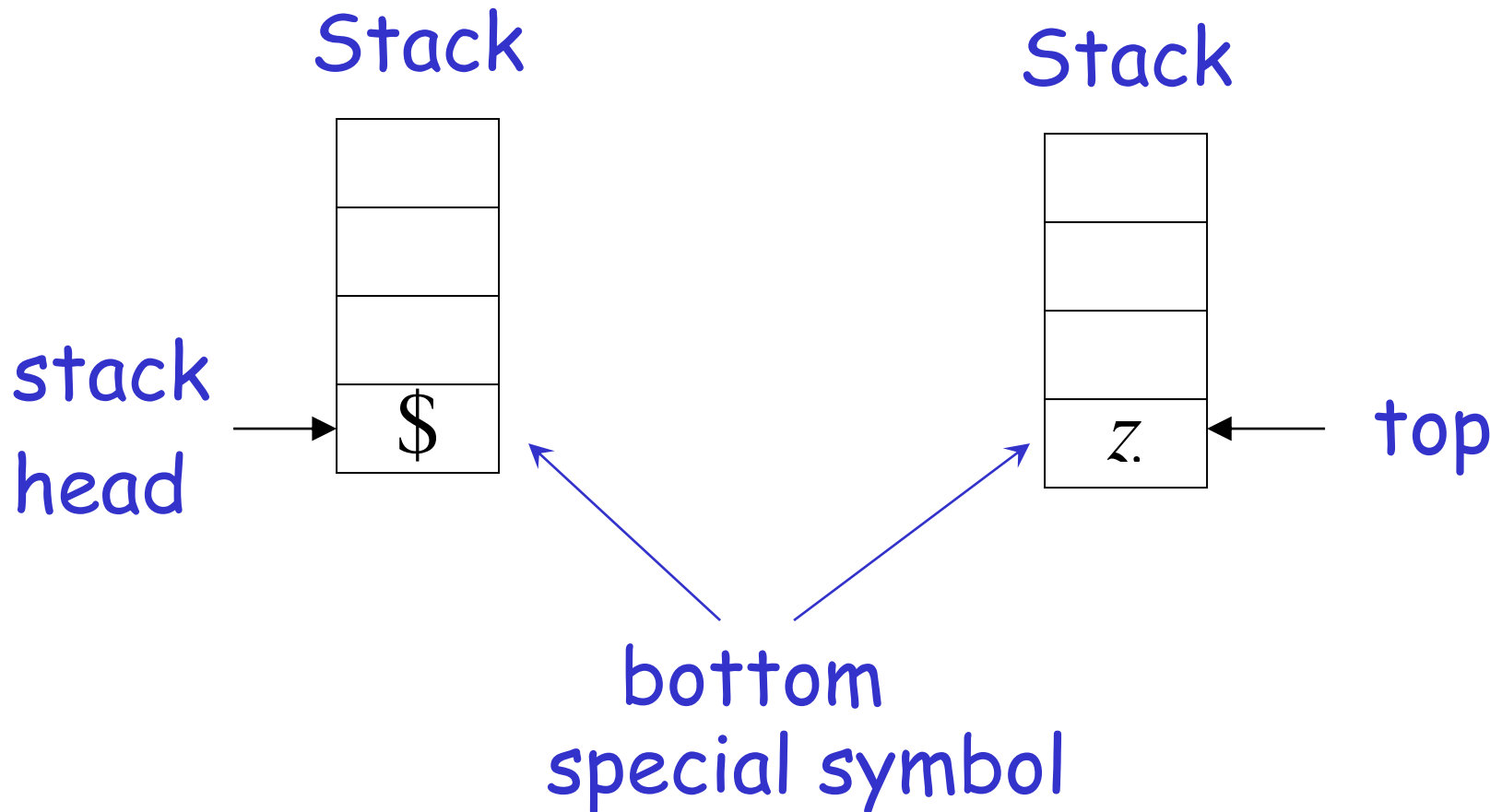
Stack



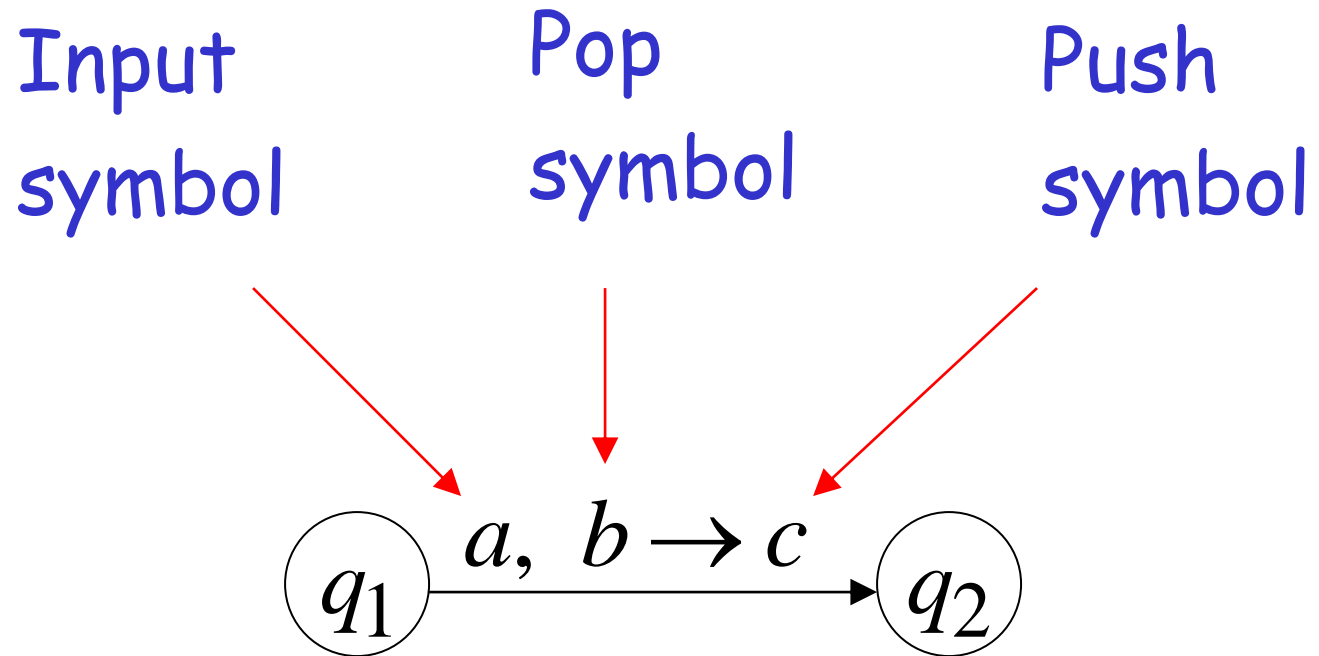
States

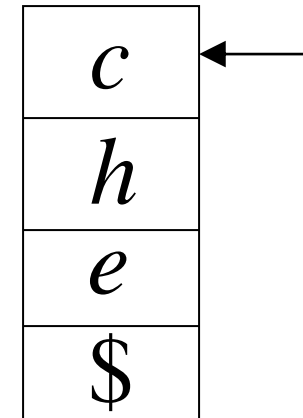
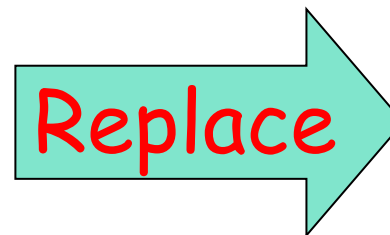
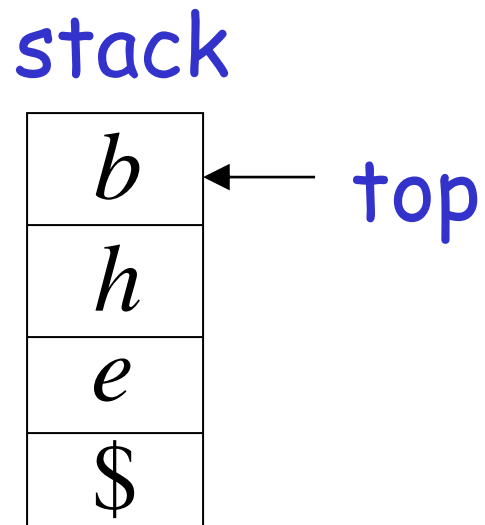
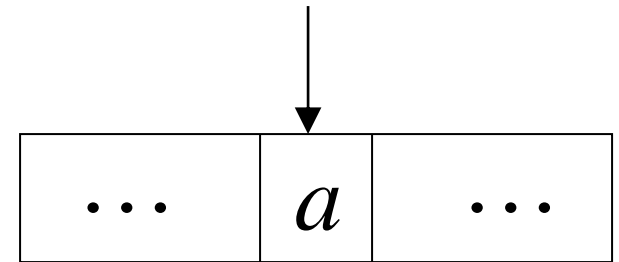
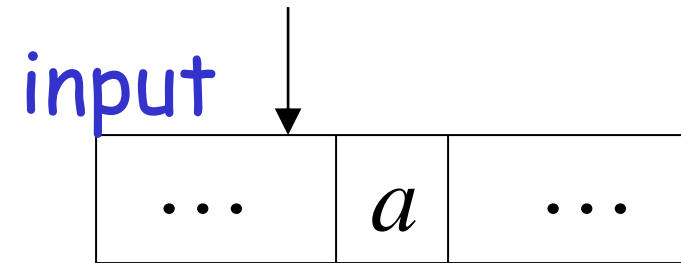
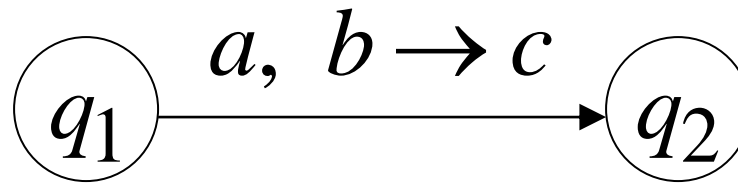


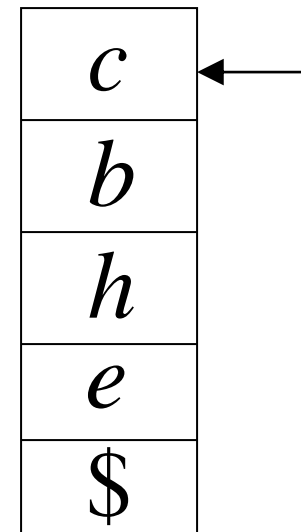
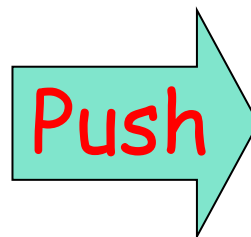
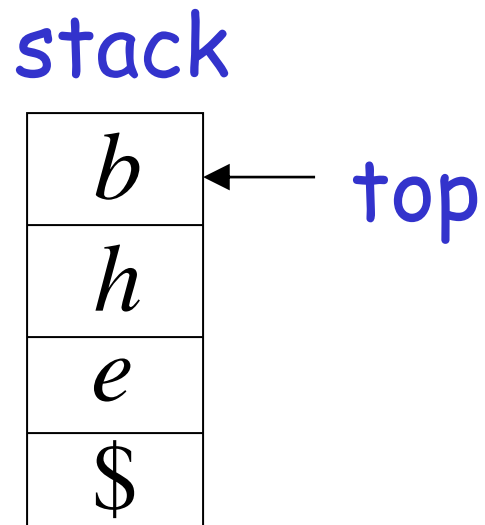
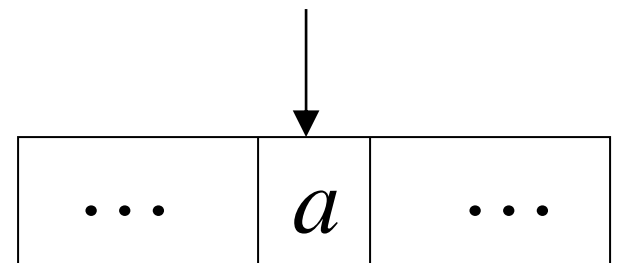
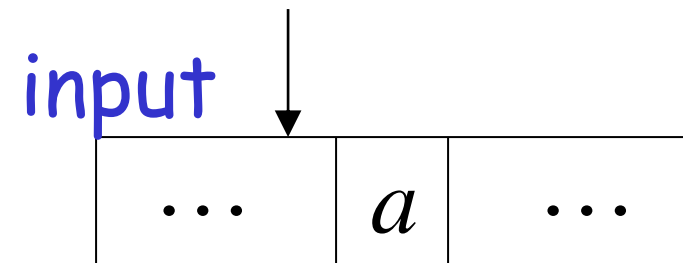
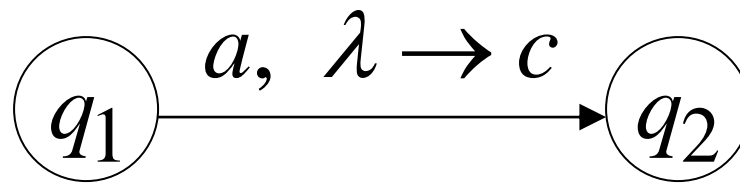
Initial Stack Symbol

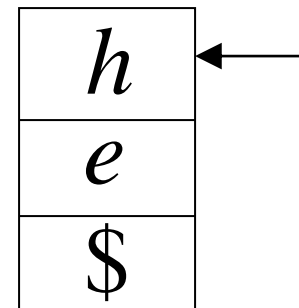
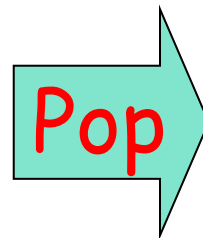
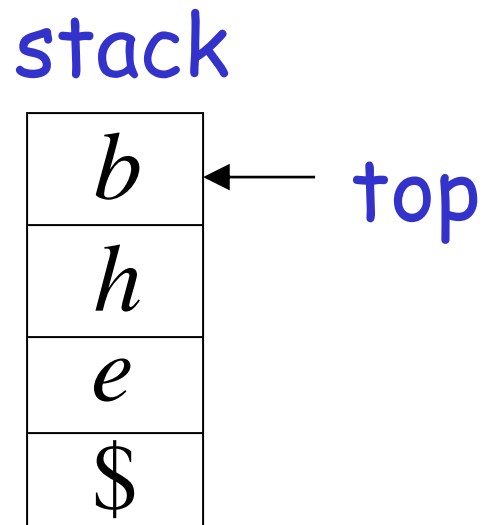
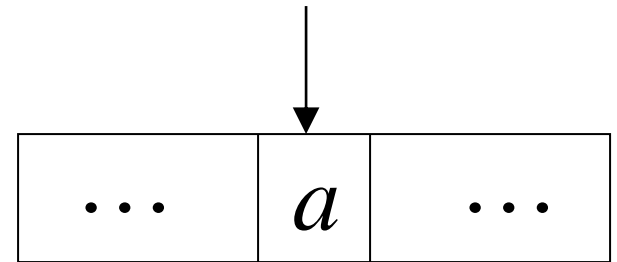
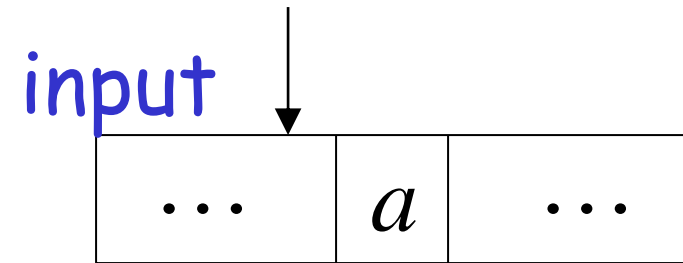
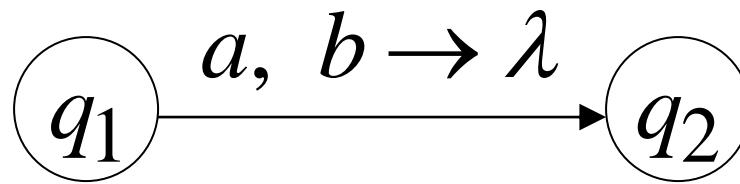


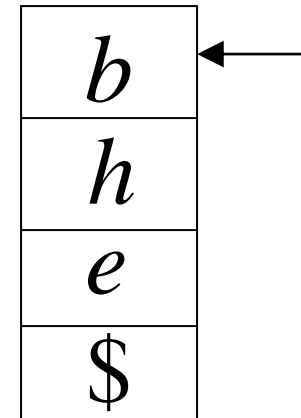
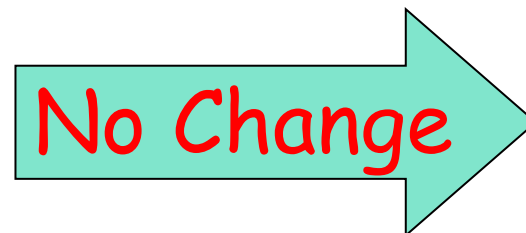
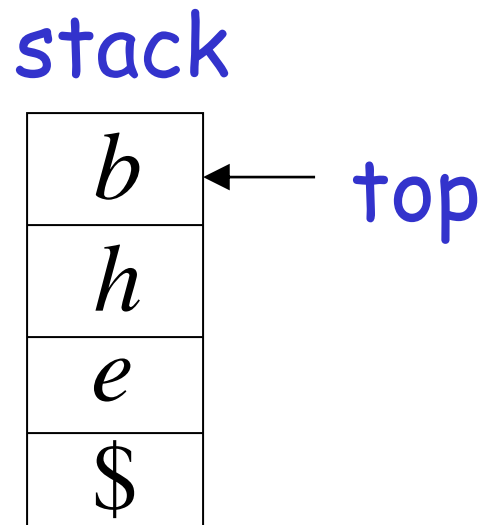
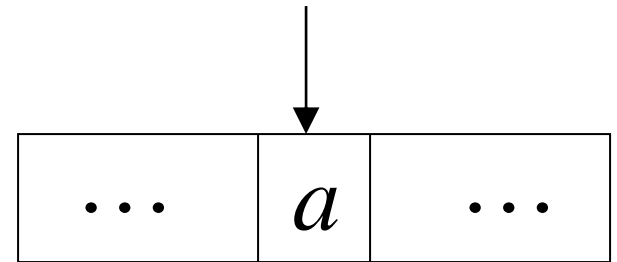
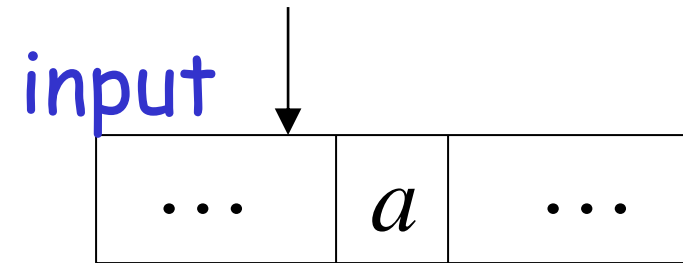
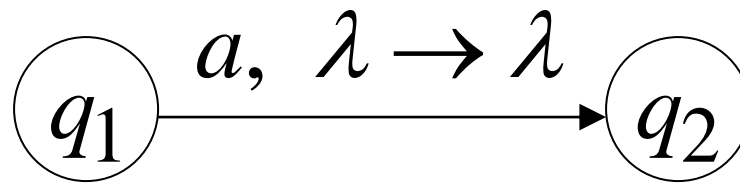
The States



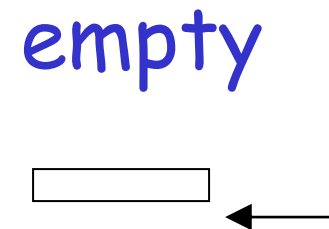
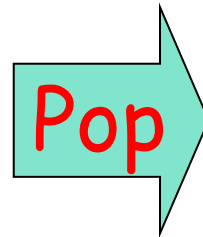
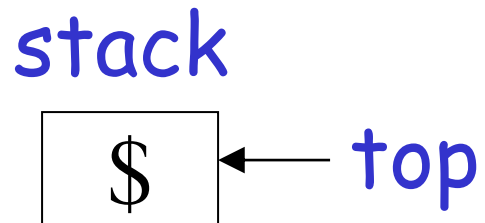
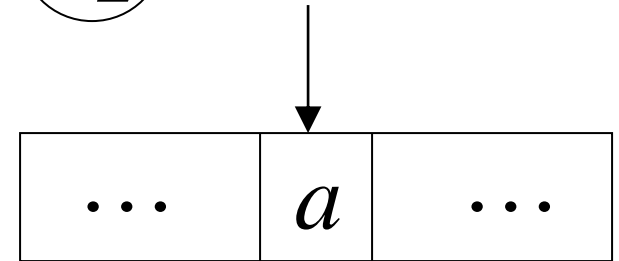
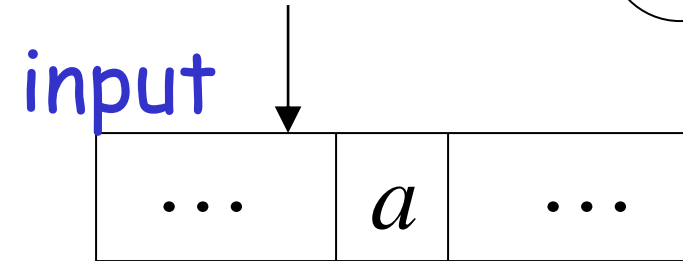
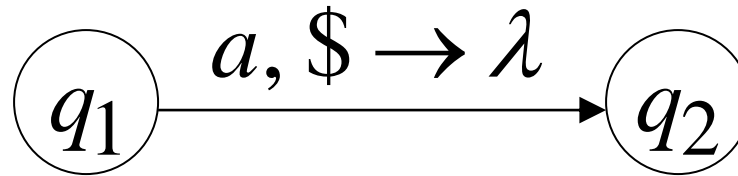




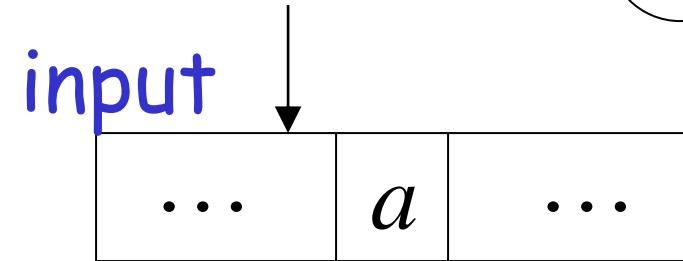
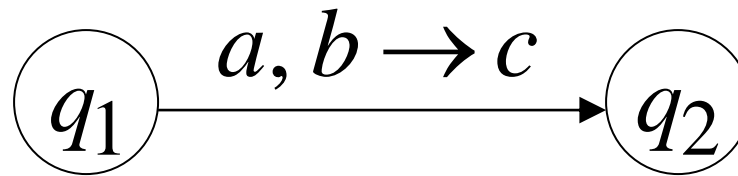




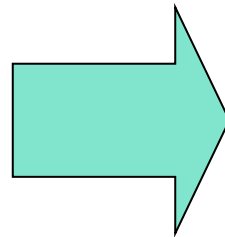
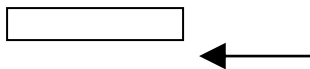
A Possible Transition



A Bad Transition



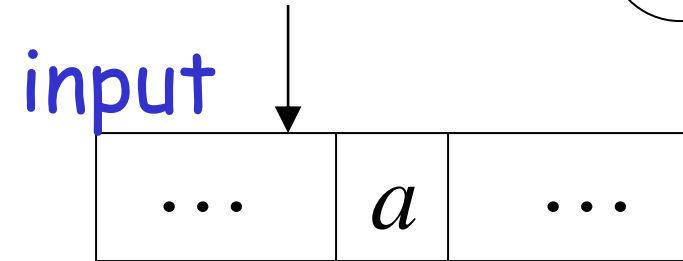
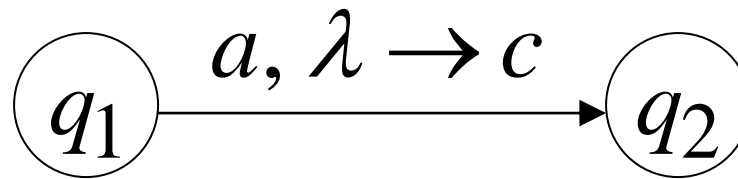
Empty stack



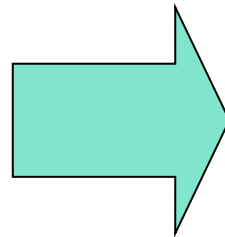
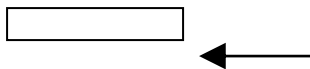
HALT

The automaton **Halts** in state q_1
and **Rejects** the input string

A Bad Transition



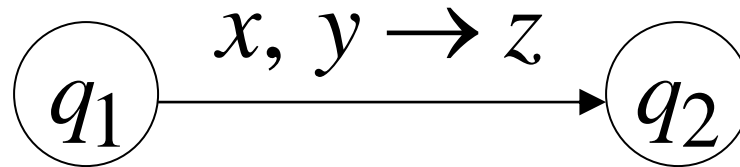
Empty stack



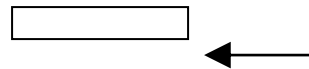
HALT

The automaton **Halts** in state q_1
and **Rejects** the input string

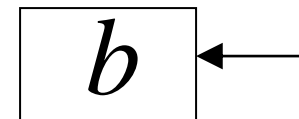
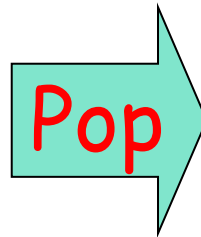
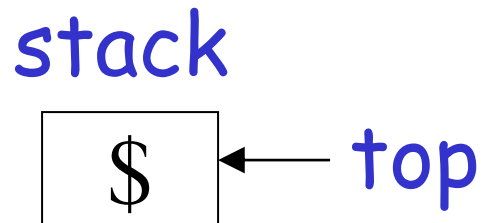
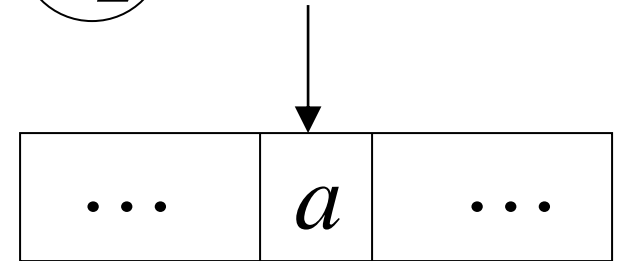
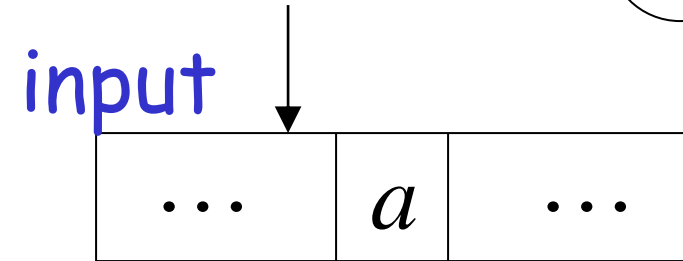
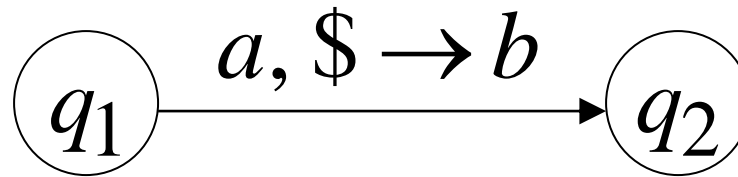
No transition is allowed to be followed
When the stack is empty



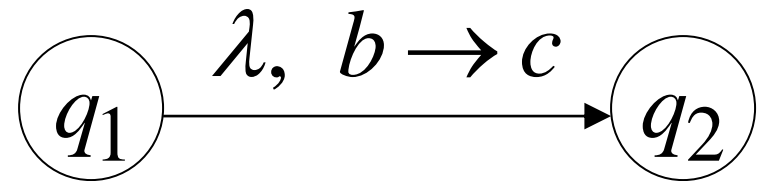
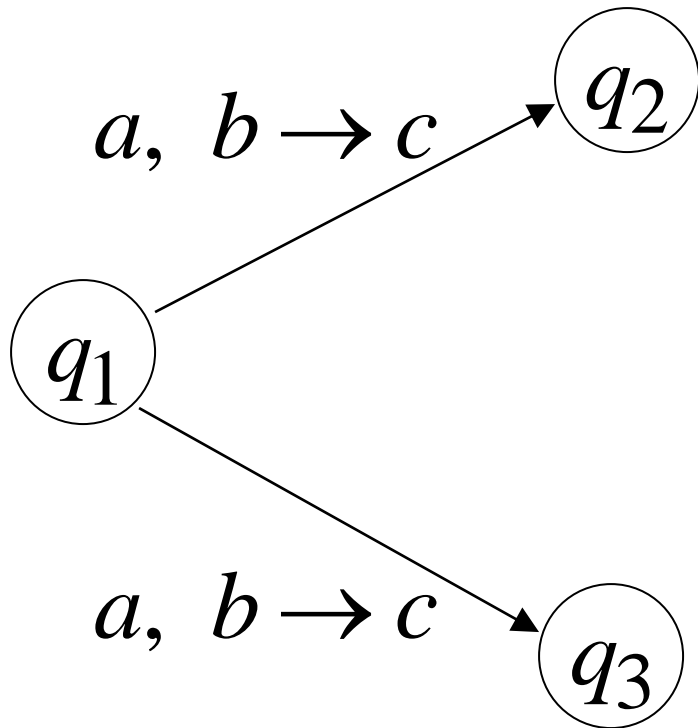
Empty stack



A Good Transition



Non-Determinism

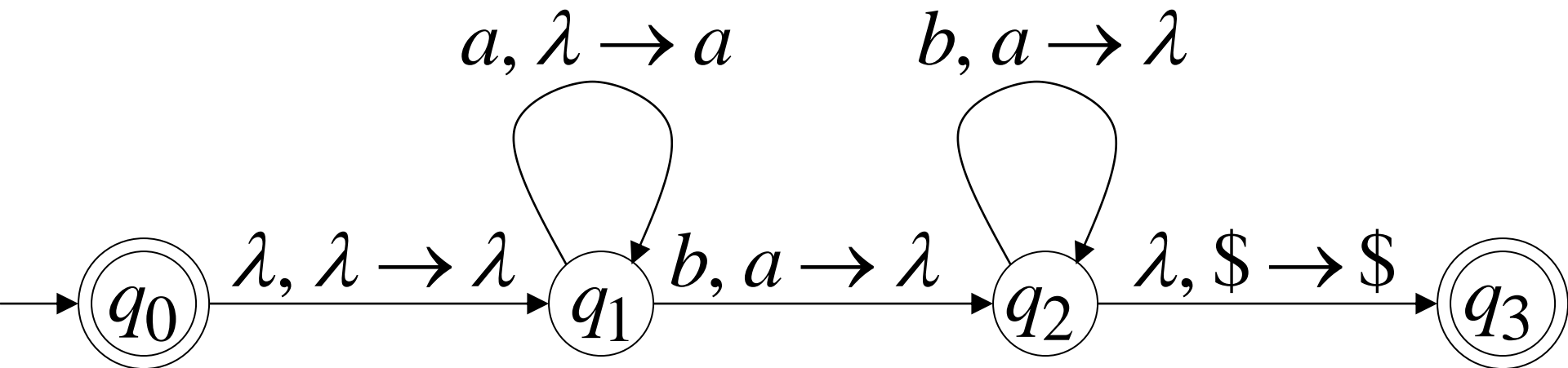


λ – transition

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

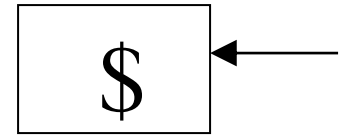
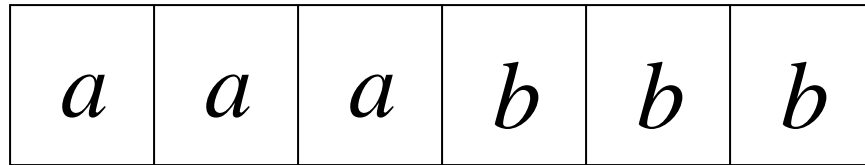
NPDA: Non-Deterministic PDA

Example:

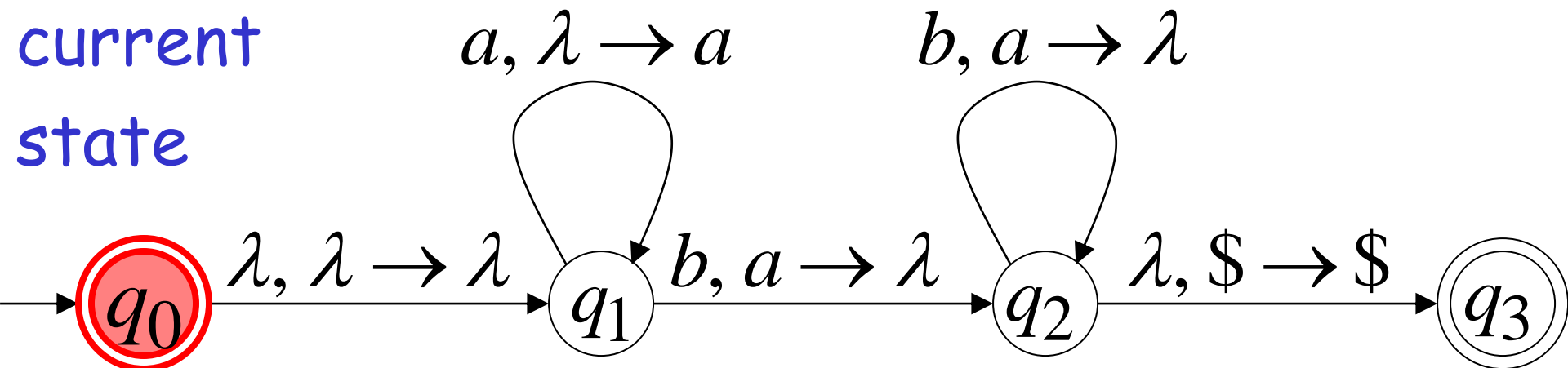


Execution Example: Time 0

Input

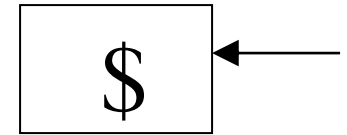
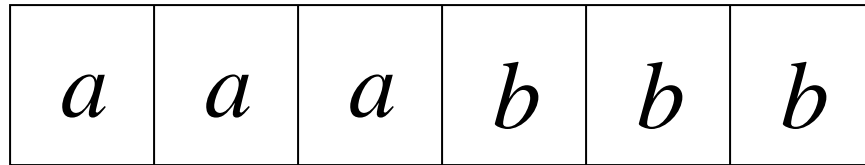


Stack

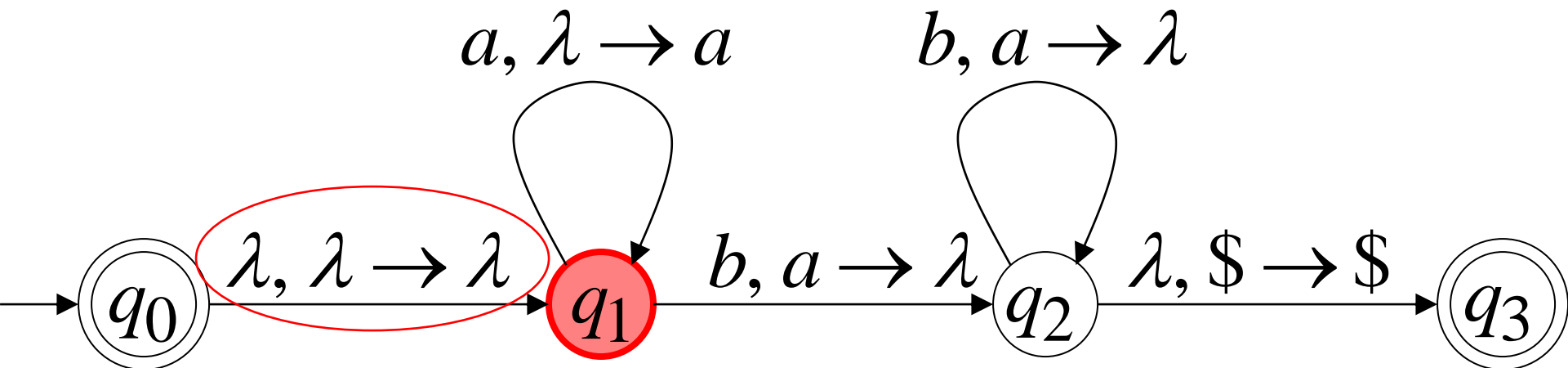


Time 1

Input

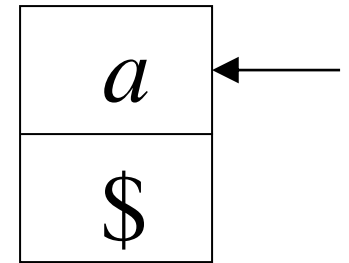
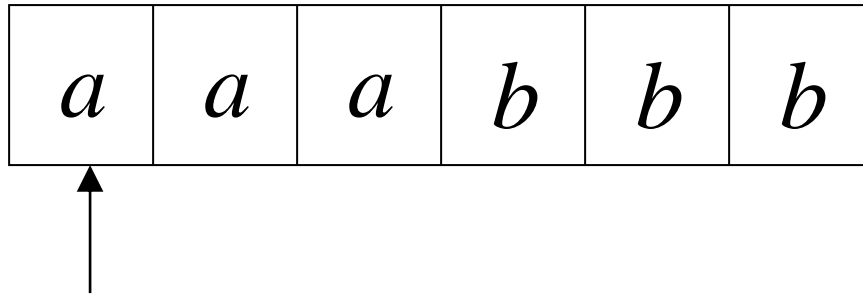


Stack

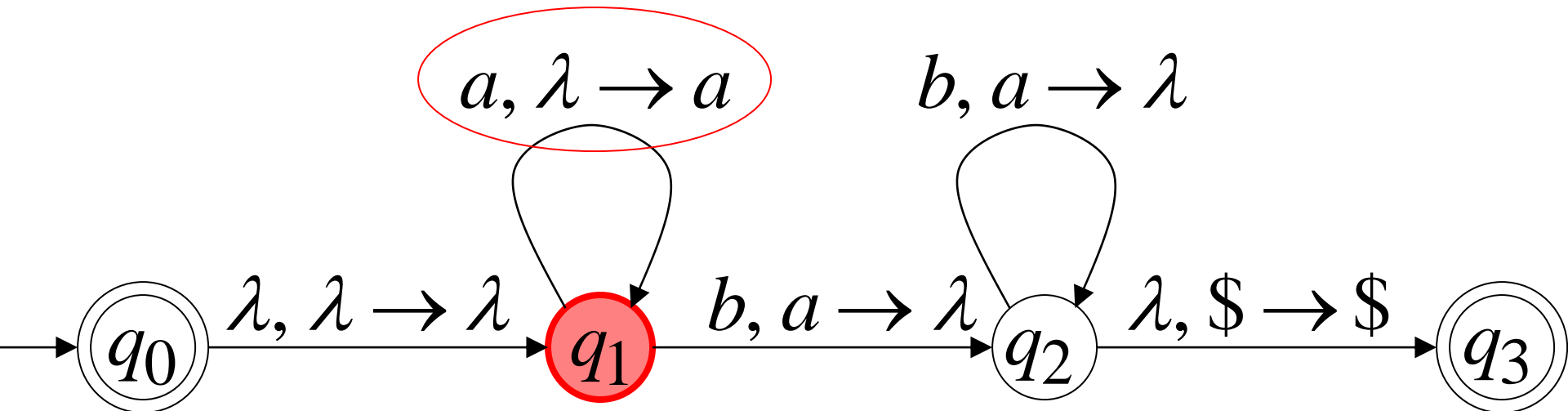


Time 2

Input

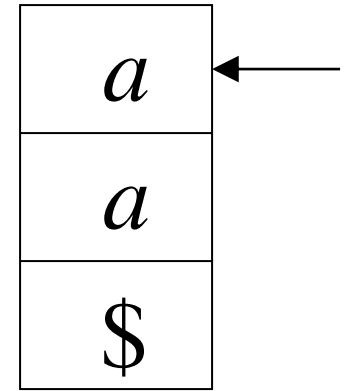
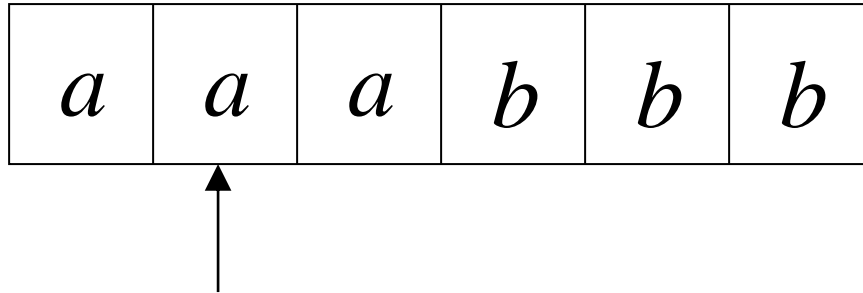


Stack

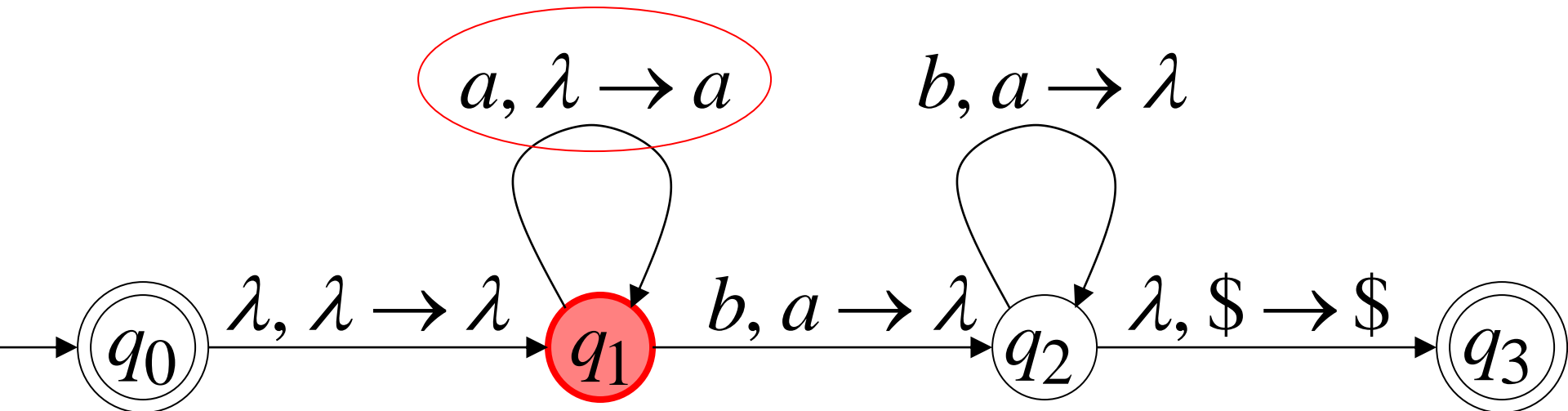


Time 3

Input

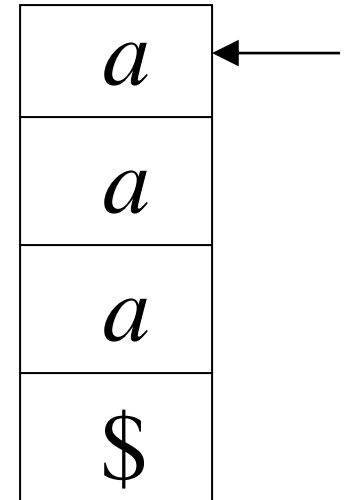
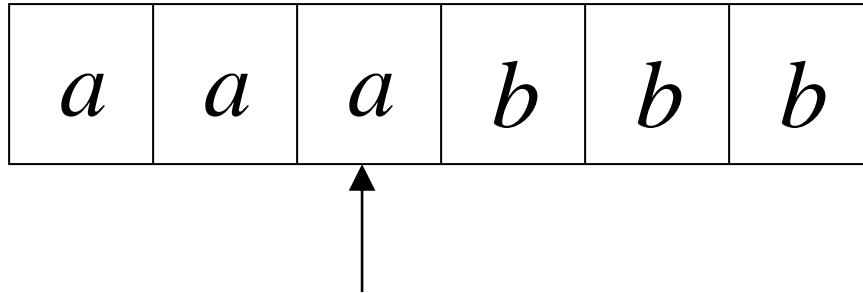


Stack

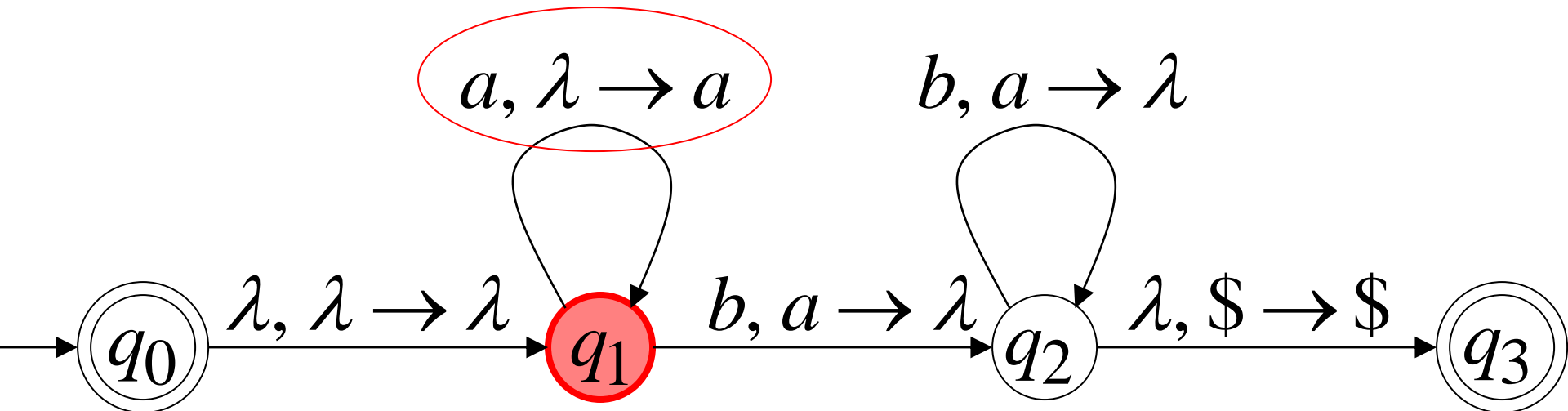


Time 4

Input

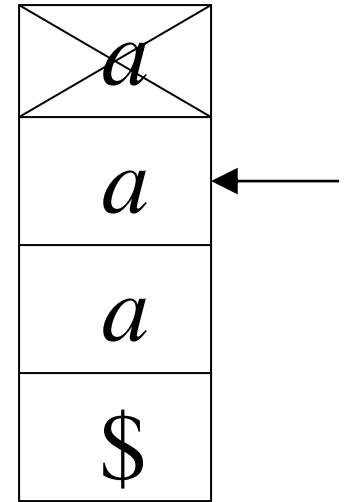
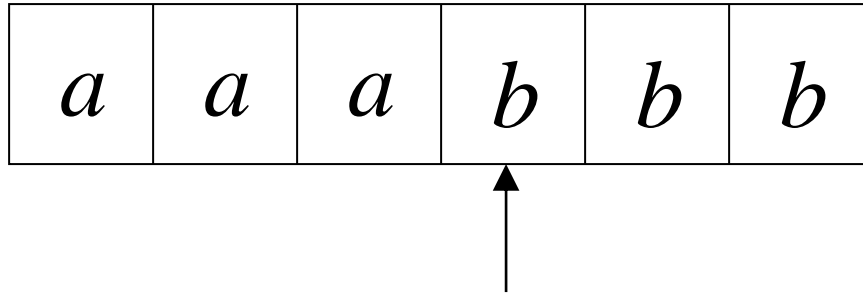


Stack

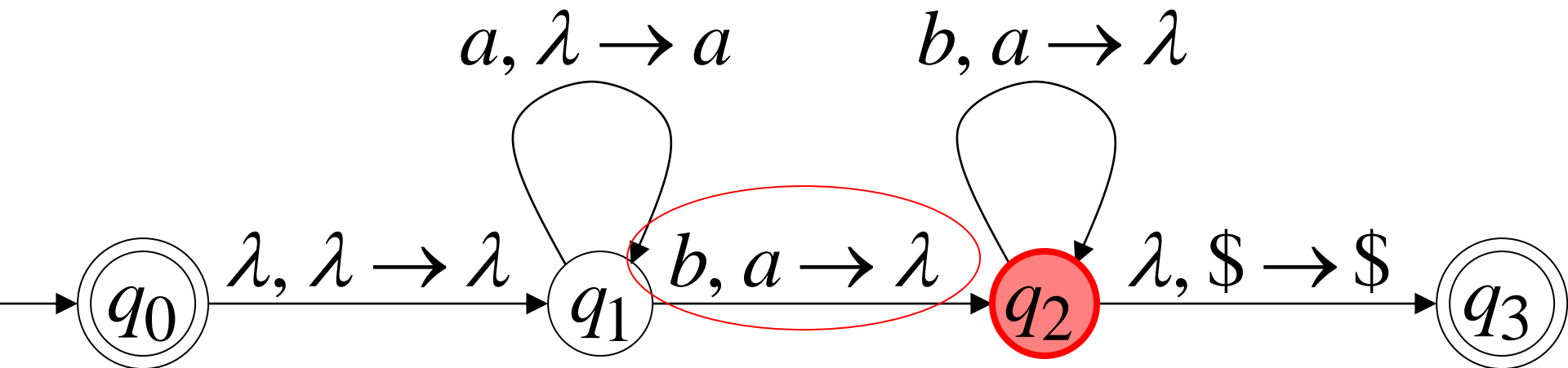


Time 5

Input

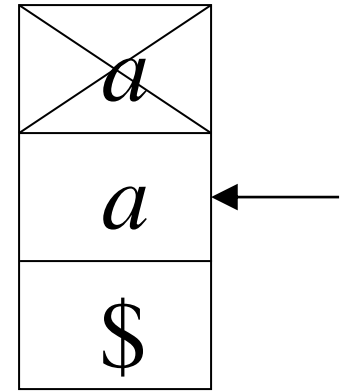
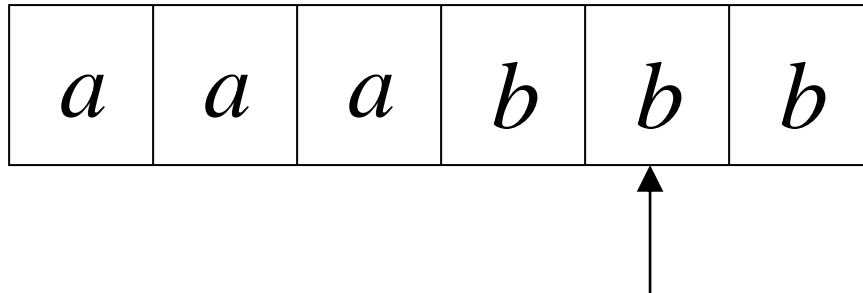


Stack

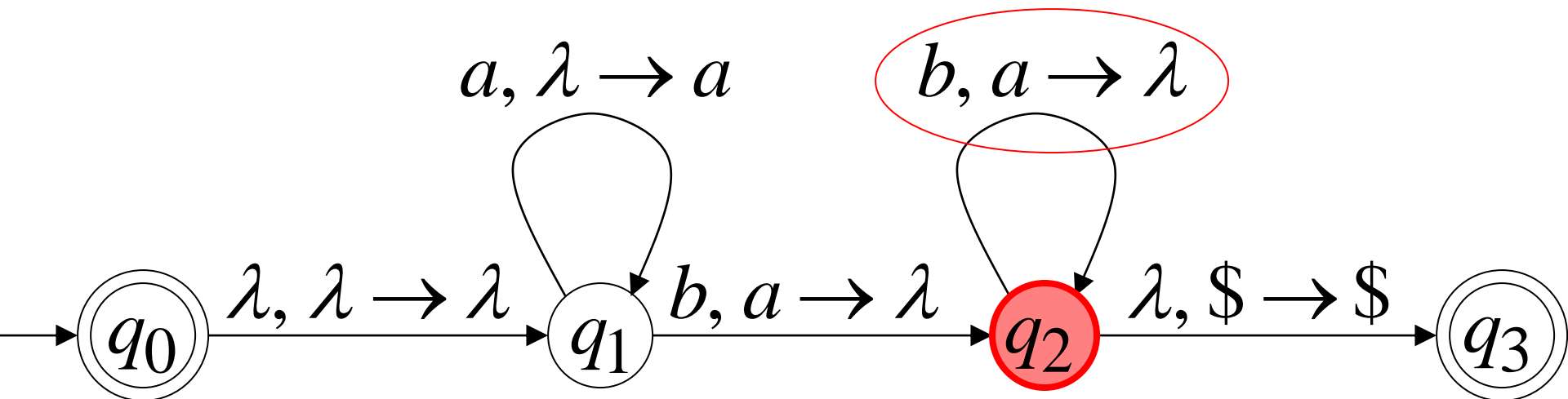


Time 6

Input

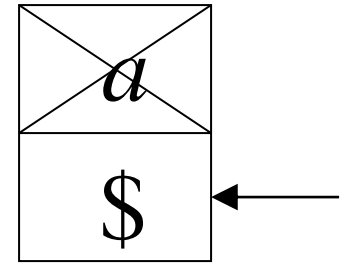
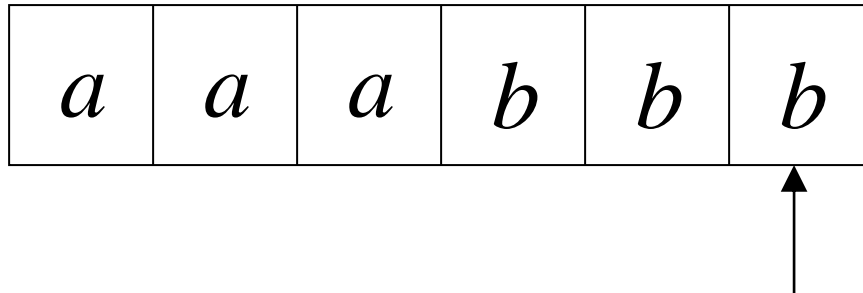


Stack

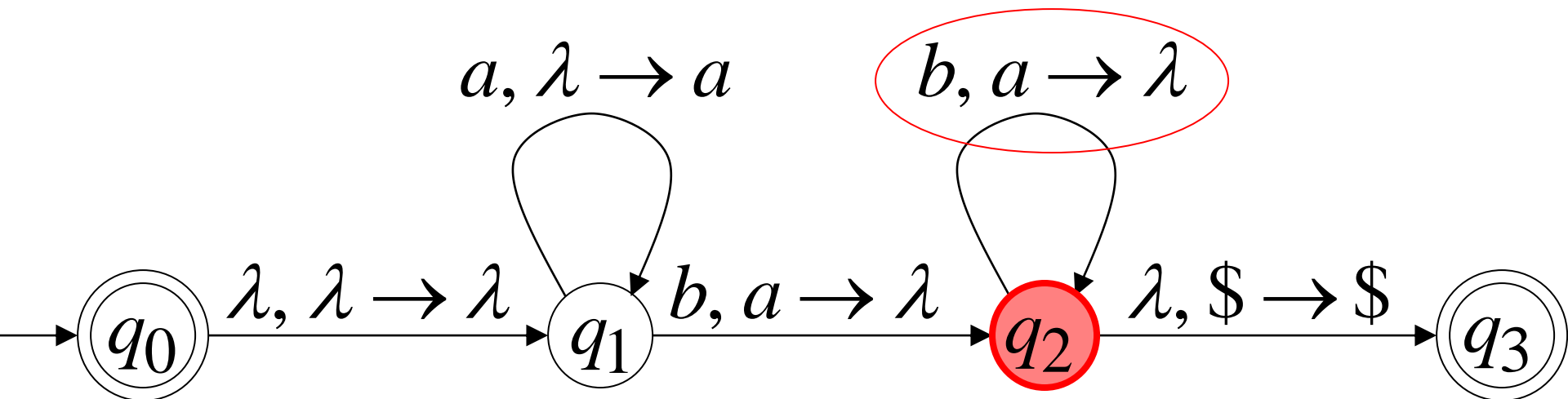


Time 7

Input

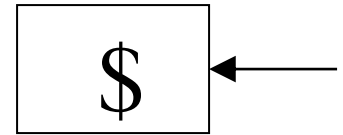
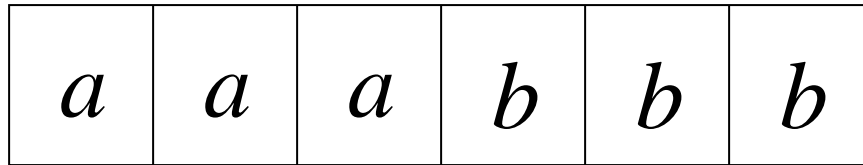


Stack

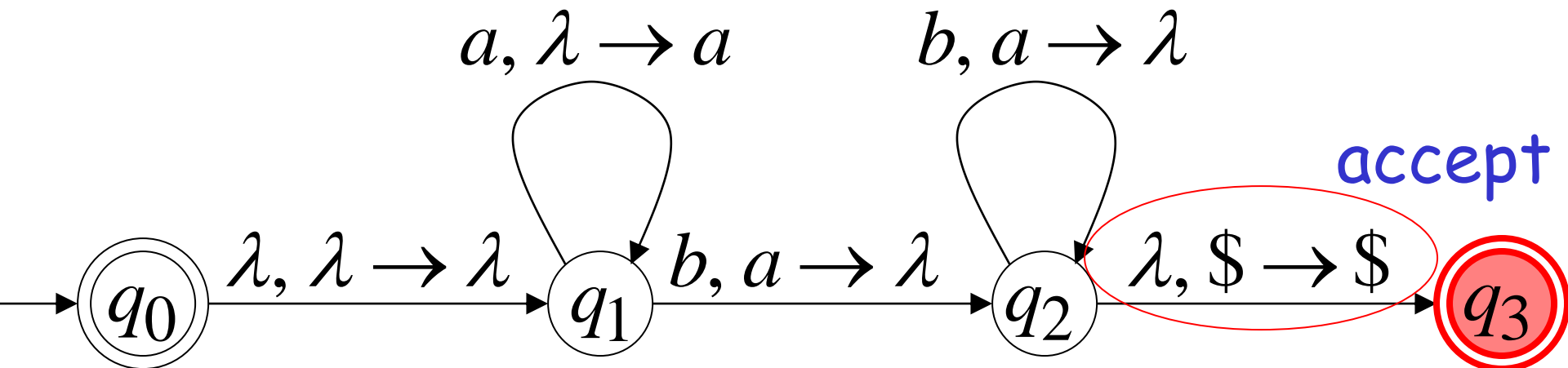


Time 8

Input



Stack



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

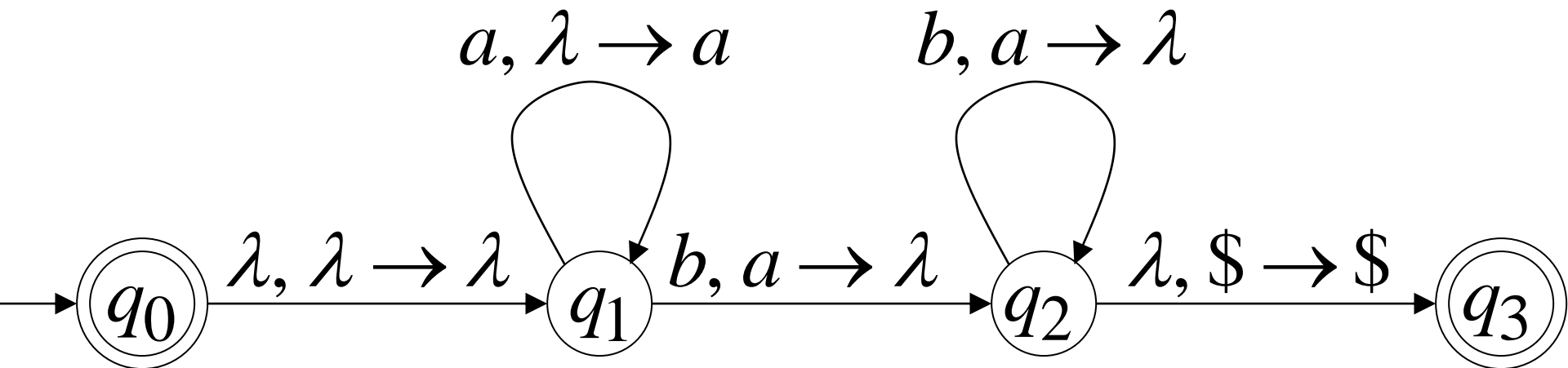
All the input is consumed

AND

The last state is a final state

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

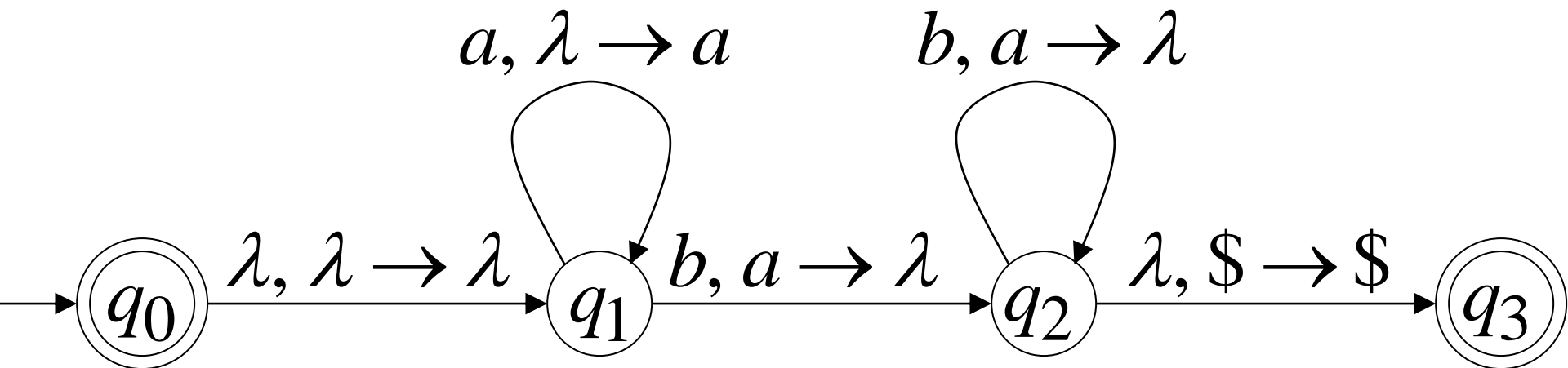
The input string *aaabbb*
is accepted by the NPDA:



In general,

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

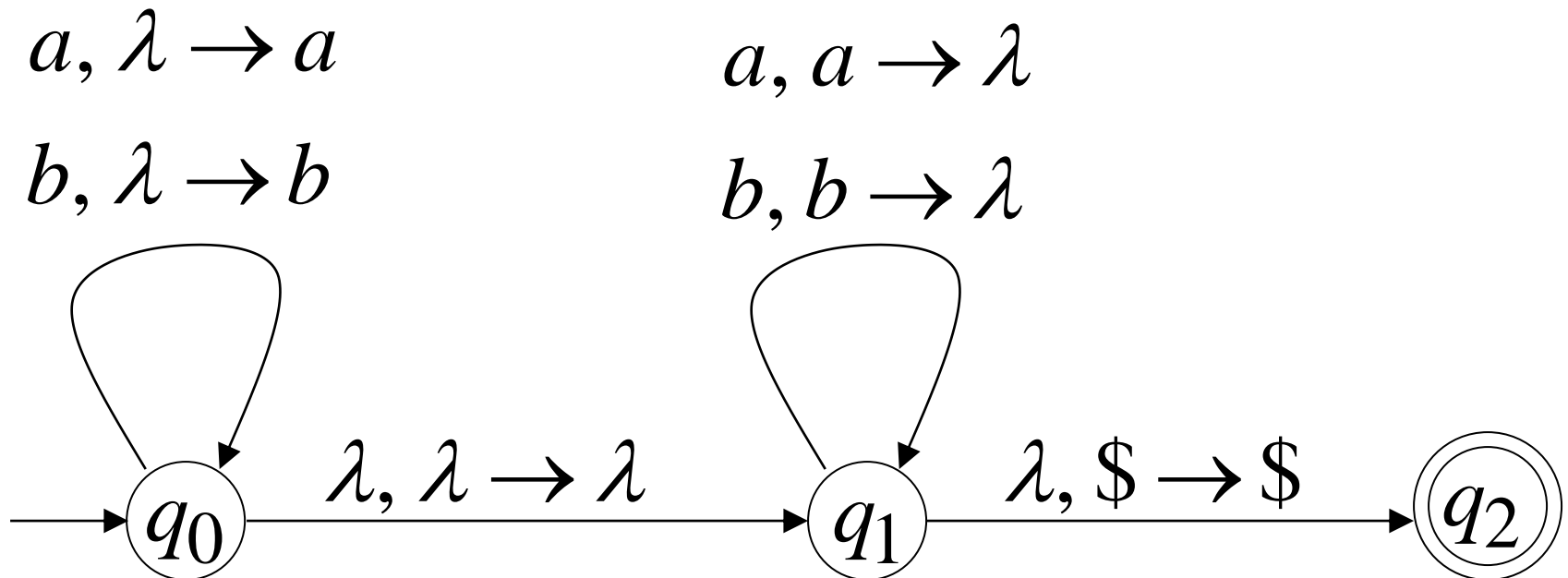
is the language accepted by the NPDA:



Another NPDA example

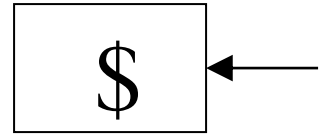
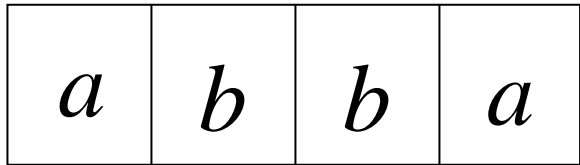
NPDA M

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



Execution Example: Time 0

Input



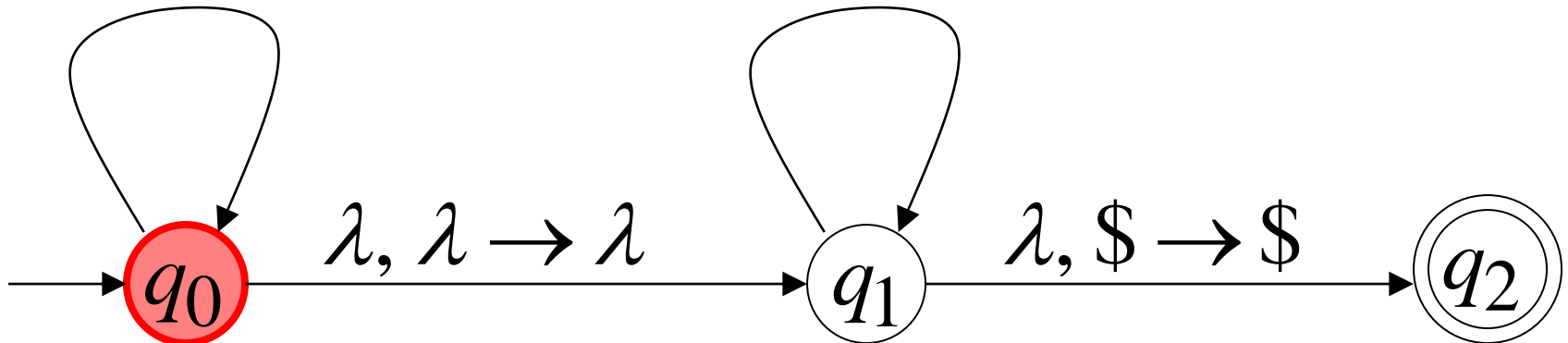
Stack

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

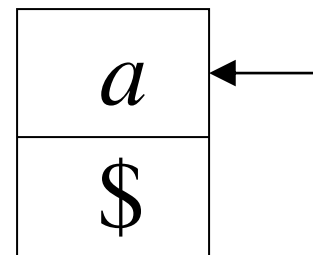
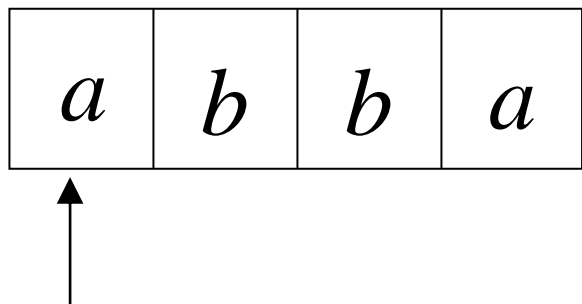
$b, \lambda \rightarrow b$

$b, b \rightarrow \lambda$



Time 1

Input



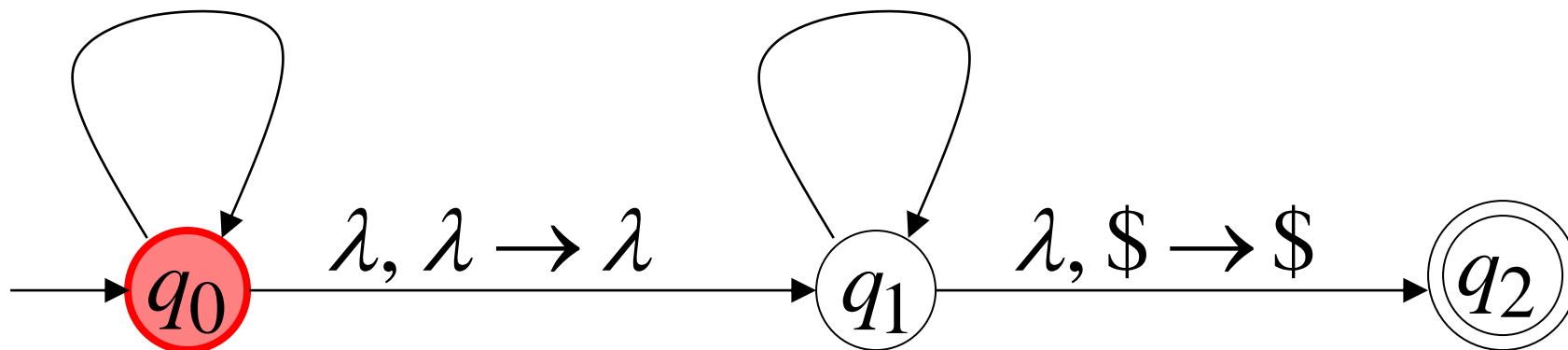
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

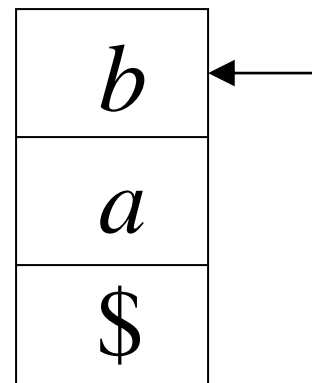
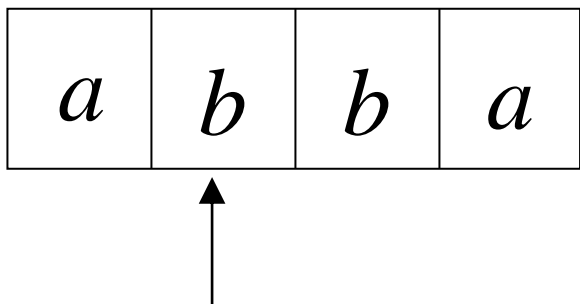
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 2

Input



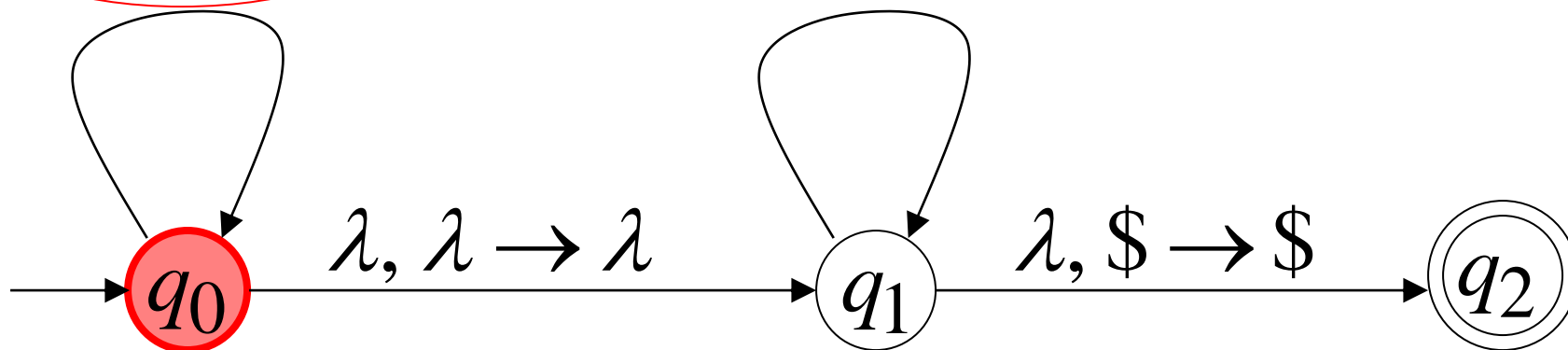
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

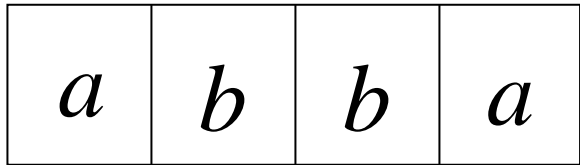
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$

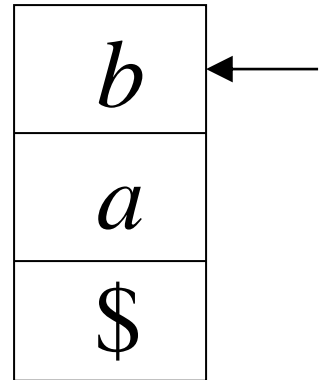


Time 3

Input



Guess the middle
of string



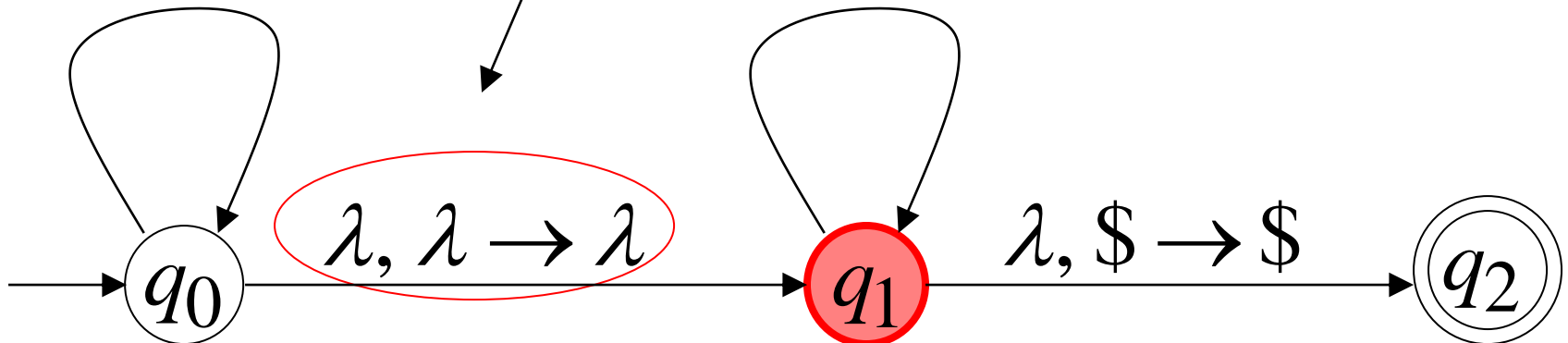
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

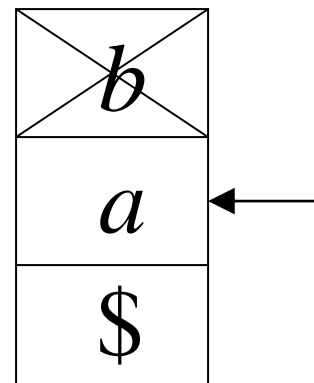
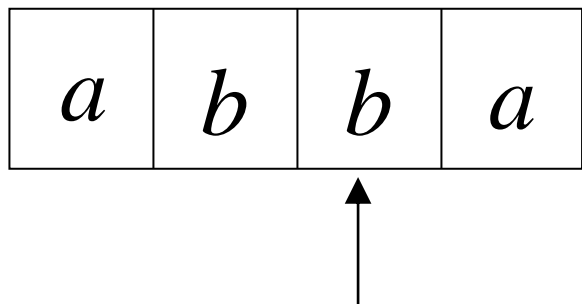
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 4

Input



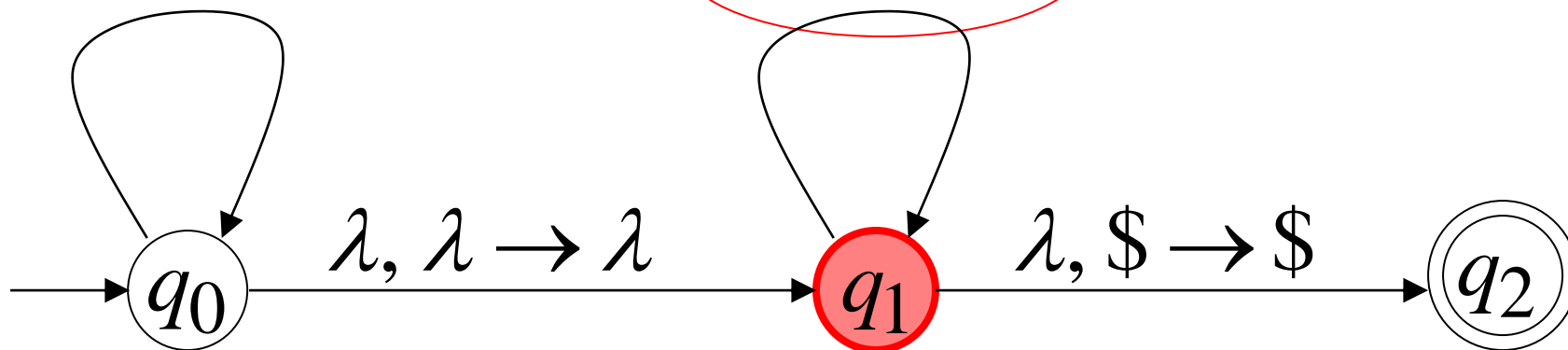
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

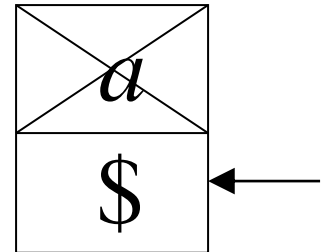
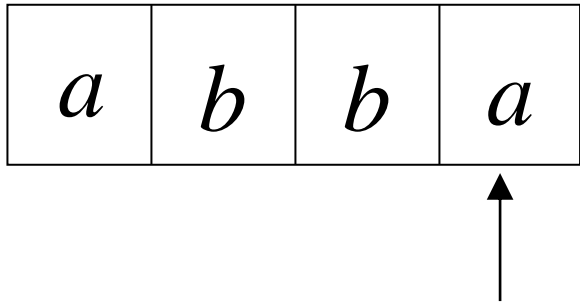
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 5

Input



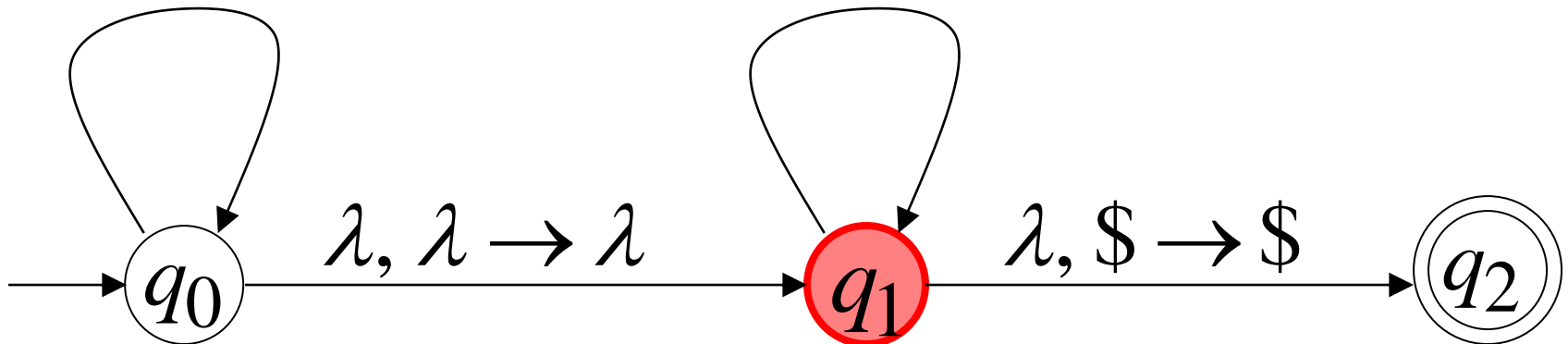
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

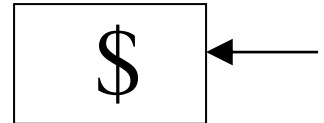
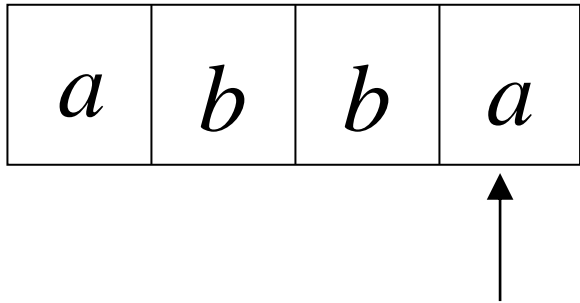
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 6

Input



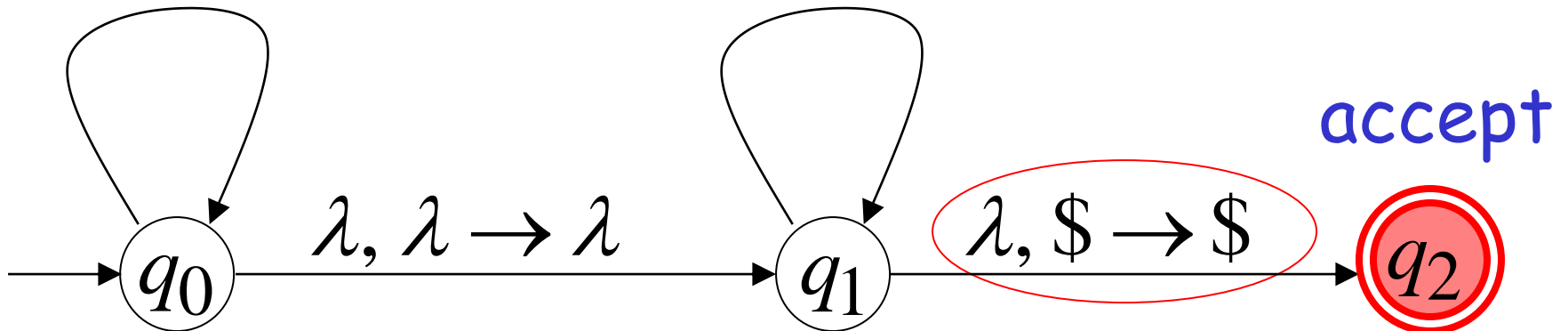
Stack

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

$b, \lambda \rightarrow b$

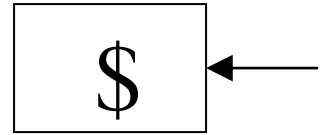
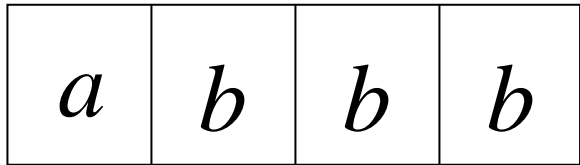
$b, b \rightarrow \lambda$



accept

Rejection Example: Time 0

Input



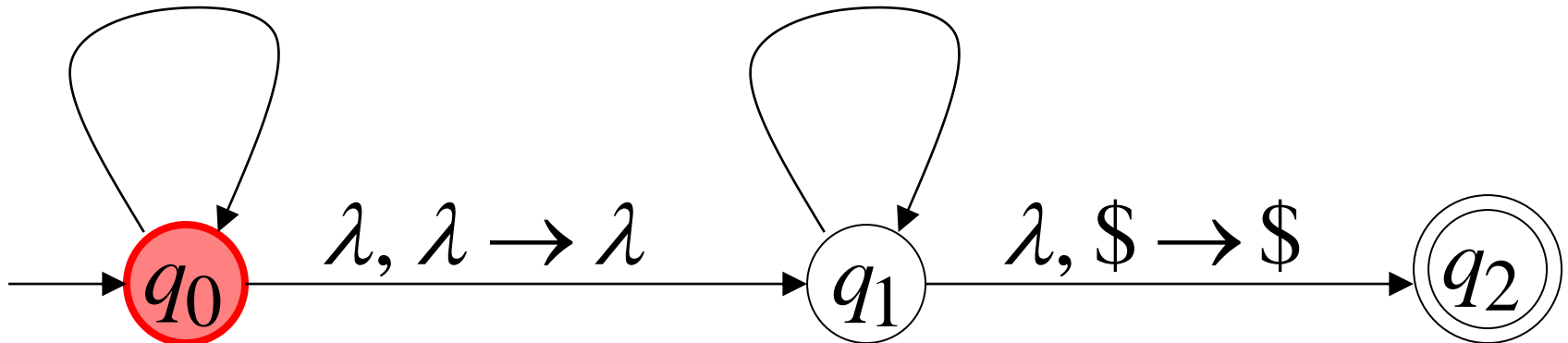
Stack

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

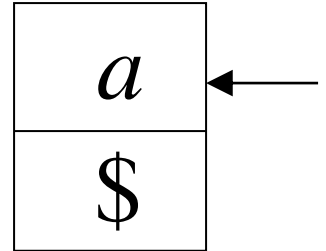
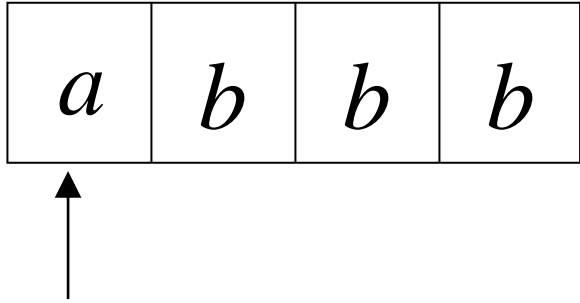
$b, \lambda \rightarrow b$

$b, b \rightarrow \lambda$



Time 1

Input



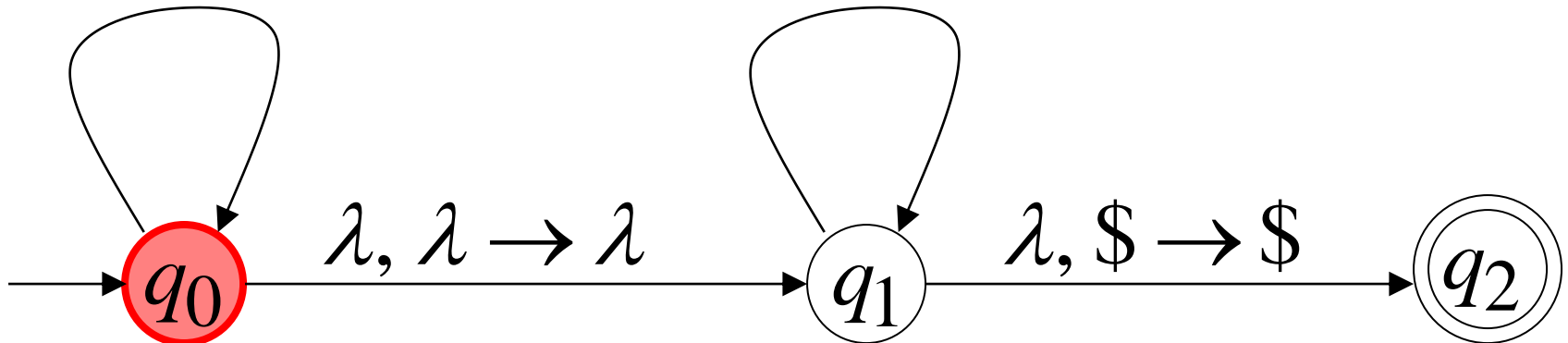
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

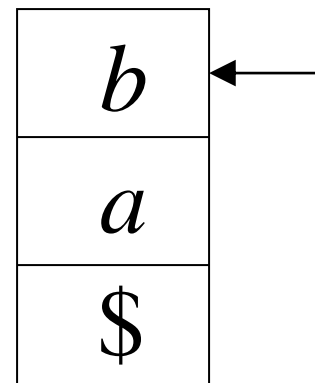
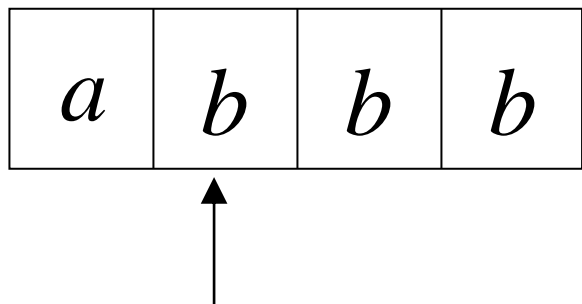
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 2

Input



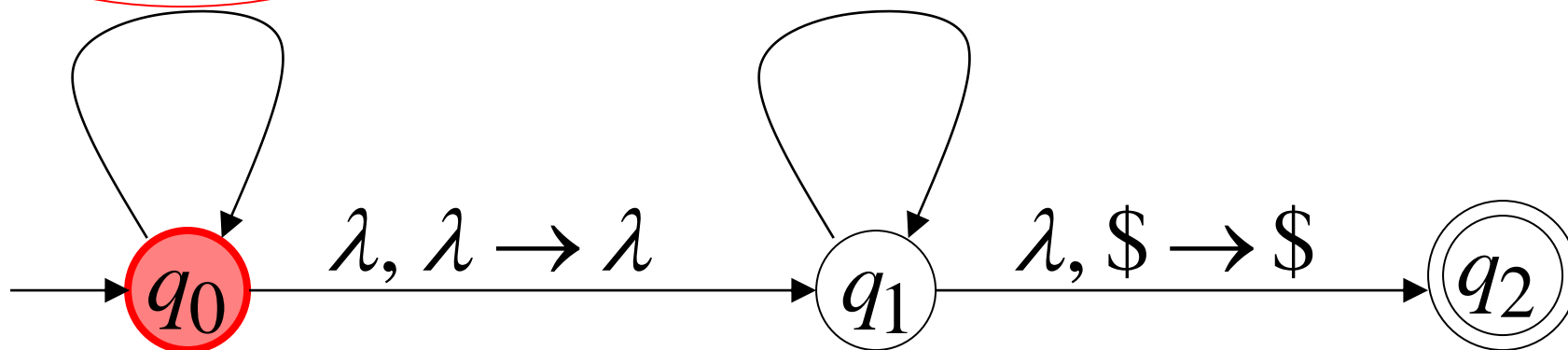
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

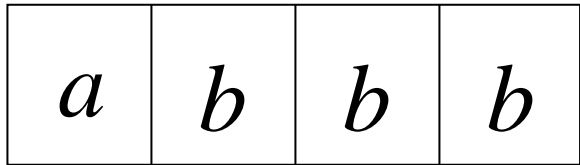
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$

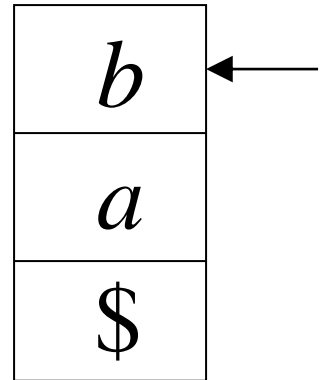


Time 3

Input



Guess the middle
of string



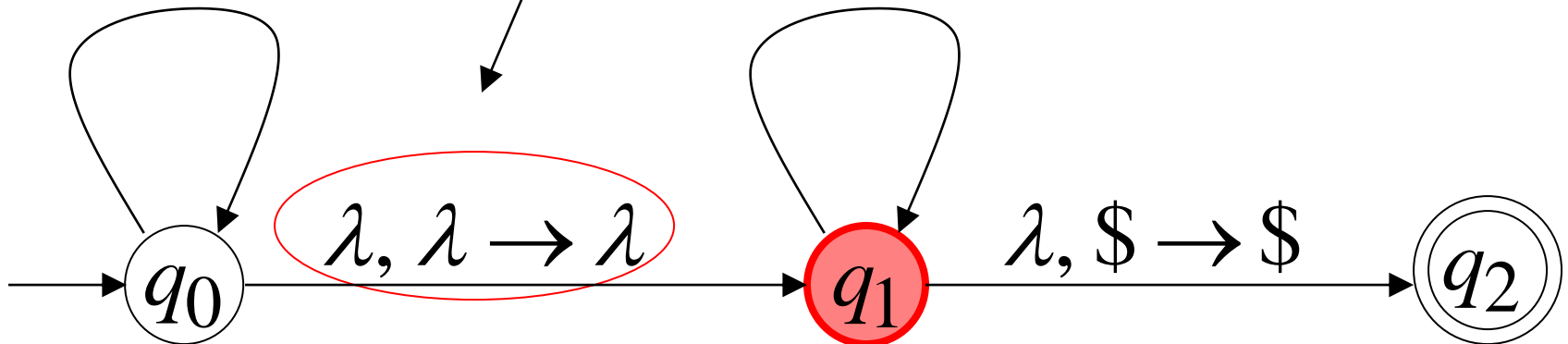
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

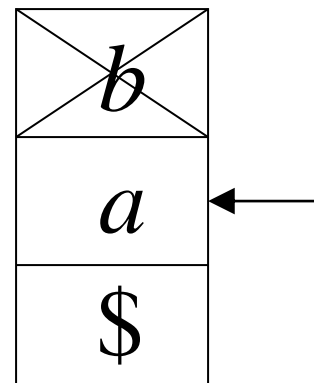
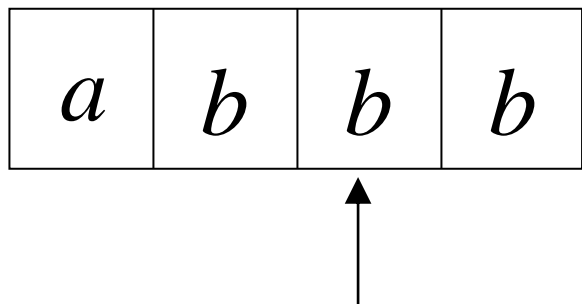
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 4

Input



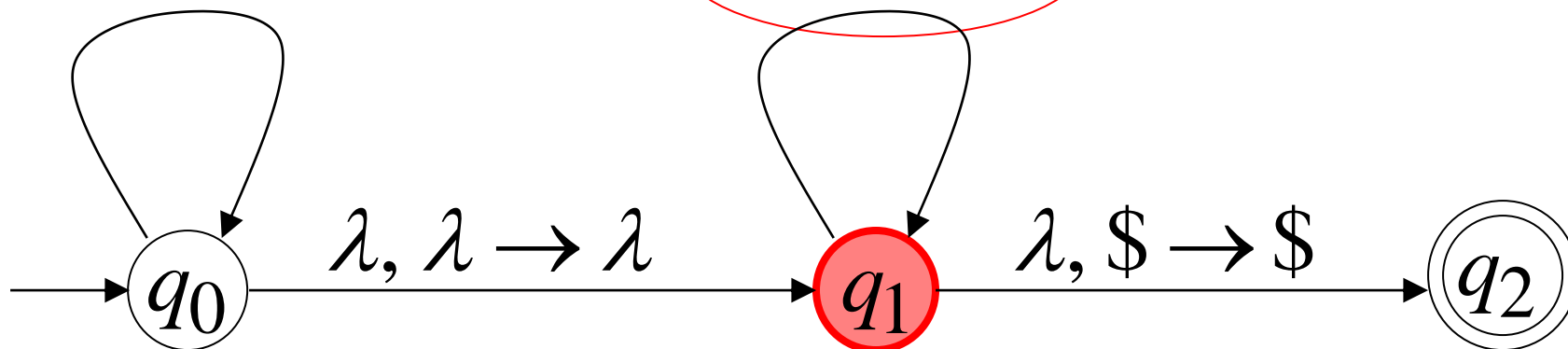
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

$a, a \rightarrow \lambda$

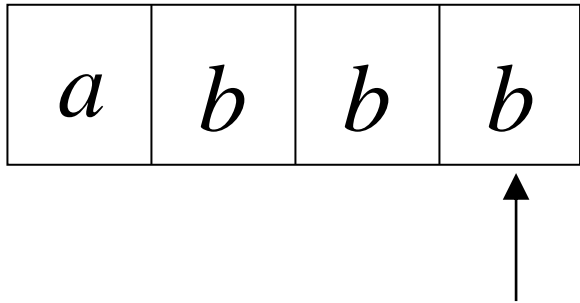
$b, b \rightarrow \lambda$



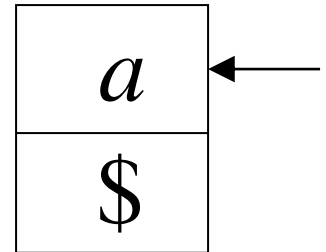
Time 5

Input

There is no possible transition.



Input is not consumed



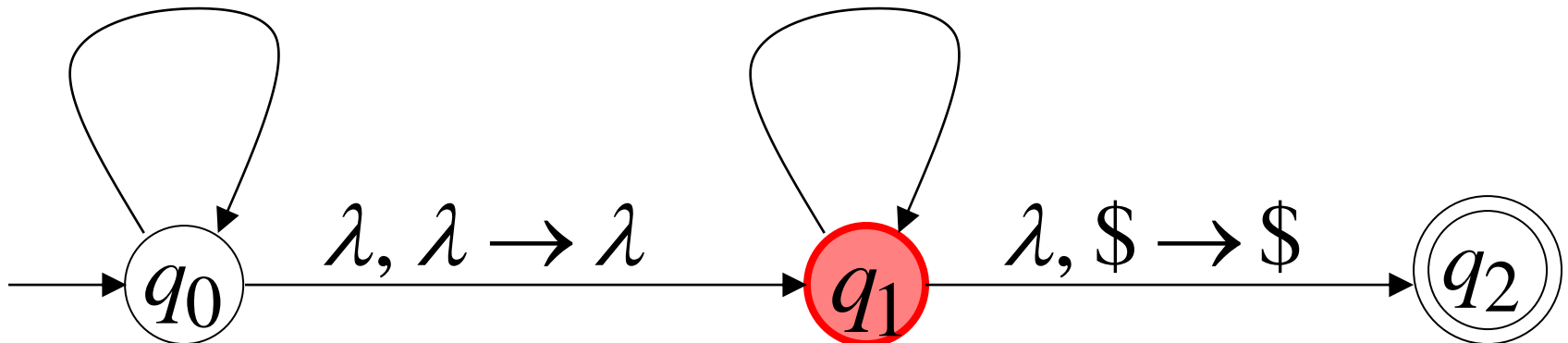
Stack

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

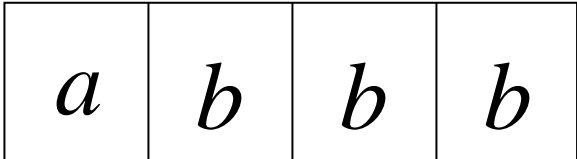
$b, \lambda \rightarrow b$

$b, b \rightarrow \lambda$

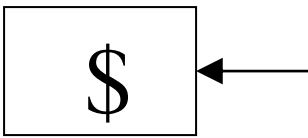


Another computation on same string:

Input



Time 0



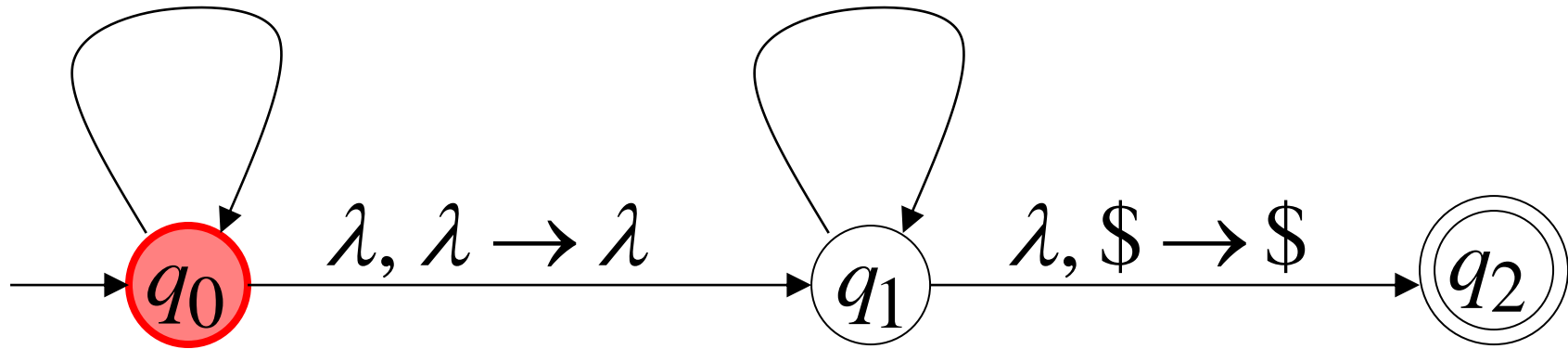
Stack

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

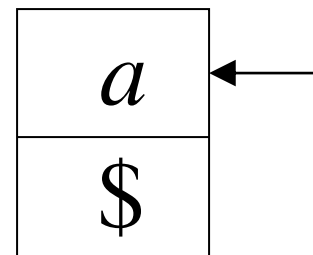
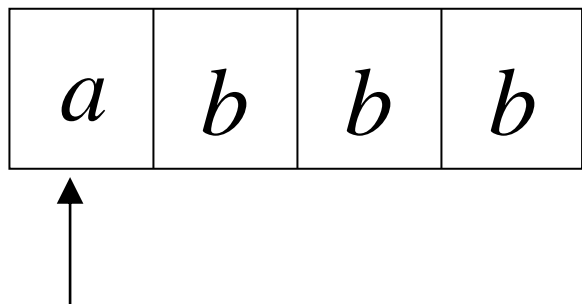
$b, \lambda \rightarrow b$

$b, b \rightarrow \lambda$



Time 1

Input



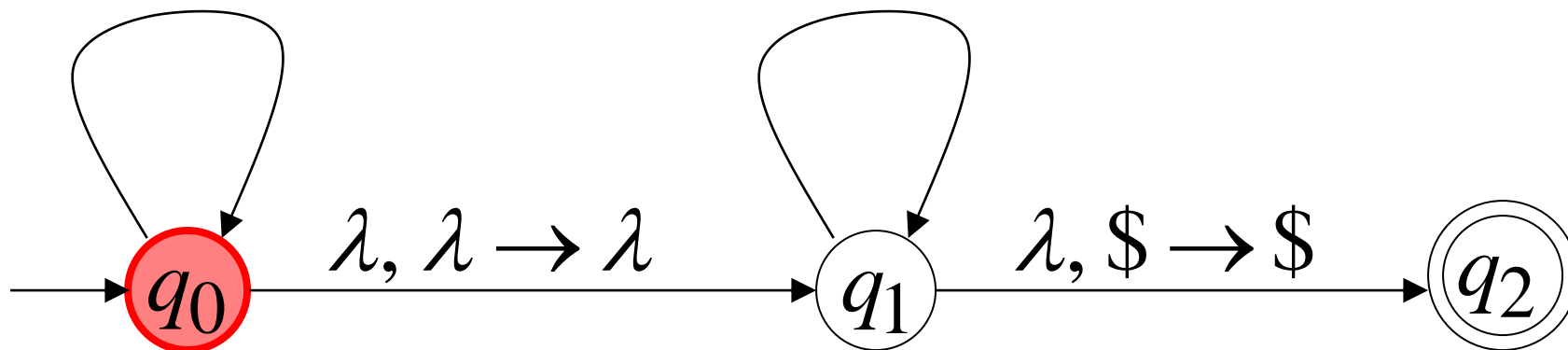
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

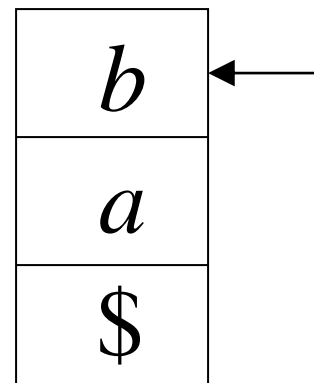
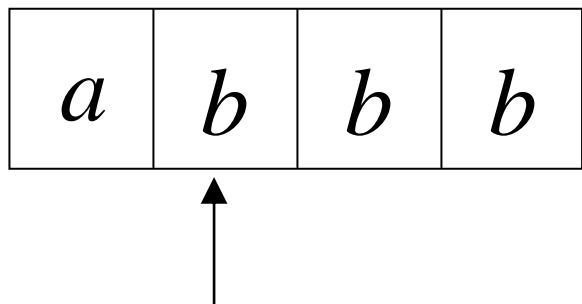
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 2

Input



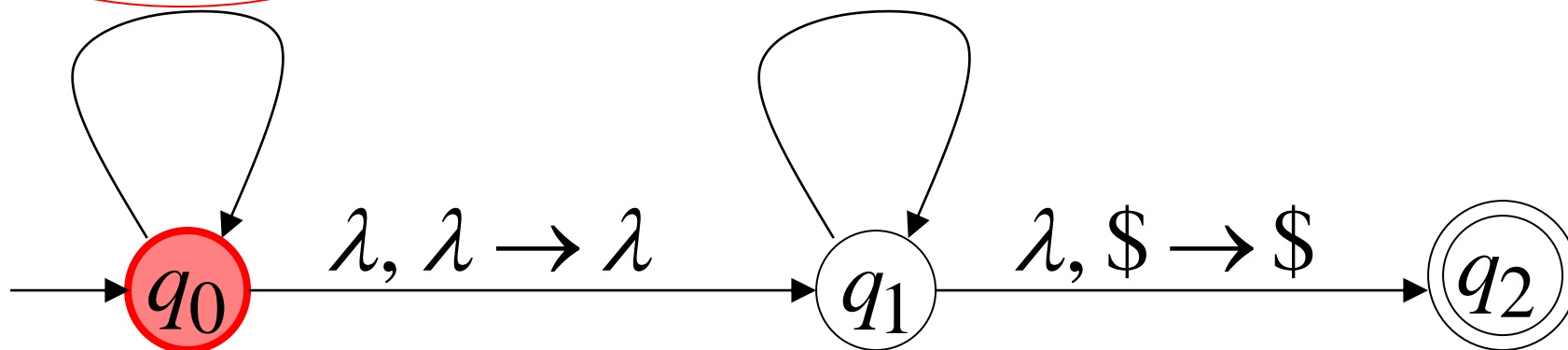
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

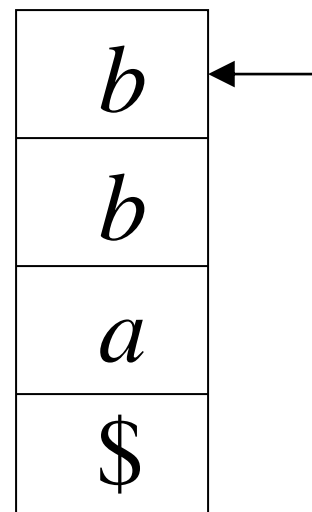
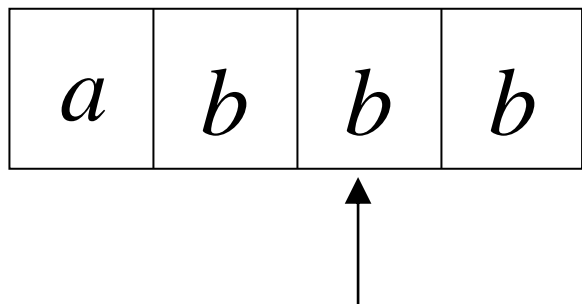
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 3

Input



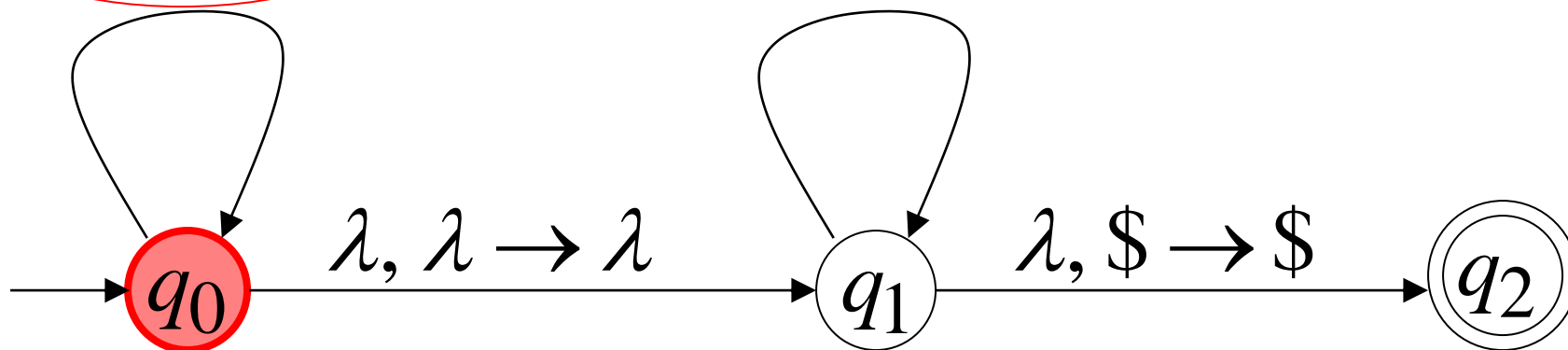
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

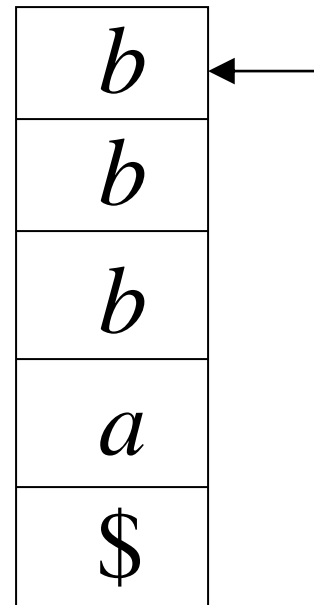
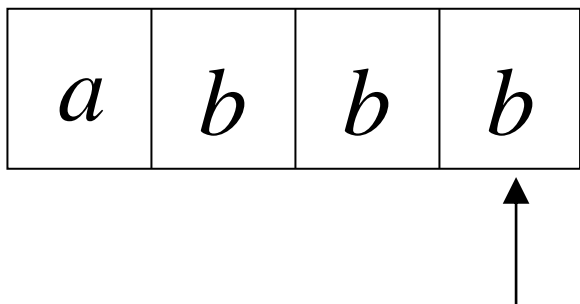
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 4

Input



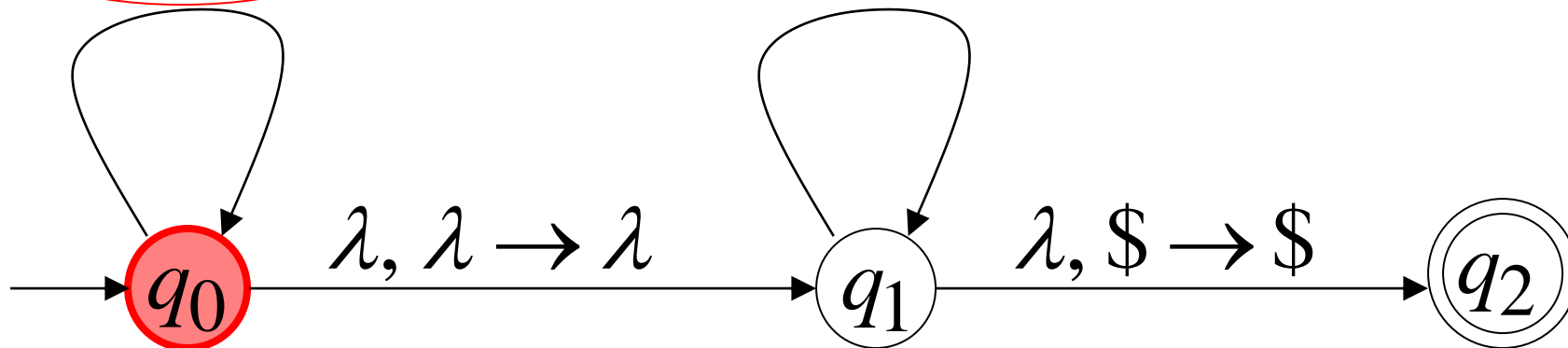
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

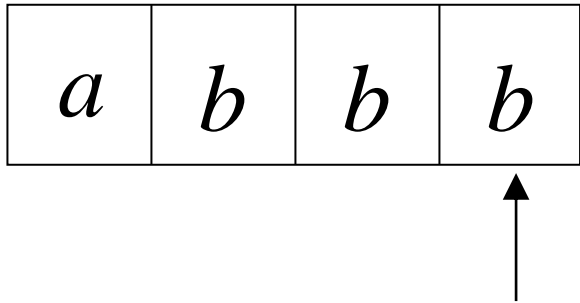
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$

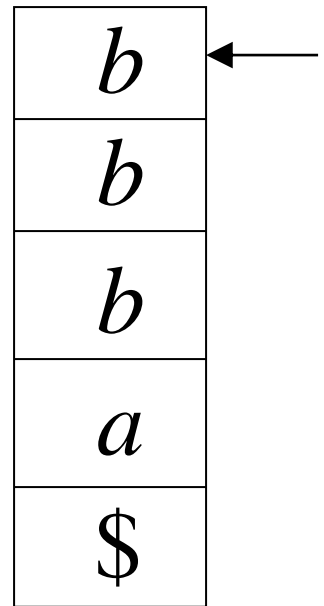


Time 5

Input



No final state
is reached



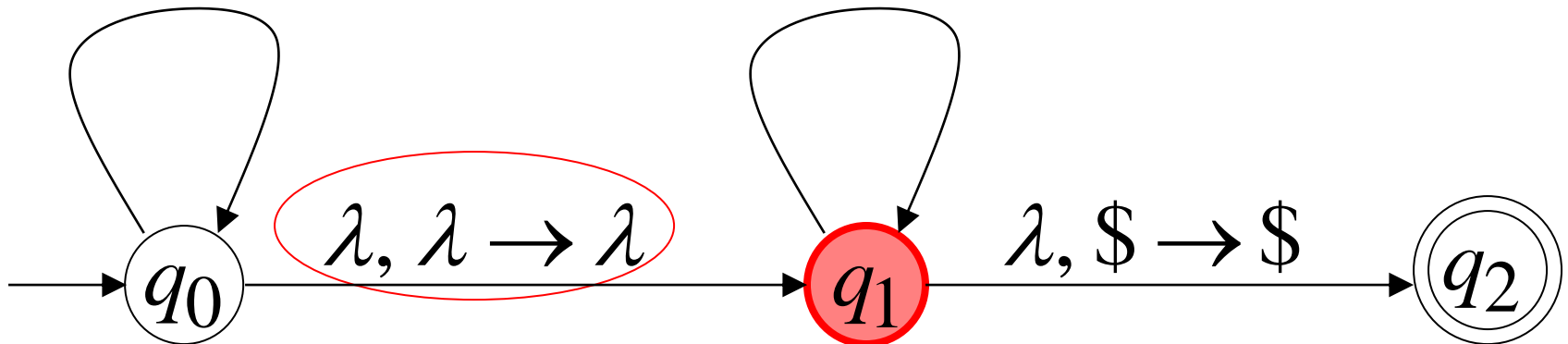
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

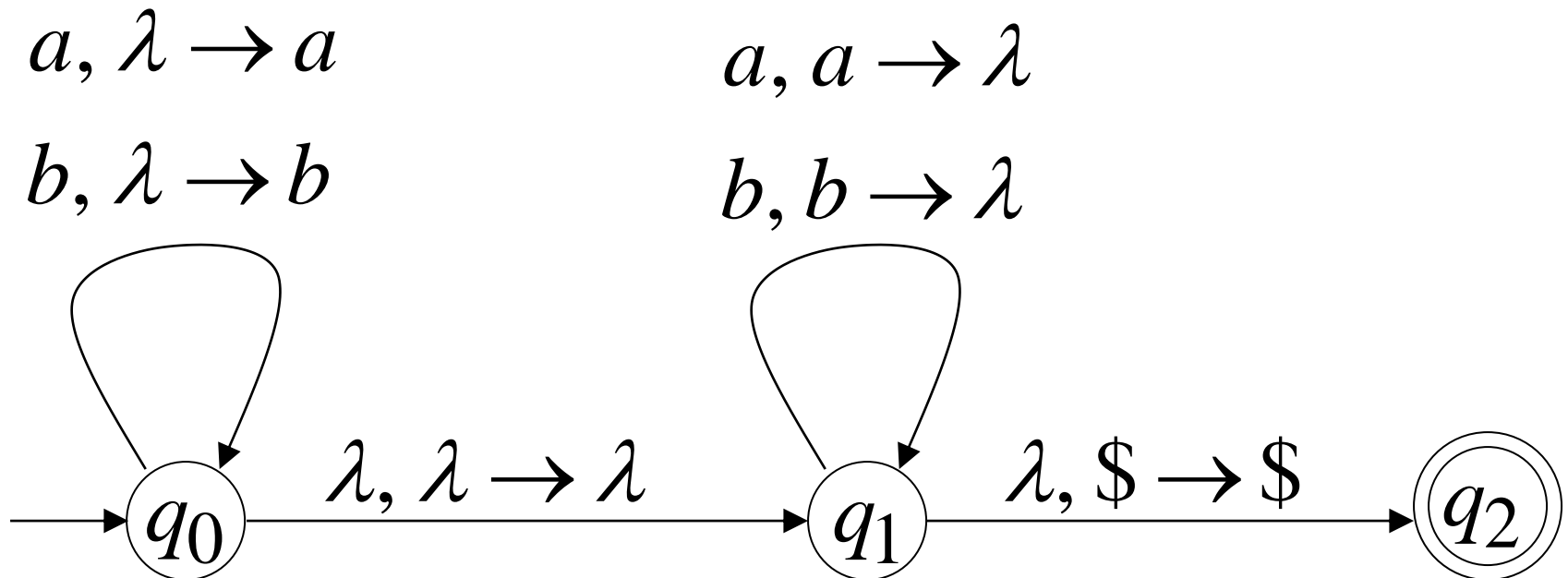
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



There is no computation
that accepts string $abbb$

$$abbb \notin L(M)$$



A string is rejected if there is
no computation such that:

All the input is consumed

AND

The last state is a final state

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

In other words, a string is rejected
if in every computation with this string:

The input cannot be consumed

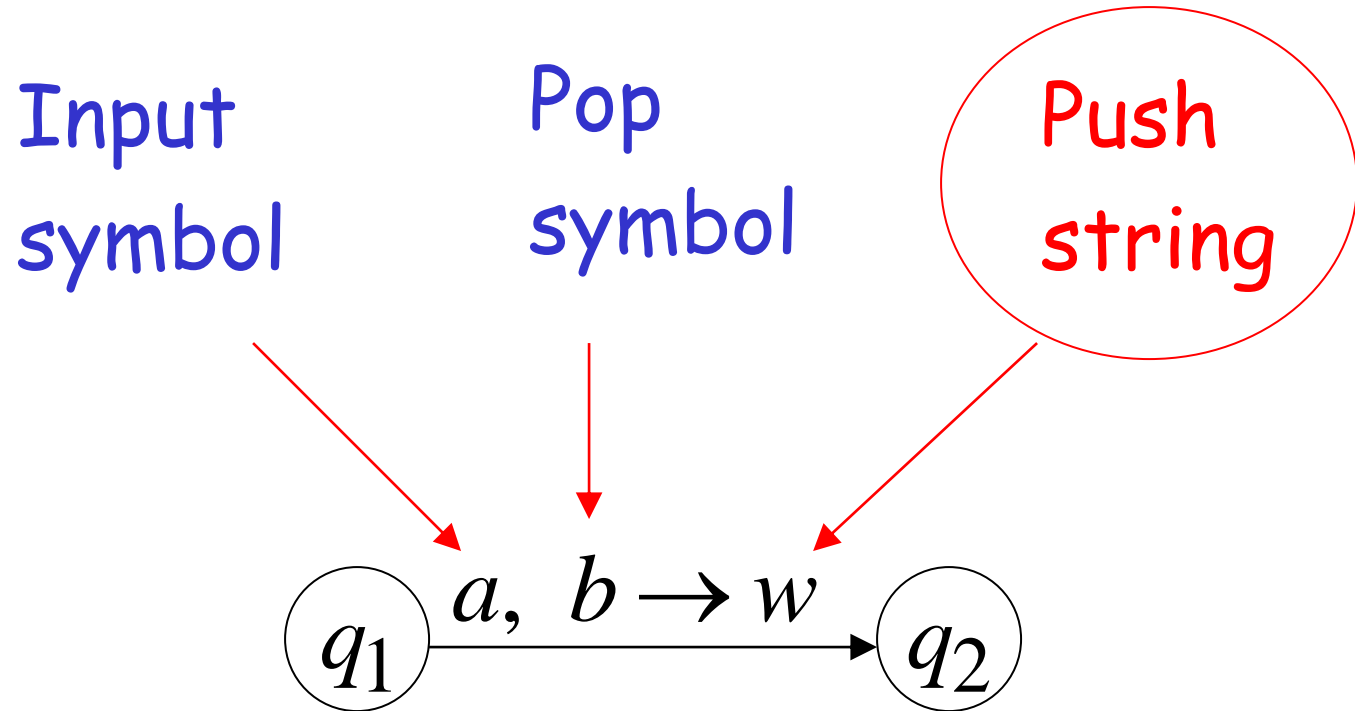
OR

The input is consumed but the last
state is not a final state

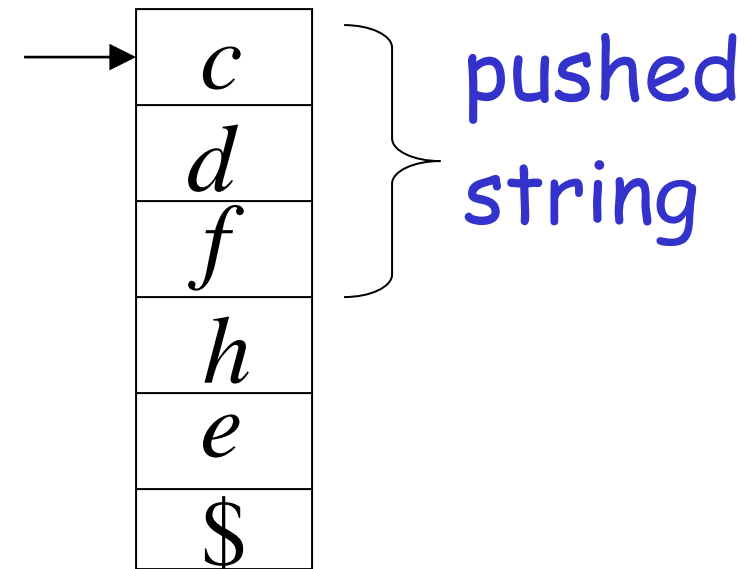
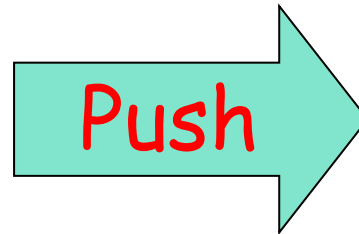
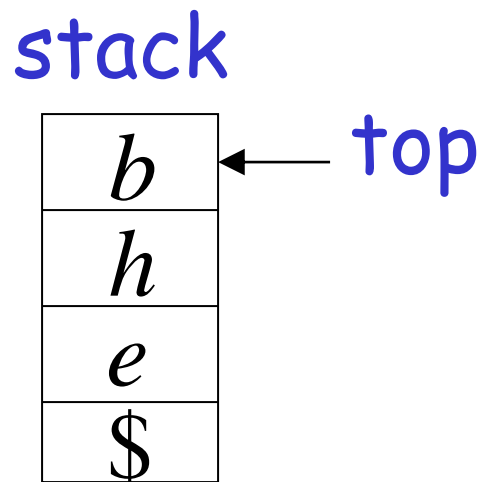
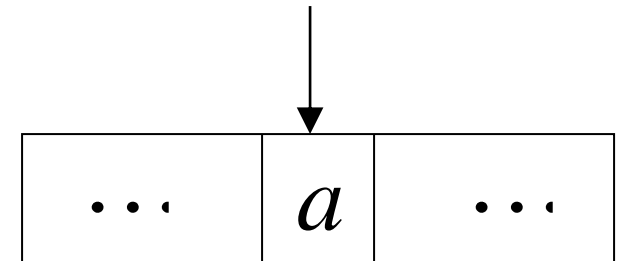
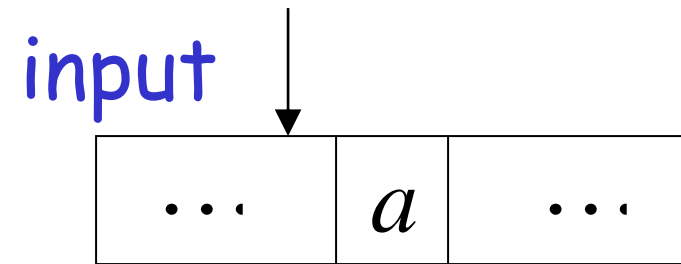
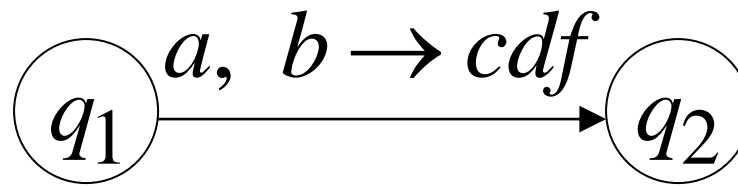
OR

The stack head moves below the
bottom of the stack

Pushing Strings



Example:



Another NPDA example

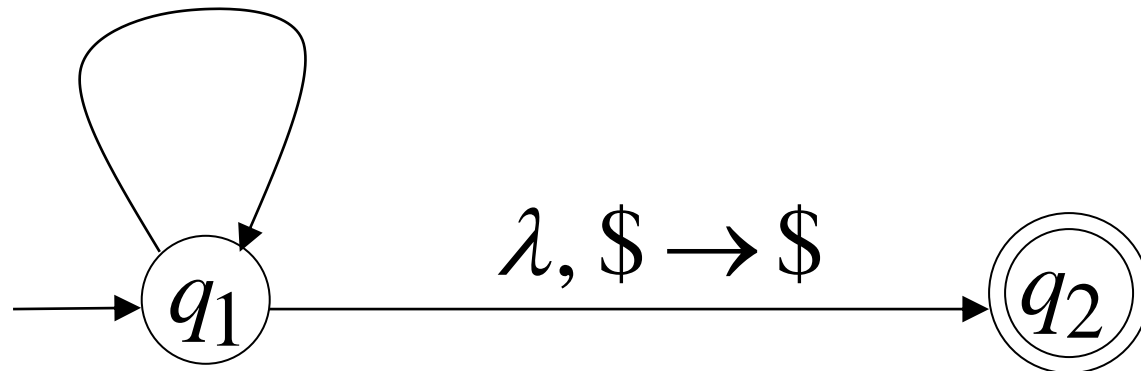
NPDA M

$$L(M) = \{w : n_a = n_b\}$$

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

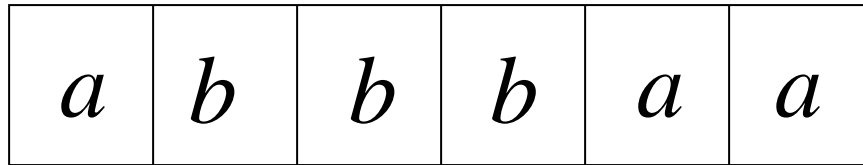
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



Execution Example: Time 0

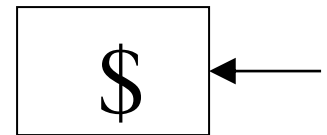
Input



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

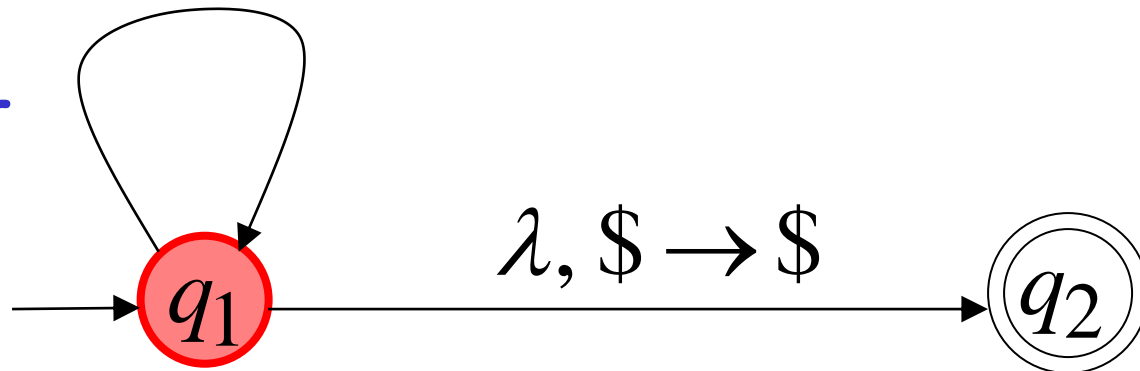
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



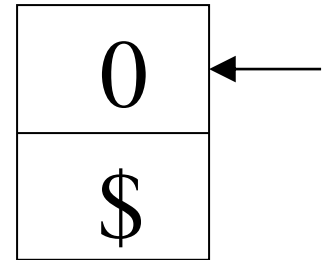
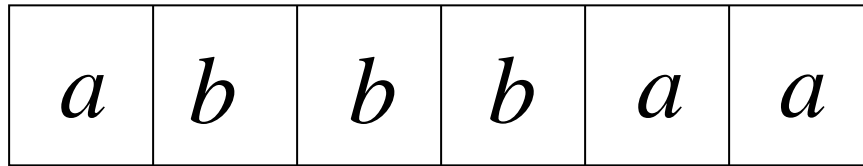
Stack

current
state



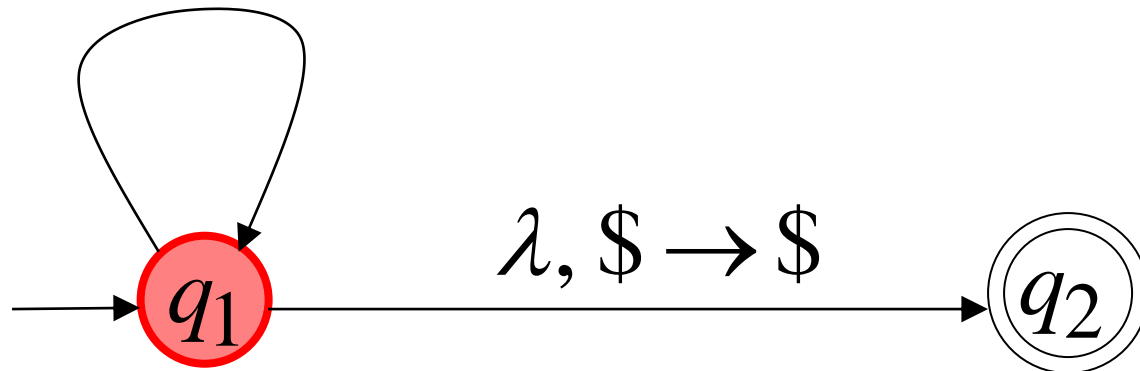
Time 1

Input



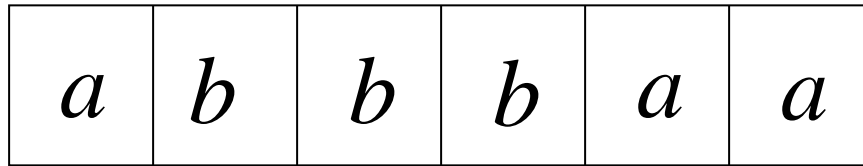
Stack

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



Time 3

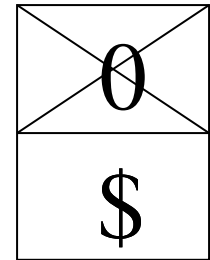
Input



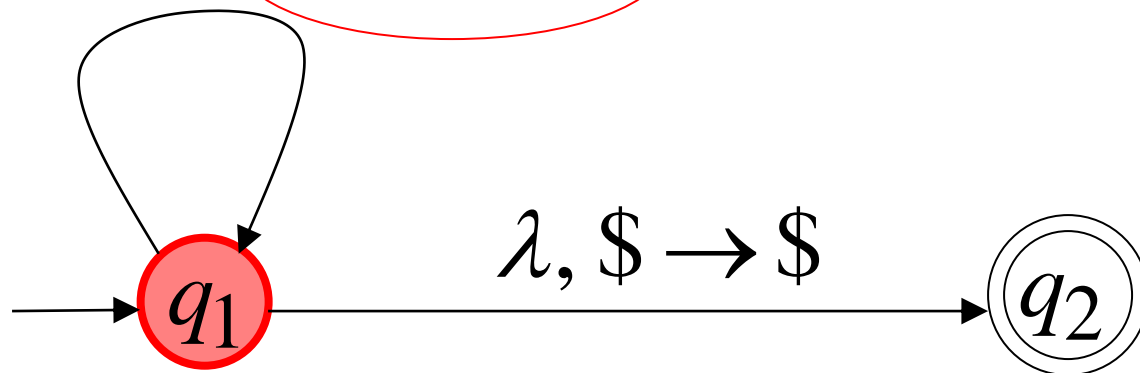
$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

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

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

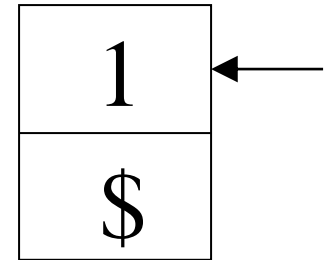
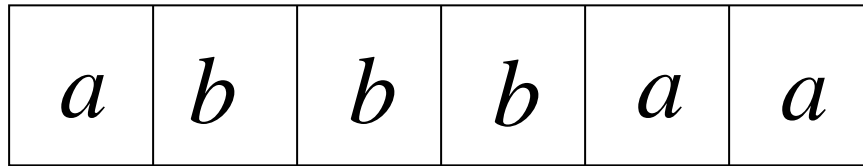


Stack



Time 4

Input

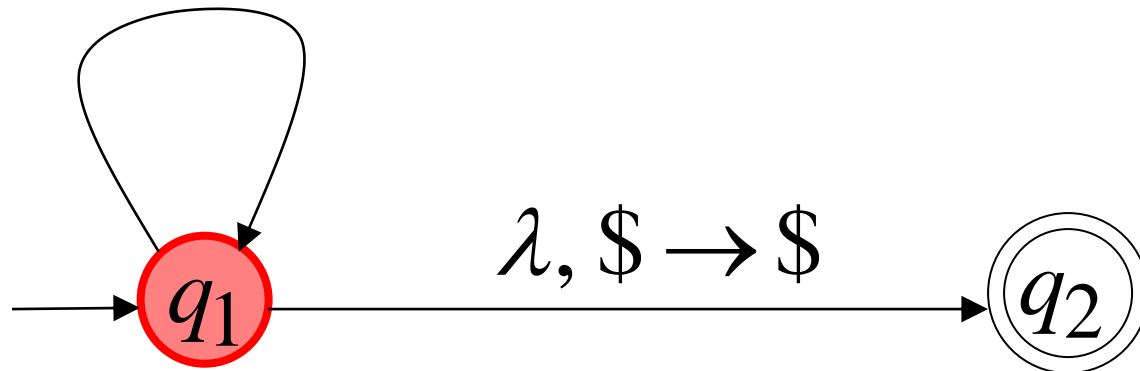


Stack

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

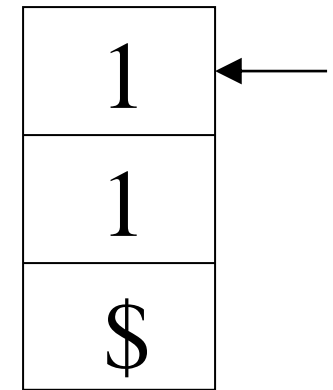
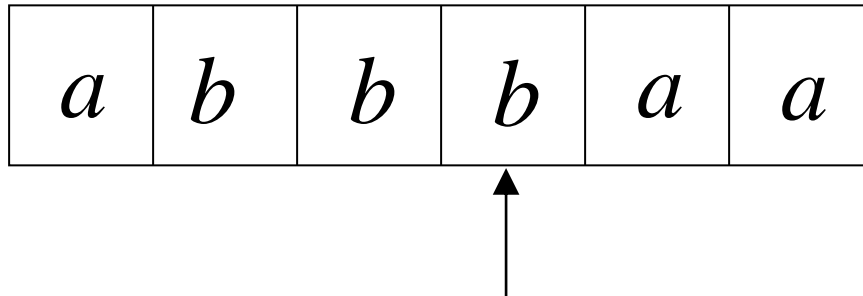
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



Time 5

Input

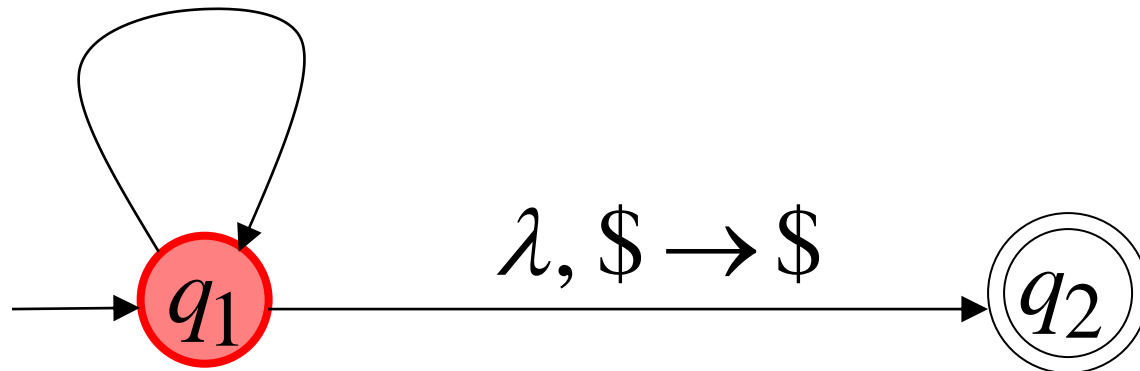


Stack

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

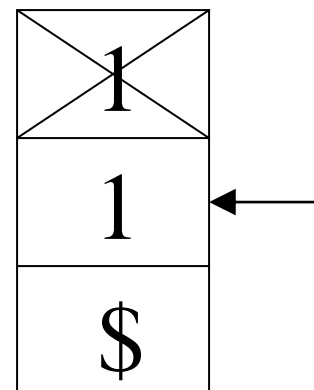
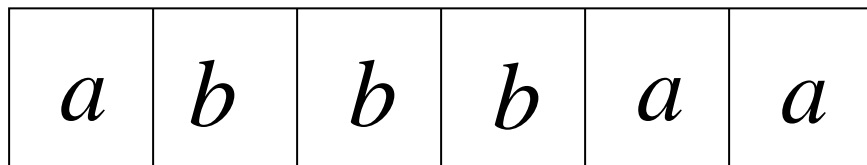
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



Time 6

Input

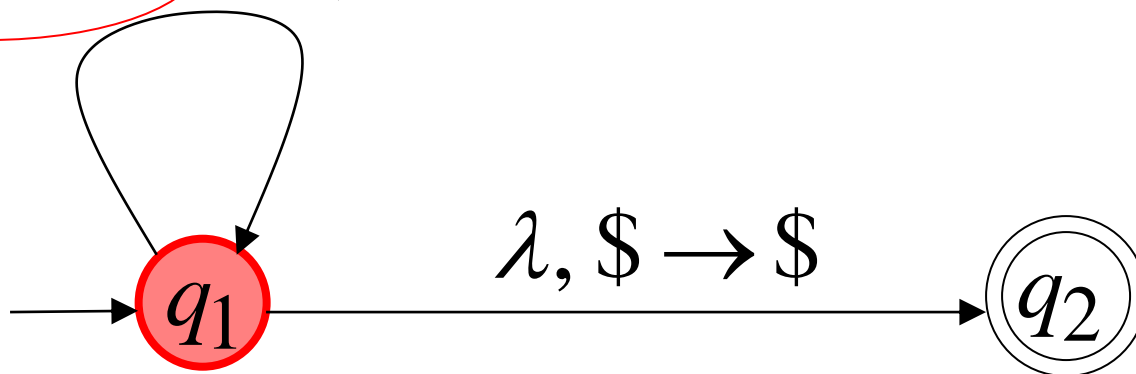


Stack

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

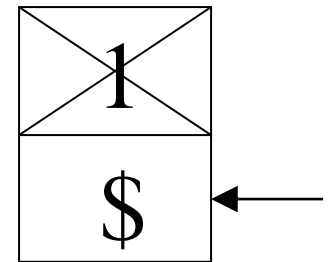
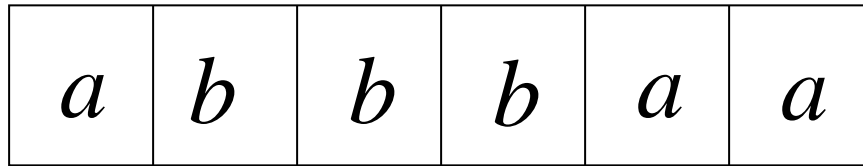
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



Time 7

Input

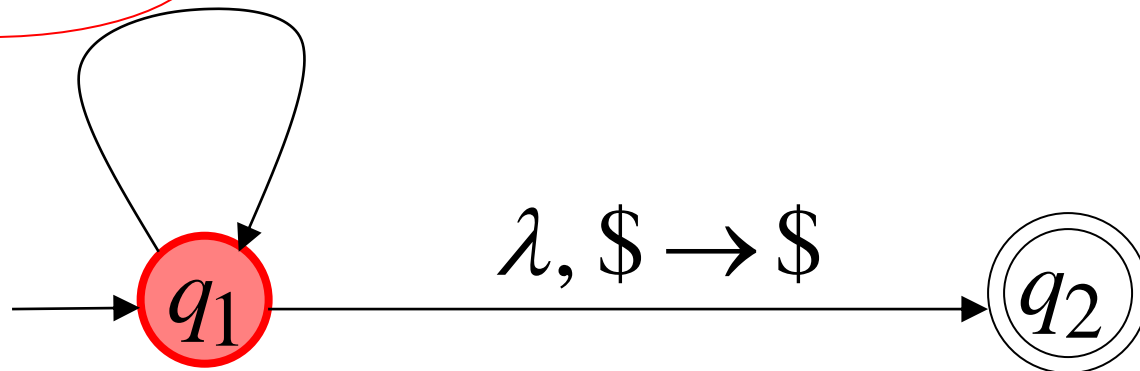


Stack

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

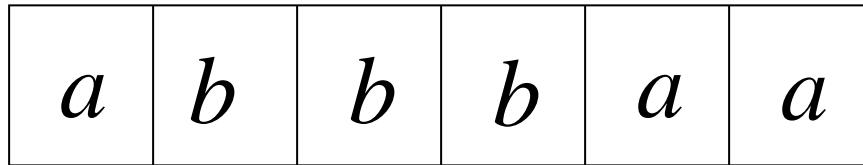
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



Time 8

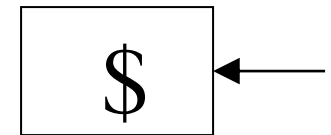
Input



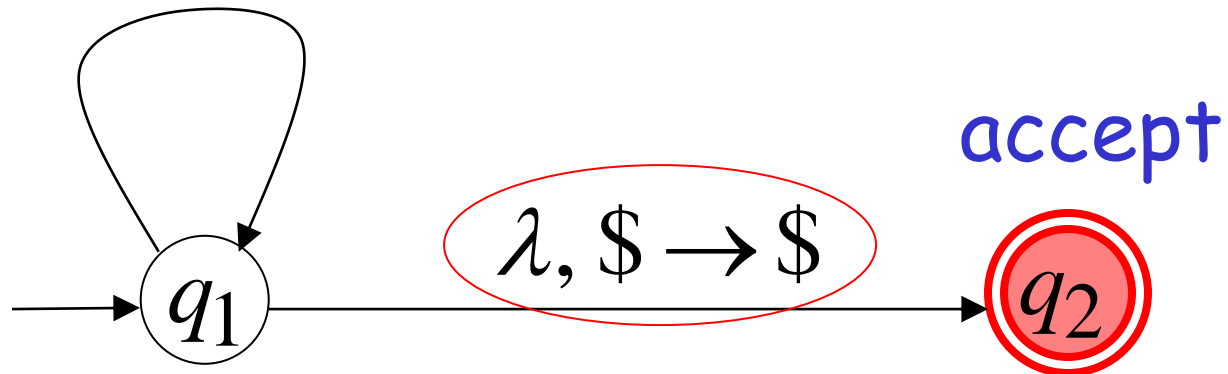
$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

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

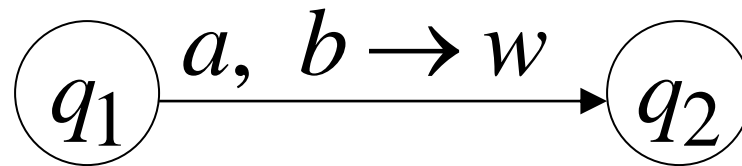
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Stack

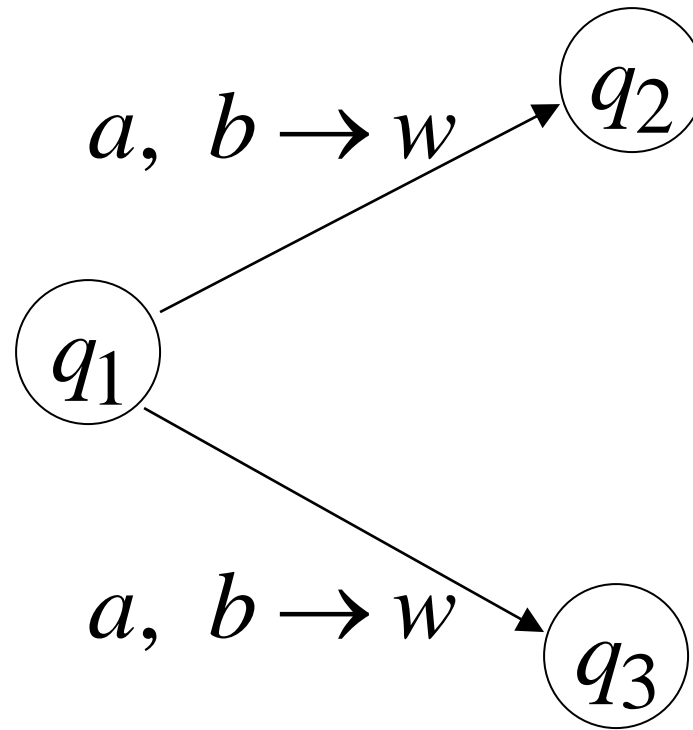


Formalities for NPDAs



Transition function:

$$\delta(q_1, a, b) = \{(q_2, w)\}$$



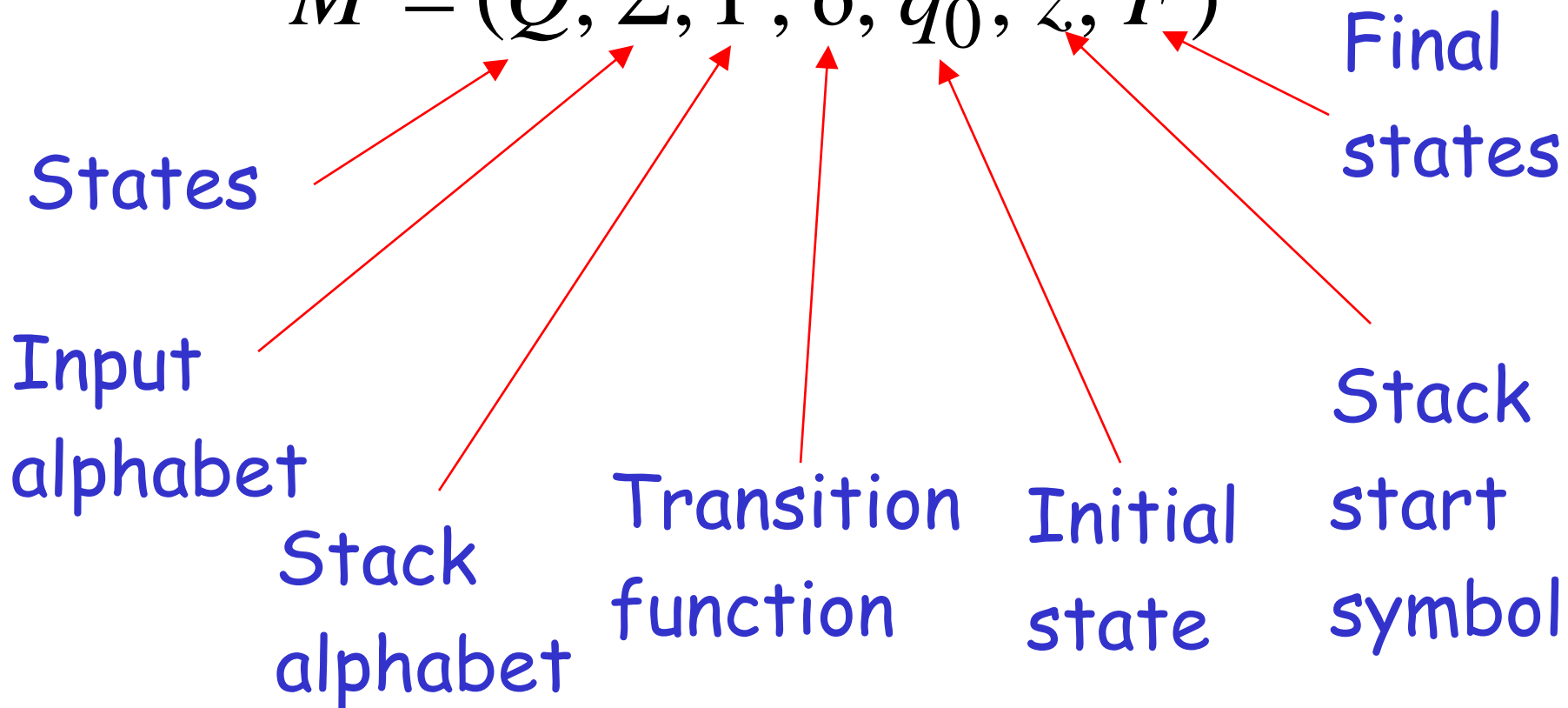
Transition function:

$$\delta(q_1, a, b) = \{(q_2, w), (q_3, w)\}$$

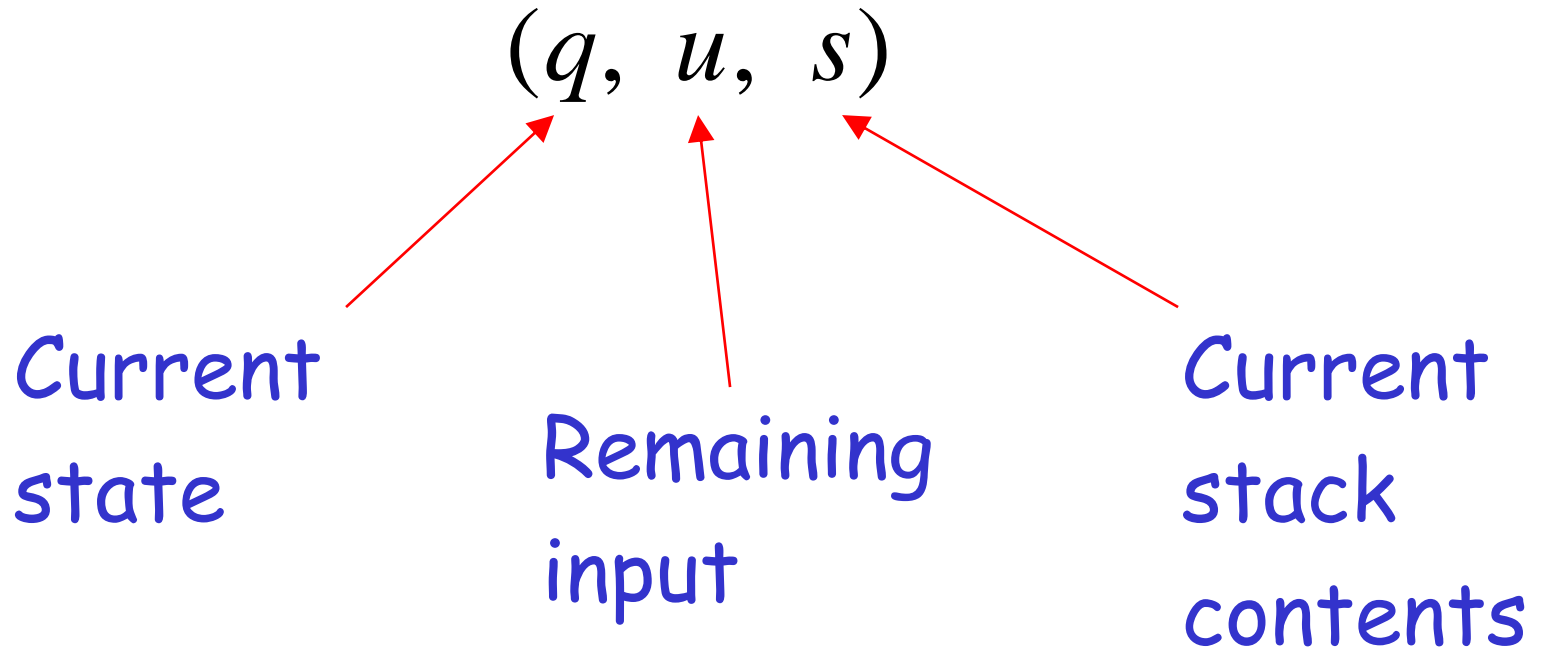
Formal Definition

Non-Deterministic Pushdown Automaton NPDA

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



Instantaneous Description



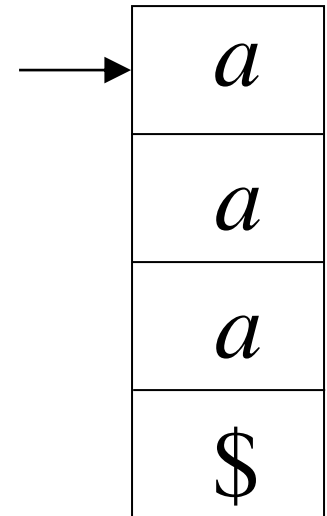
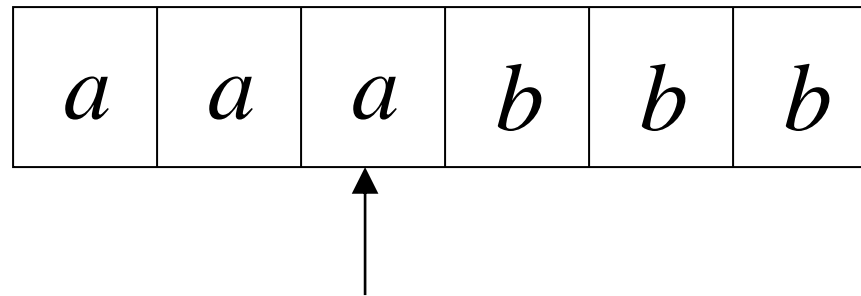
Example:

Instantaneous Description

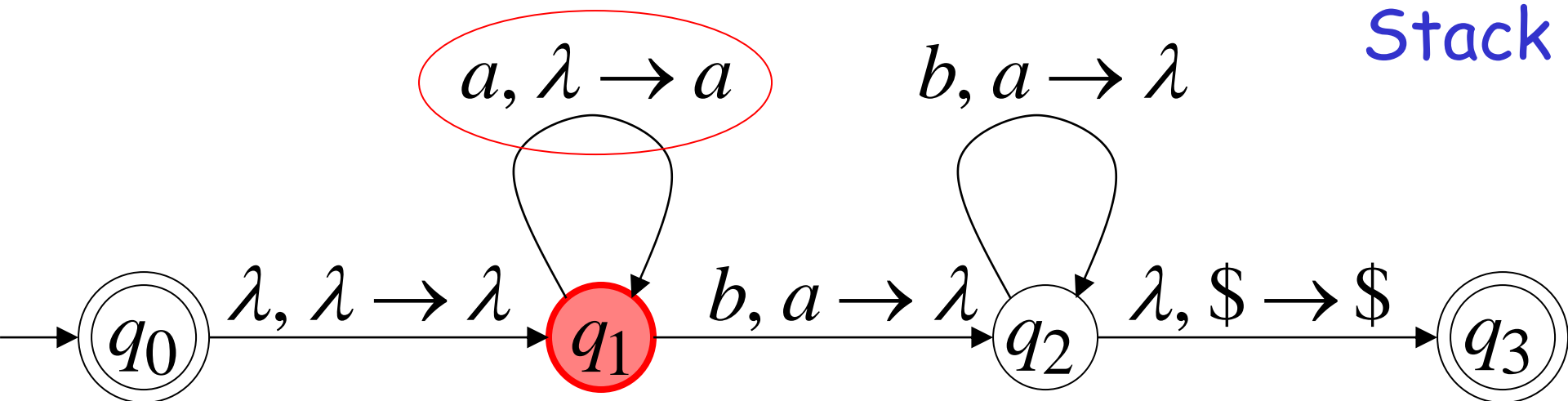
$(q_1, bbb, aaa\$)$

Time 4:

Input



Stack



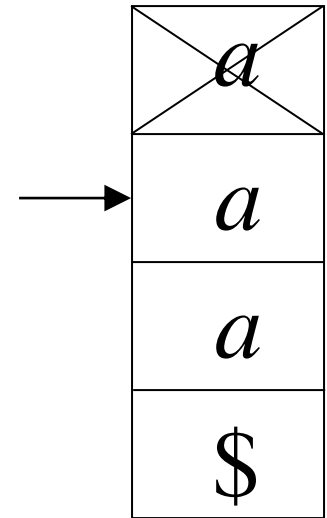
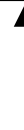
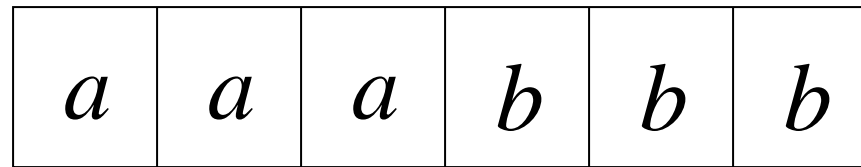
Example:

Instantaneous Description

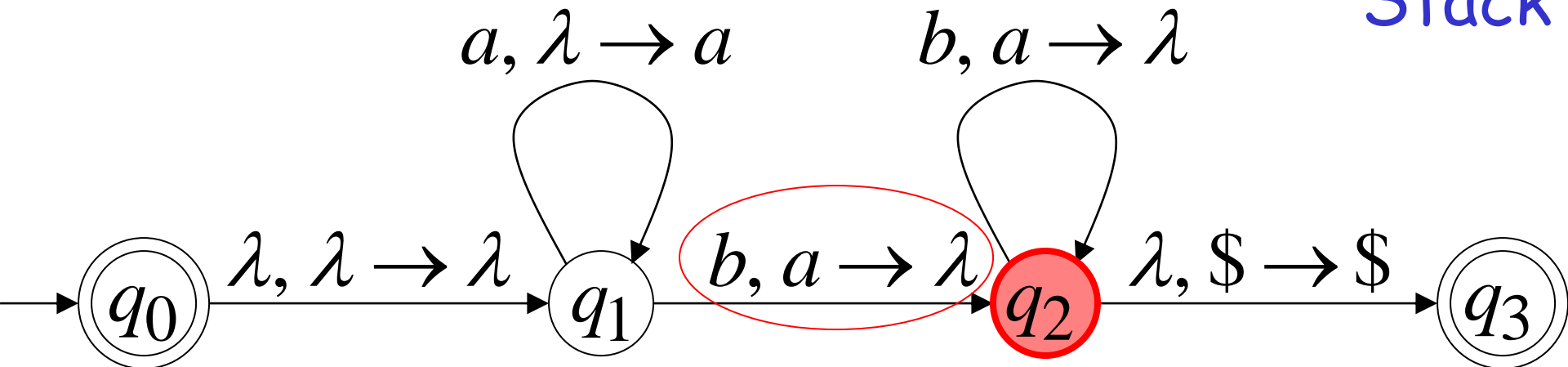
$(q_2, bb, aa\$)$

Time 5:

Input



Stack



We write:

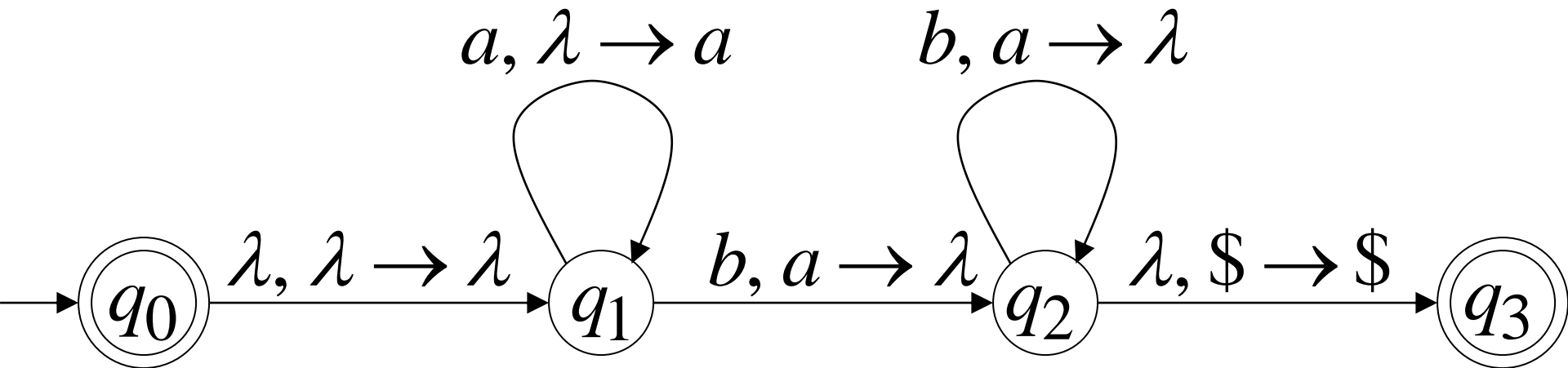
$$(q_1, bbb, aaa\$) \succ (q_2, bb, aa\$)$$

Time 4

Time 5

A computation:

$(q_0, aaabbbb, \$) \succ (q_1, aaabbbb, \$) \succ$
 $(q_1, aabbbb, a\$) \succ (q_1, abbbb, aa\$) \succ (q_1, bbb, aaa\$) \succ$
 $(q_2, bb, aa\$) \succ (q_2, b, a\$) \succ (q_2, \lambda, \$) \succ (q_3, \lambda, \$)$



$$\begin{aligned}
& (q_0, aaabbbb, \$) \succ (q_1, aaabbbb, \$) \succ \\
& (q_1, aabbbb, a\$) \succ (q_1, abbbb, aa\$) \succ (q_1, bbb, aaa\$) \succ \\
& (q_2, bb, aa\$) \succ (q_2, b, a\$) \succ (q_2, \lambda, \$) \succ (q_3, \lambda, \$)
\end{aligned}$$

For convenience we write:

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

Formal Definition

Language $L(M)$ of NPDA M :

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

Initial state



Final state



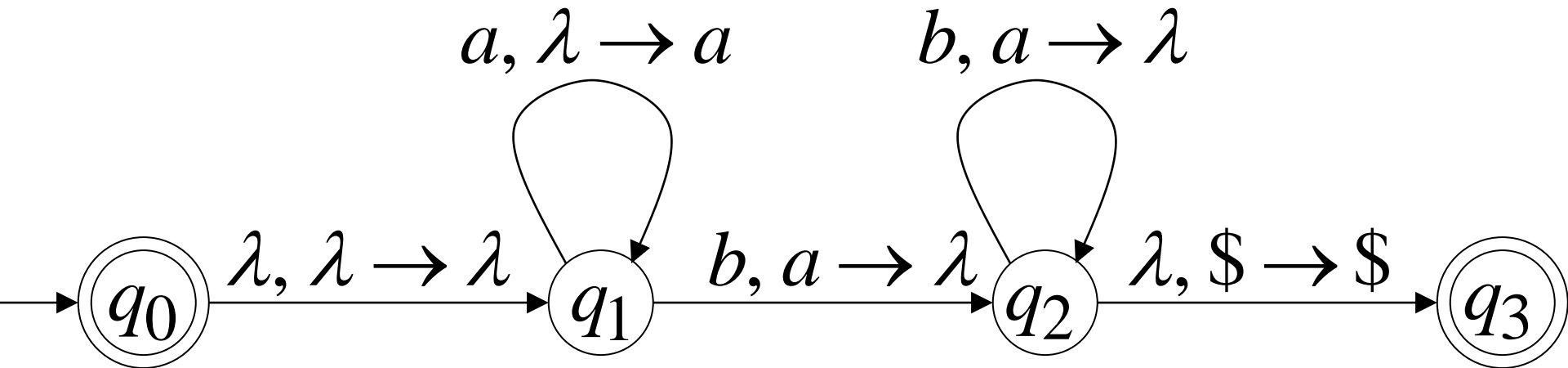
Example:

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



$$aaabbb \in L(M)$$

NPDA M :

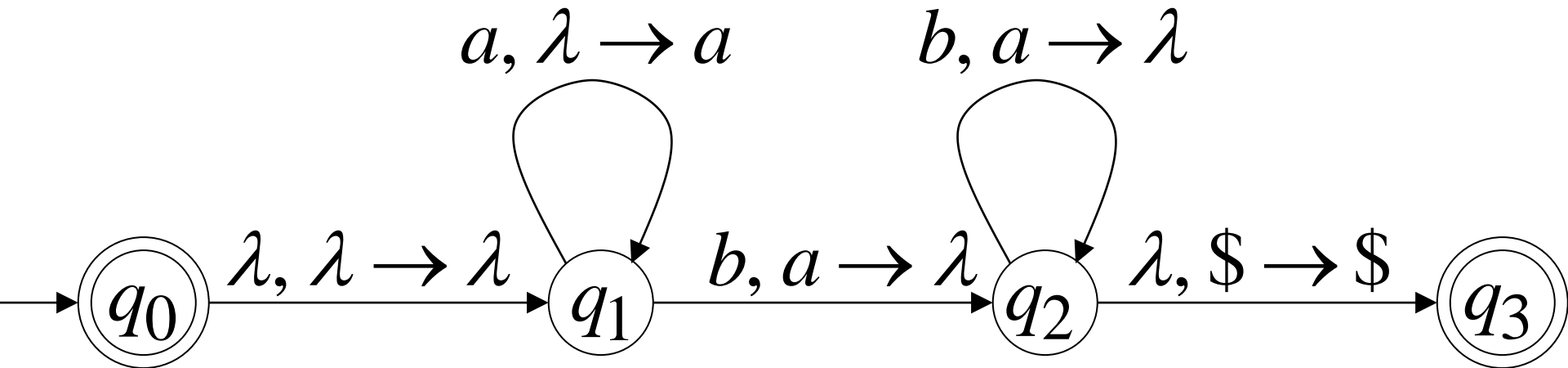


$$(q_0, a^n b^n, \$) \stackrel{*}{\succ} (q_3, \lambda, \$)$$



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

NPDA M :



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

NPDA M :

