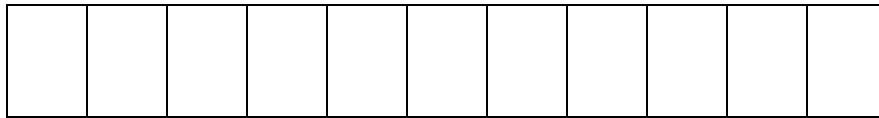
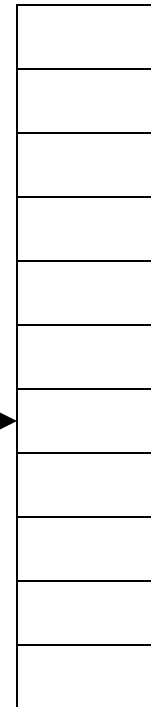


Pushdown Automaton -- PDA

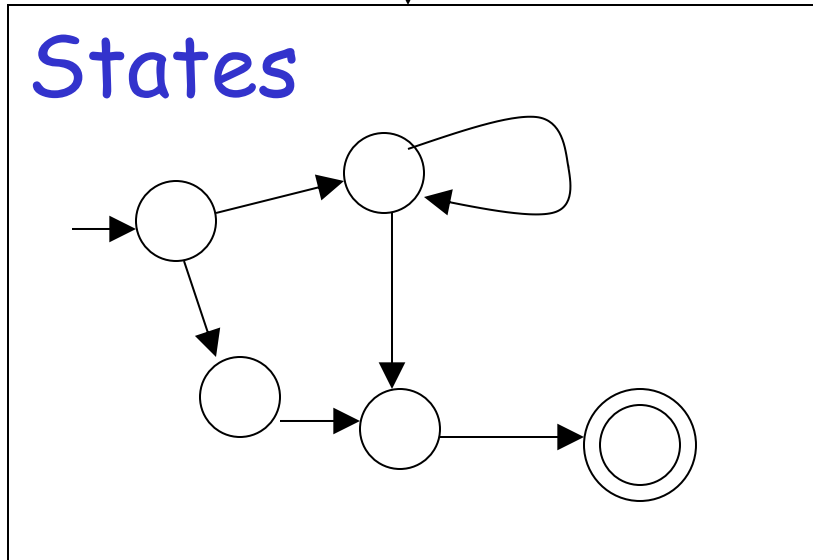
Input String



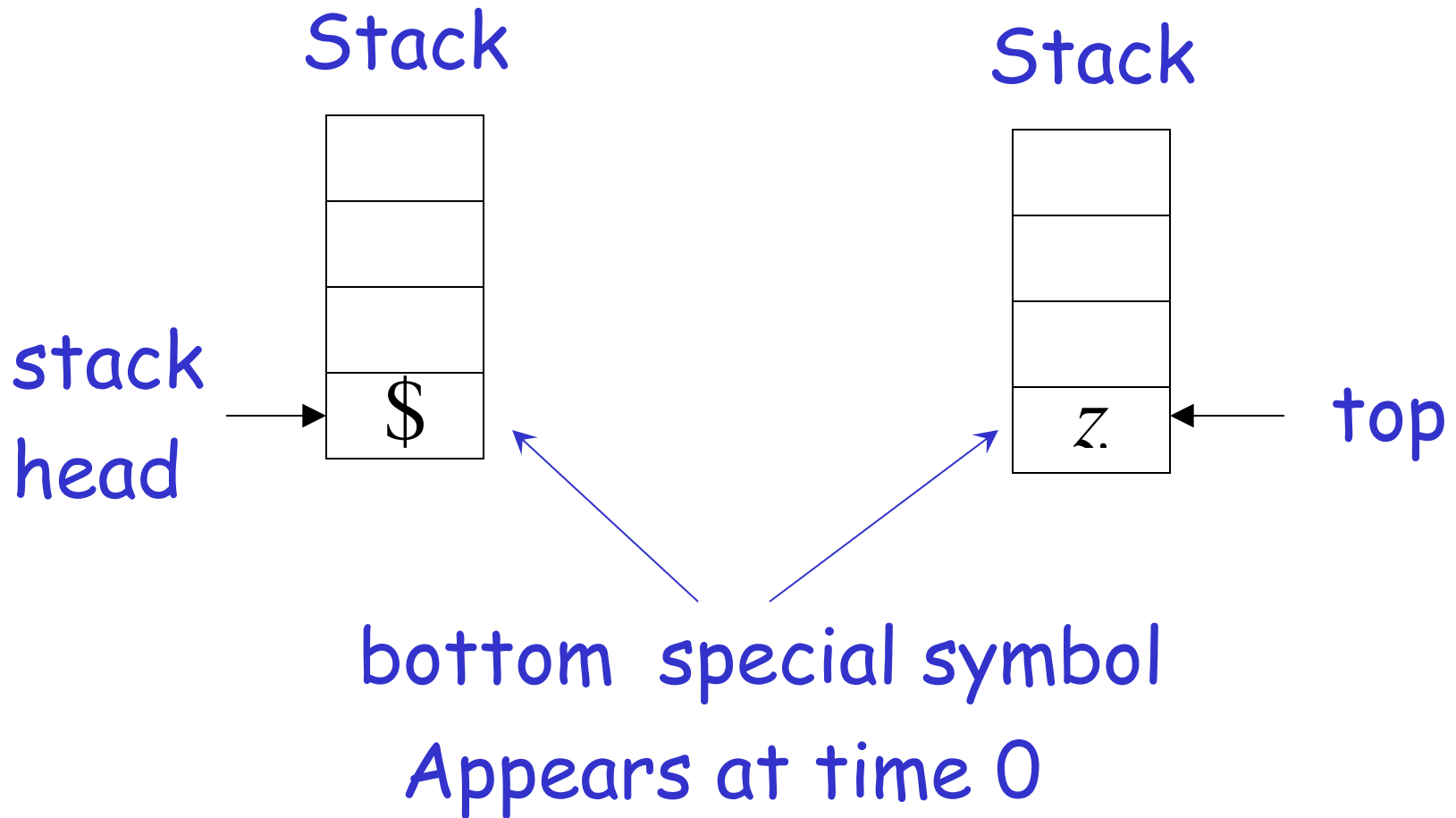
Stack



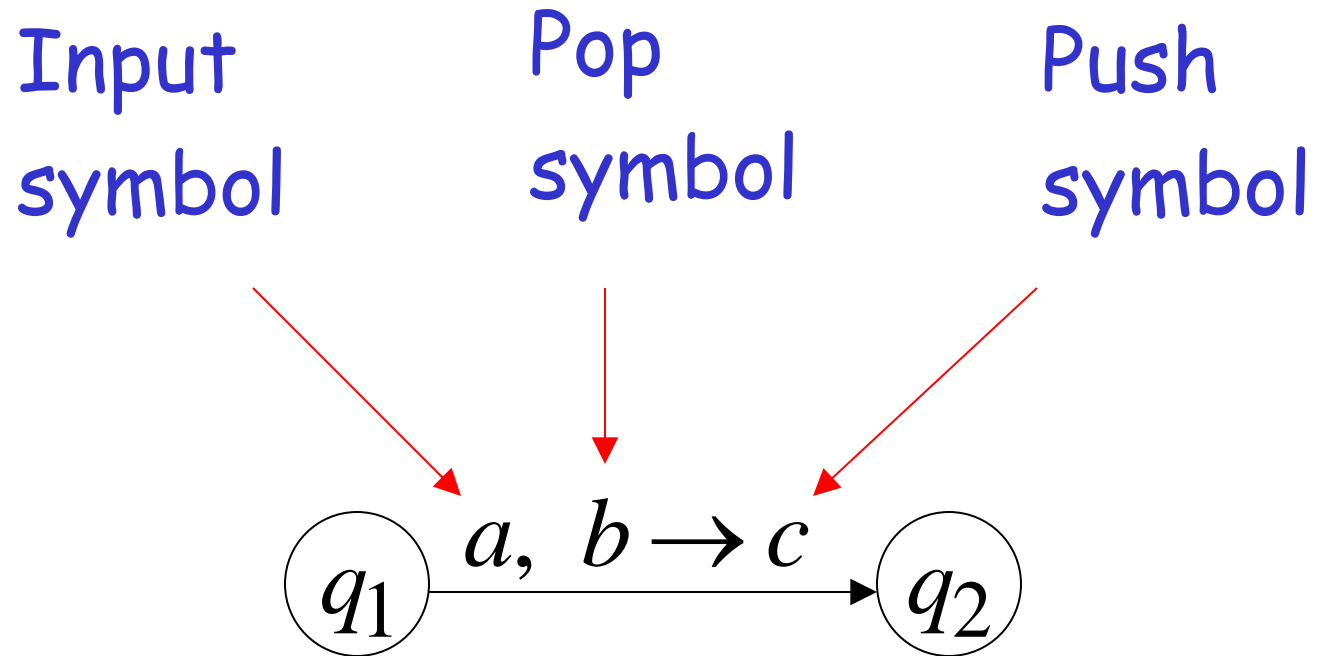
States

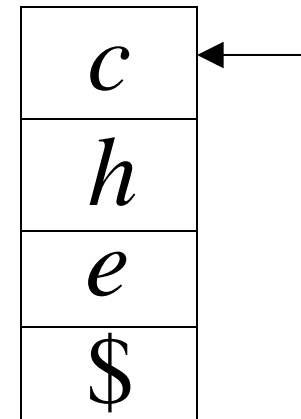
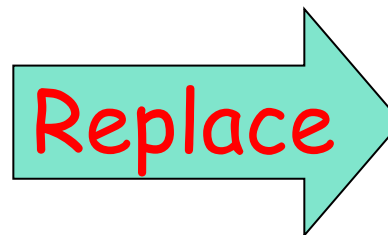
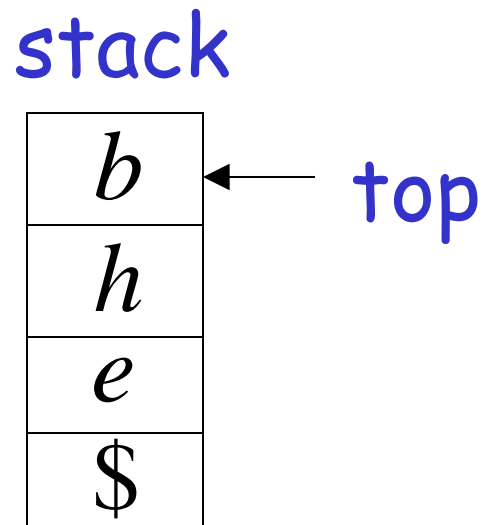
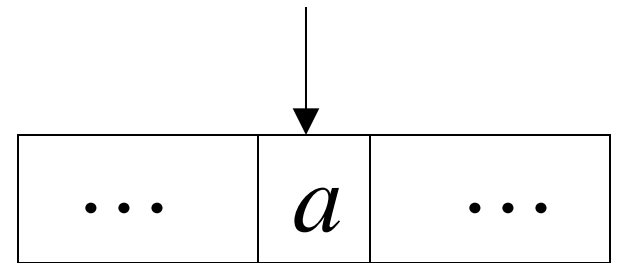
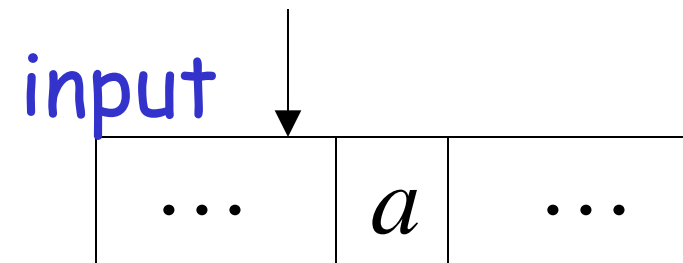
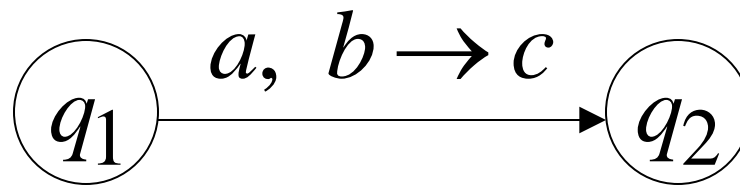


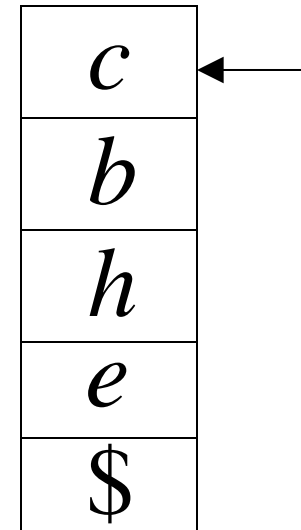
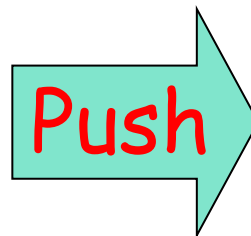
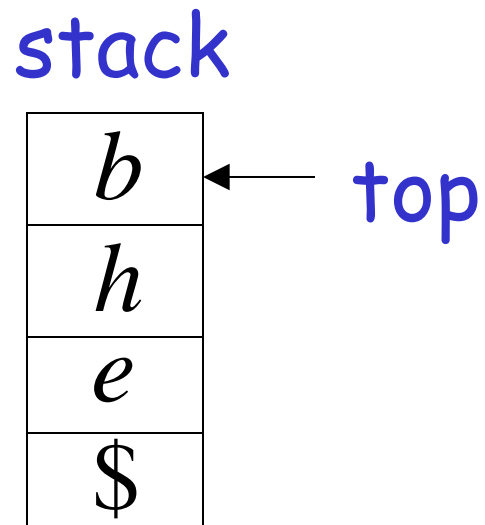
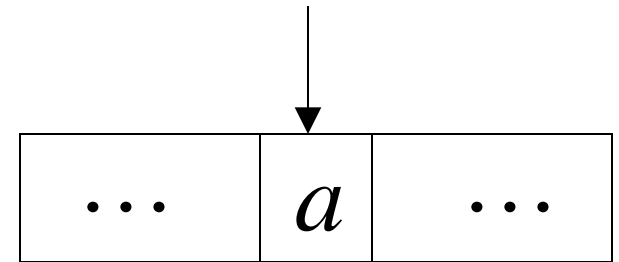
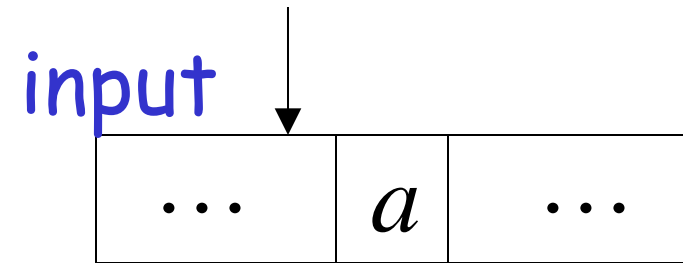
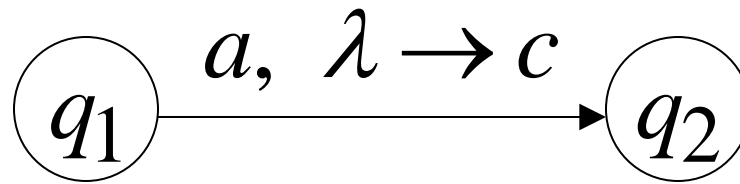
Initial Stack Symbol

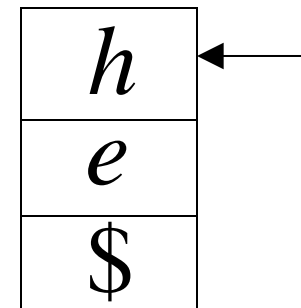
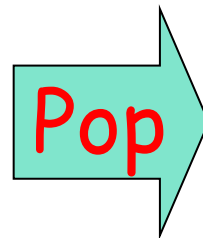
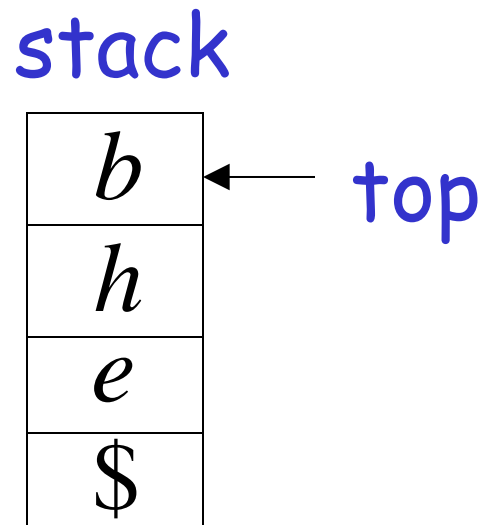
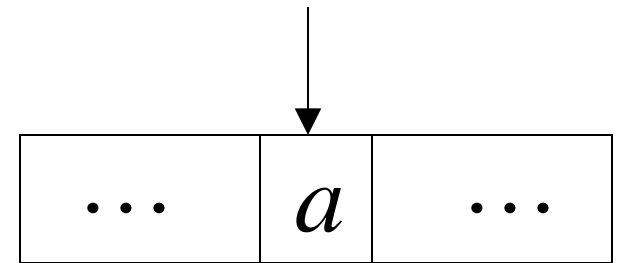
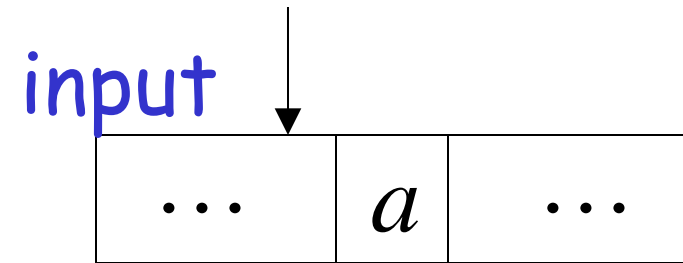
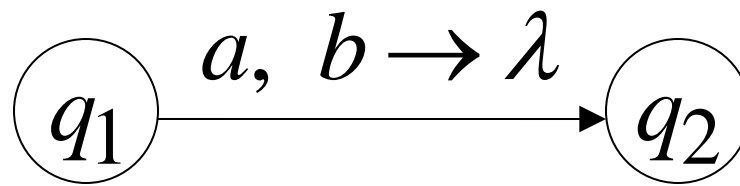


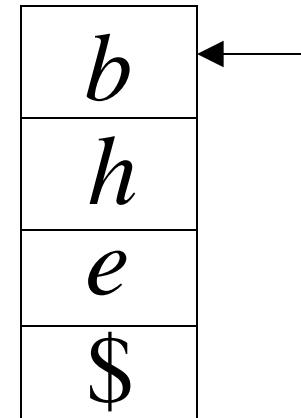
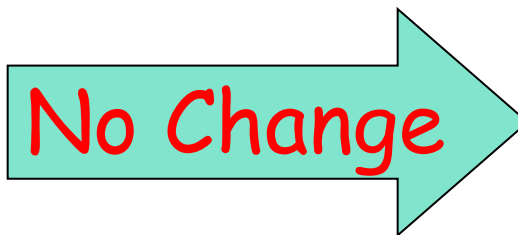
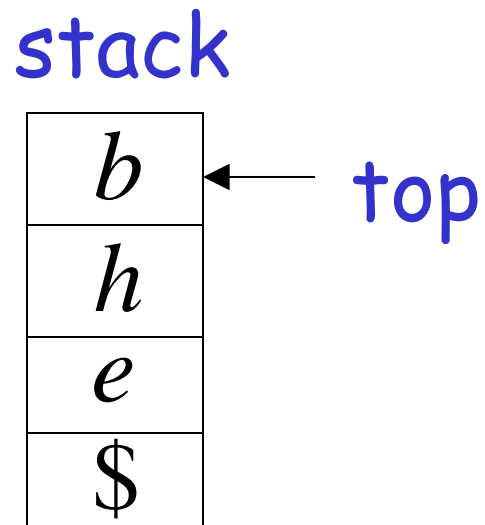
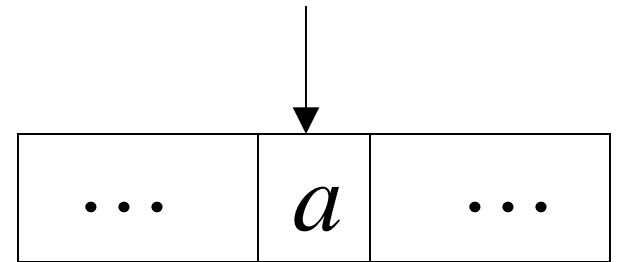
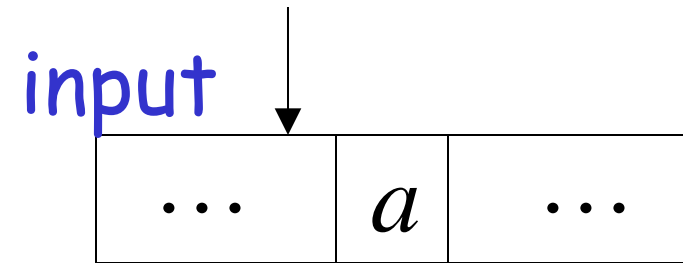
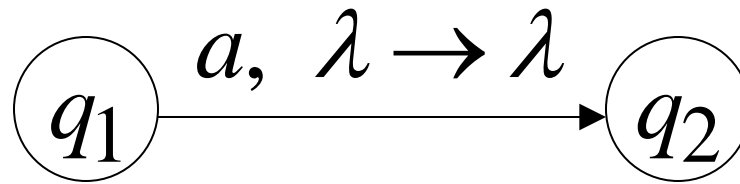
The States



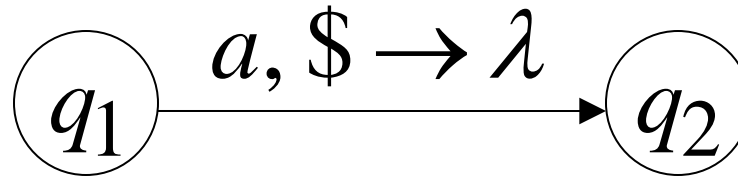




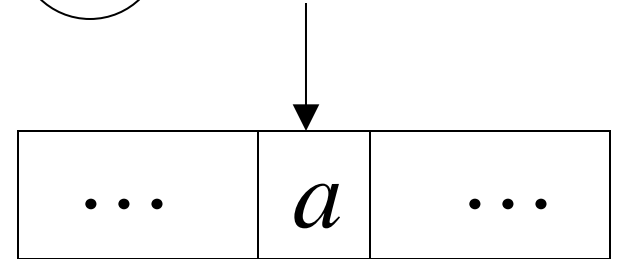
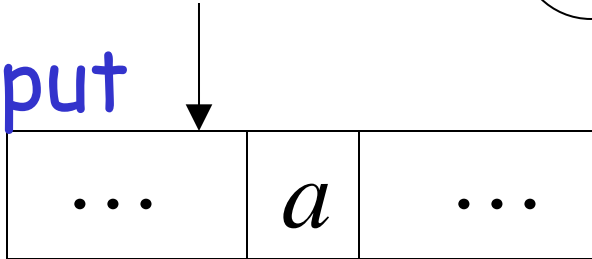




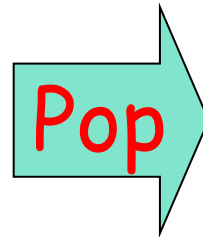
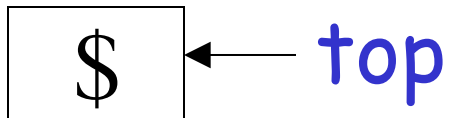
Empty Stack



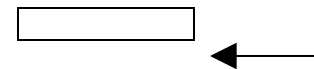
input



stack

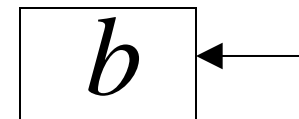
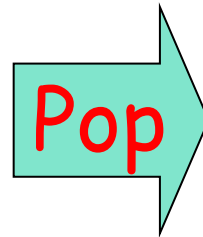
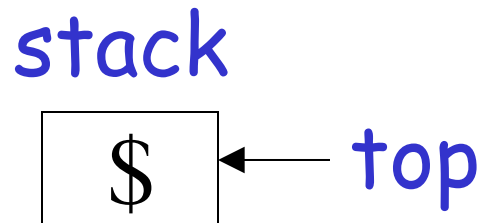
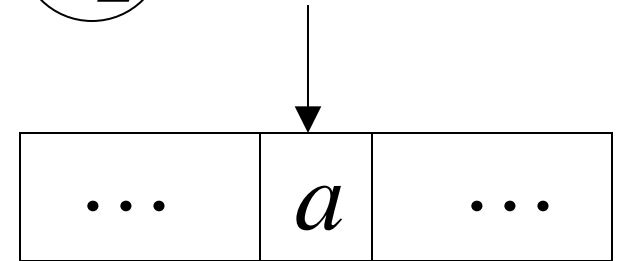
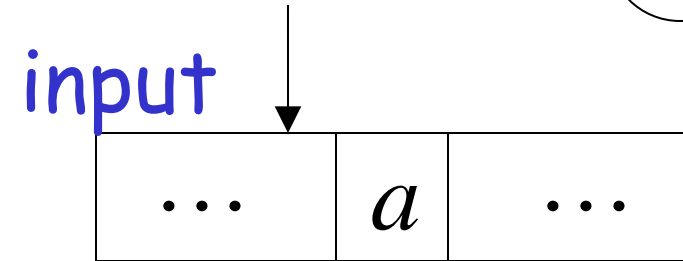
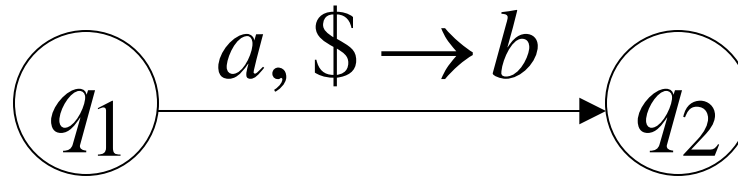


empty



The automaton **HALTS**
No possible transition after q_2

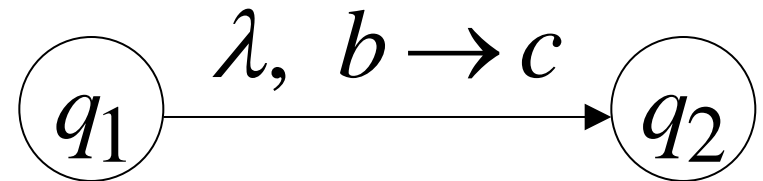
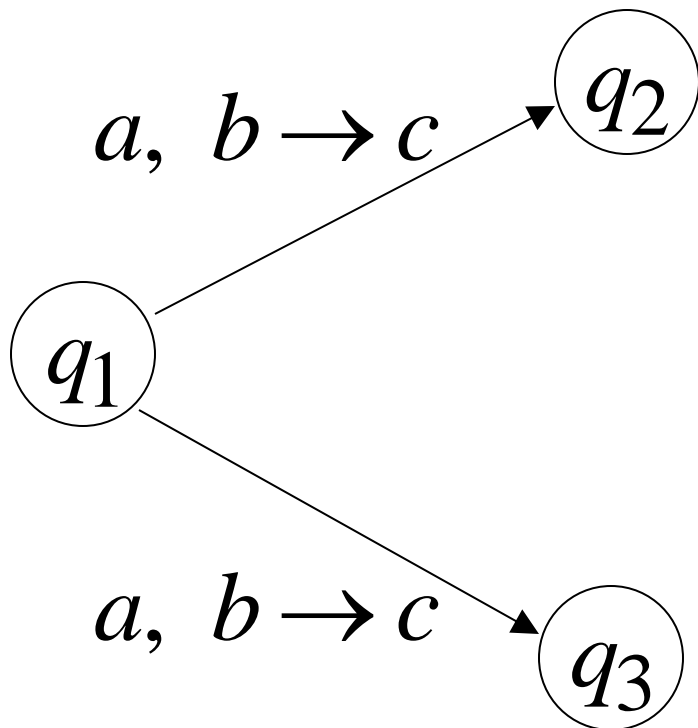
A Possible Transition



Non-Determinism

PDAs are non-deterministic

Allow non-deterministic transitions

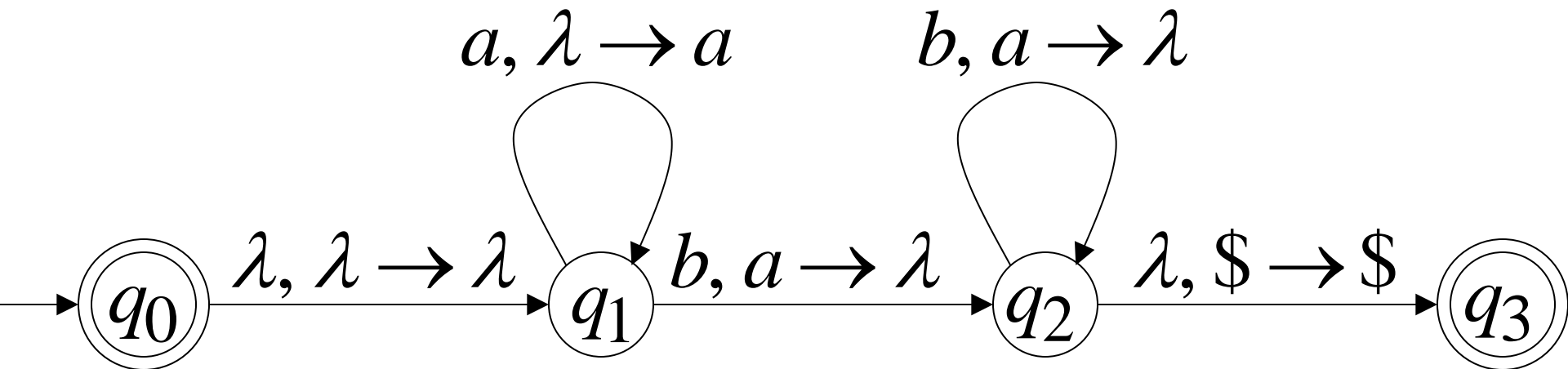


λ – transition

Example PDA: Basic Idea?

PDA M

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



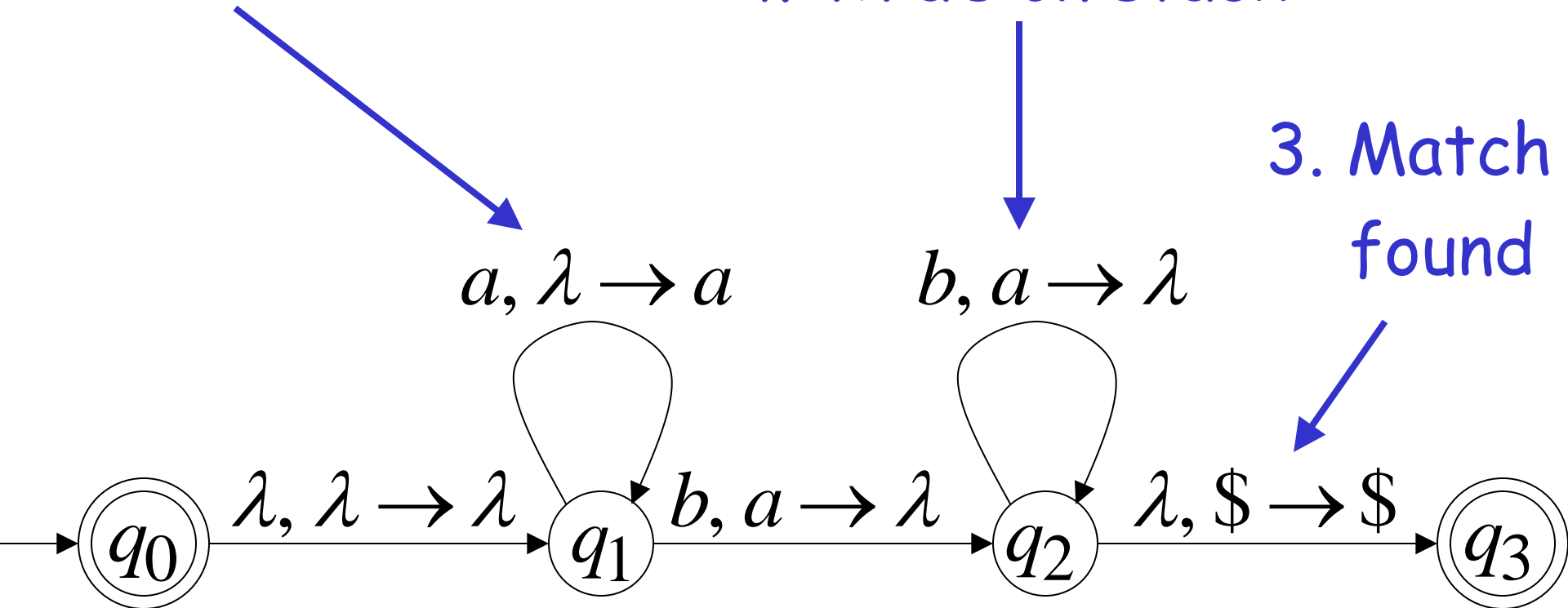
$$L(M) = \{a^n b^n : n \geq 0\}$$

Basic Idea:

1. Push the a's
on the stack

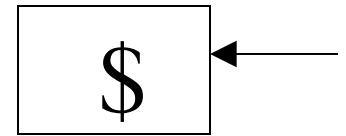
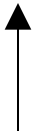
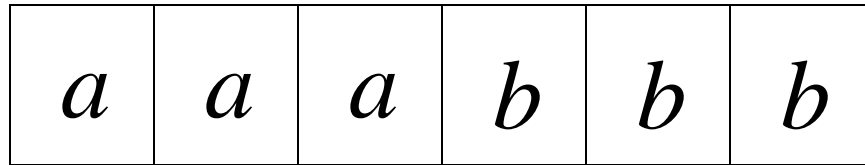
2. Match the b's on input
with a's on stack

3. Match
found

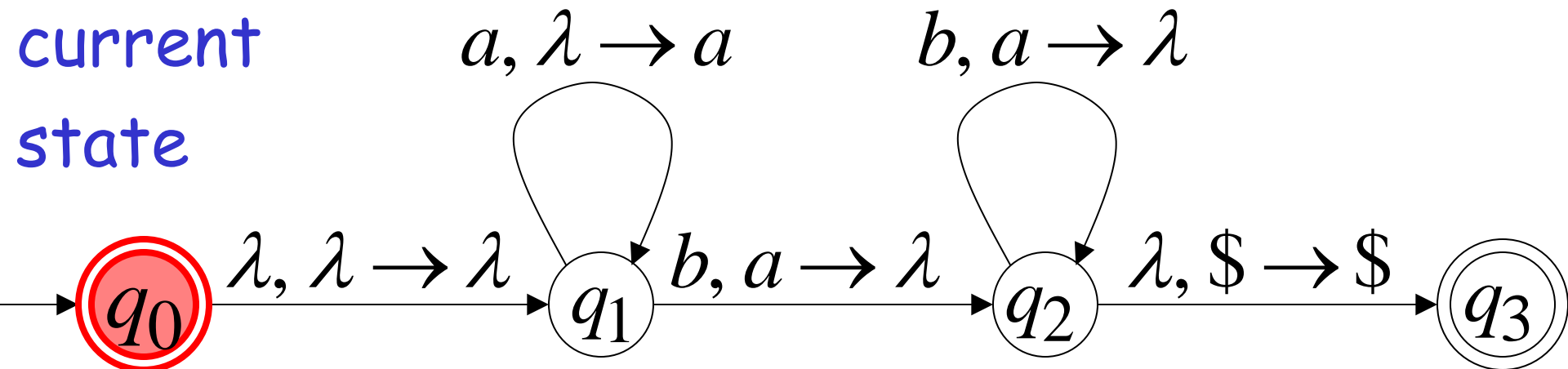


Execution Example: Time 0

Input

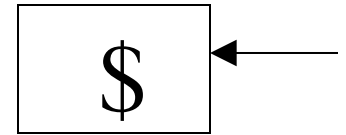
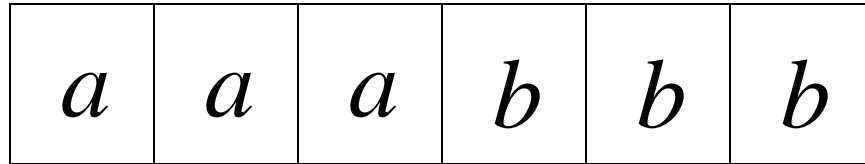


Stack

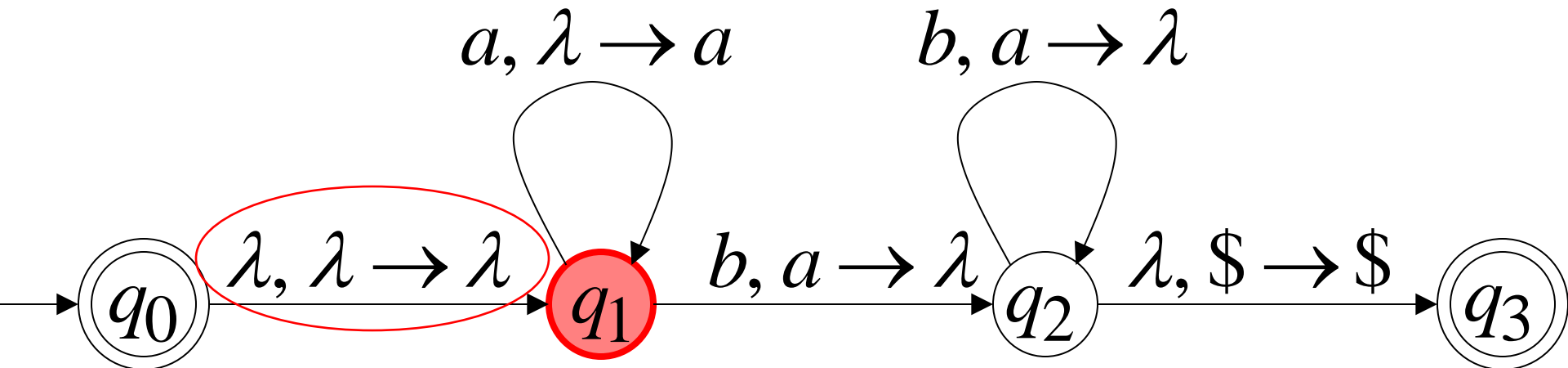


Time 1

Input

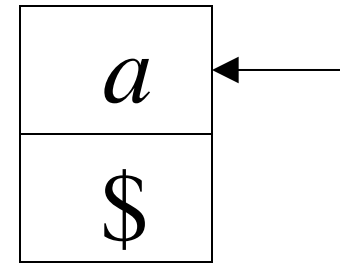
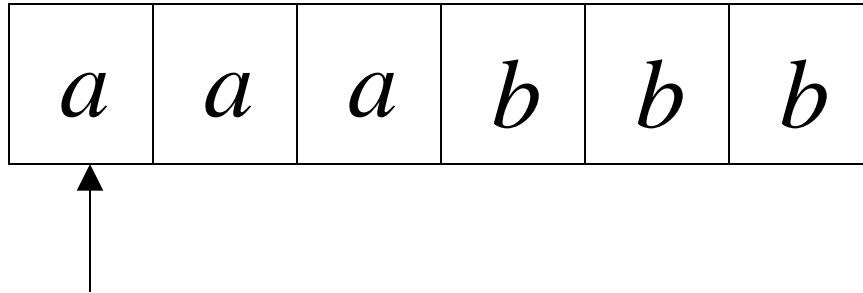


Stack

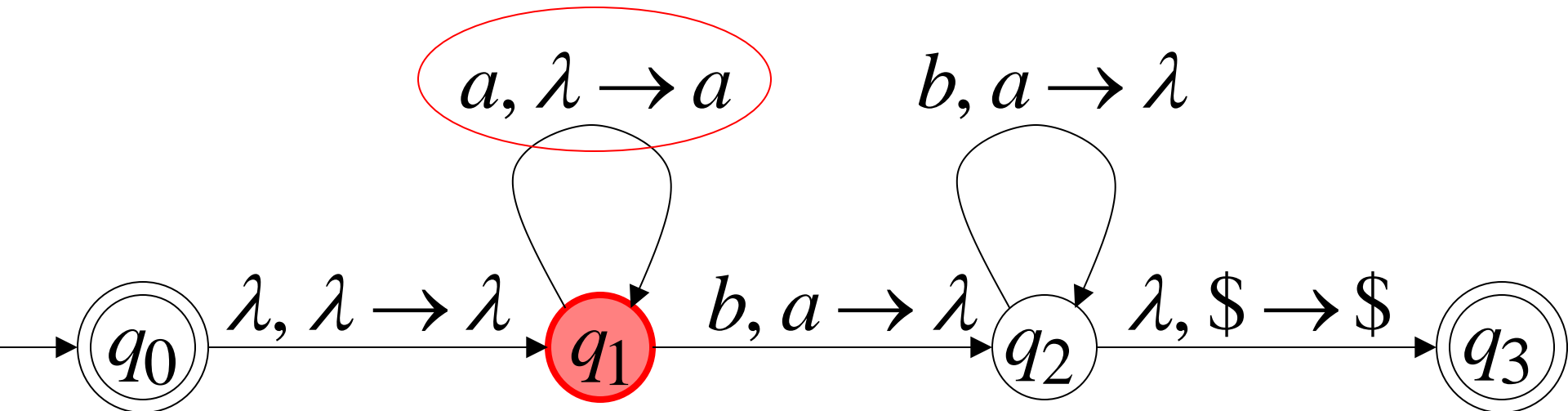


Time 2

Input

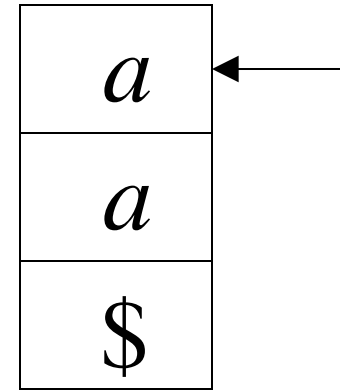
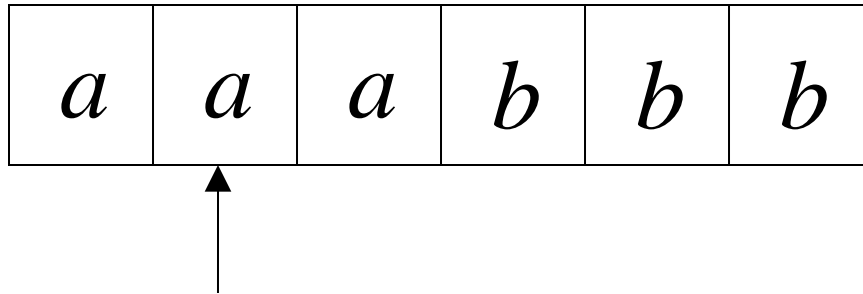


Stack

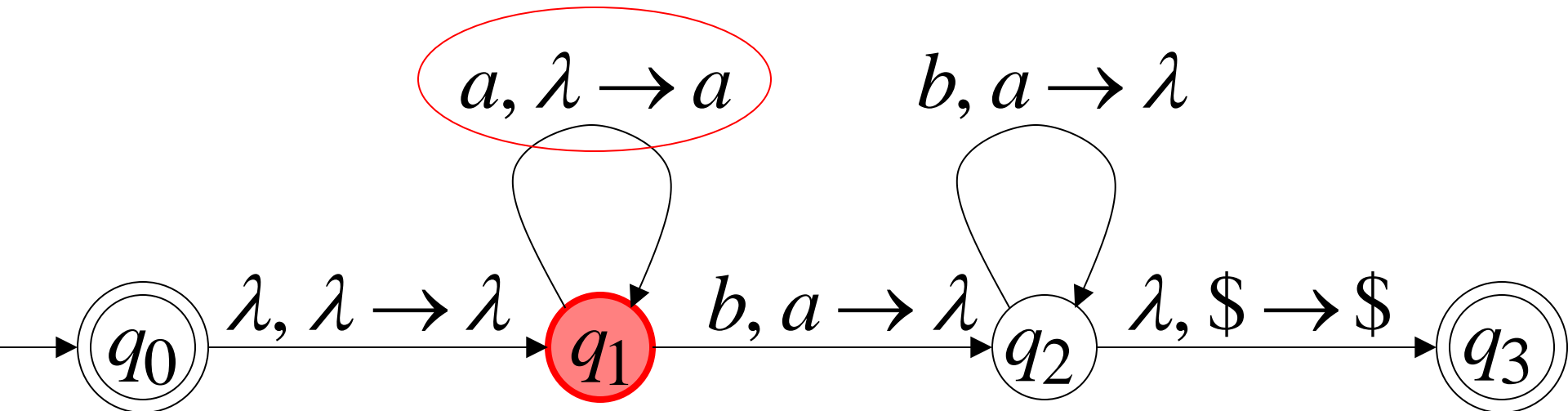


Time 3

Input

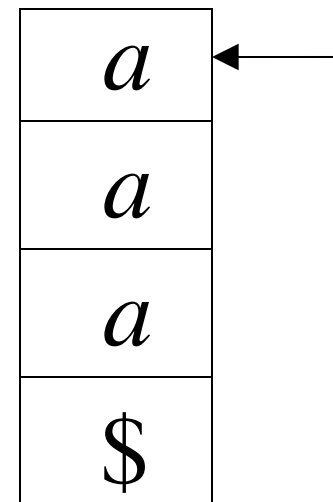
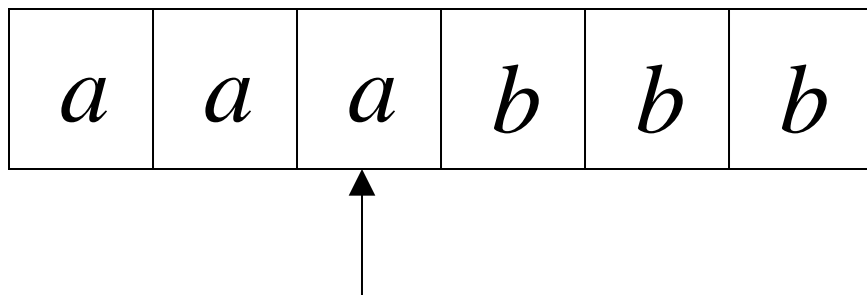


Stack

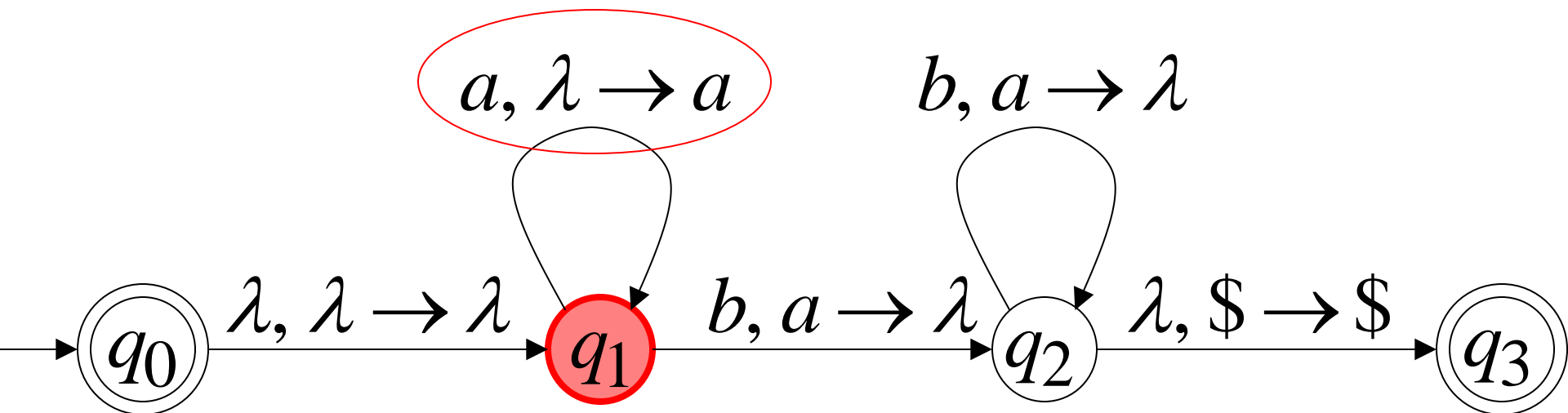


Time 4

Input

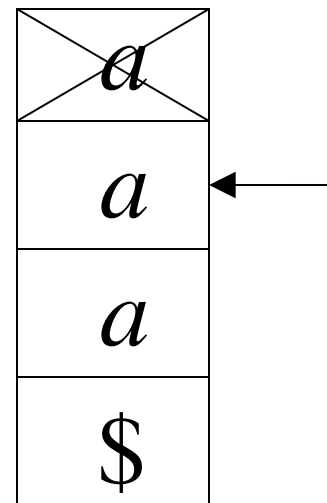
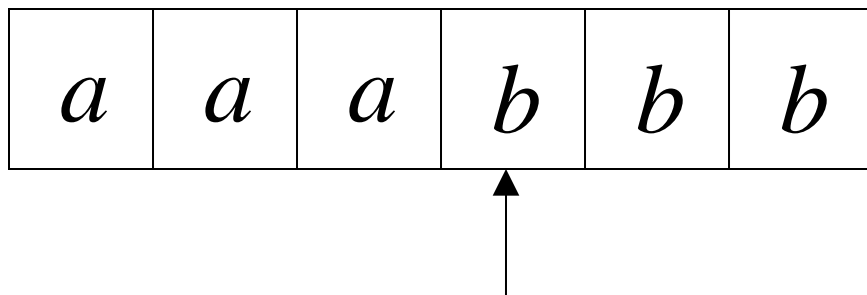


Stack

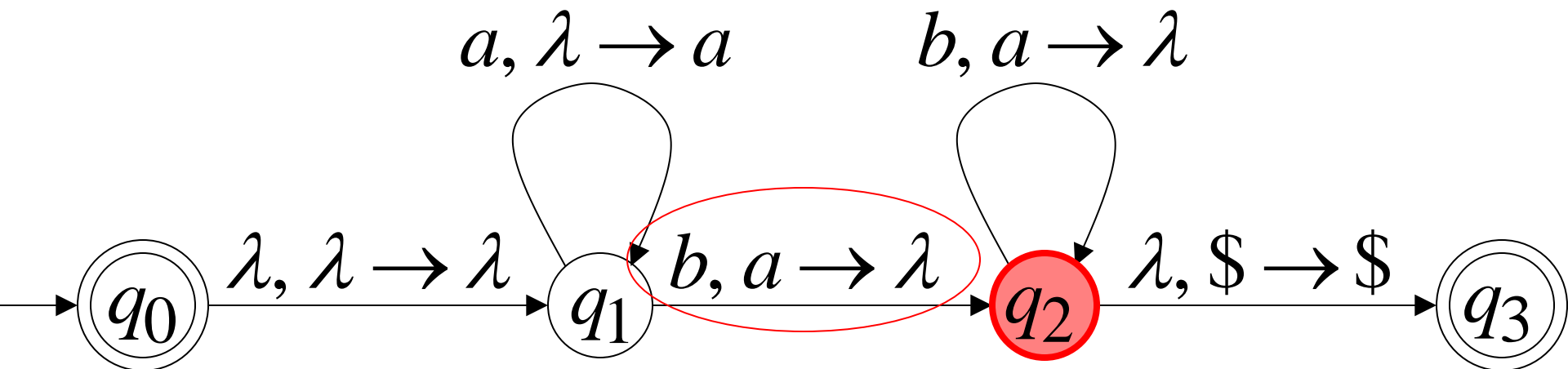


Time 5

Input

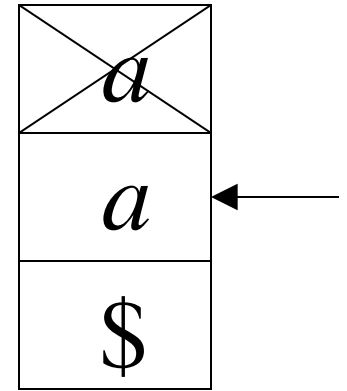
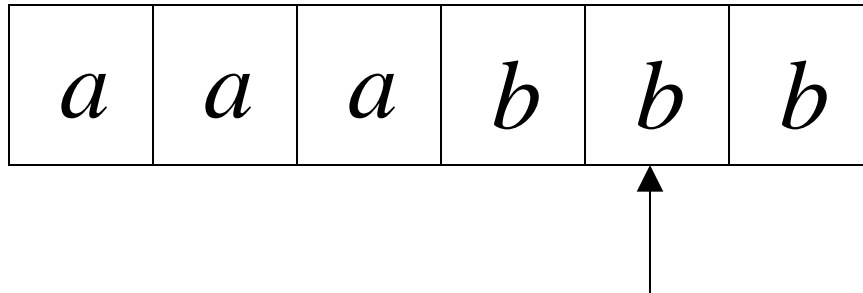


Stack

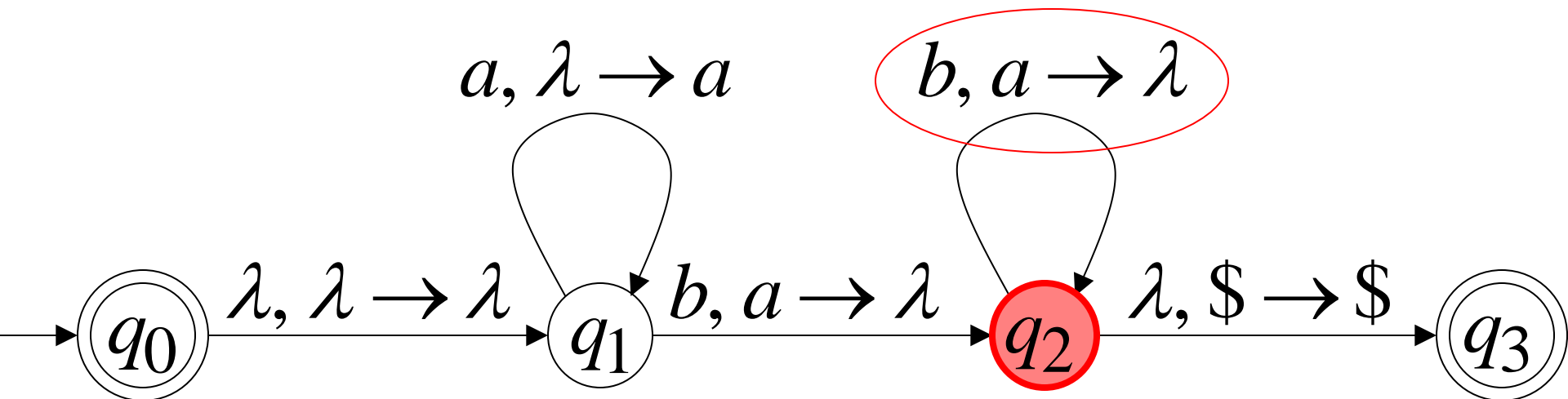


Time 6

Input

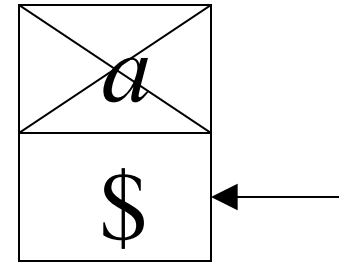
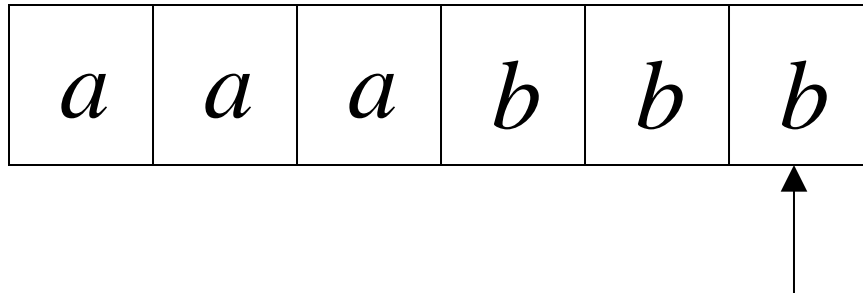


Stack

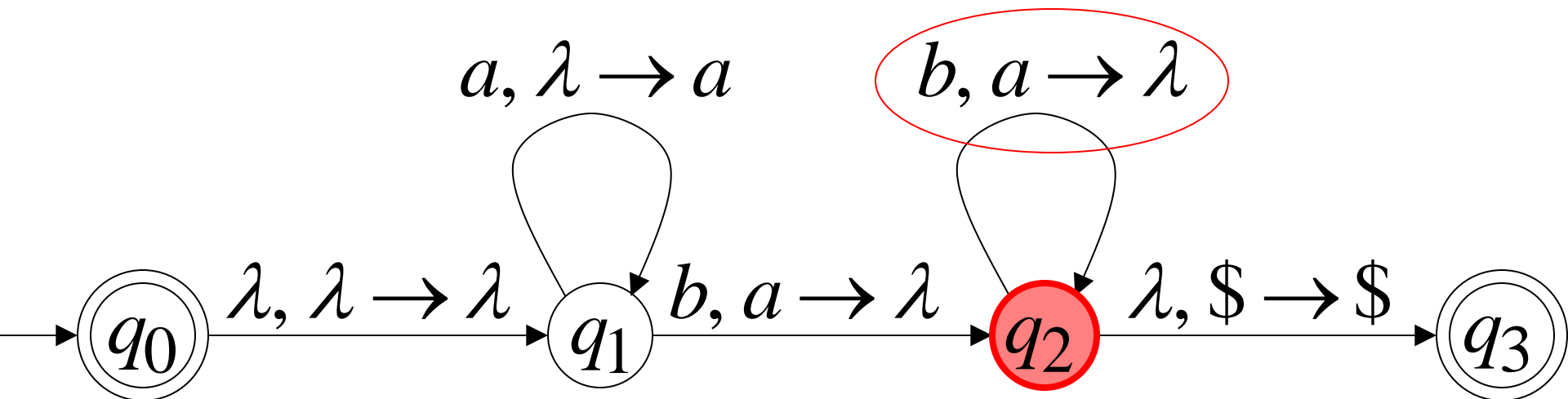


Time 7

Input

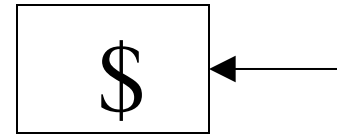
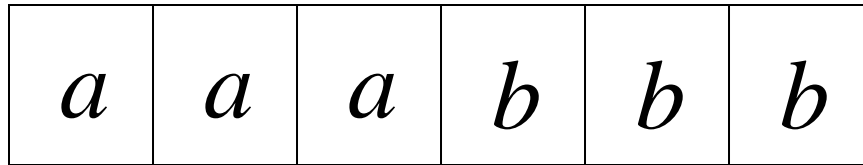


Stack

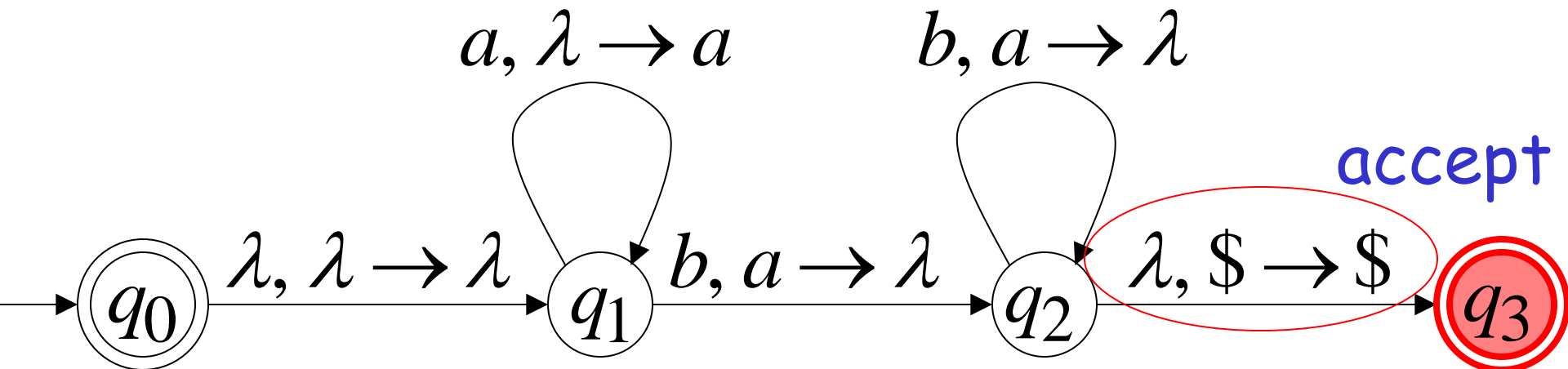


Time 8

Input



Stack



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

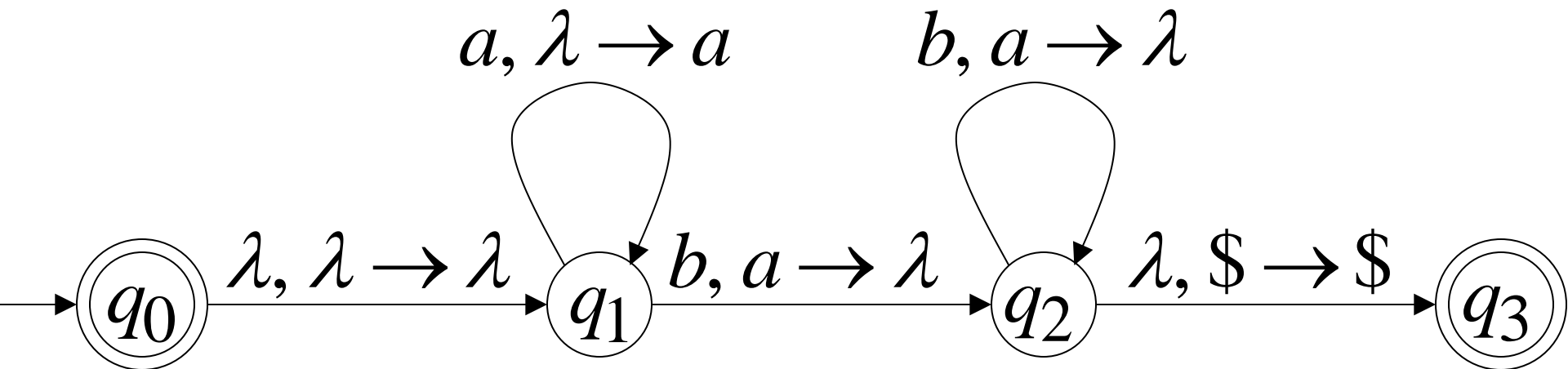
All the input is consumed

AND

The last state is an accepting state

At the end of the computation,
we do not care about the stack contents
(the stack can be empty at the last state)

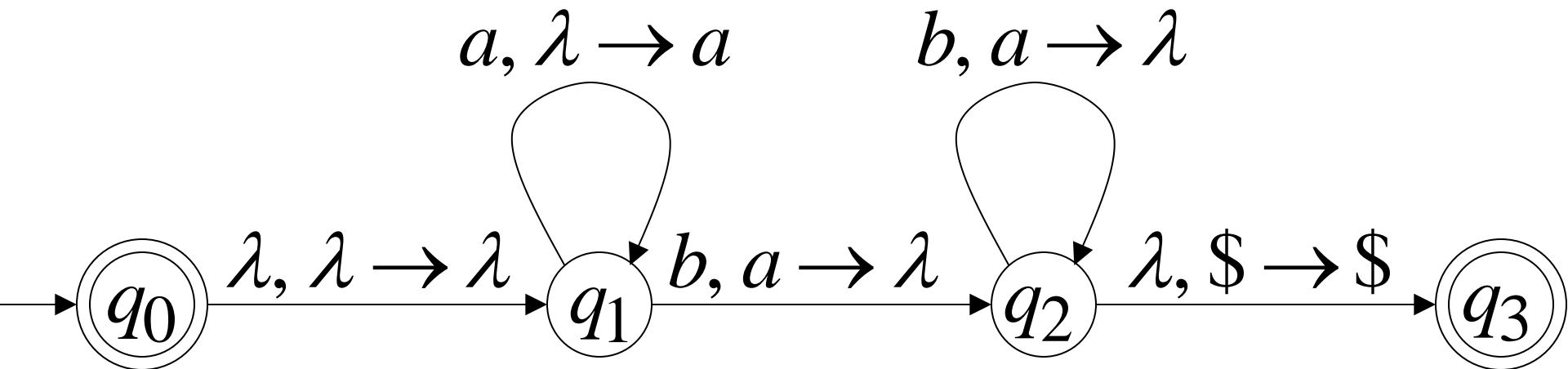
The input string *aaabbb*
is accepted by the PDA:



In general,

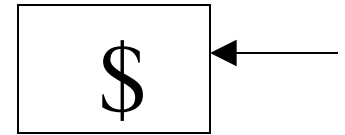
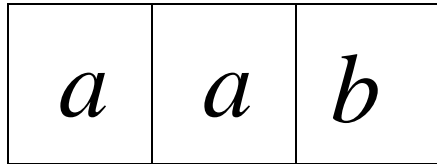
$$L = \{a^n b^n : n \geq 0\}$$

is the language accepted by the PDA:



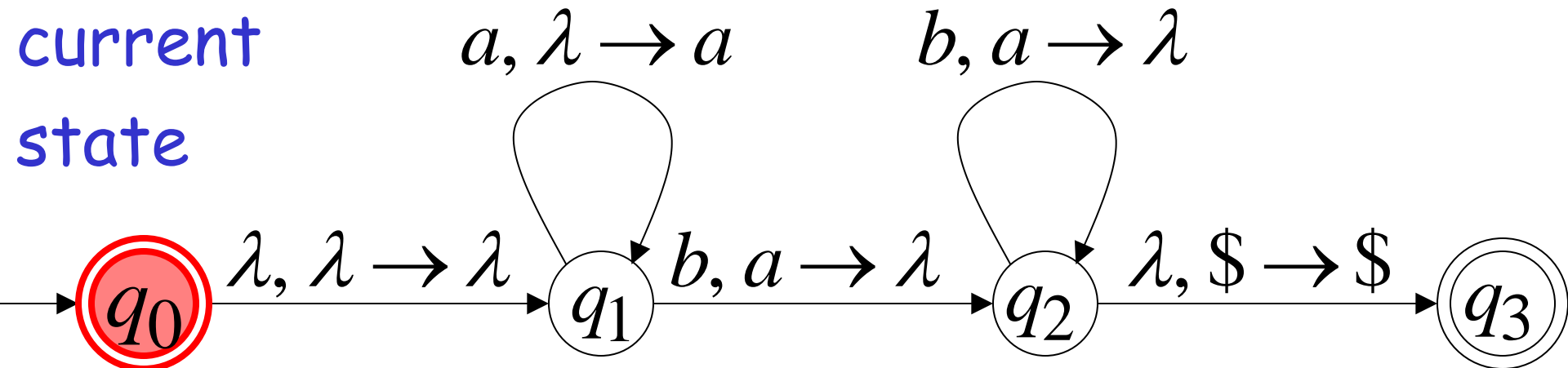
Rejection Example: Time 0

Input



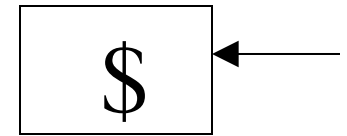
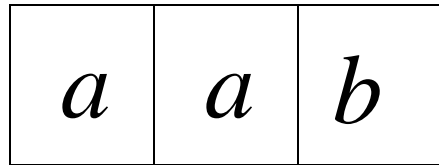
Stack

current
state



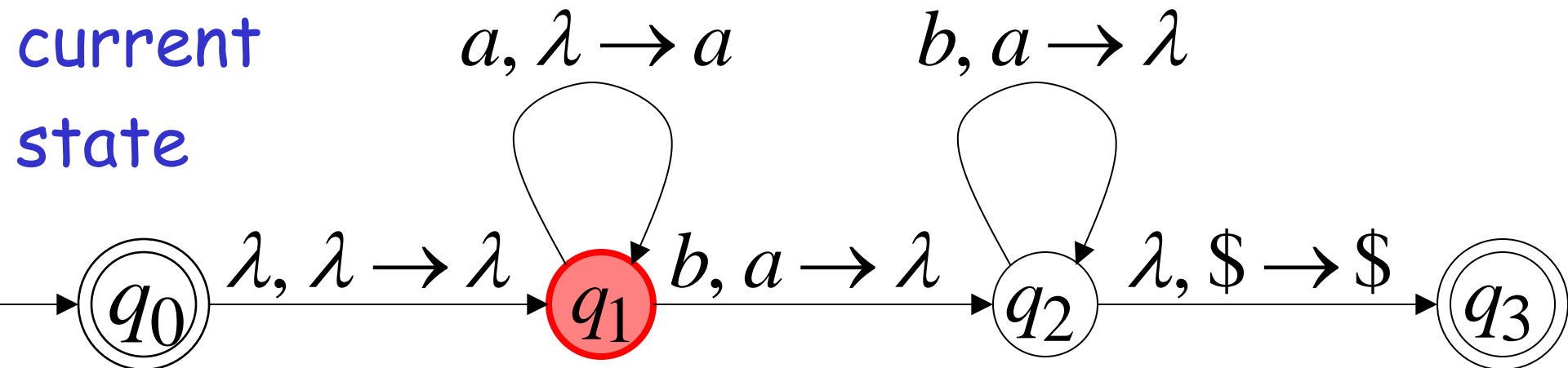
Rejection Example: Time 1

Input



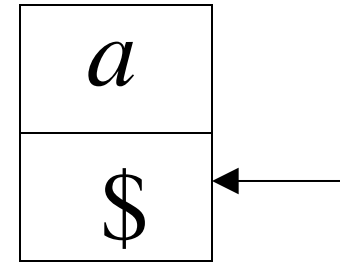
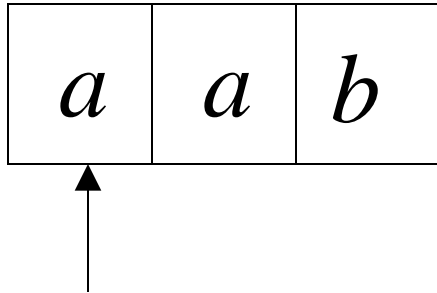
Stack

current
state

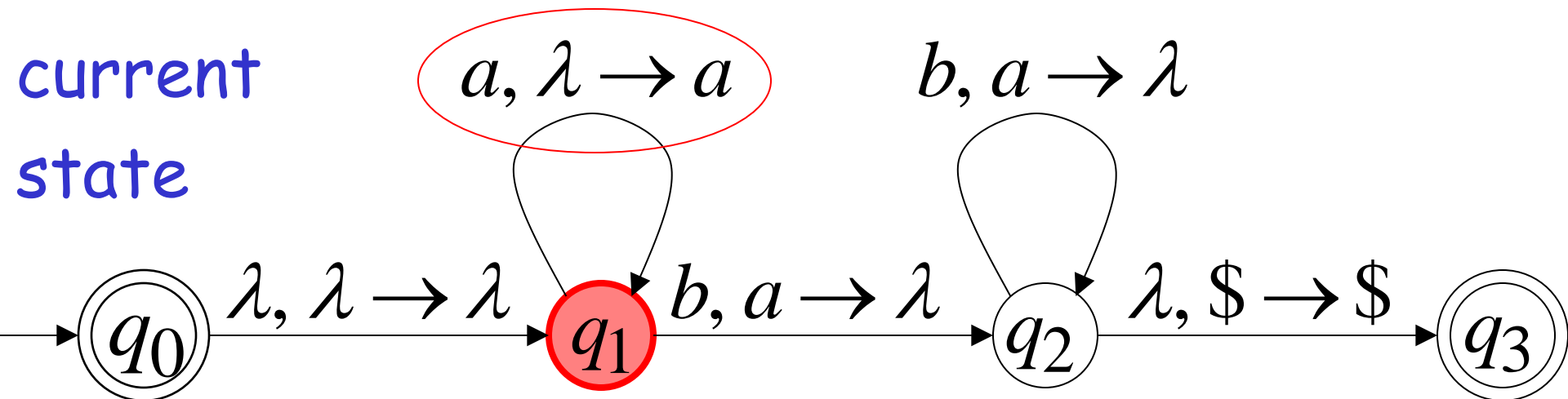


Rejection Example: Time 2

Input

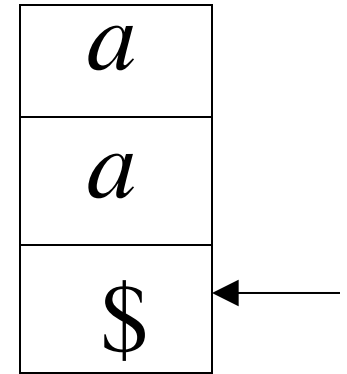
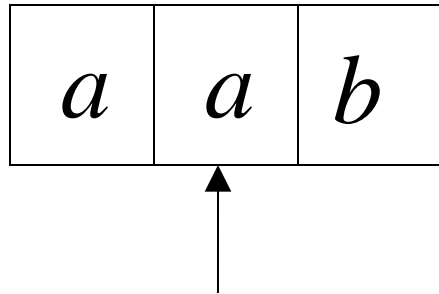


Stack

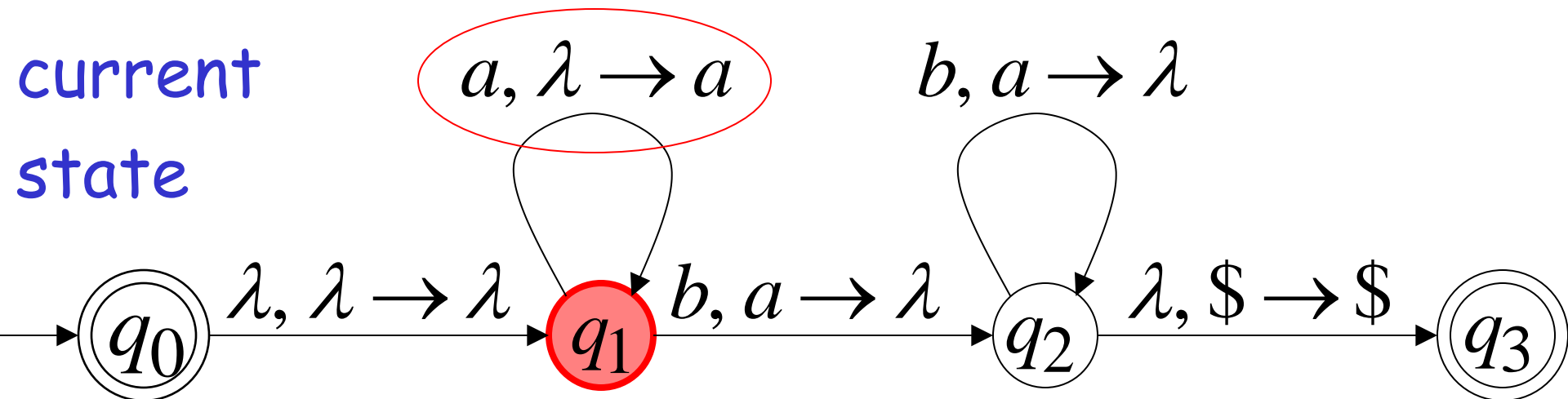


Rejection Example: Time 3

Input

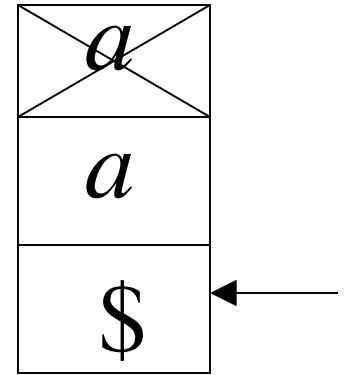
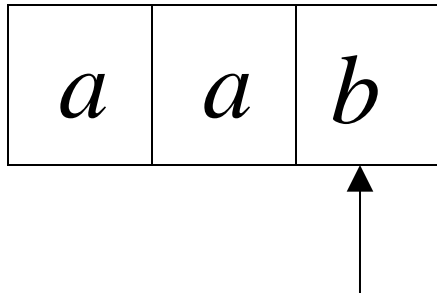


Stack



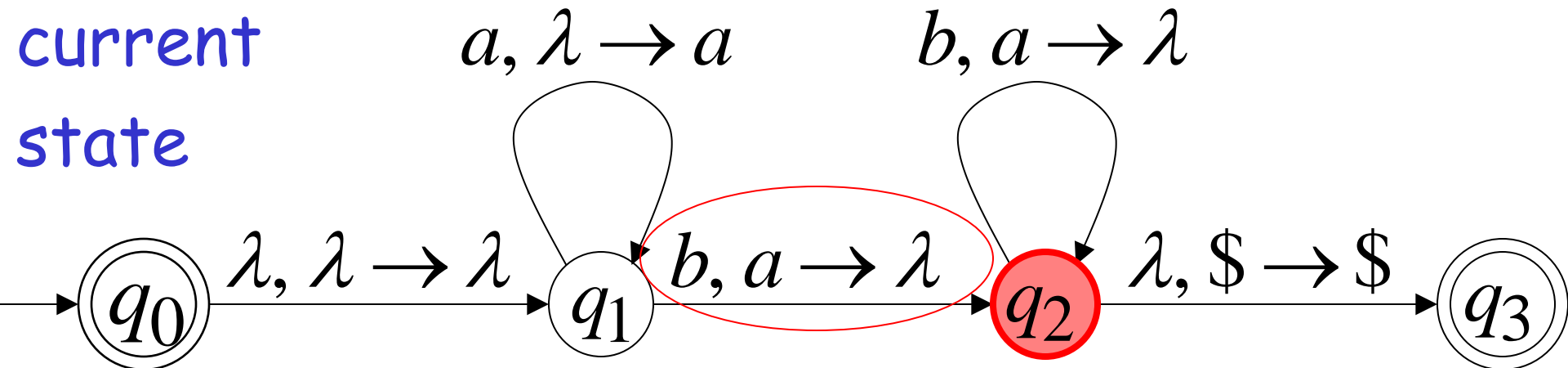
Rejection Example: Time 4

Input



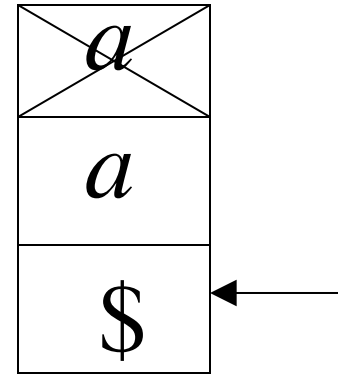
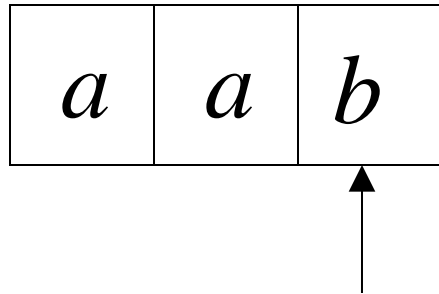
Stack

current
state



Rejection Example: Time 4

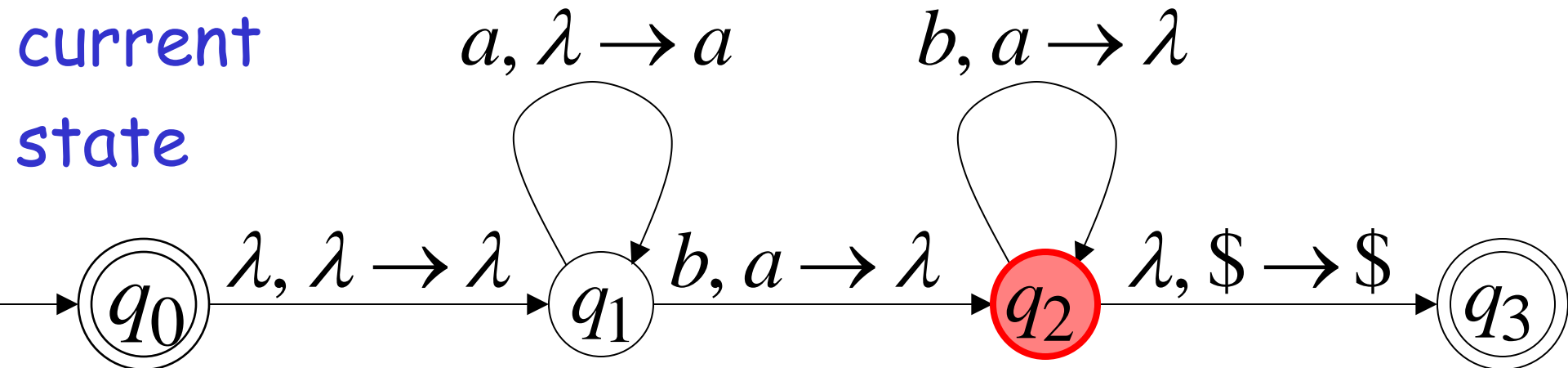
Input



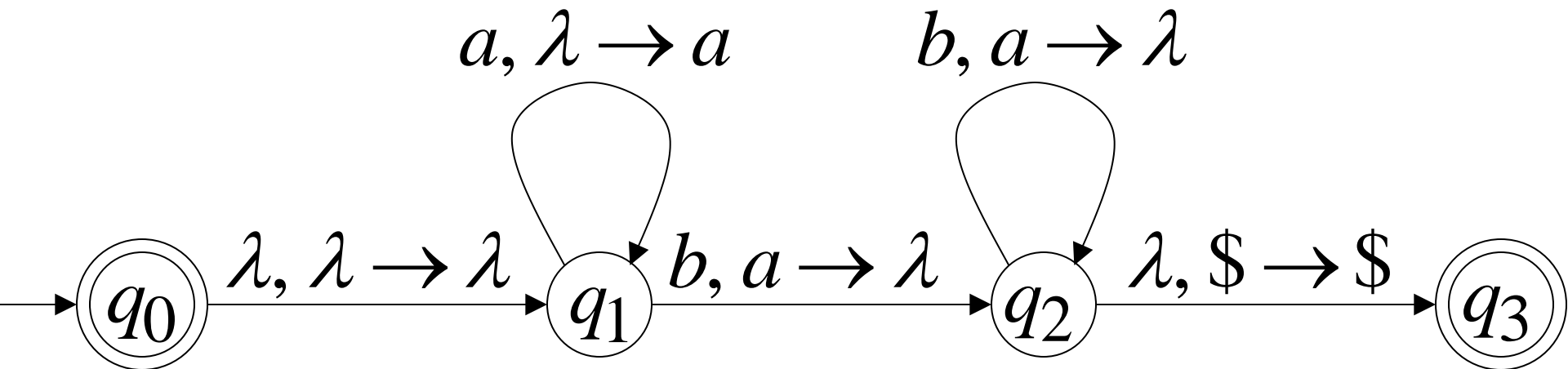
Stack

reject

current
state



The input string aab
is rejected by the PDA:



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

All the input is consumed

AND

The last state is an accept state

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

Another PDA example: Language?

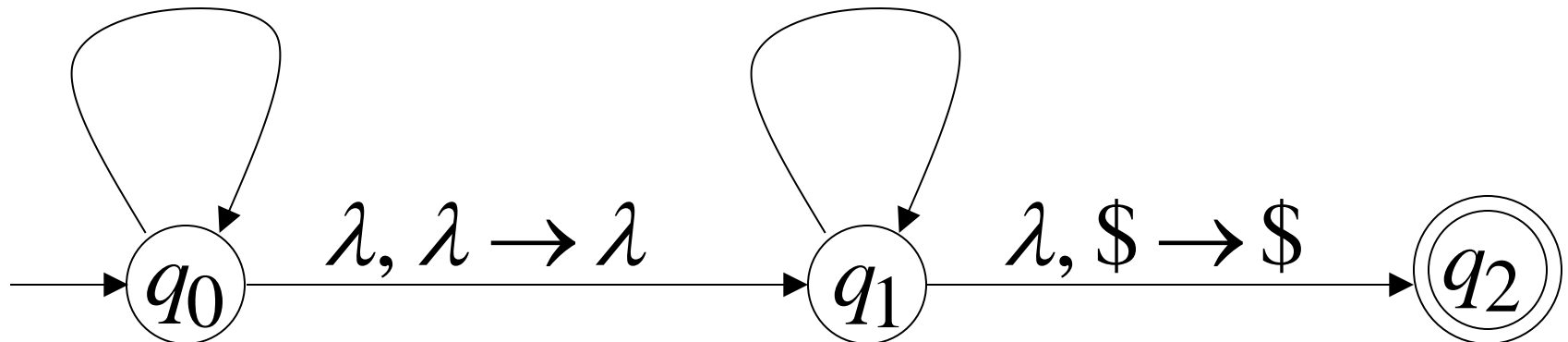
PDA M

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

$b, \lambda \rightarrow b$

$b, b \rightarrow \lambda$



Another PDA example

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

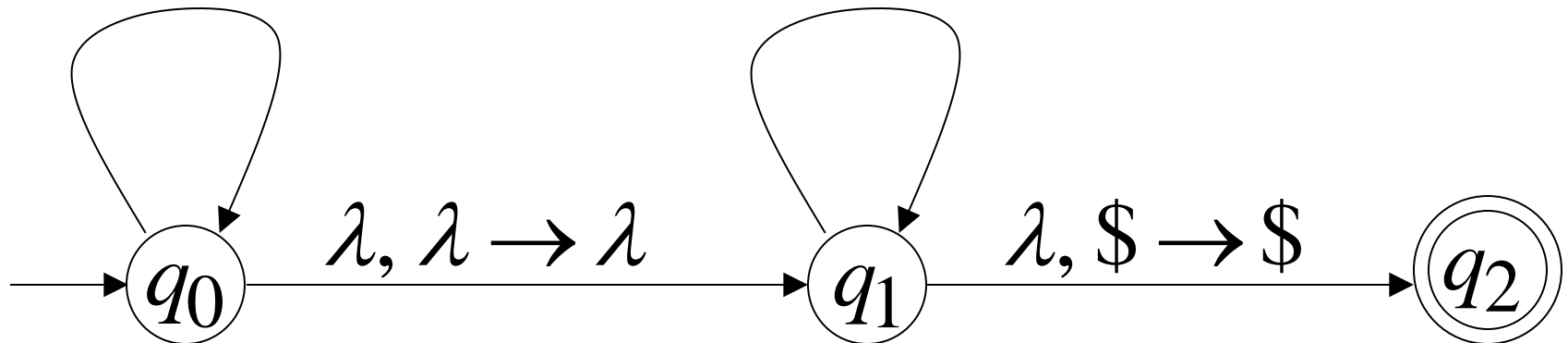
PDA M

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

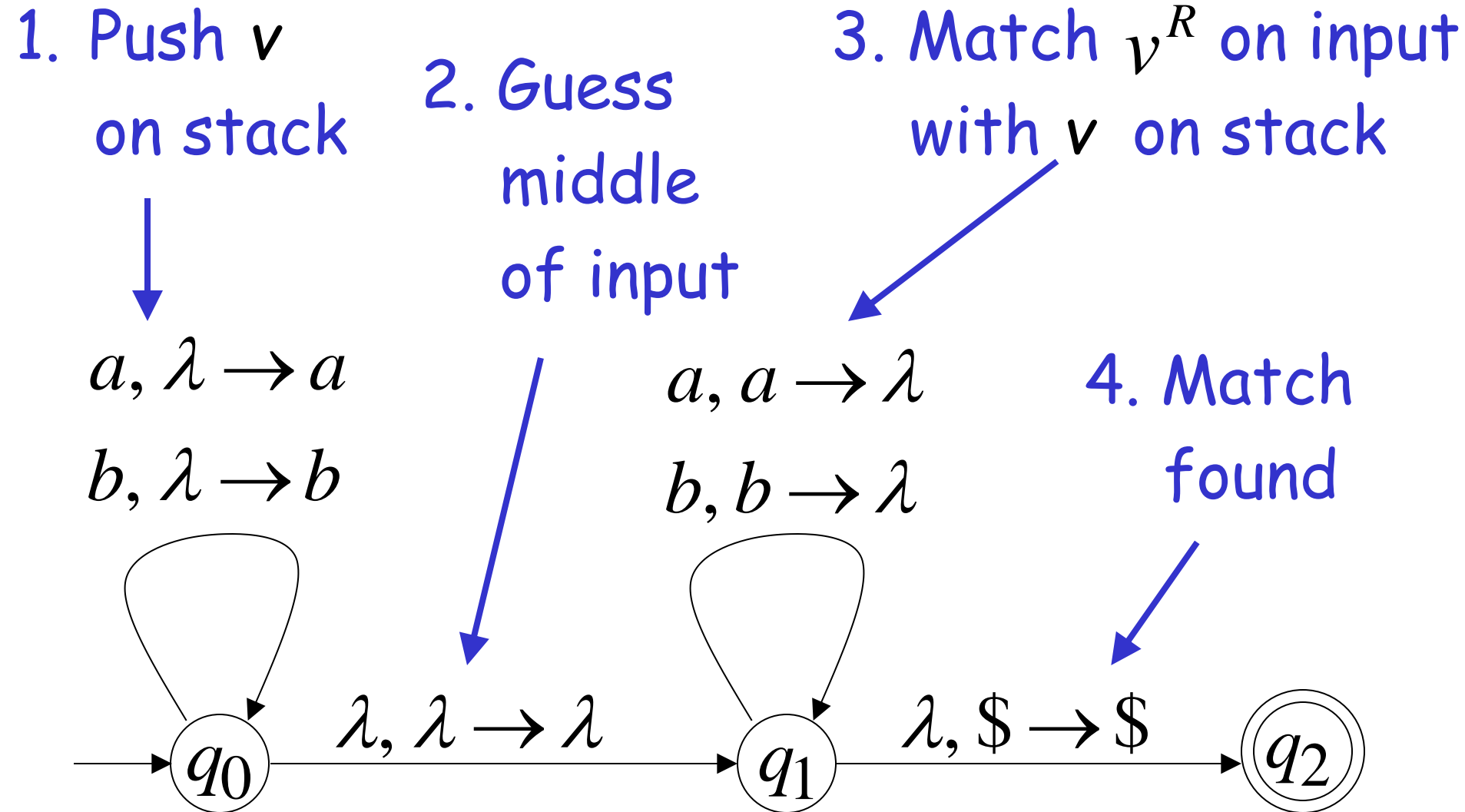
$b, \lambda \rightarrow b$

$b, b \rightarrow \lambda$



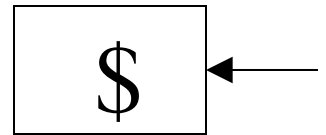
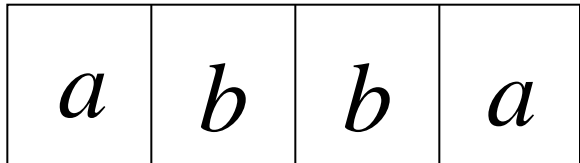
Basic Idea:

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



Execution Example: Time 0

Input



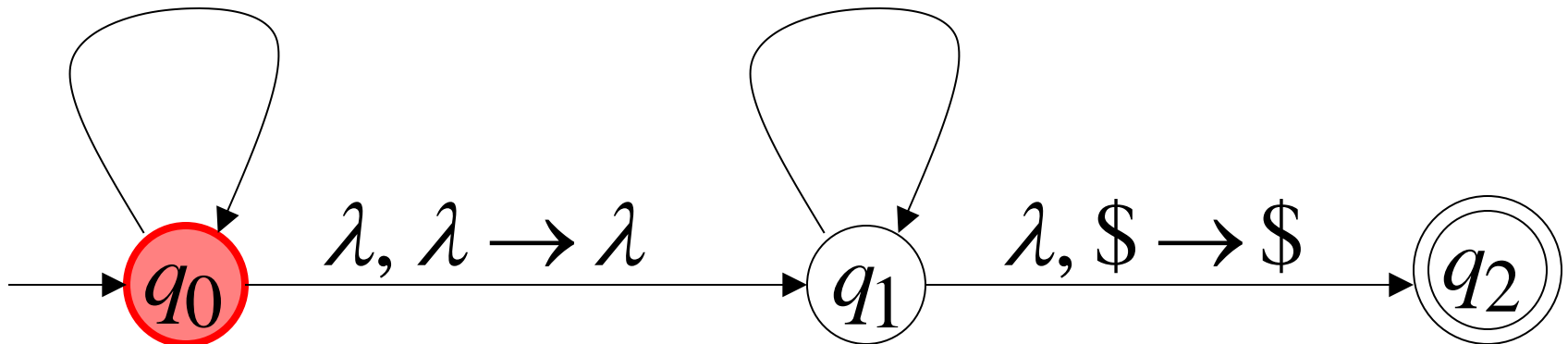
Stack

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

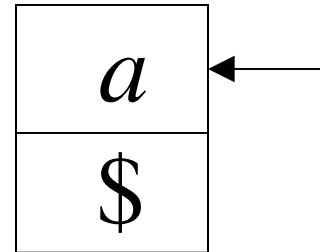
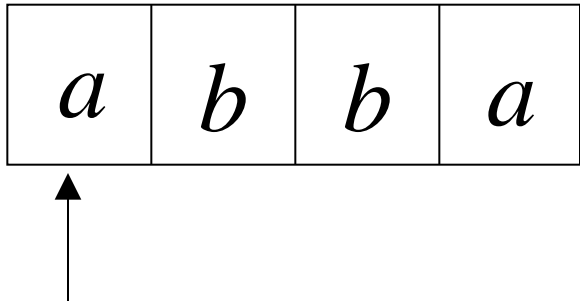
$b, \lambda \rightarrow b$

$b, b \rightarrow \lambda$



Time 1

Input



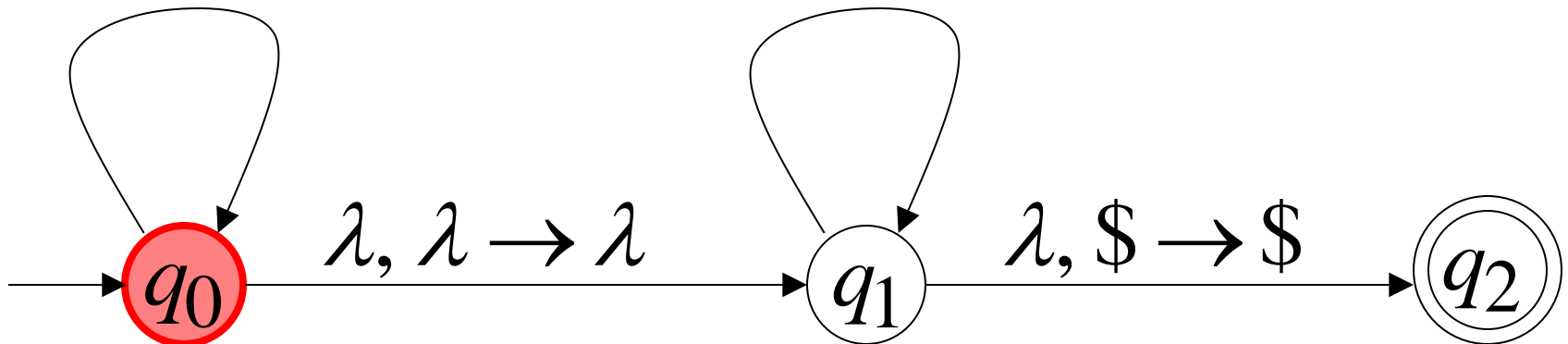
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

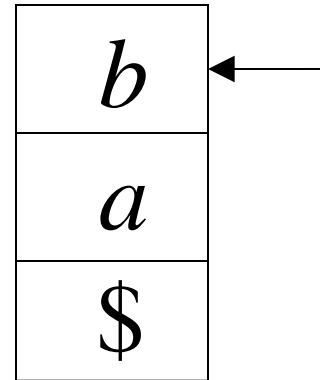
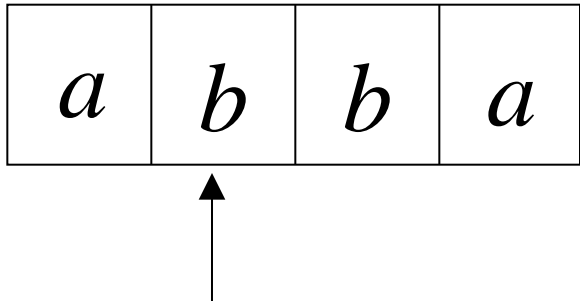
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 2

Input



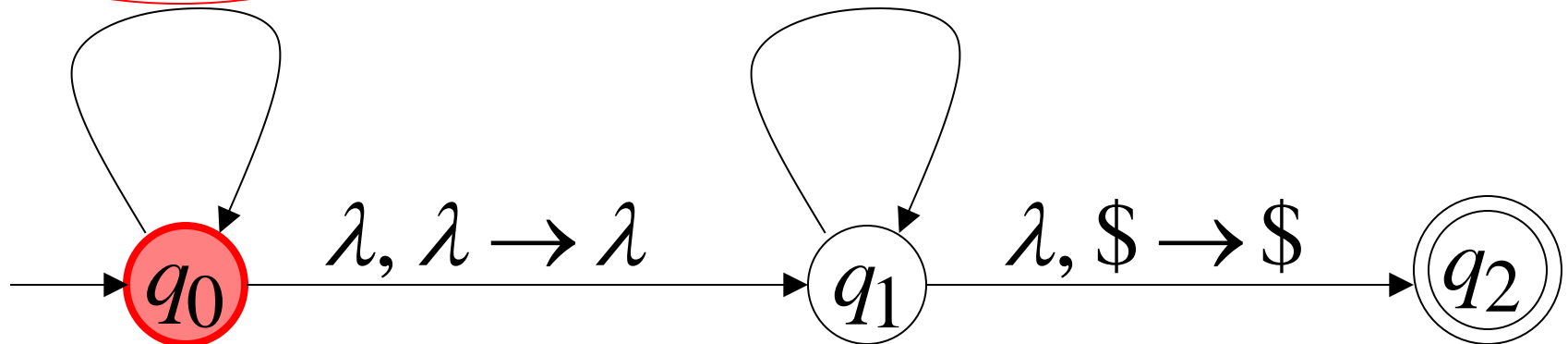
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

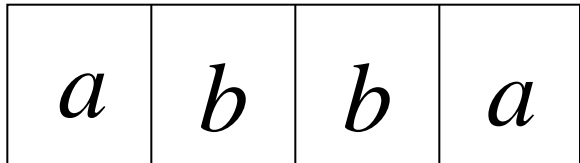
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$

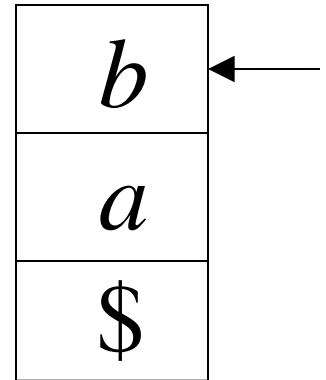


Time 3

Input



Guess the middle
of string



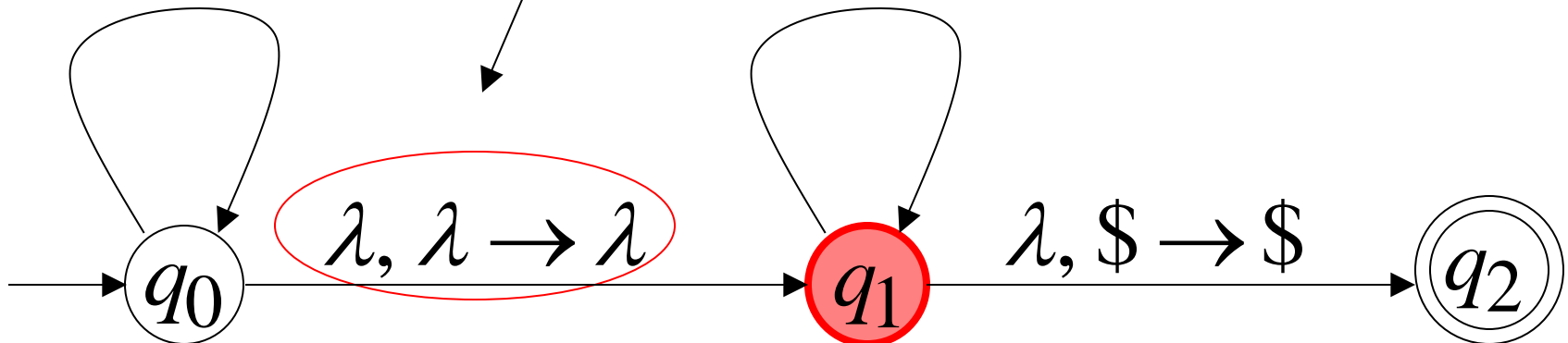
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

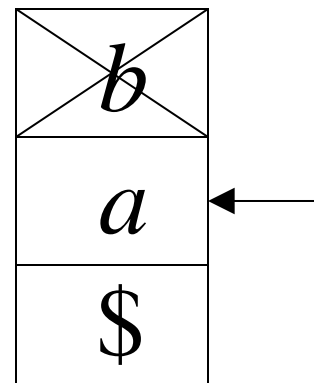
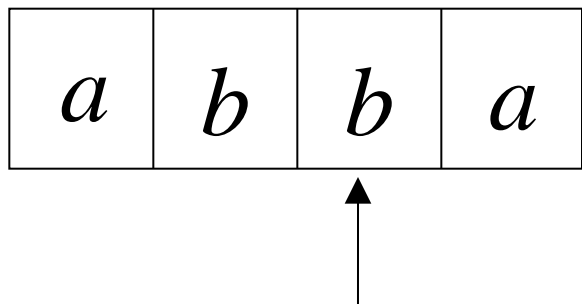
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 4

Input



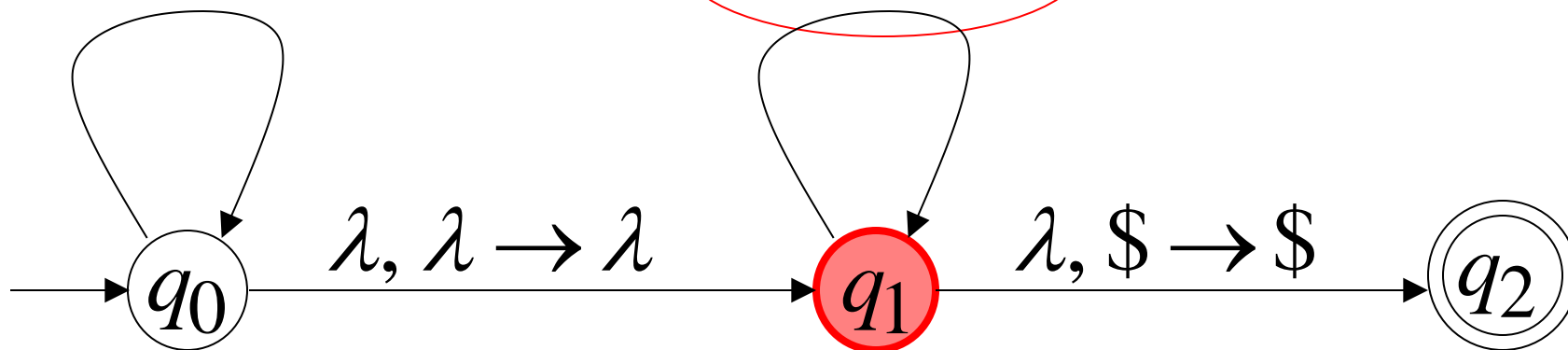
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

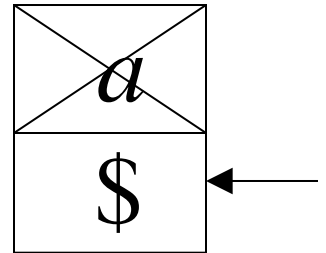
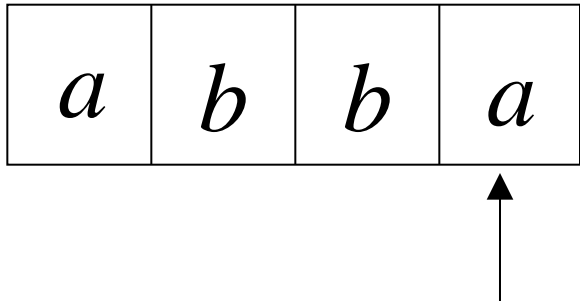
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 5

Input



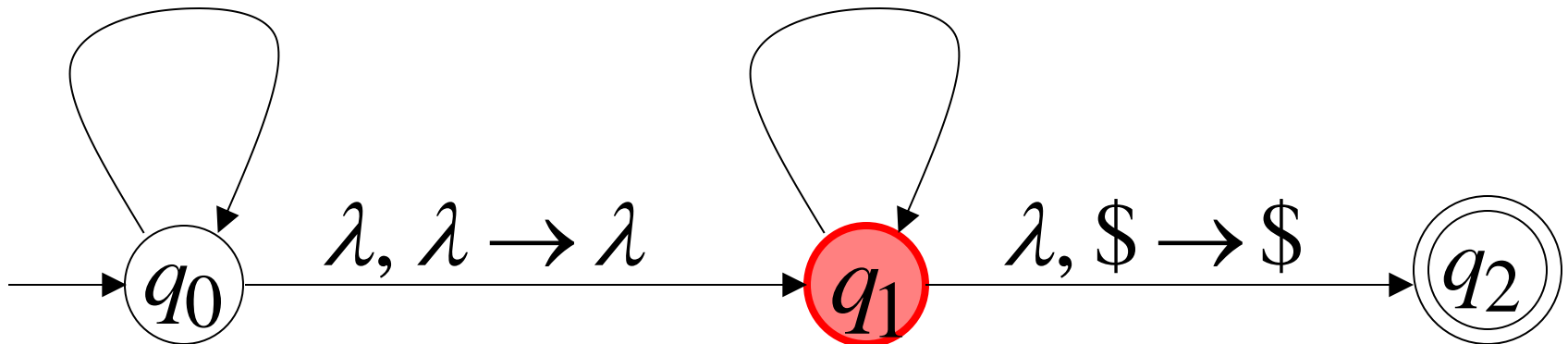
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

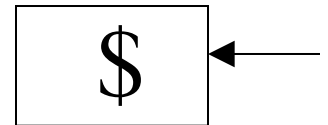
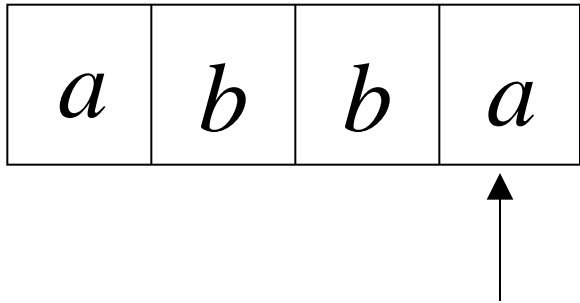
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 6

Input



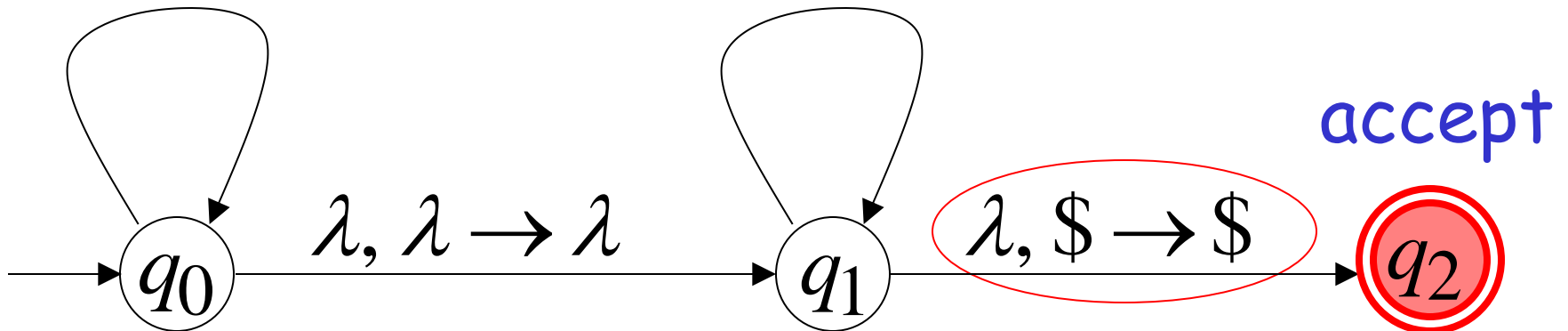
Stack

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

$b, \lambda \rightarrow b$

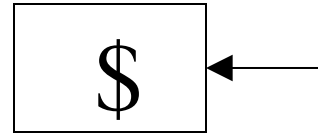
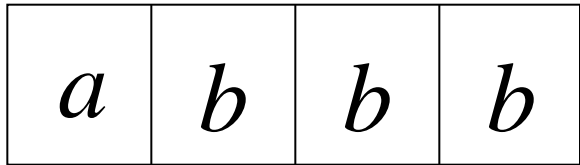
$b, b \rightarrow \lambda$



accept

Rejection Example: Time 0

Input



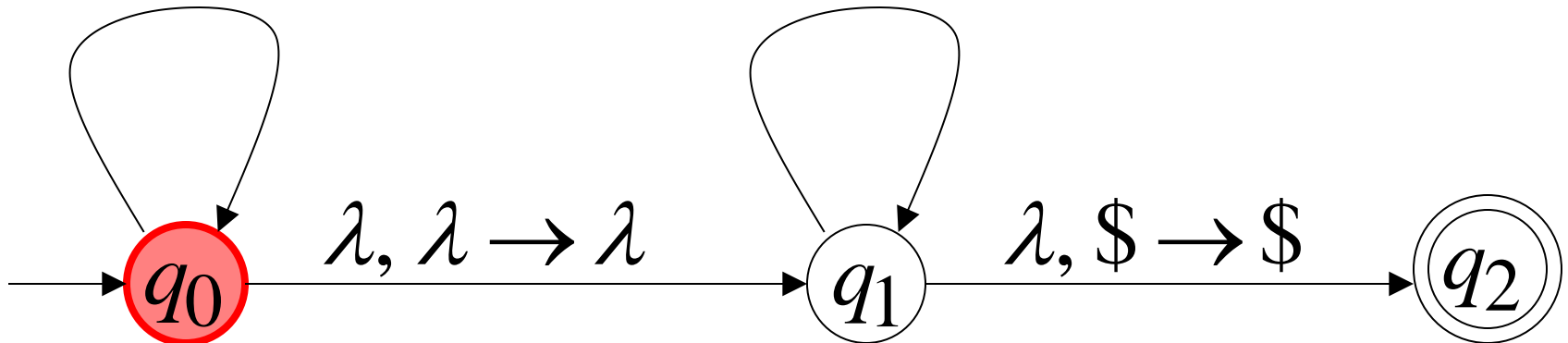
Stack

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

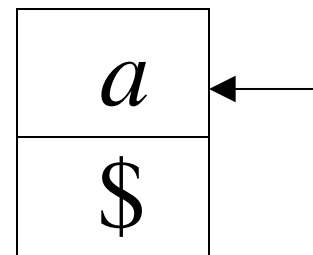
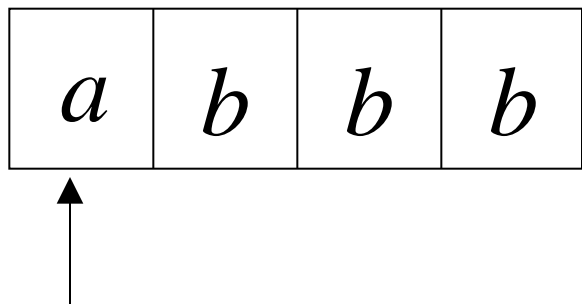
$b, \lambda \rightarrow b$

$b, b \rightarrow \lambda$



Time 1

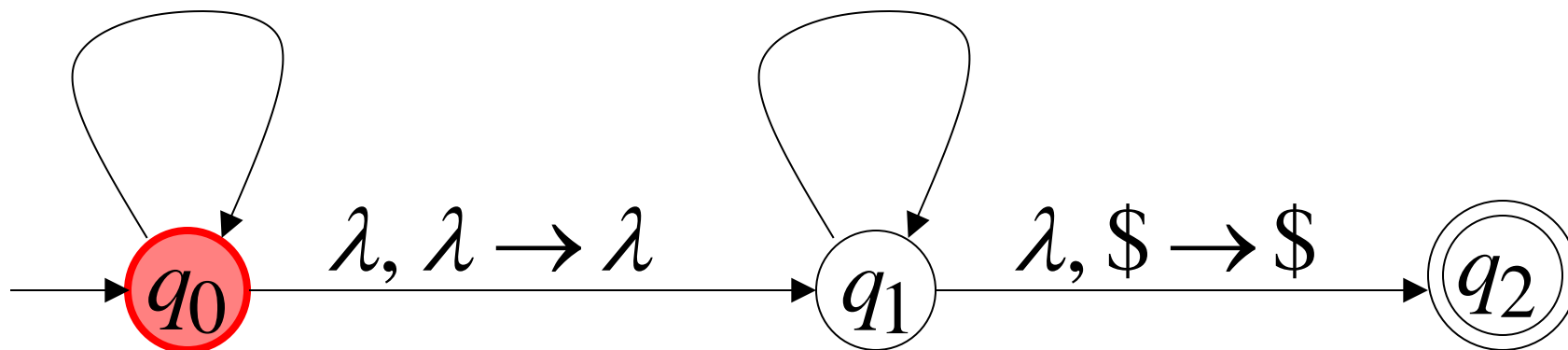
Input



Stack

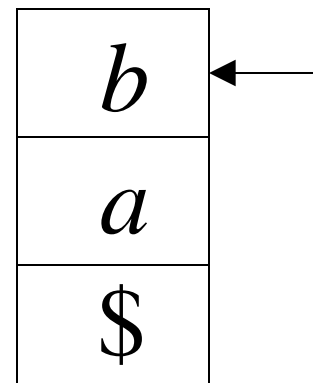
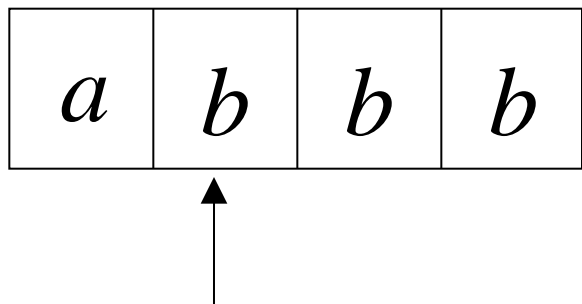
$a, \lambda \rightarrow a$
 $b, \lambda \rightarrow b$

$a, a \rightarrow \lambda$
 $b, b \rightarrow \lambda$



Time 2

Input



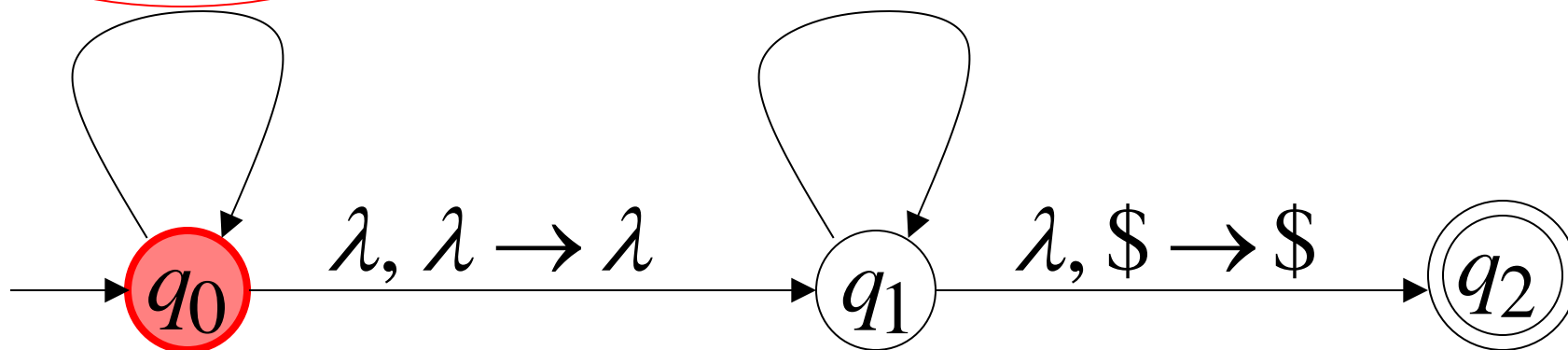
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

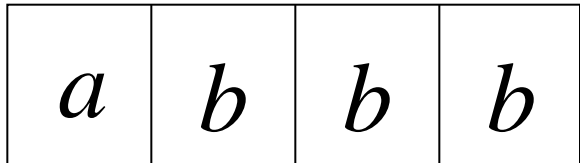
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$

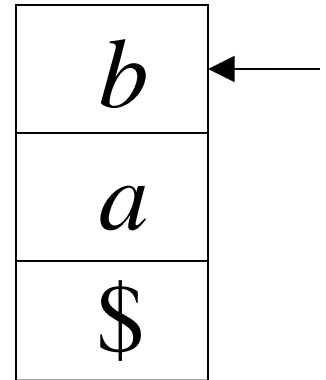


Time 3

Input



Guess the middle
of string



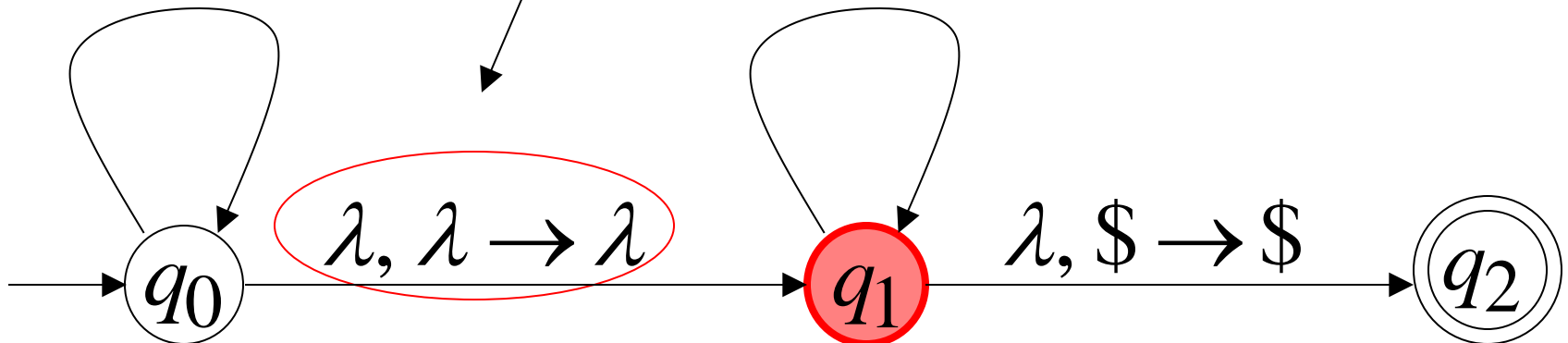
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

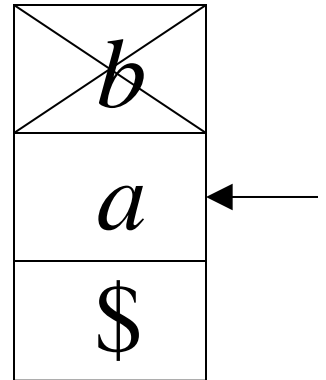
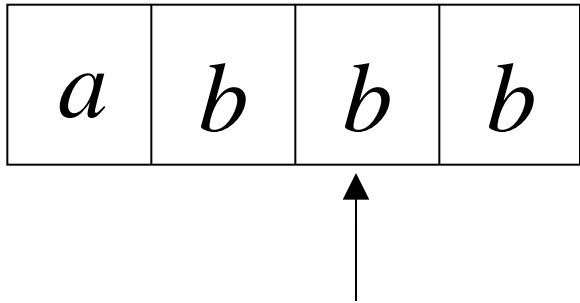
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 4

Input



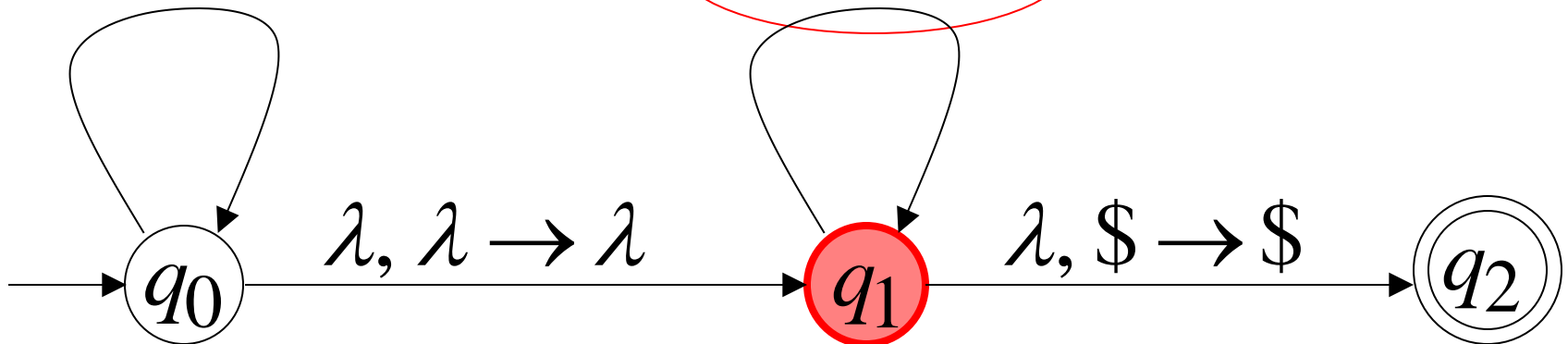
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

$a, a \rightarrow \lambda$

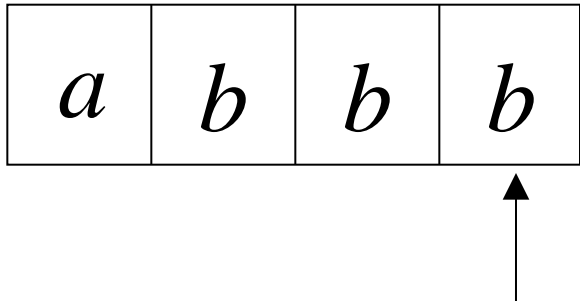
$b, b \rightarrow \lambda$



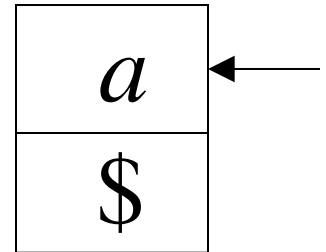
Time 5

Input

There is no possible transition.



Input is not consumed



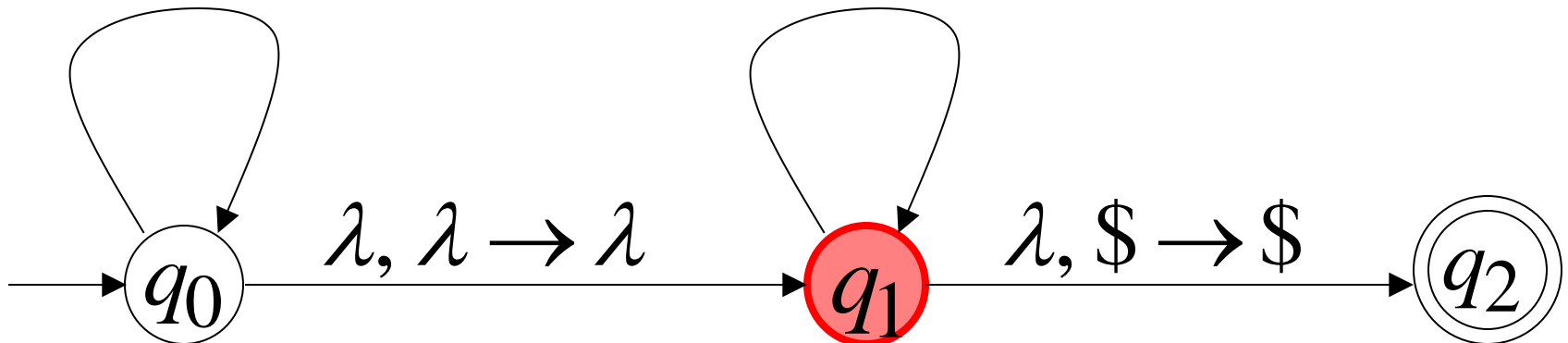
Stack

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

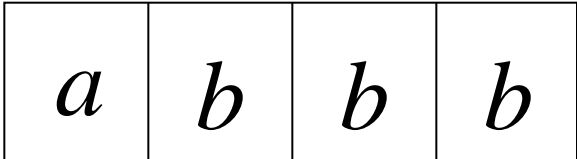
$b, \lambda \rightarrow b$

$b, b \rightarrow \lambda$

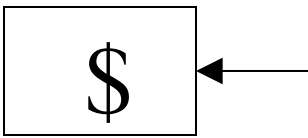


Another computation on same string:

Input



Time 0



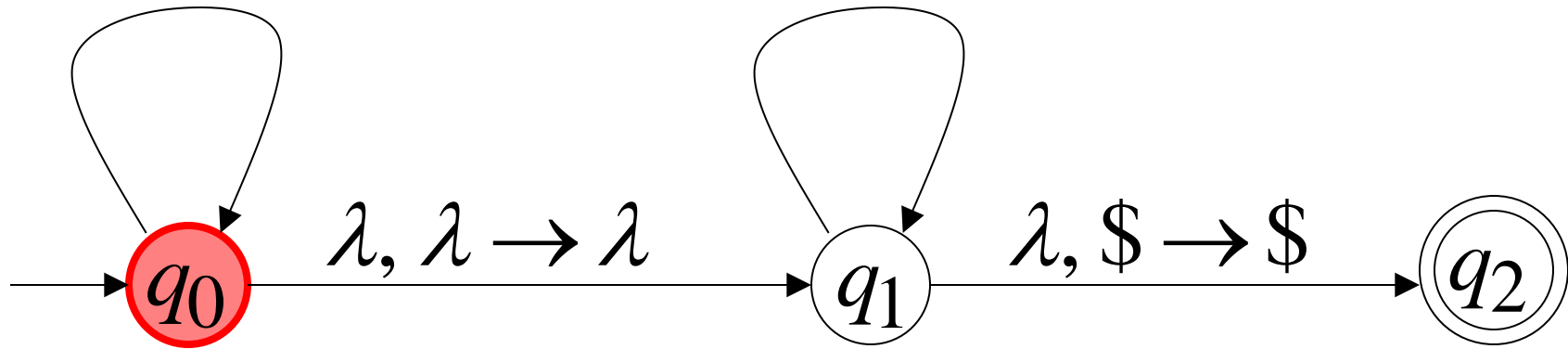
Stack

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

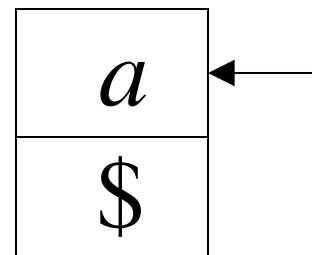
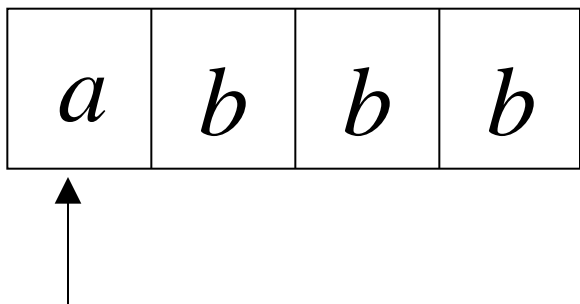
$b, \lambda \rightarrow b$

$b, b \rightarrow \lambda$



Time 1

Input



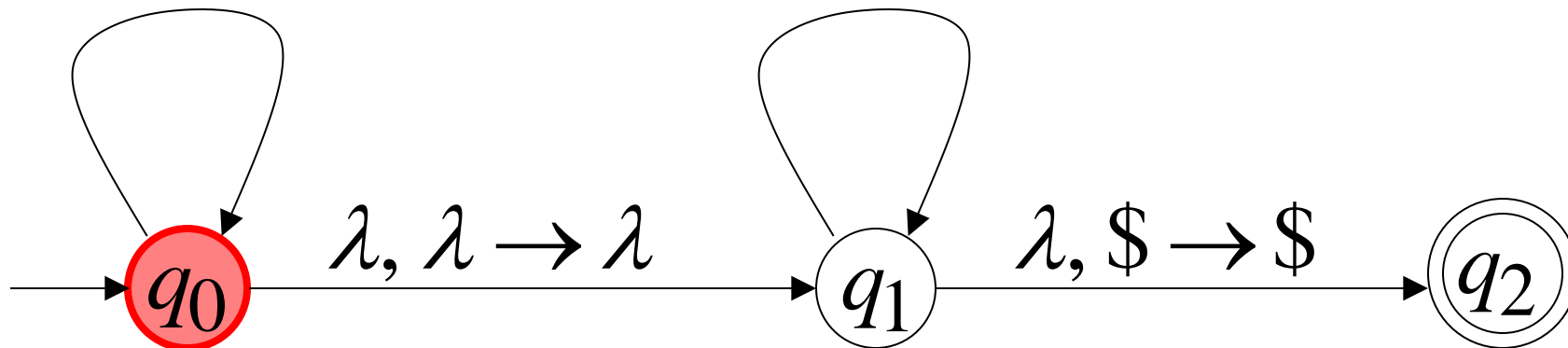
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

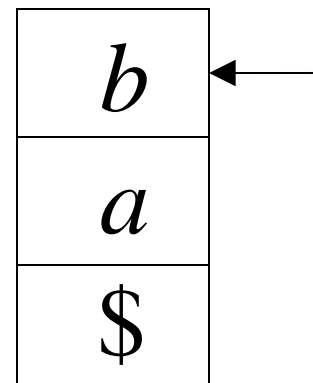
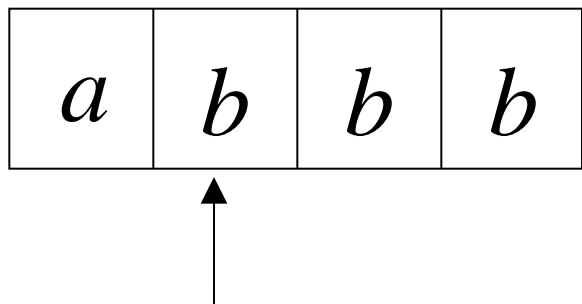
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 2

Input



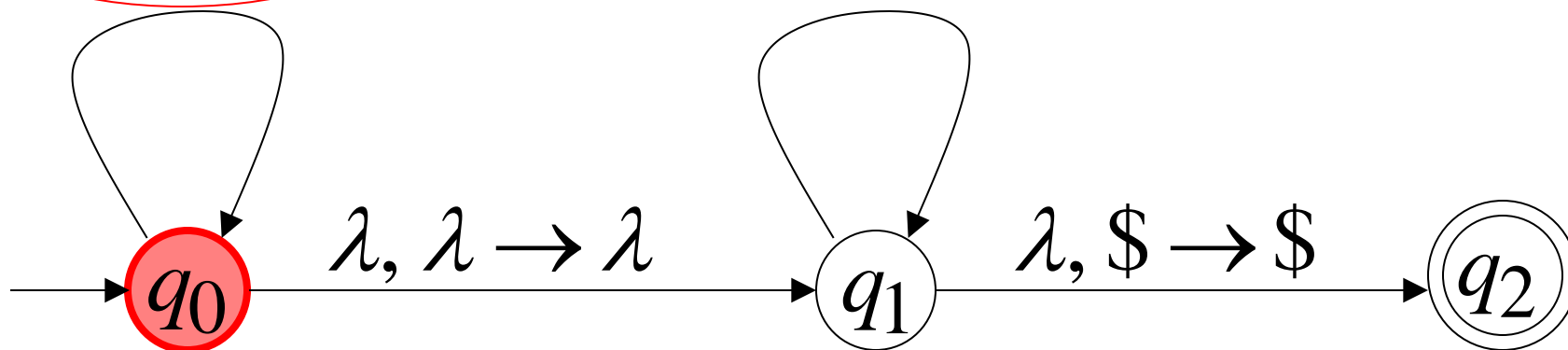
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

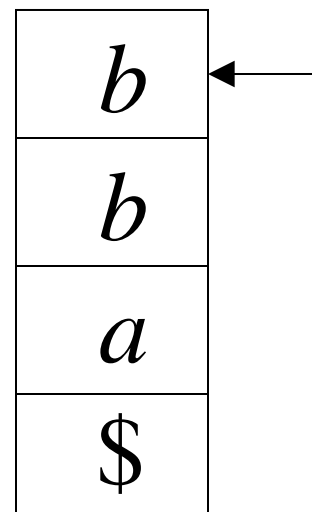
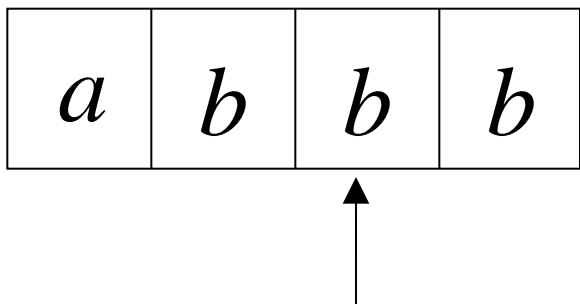
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 3

Input



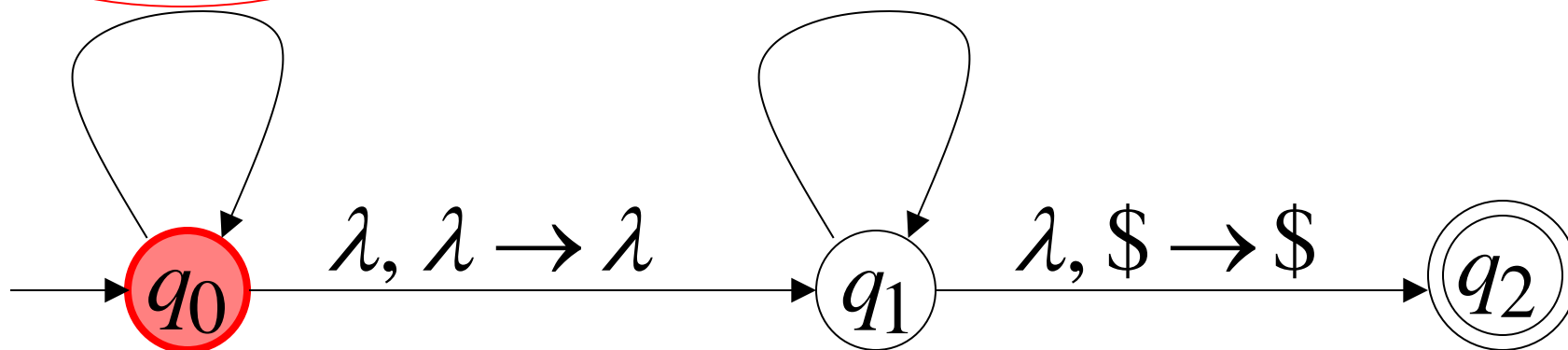
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

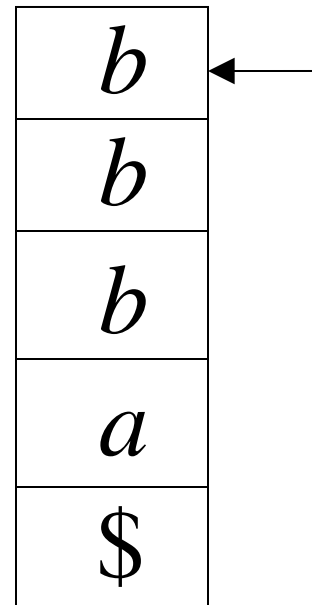
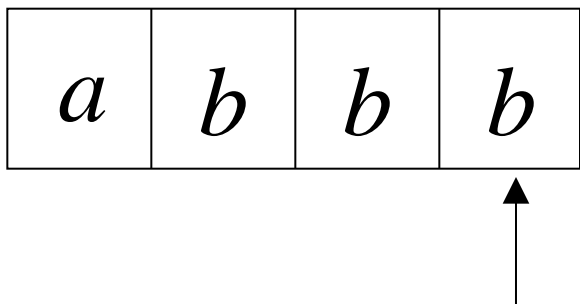
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



Time 4

Input



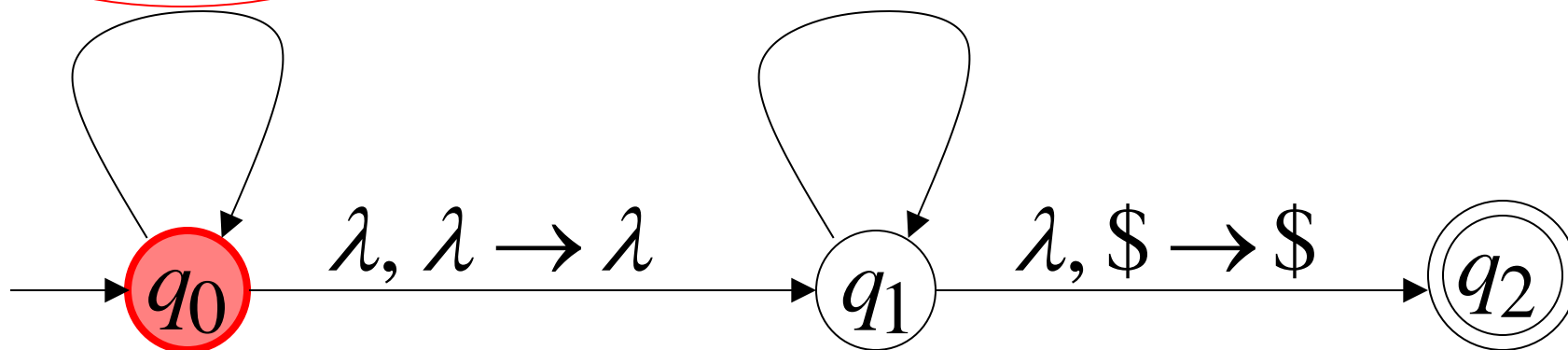
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

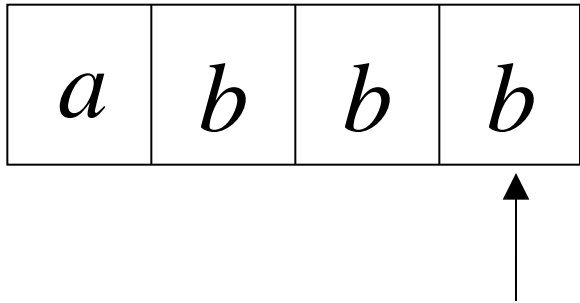
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$

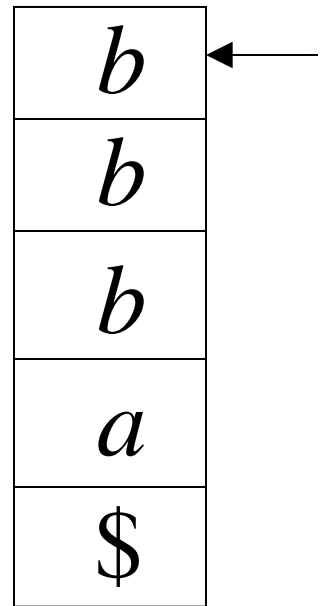


Time 5

Input



No final state
is reached



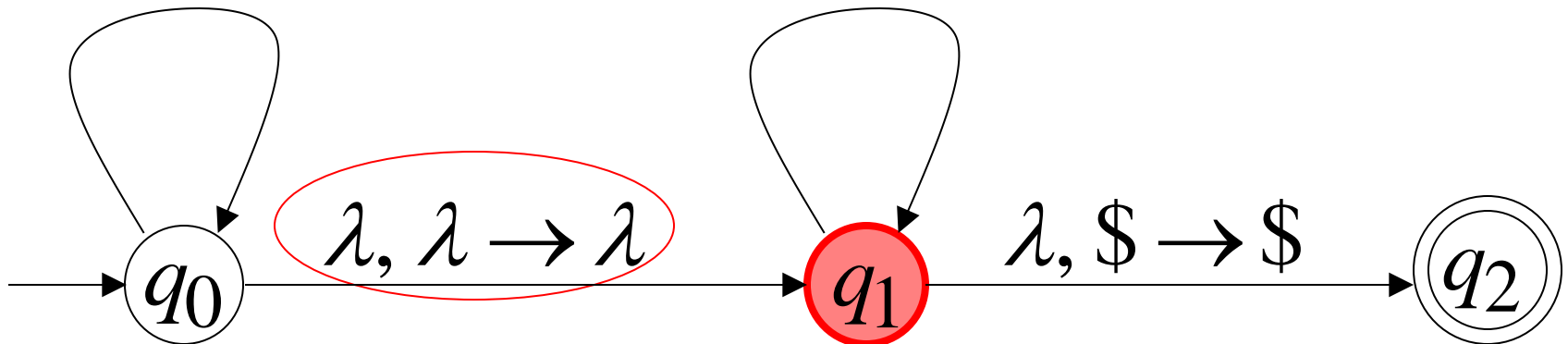
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

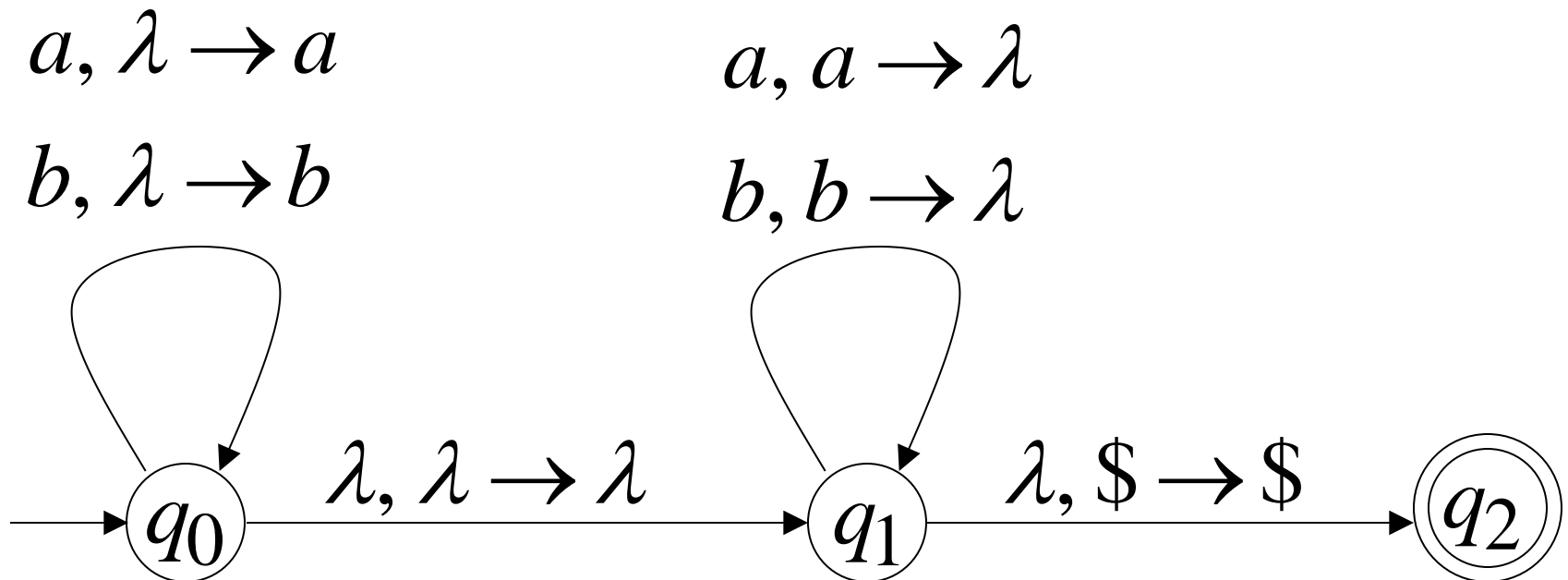
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$



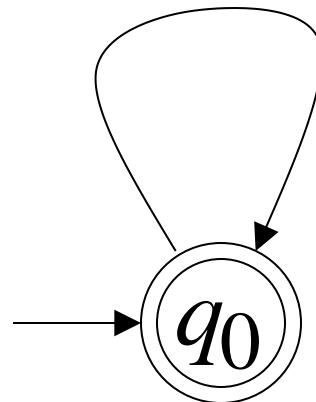
There is no computation
that accepts string $abbb$

$$abbb \notin L(M)$$



Another PDA example: Language?

PDA M



$a, \lambda \rightarrow a$

$b, a \rightarrow \lambda$

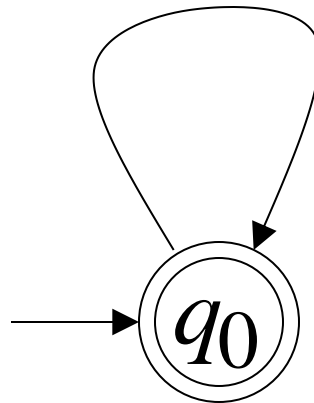
Another PDA example

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

in every prefix v , $n_a(v) \geq n_b(v)\}$

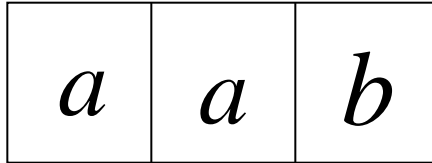
$$a, \lambda \rightarrow a$$
$$b, a \rightarrow \lambda$$

PDA M



Execution Example: Time 0

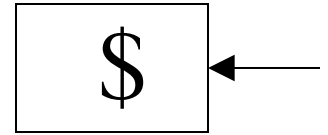
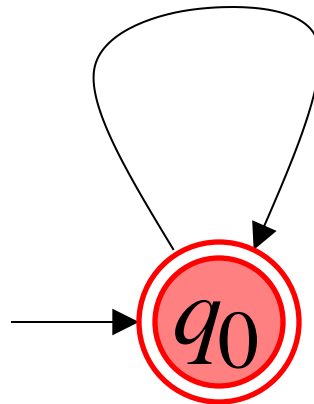
Input



$a, \lambda \rightarrow a$

$b, a \rightarrow \lambda$

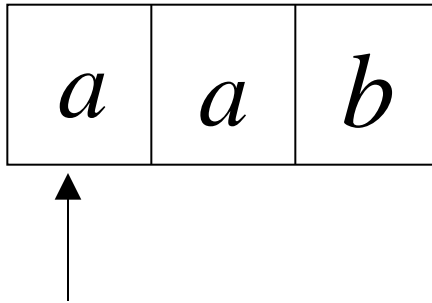
$b, \$ \rightarrow \lambda$



Stack

Time 1

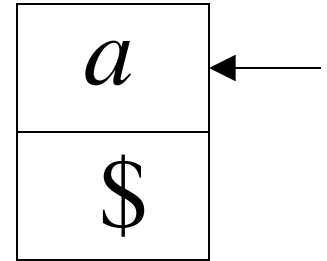
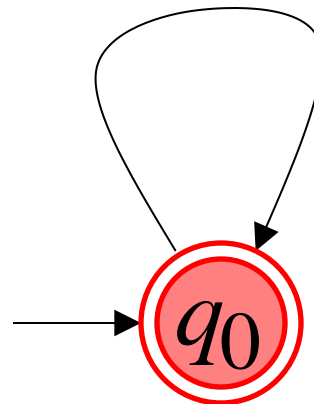
Input



$a, \lambda \rightarrow a$

$b, a \rightarrow \lambda$

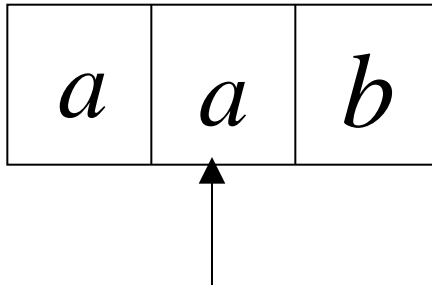
$b, \$ \rightarrow \lambda$



Stack

Time 2

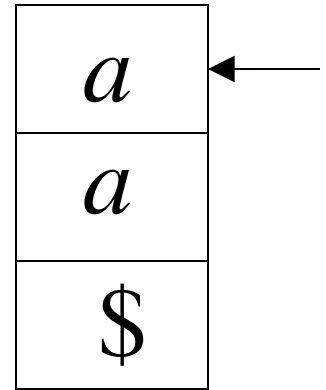
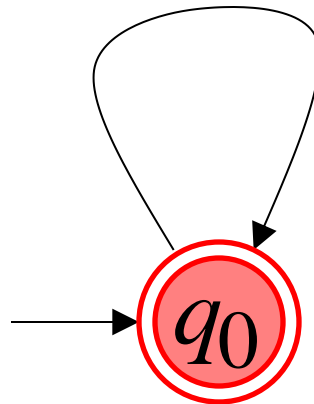
Input



$$a, \lambda \rightarrow a$$

$$b, a \rightarrow \lambda$$

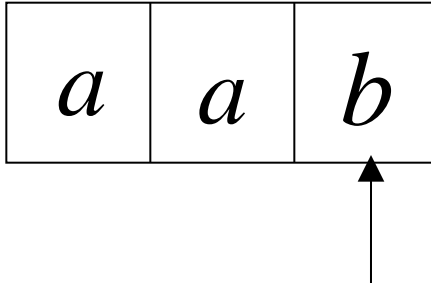
$$b, \$ \rightarrow \lambda$$



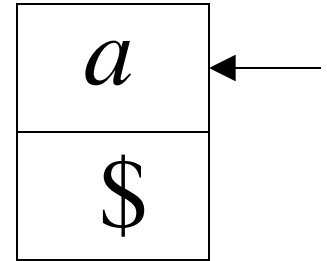
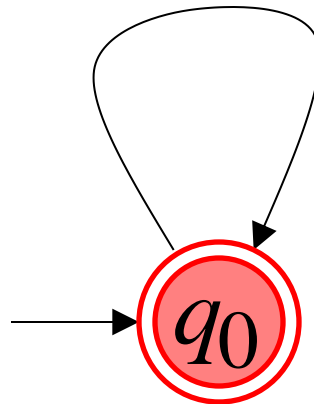
Stack

Time 3

Input



$a, \lambda \rightarrow a$
 $b, a \rightarrow \lambda$
 $b, \$ \rightarrow \lambda$

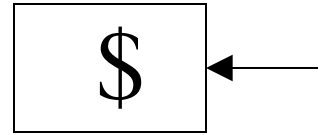
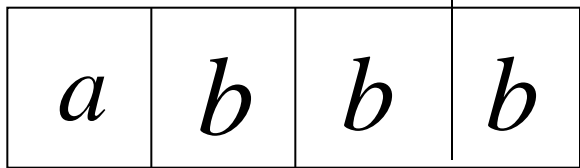


Stack

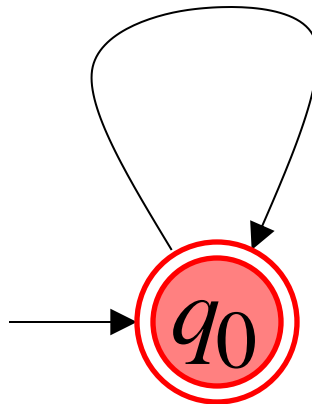
accept

Rejection example: Time 0

Input

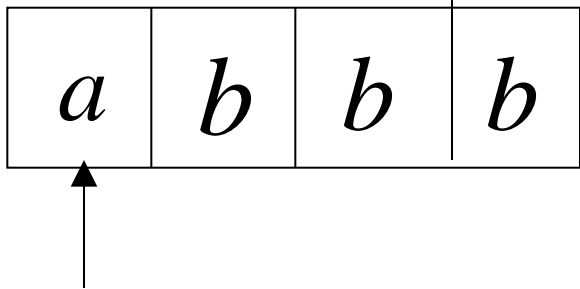


Stack



Time 1

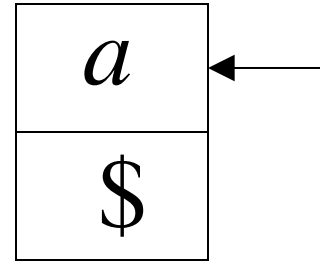
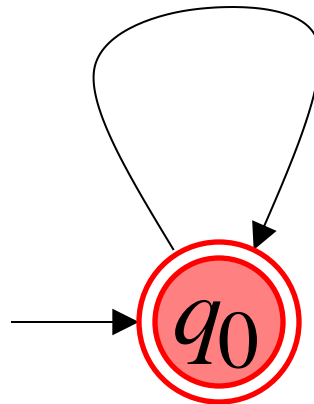
Input



$$a, \lambda \rightarrow a$$

$$b, a \rightarrow \lambda$$

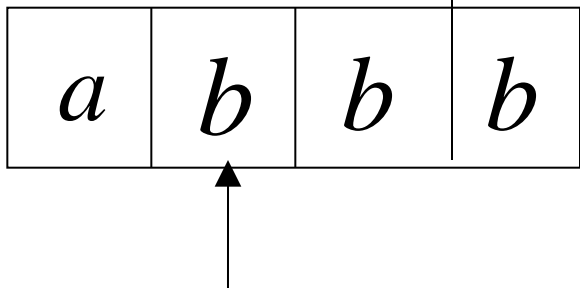
$$b, \$ \rightarrow \lambda$$



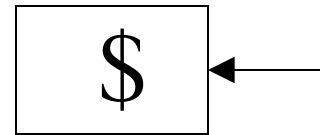
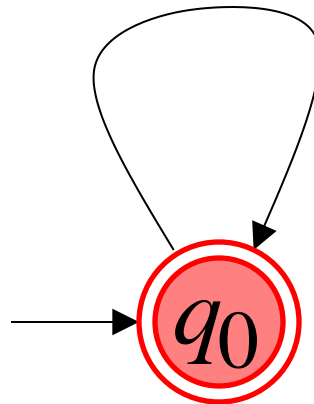
Stack

Time 2

Input



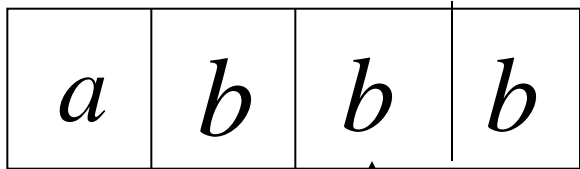
$a, \lambda \rightarrow a$
 $b, a \rightarrow \lambda$
 $b, \$ \rightarrow \lambda$



Stack

Time 3

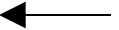
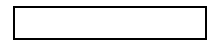
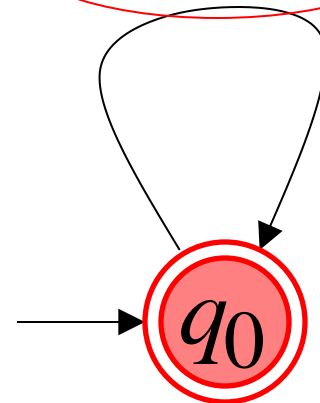
Input



$a, \lambda \rightarrow a$

$b, a \rightarrow \lambda$

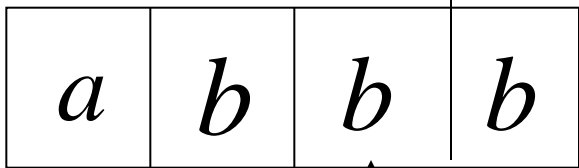
$b, \$ \rightarrow \lambda$



Stack

Time 4

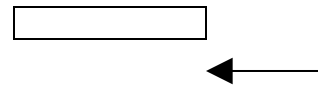
Input



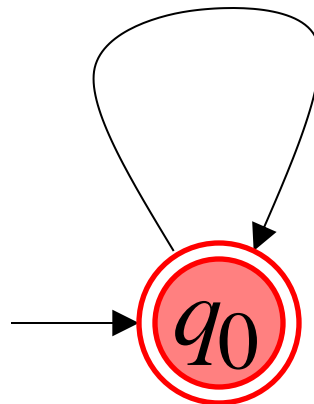
$a, \lambda \rightarrow a$

$b, a \rightarrow \lambda$

$b, \$ \rightarrow \lambda$

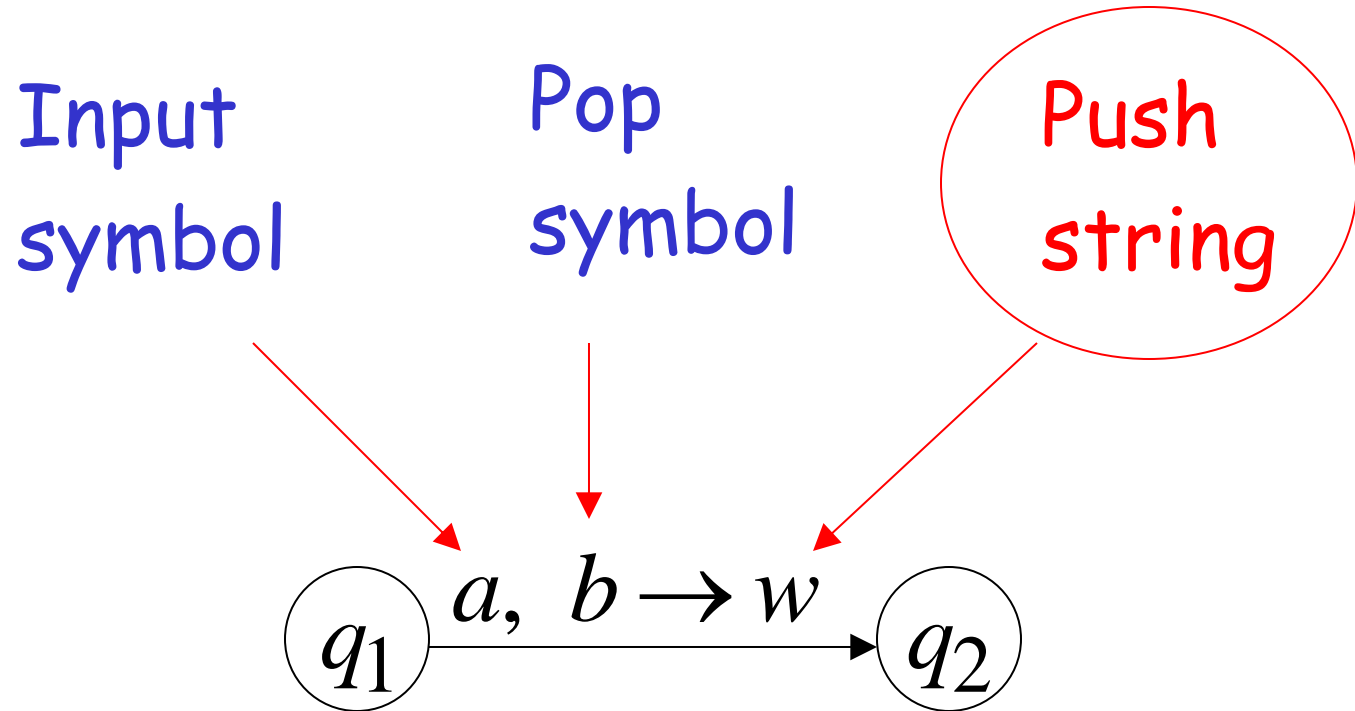


Stack

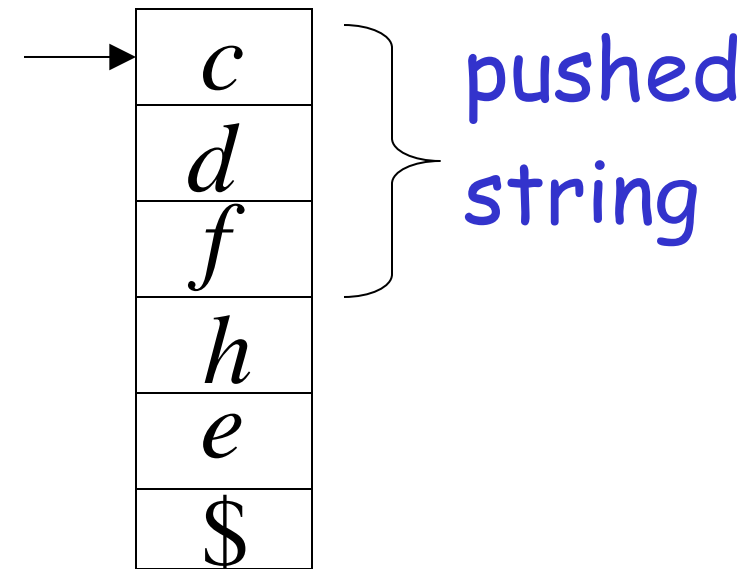
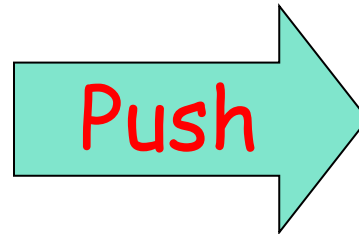
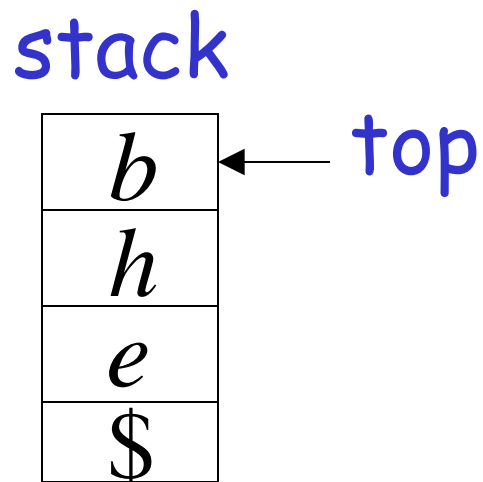
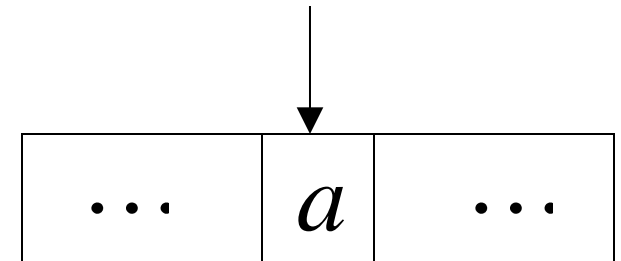
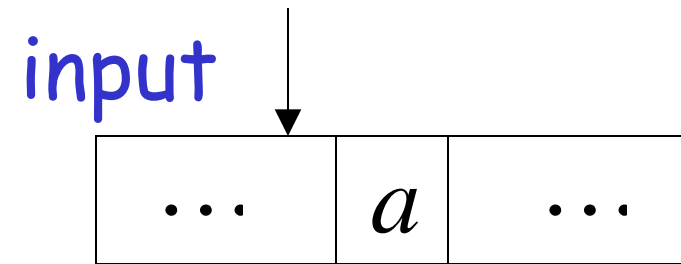
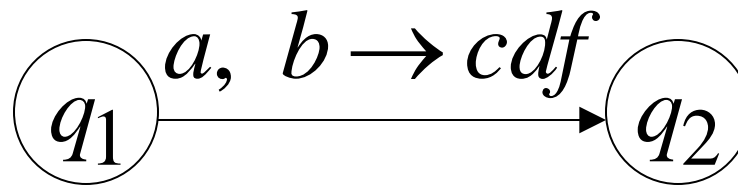


Halt and Reject

Pushing Strings



Example:



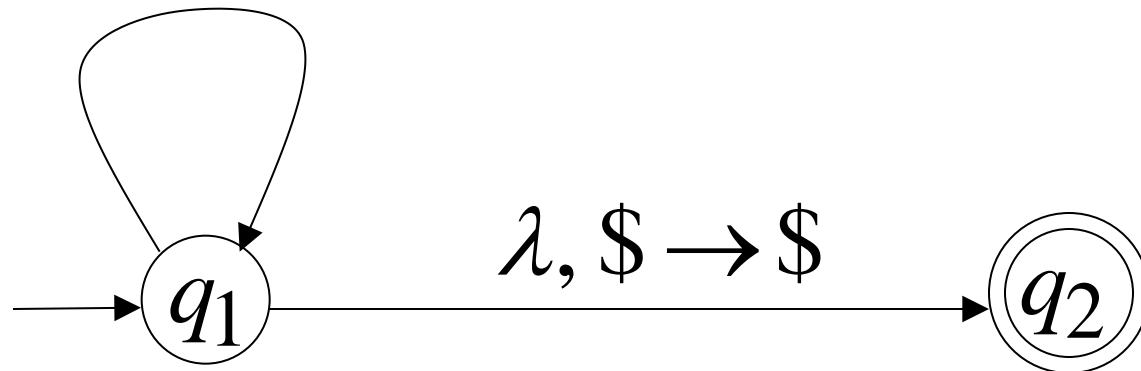
Another PDA example: Language?

PDA M

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

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

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



Another PDA example

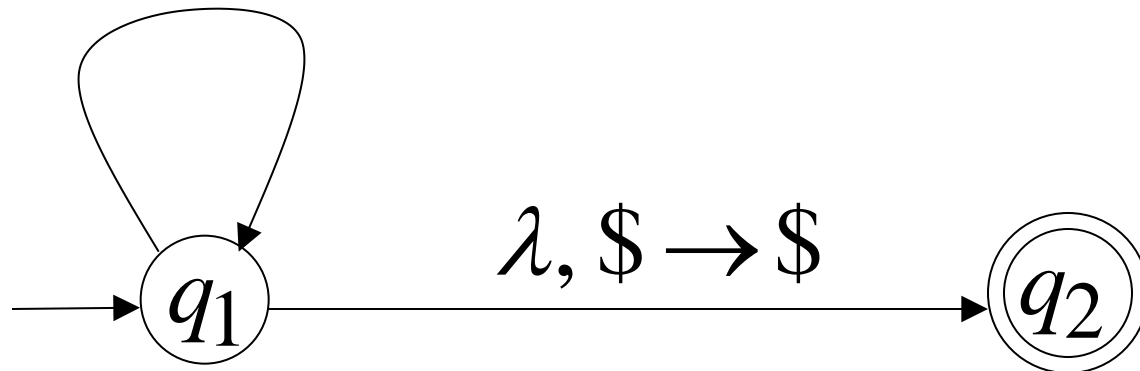
$$L(M) = \{w \in \{a,b\}^* : n_a(w) = n_b(w)\}$$

PDA M

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

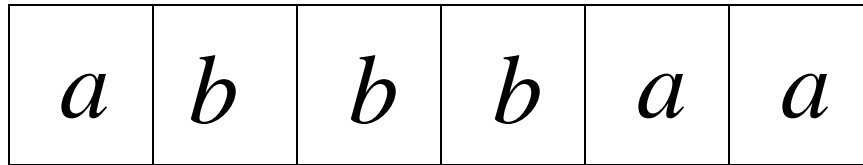
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



Execution Example: Time 0

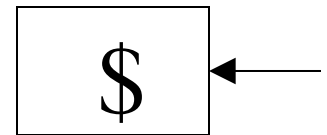
Input



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

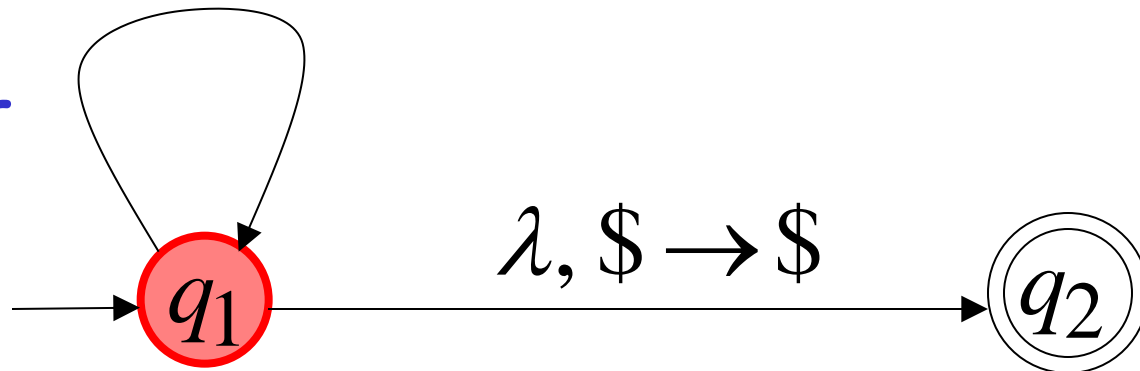
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



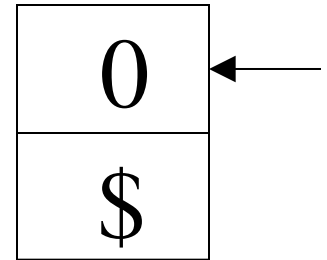
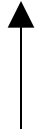
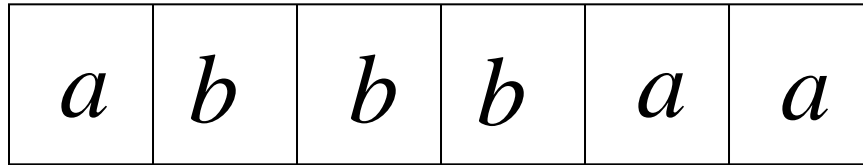
Stack

current
state



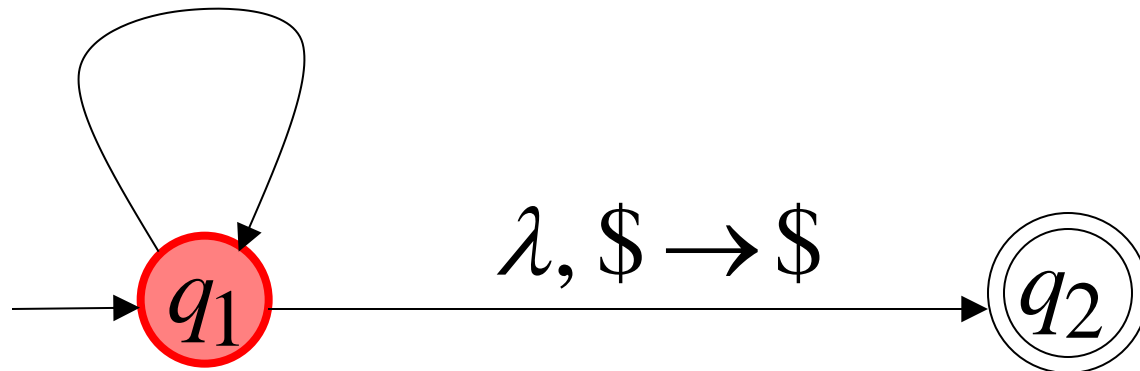
Time 1

Input



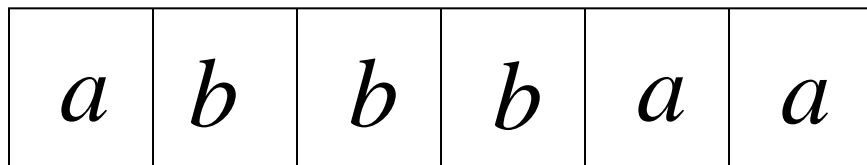
Stack

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



Time 3

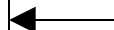
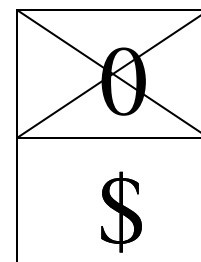
Input



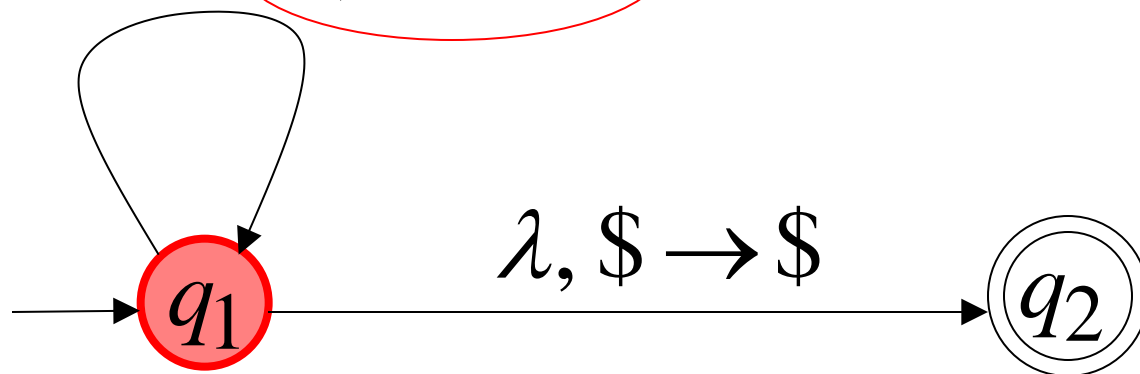
$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

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

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

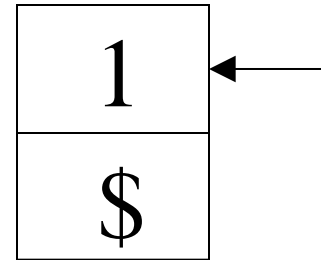
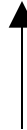
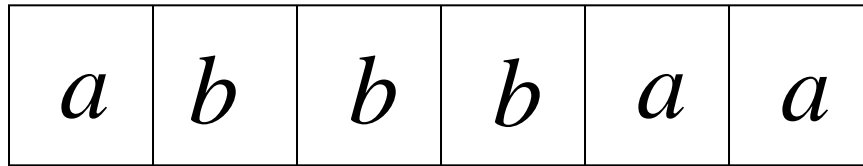


Stack



Time 4

Input

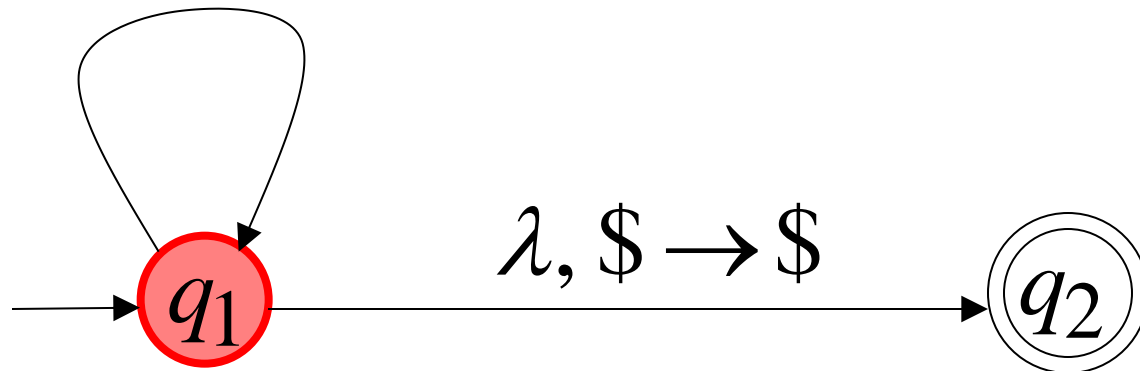


Stack

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

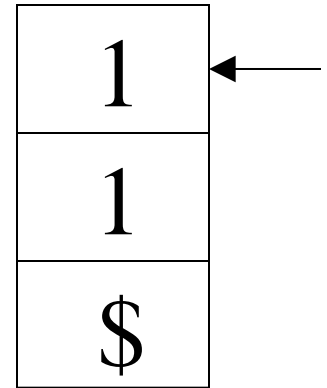
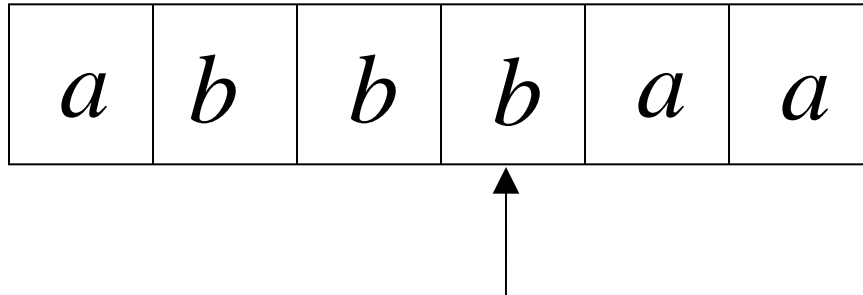
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



Time 5

Input

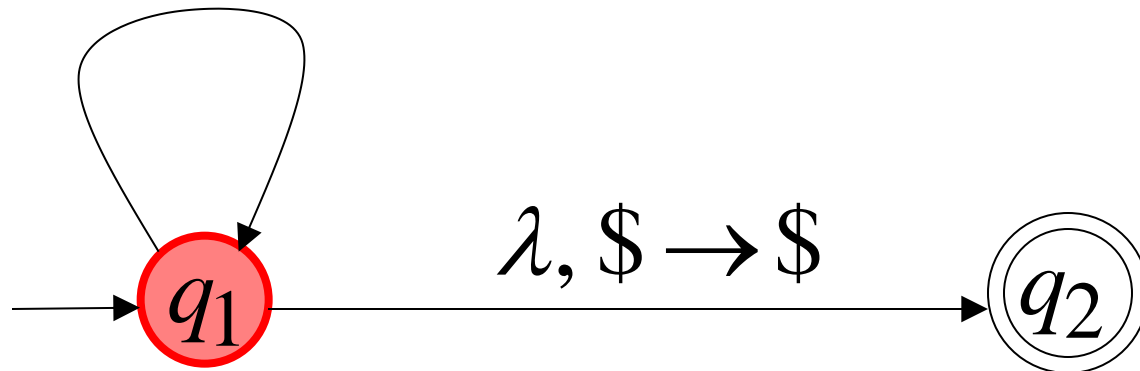


Stack

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

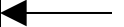
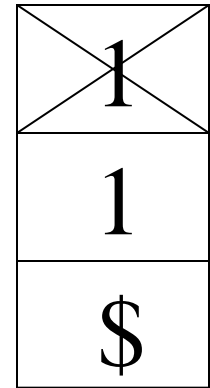
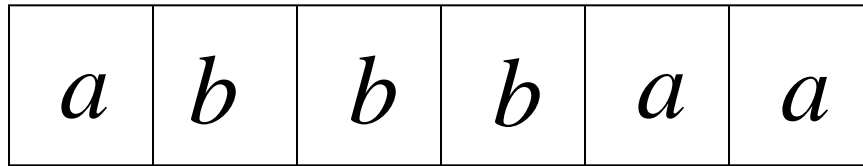
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



Time 6

Input

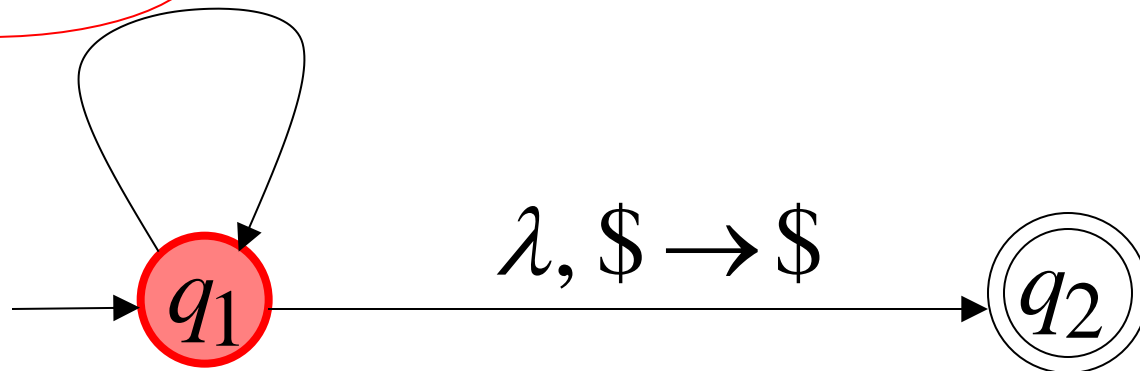


Stack

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

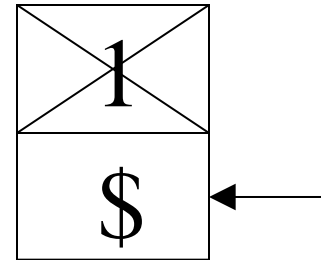
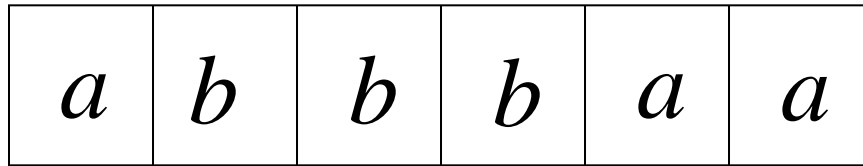
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



Time 7

Input

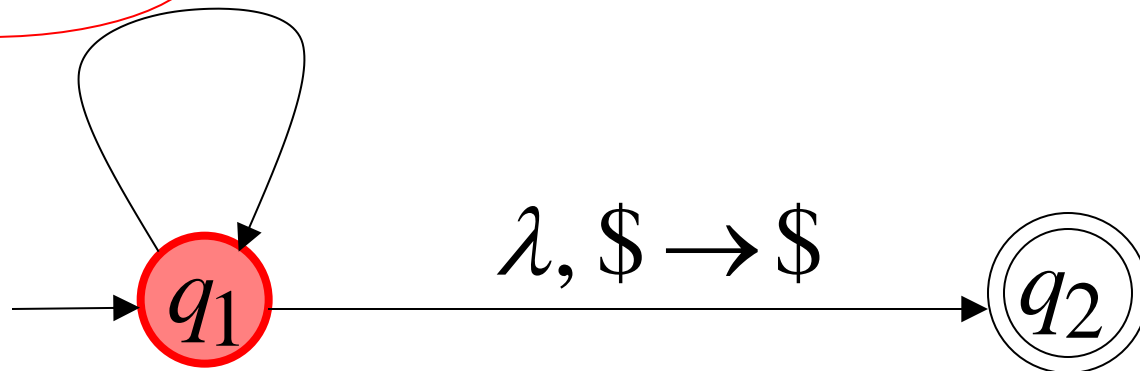


Stack

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

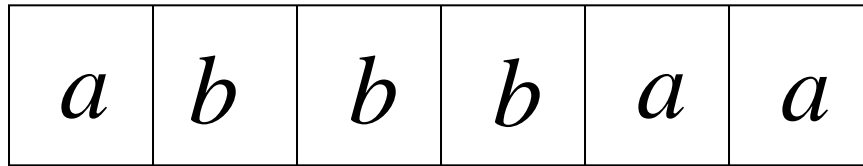
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



Time 8

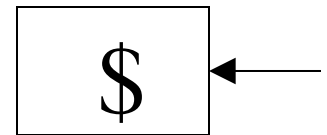
Input



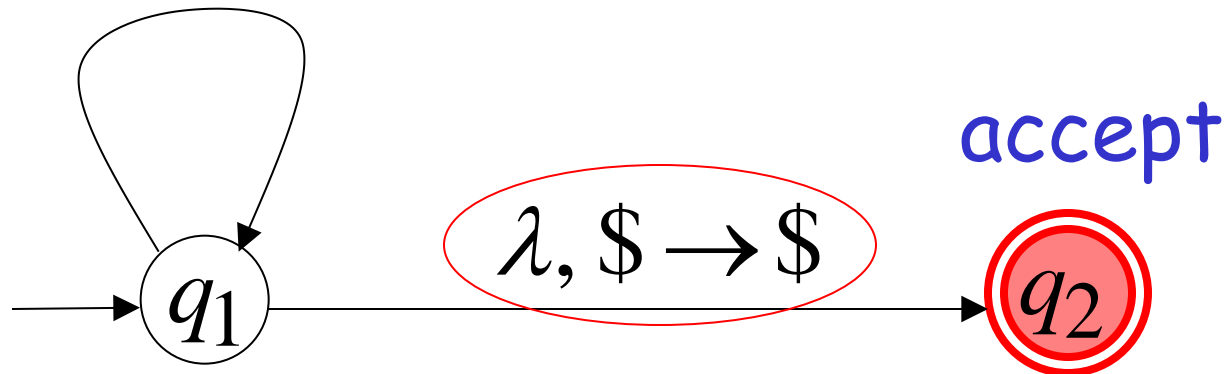
$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

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

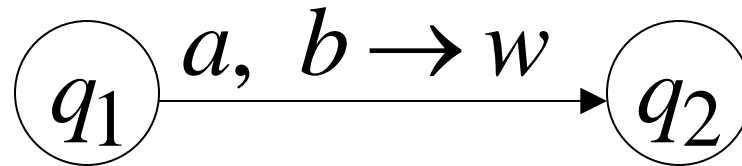
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Stack

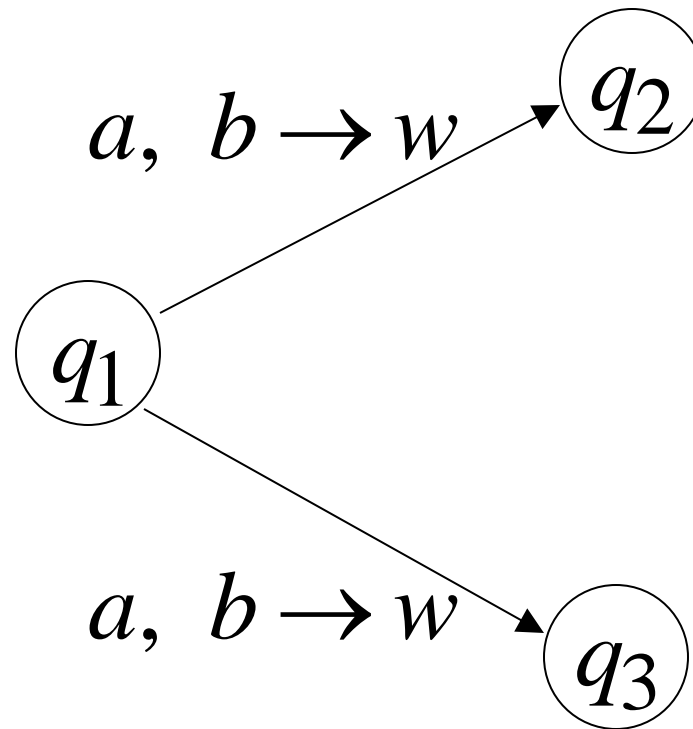


PDAs: Formal Definition



Transition function:

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



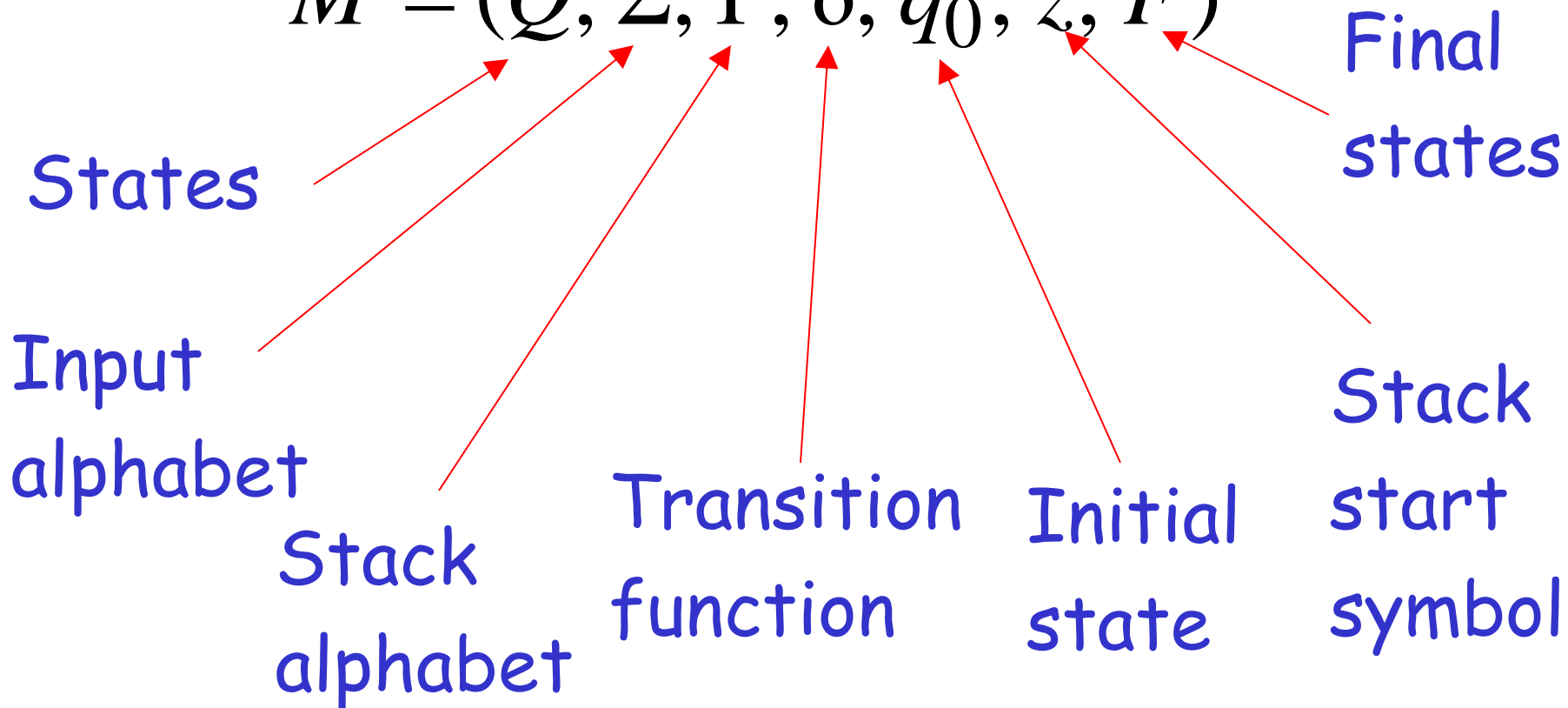
Transition function:

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

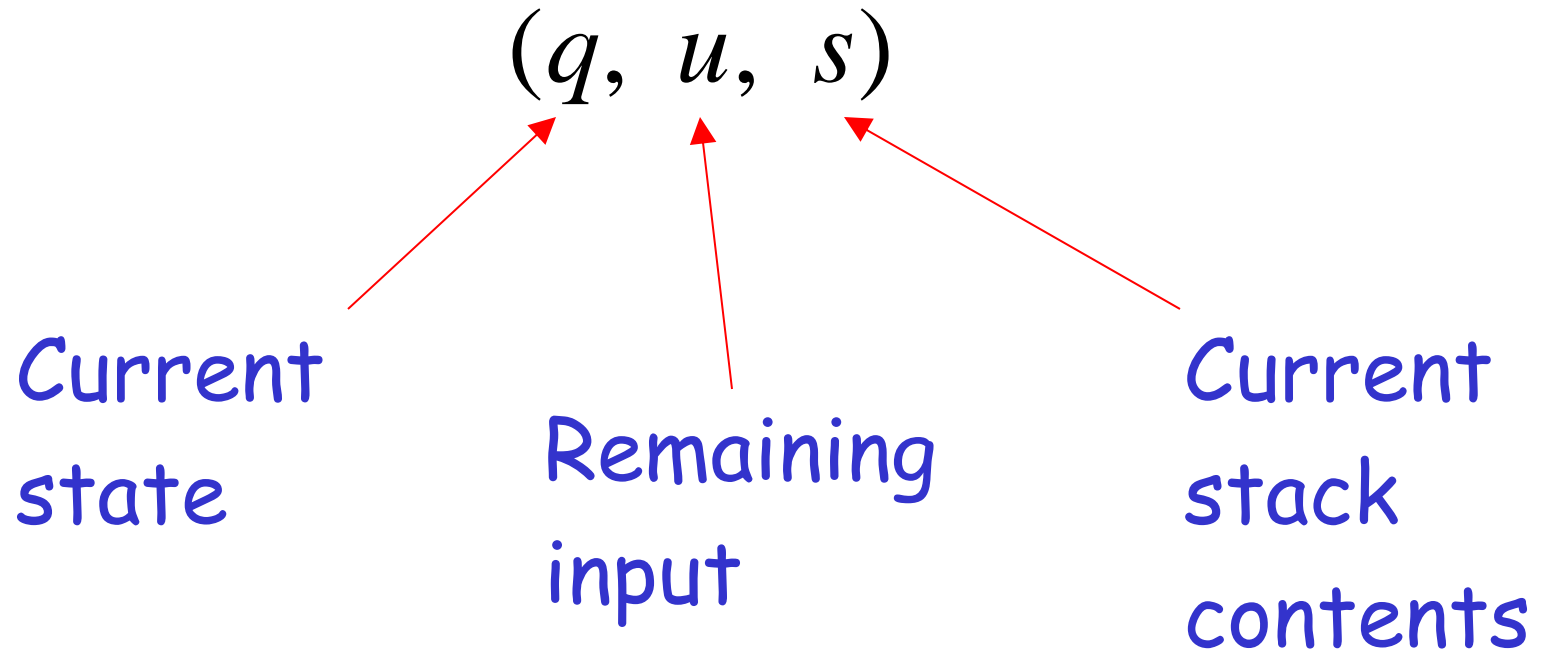
Formal Definition

Pushdown Automaton (PDA)

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



Instantaneous Description



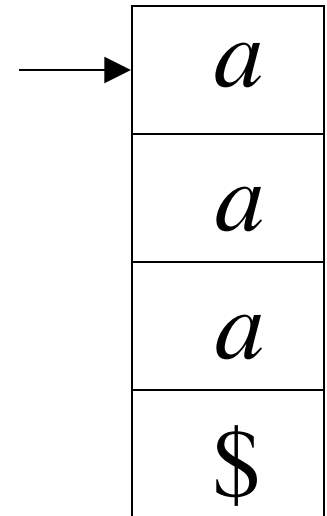
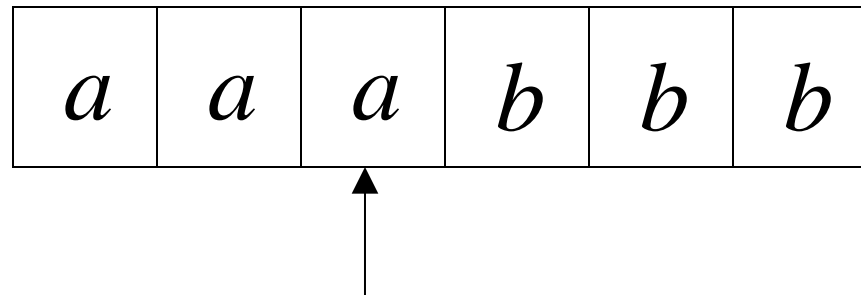
Example:

Instantaneous Description

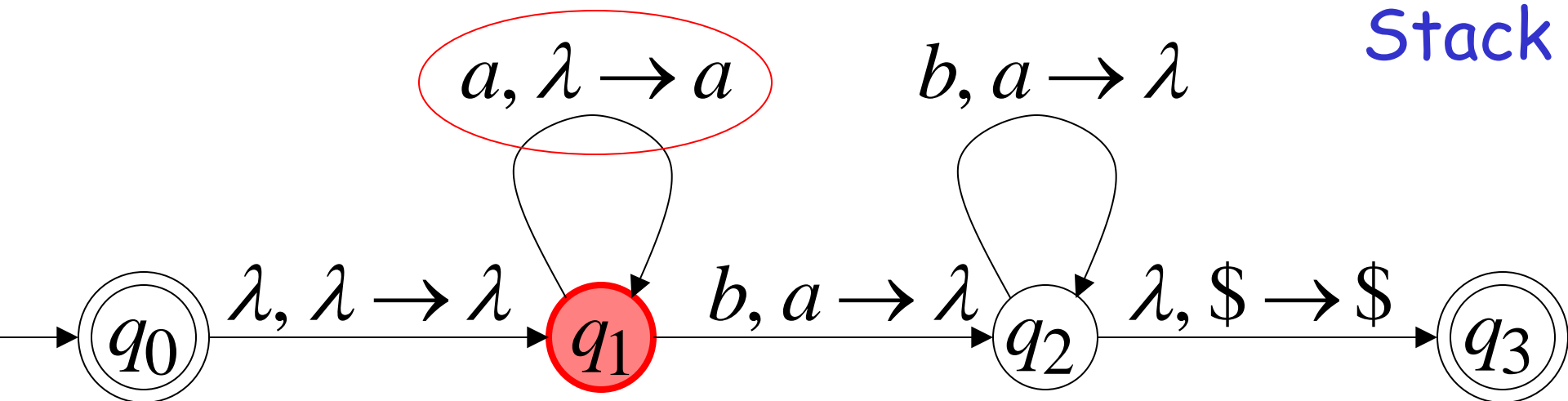
$(q_1, bbb, aaa\$)$

Time 4:

Input



Stack



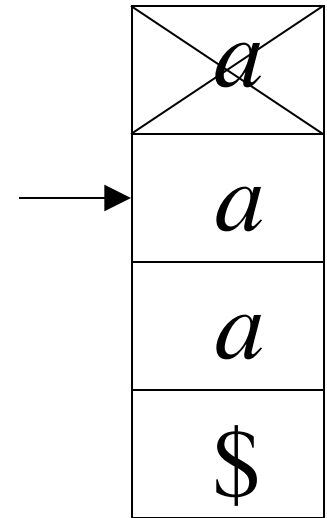
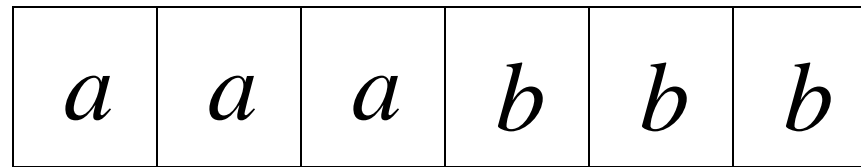
Example:

Instantaneous Description

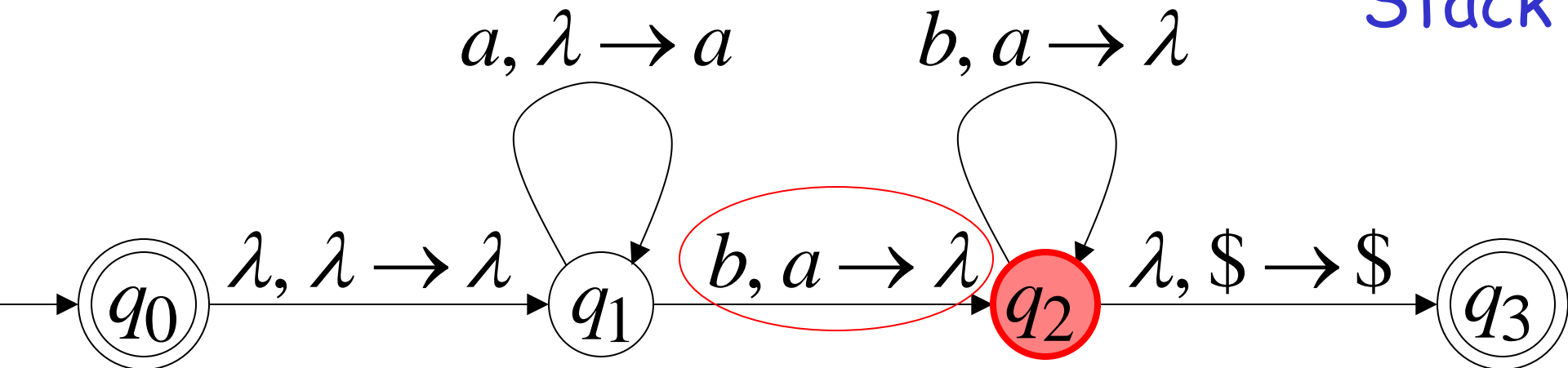
$(q_2, bb, aa\$)$

Time 5:

Input



Stack



We write:

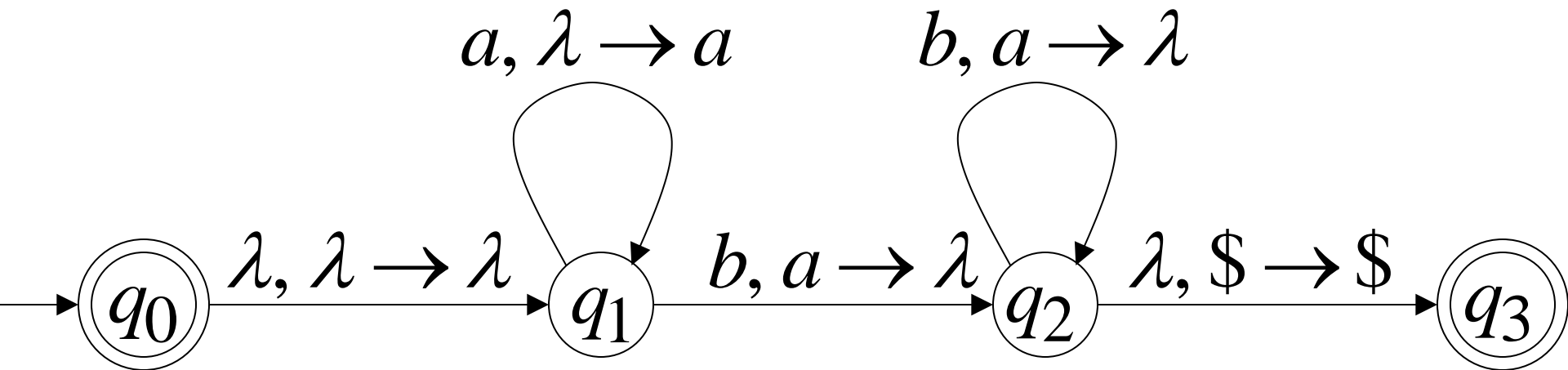
$$(q_1, bbb, aaa\$) \succ (q_2, bb, aa\$)$$

Time 4

Time 5

A computation:

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



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

For convenience we write:

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

Formal Definition

Language $L(M)$ of PDA M :

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

Initial state



Final state



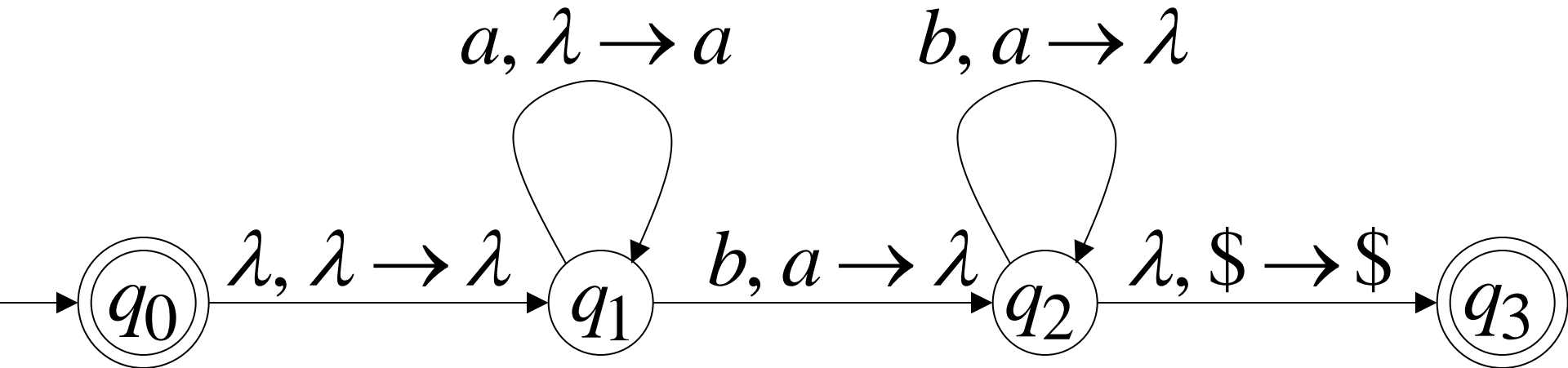
Example:

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



$$aaabbb \in L(M)$$

PDA M :

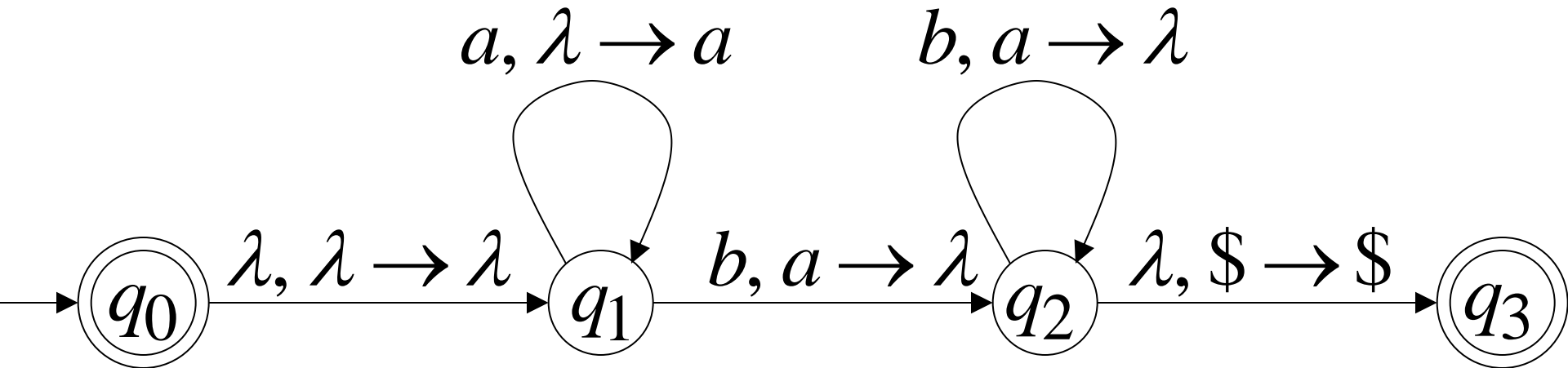


$$(q_0, a^n b^n, \$) \stackrel{*}{\succ} (q_3, \lambda, \$)$$



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

PDA M :



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

PDA M :

