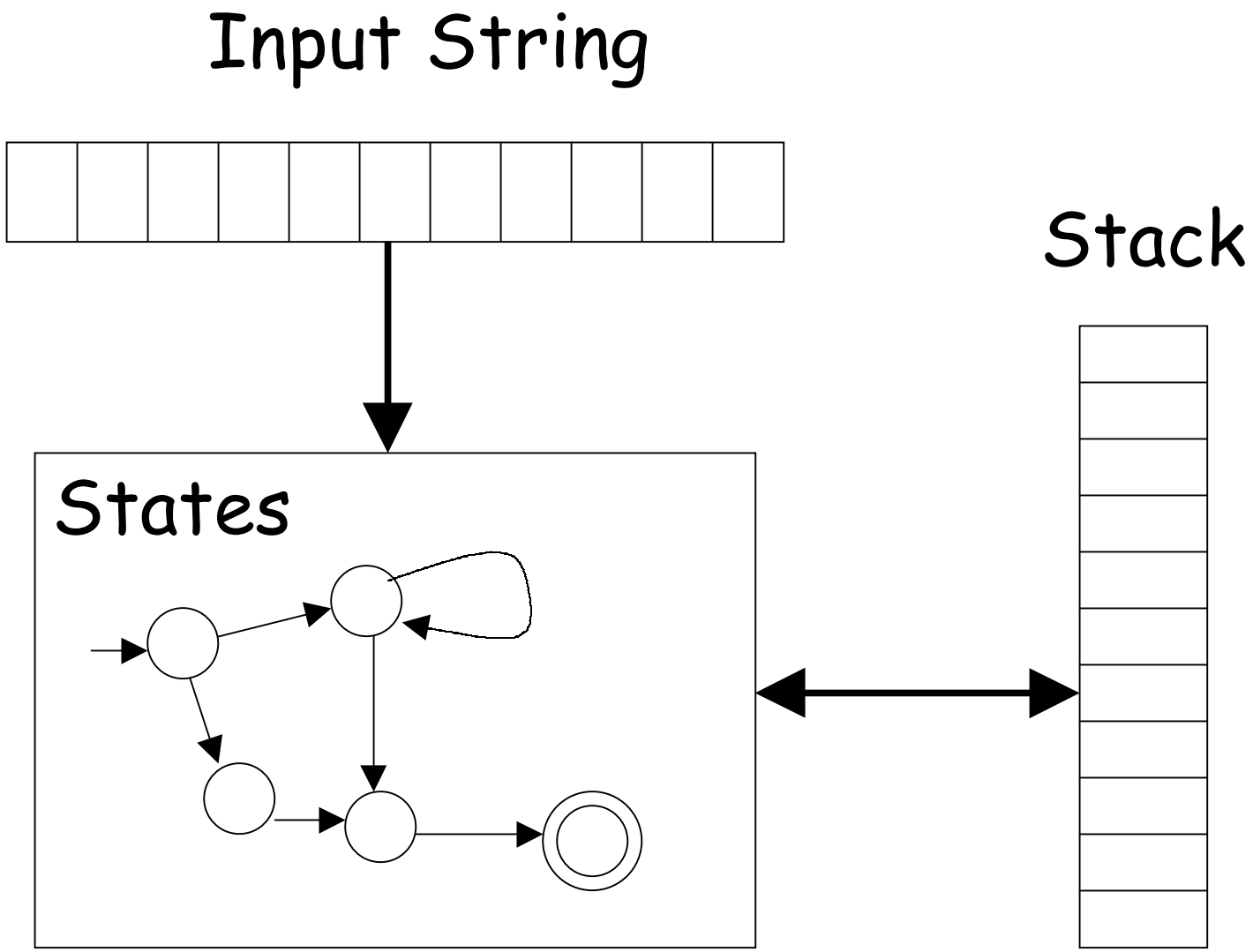
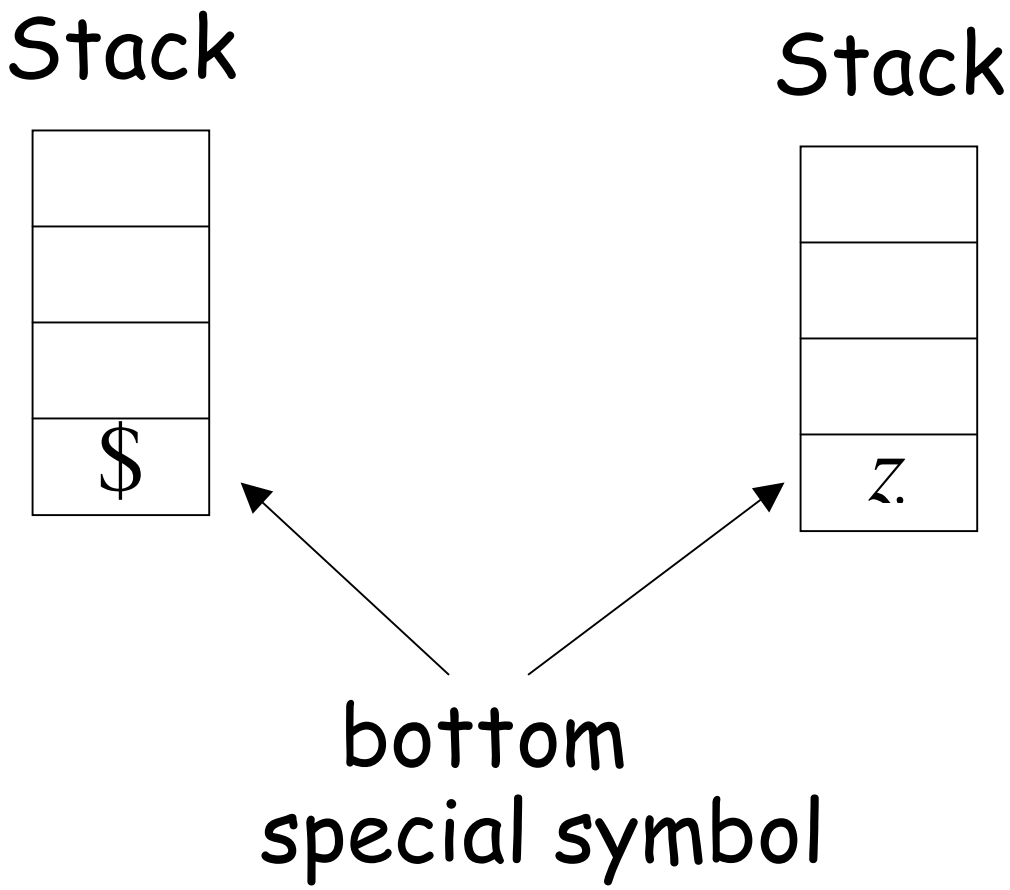
Pushdown Automata
PDAs

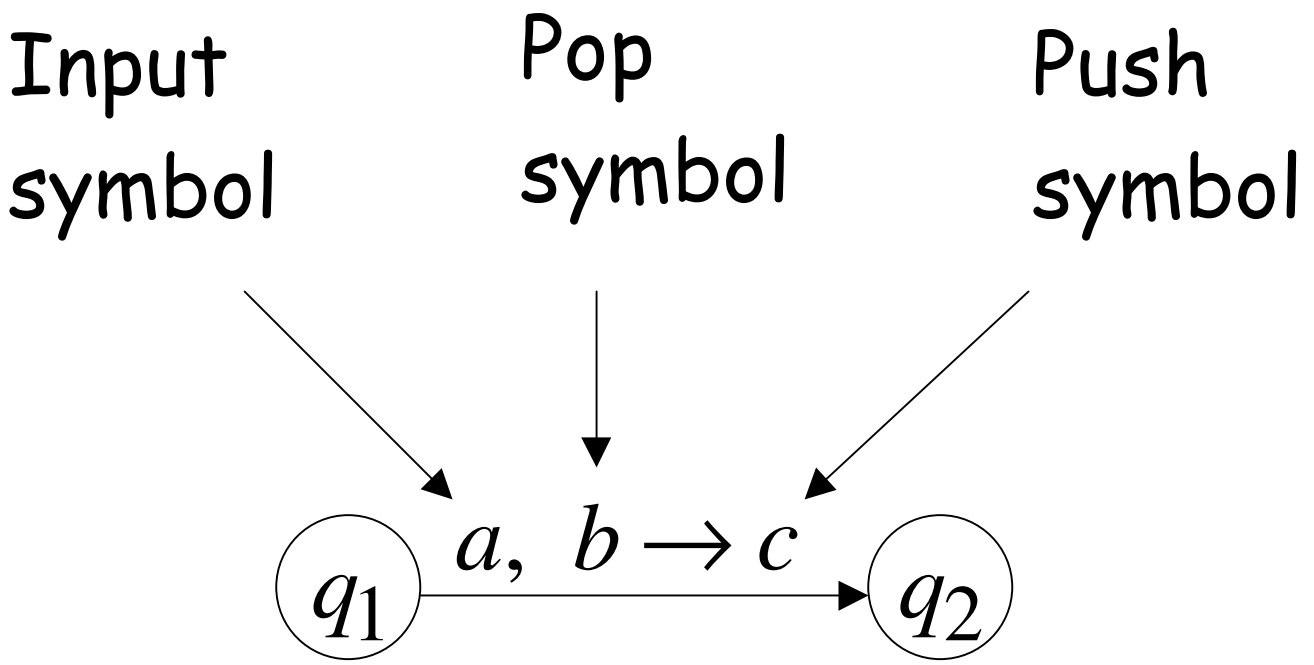
Pushdown Automaton -- PDA

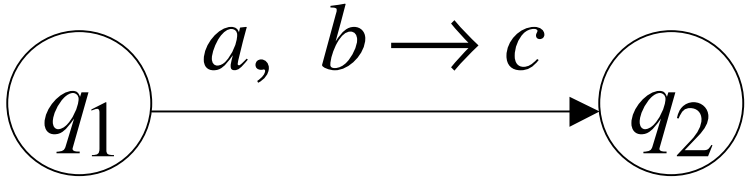


Initial Stack Symbol

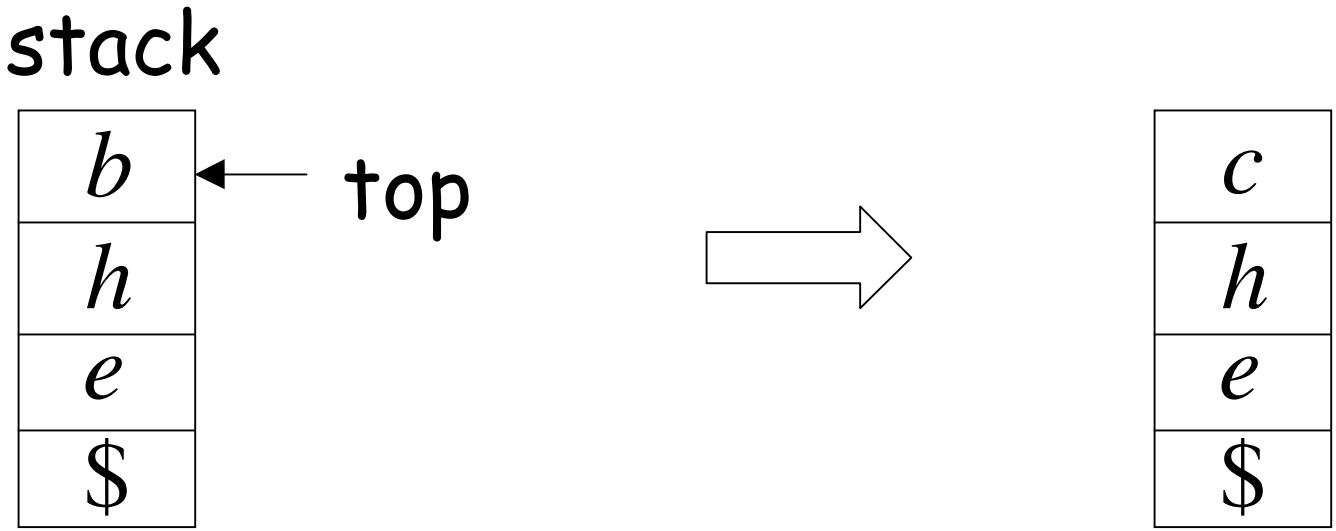


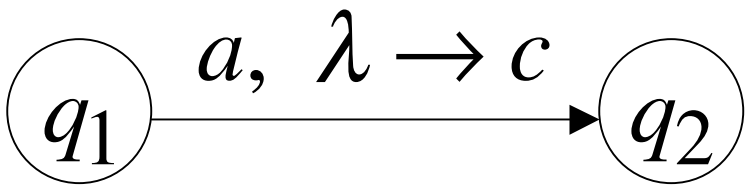
The States



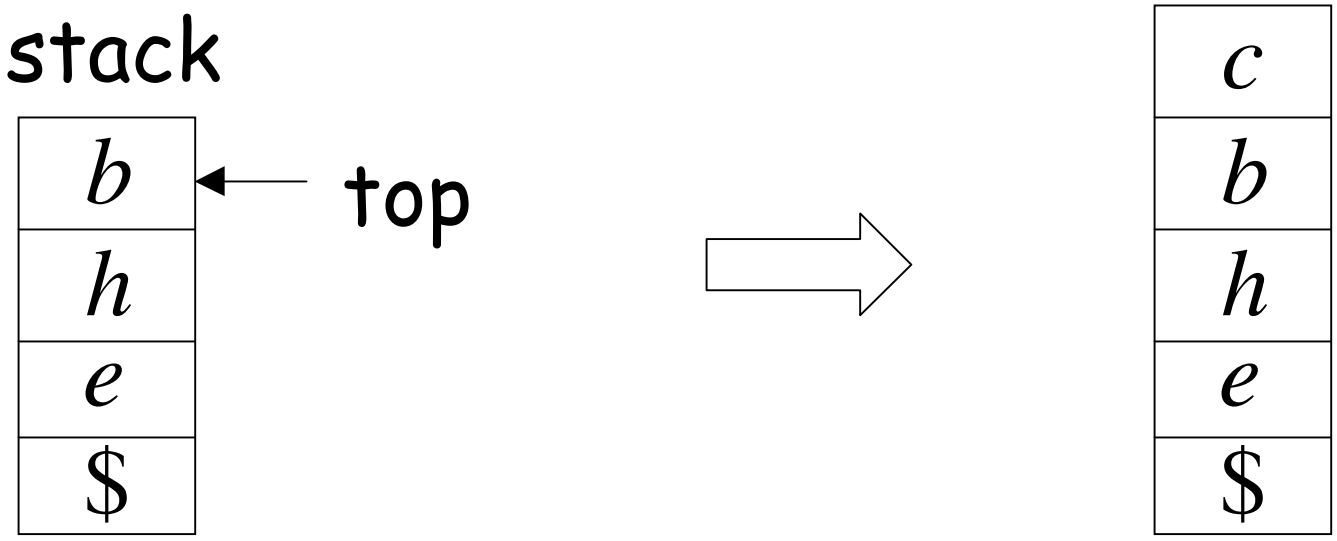

$a,\ b \rightarrow c$
q_1
q_2

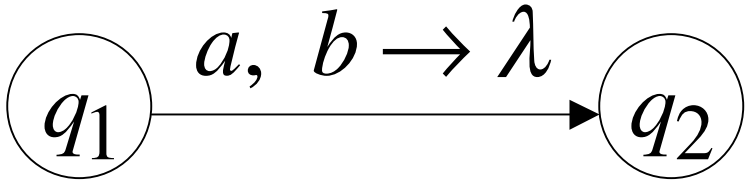
input
…
a
…
…
a
…


stack
b
top
h
e
$
c
h
e
$

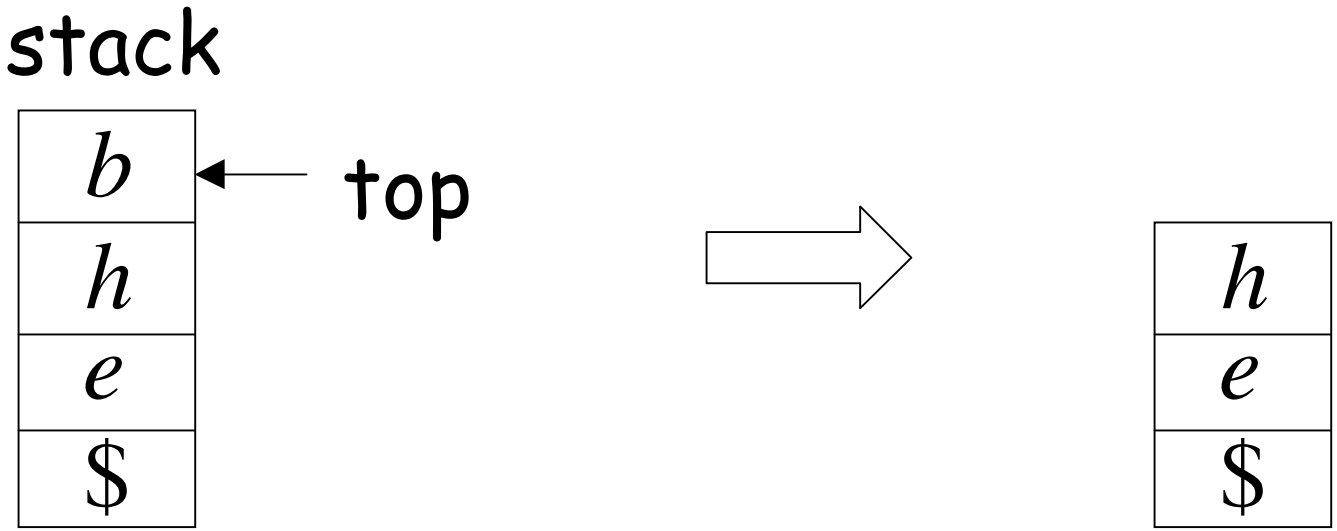

$a,\ \lambda \rightarrow c$
q_1
q_2

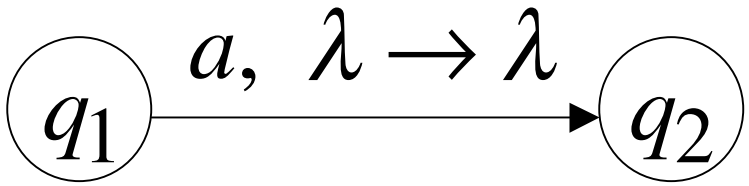
input
…
a
…
…
a
…


stack
b
top
h
e
$\$$
c
b
h
e
$\$$

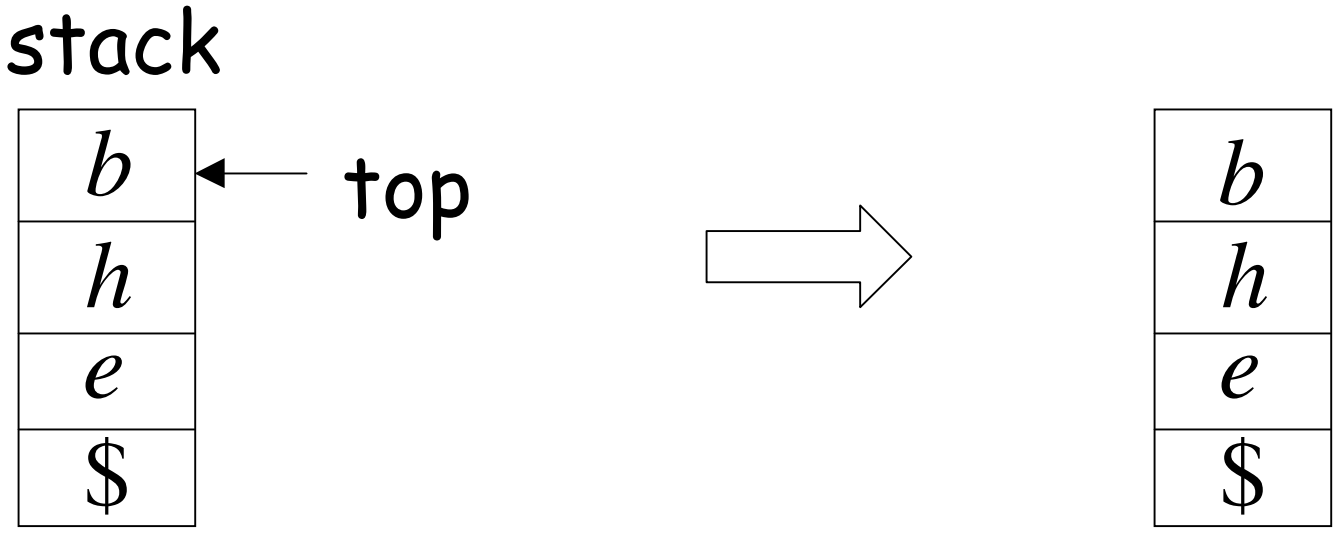

$a,\ b \to \lambda$
q_1
q_2

input
…
a
…
…
a
…

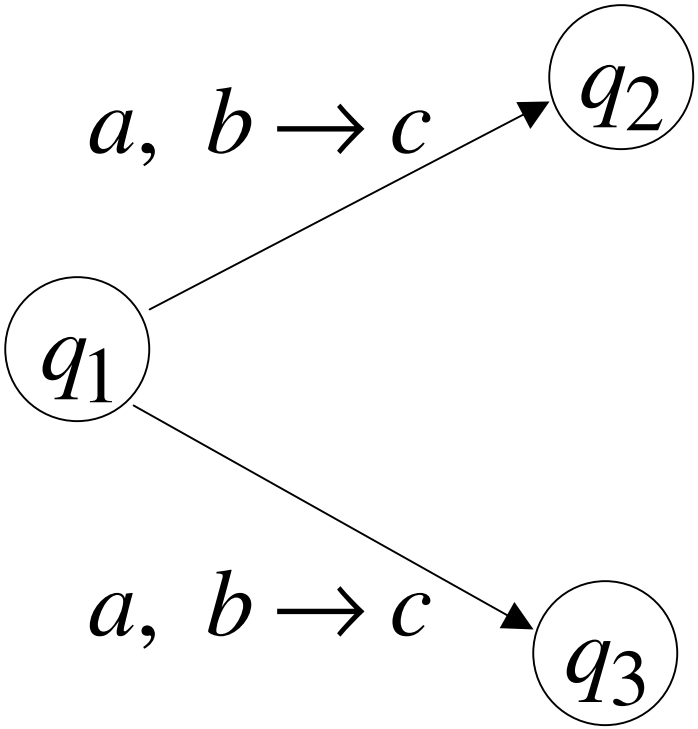

stack
b
top
h
e
$\$$
h
e
$\$$

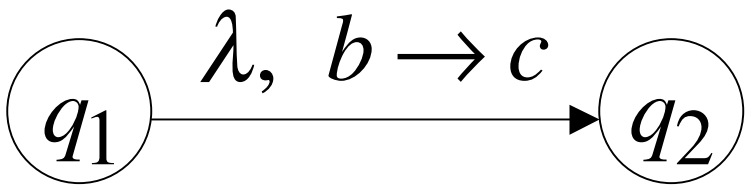

$a,\ \lambda \rightarrow \lambda$
q_1
q_2

input
…
a
…
…
a
…


stack
b
top
h
e
$\$$
b
h
e
$\$$

Non-Determinism

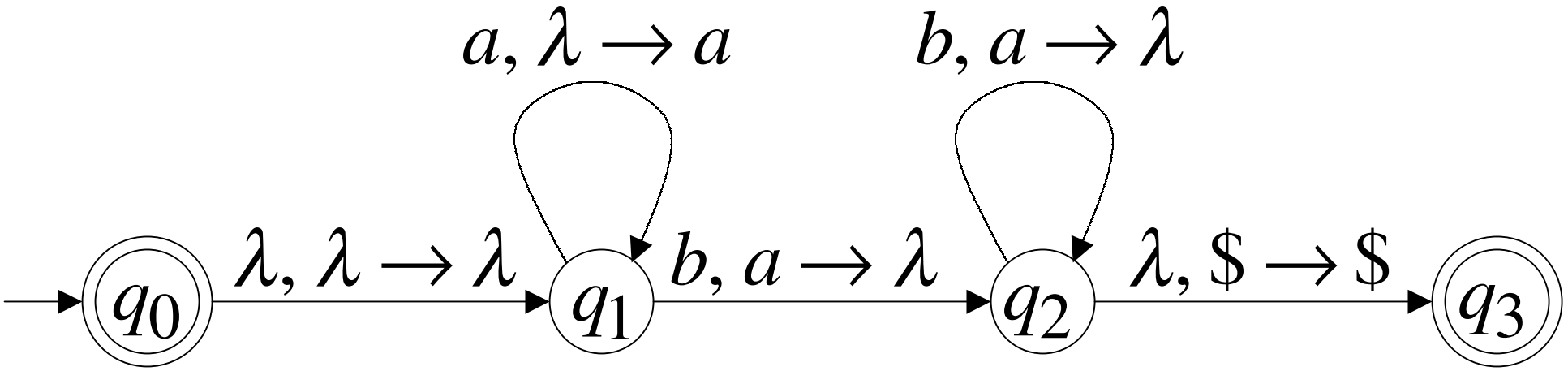




$\lambda-$transition

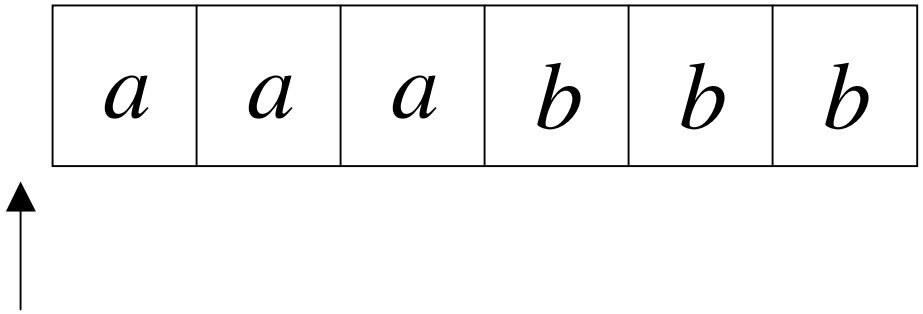
NPDA: Non-Deterministic PDA

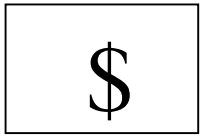
Example:



Execution Example: Time 0

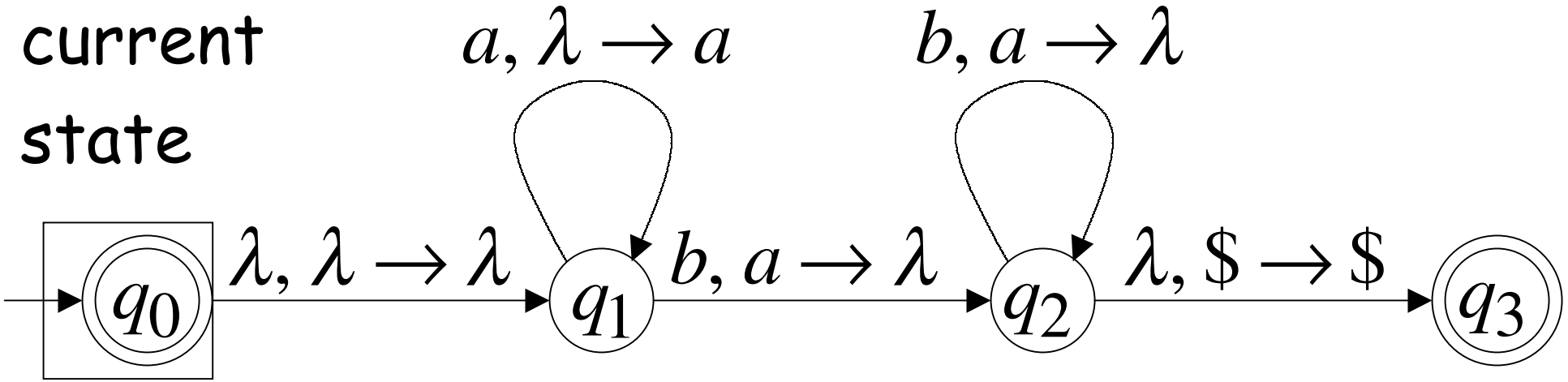
Input





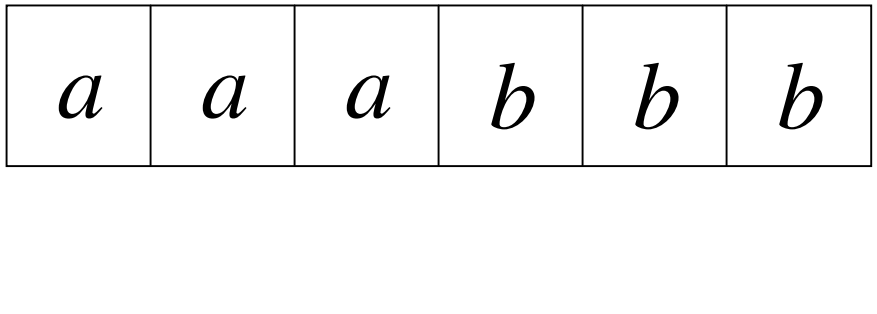
Stack

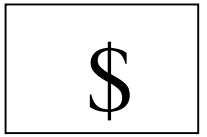
current state



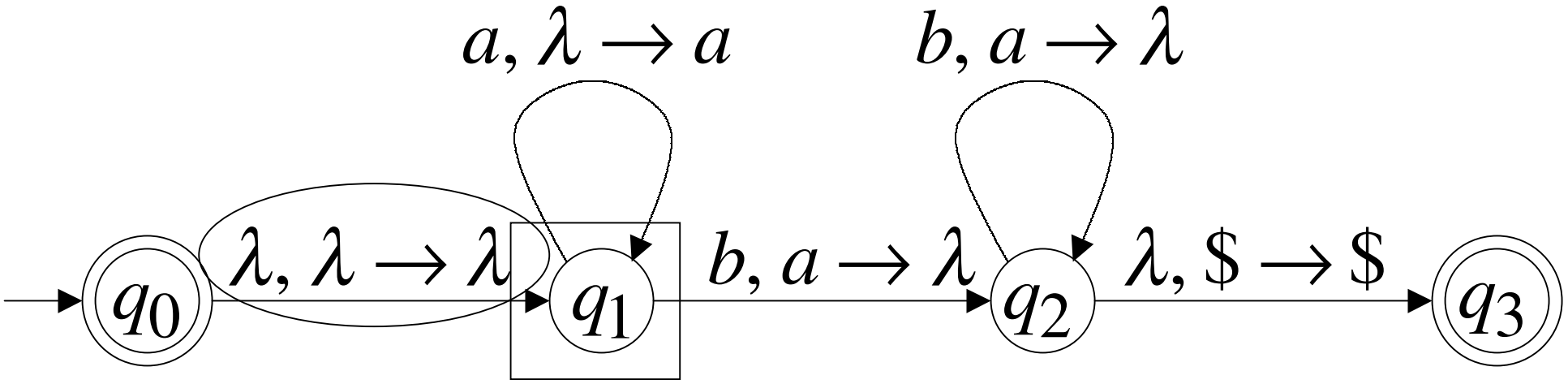
Time 1

Input



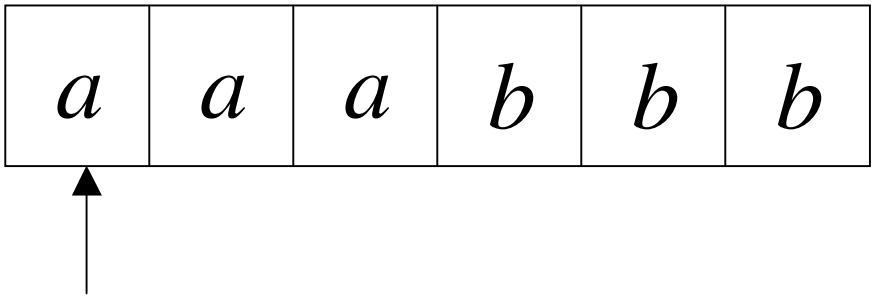


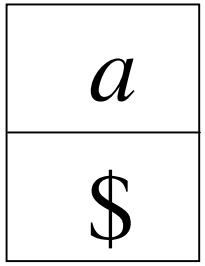
Stack



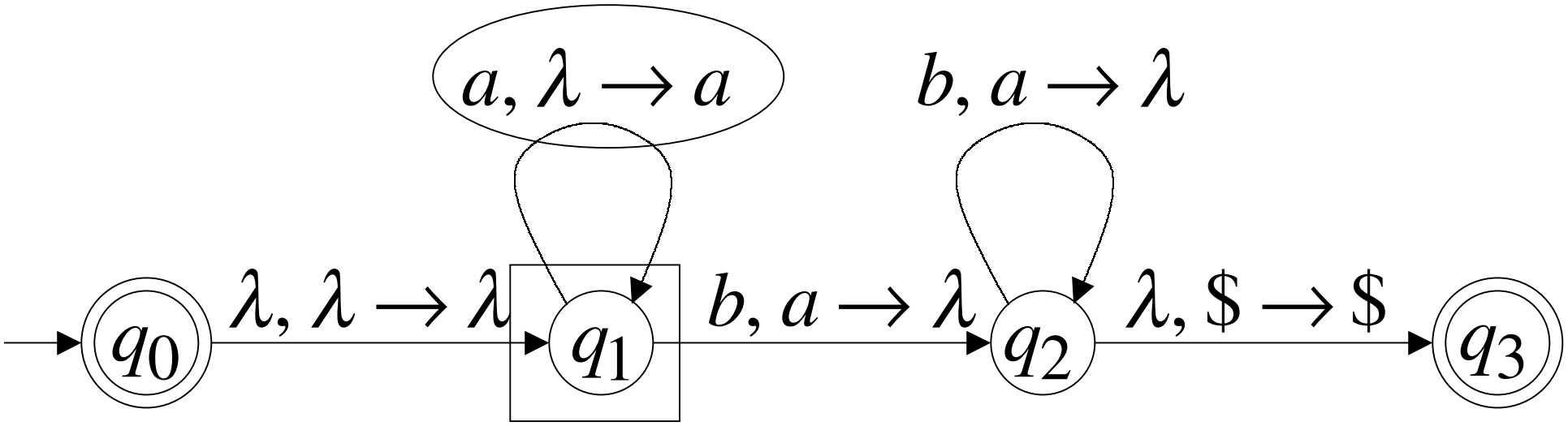
Time 2

Input



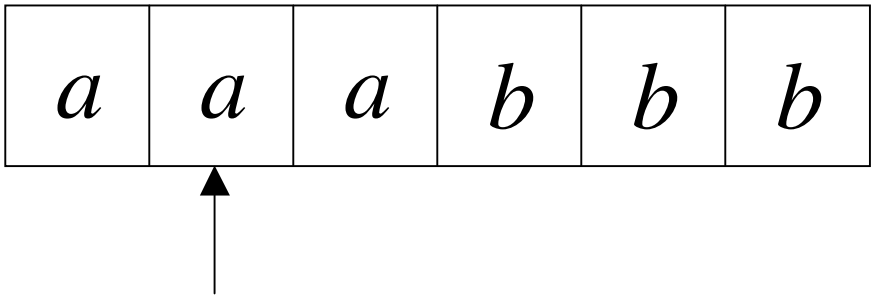


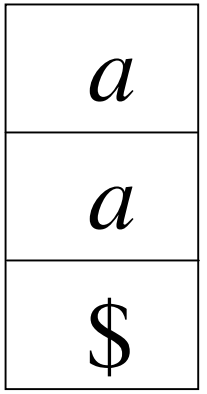
Stack



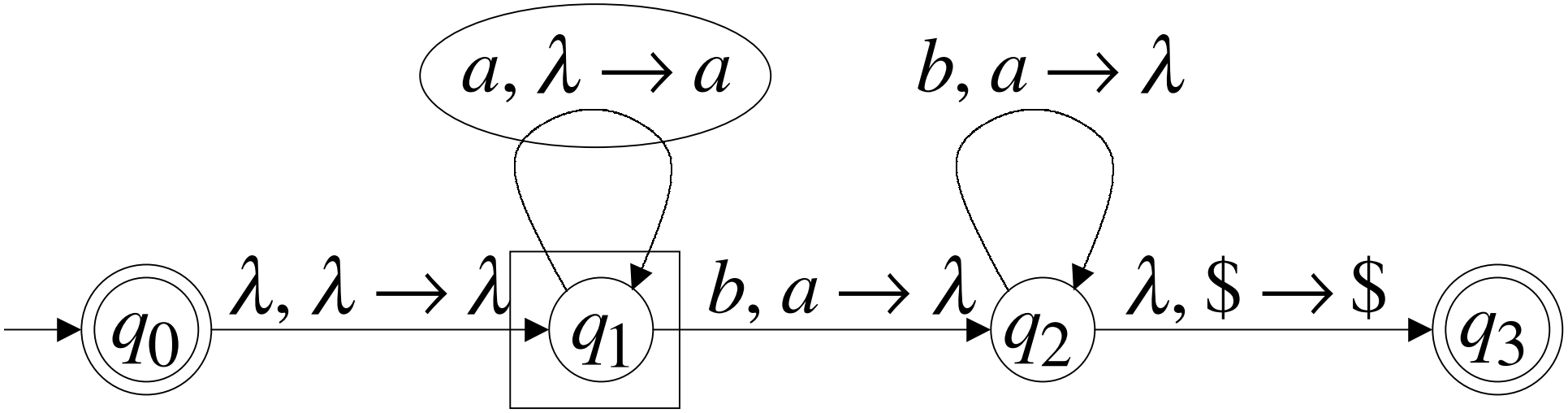
Time 3

Input



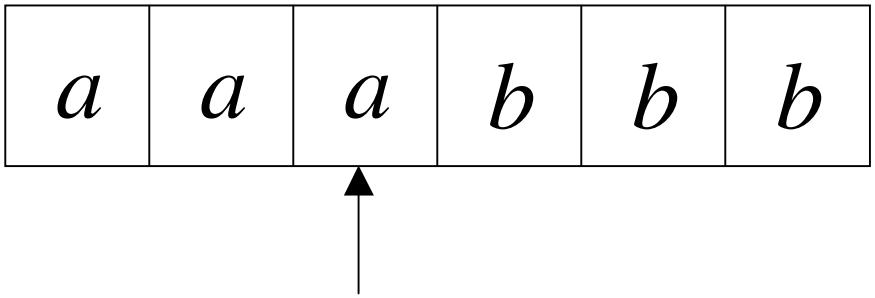


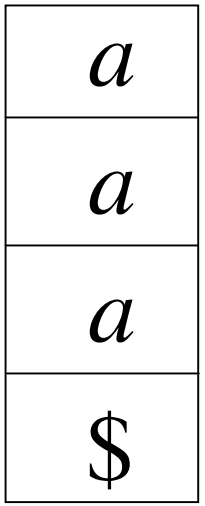
Stack



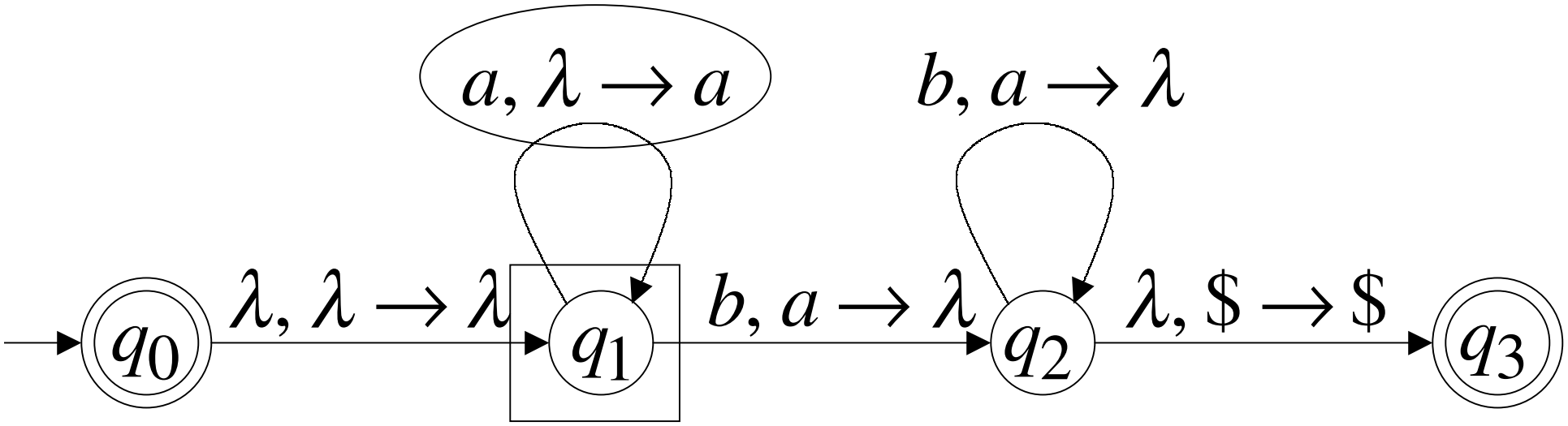
Time 4

Input



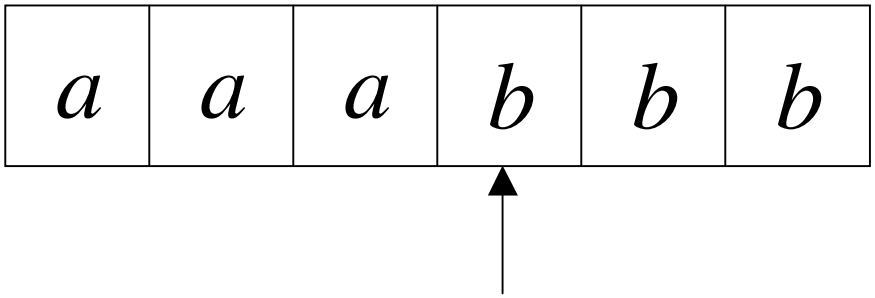


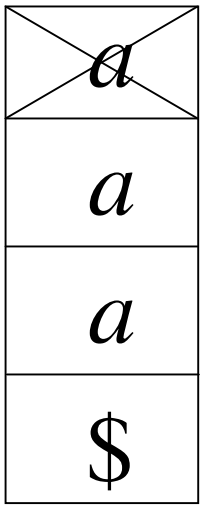
Stack



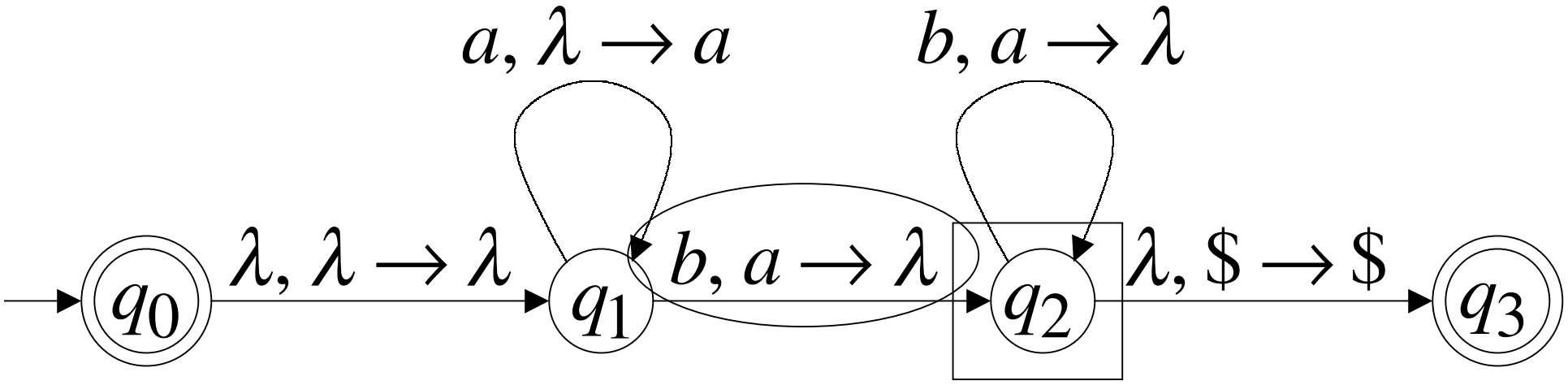
Time 5

Input



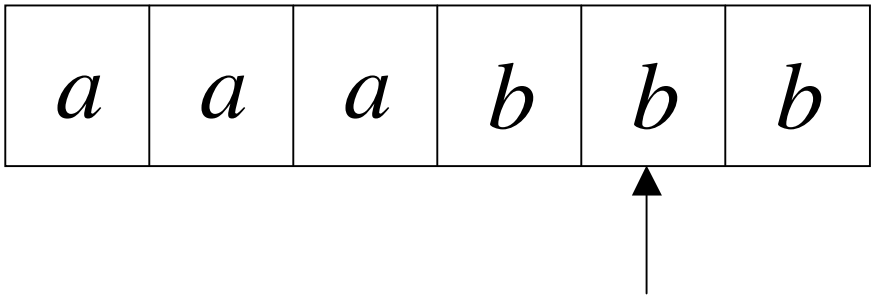


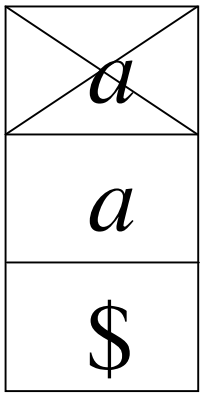
Stack



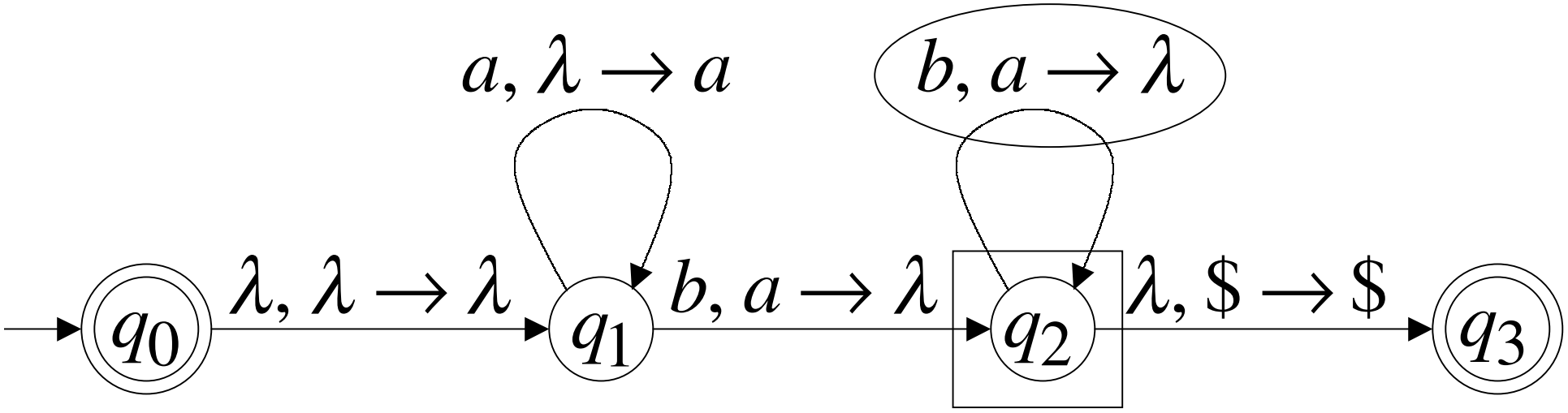
Time 6

Input



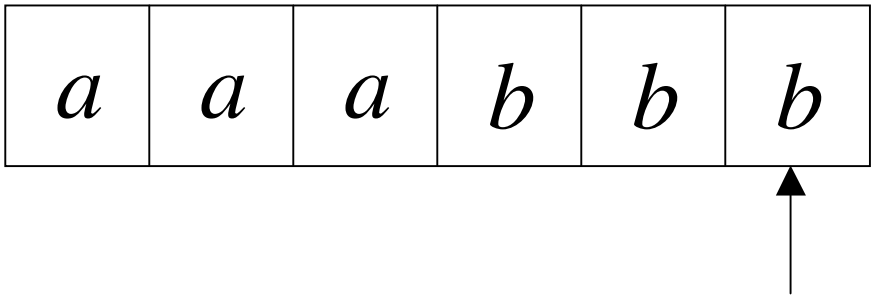


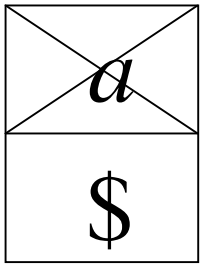
Stack



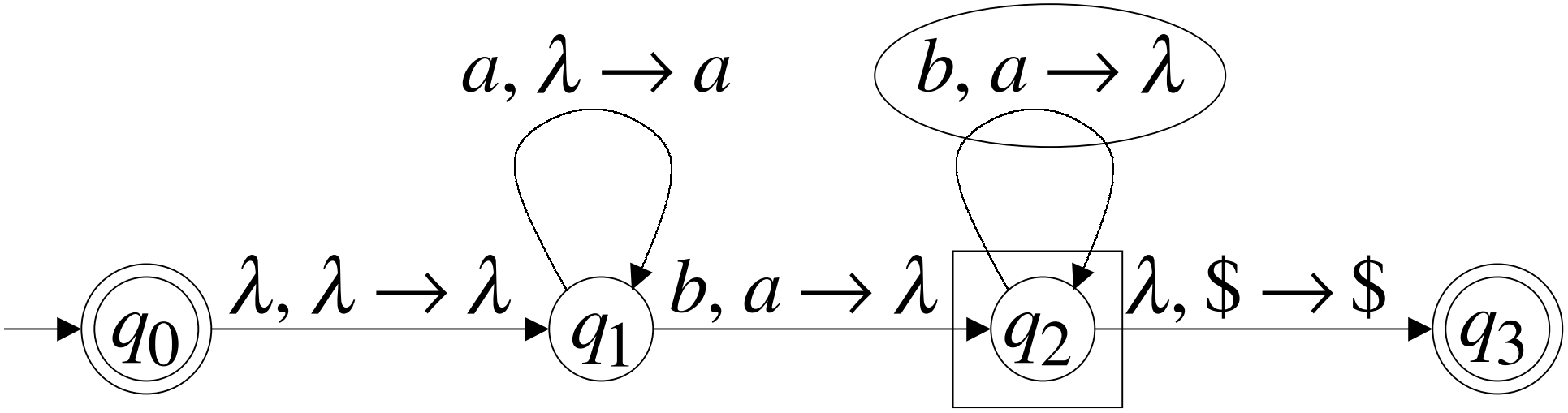
Time 7

Input



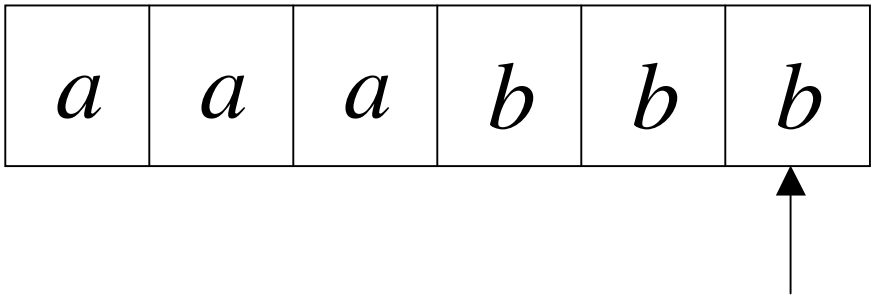


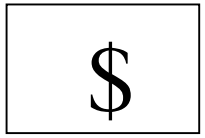
Stack



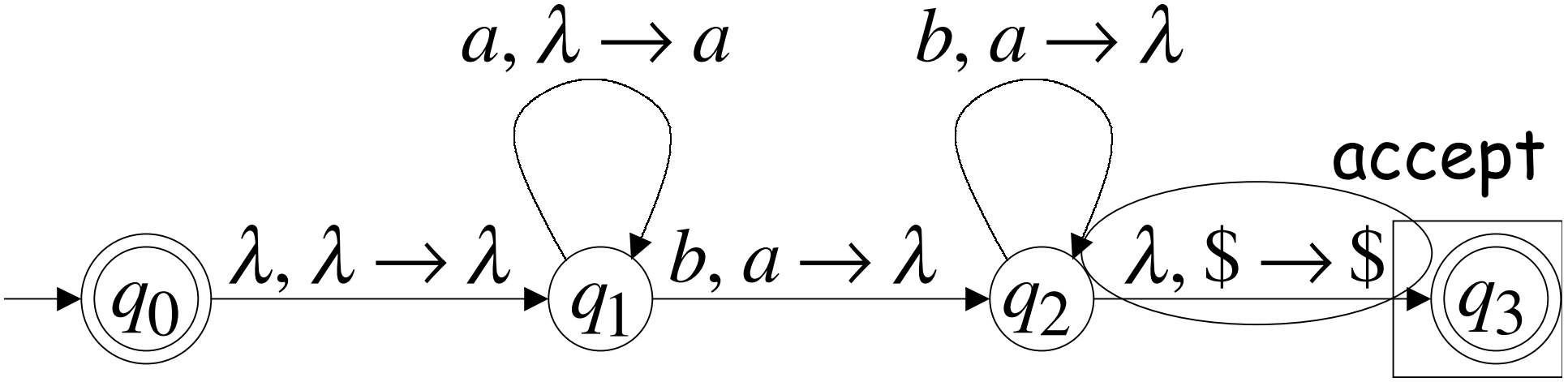
Time 8

Input





Stack

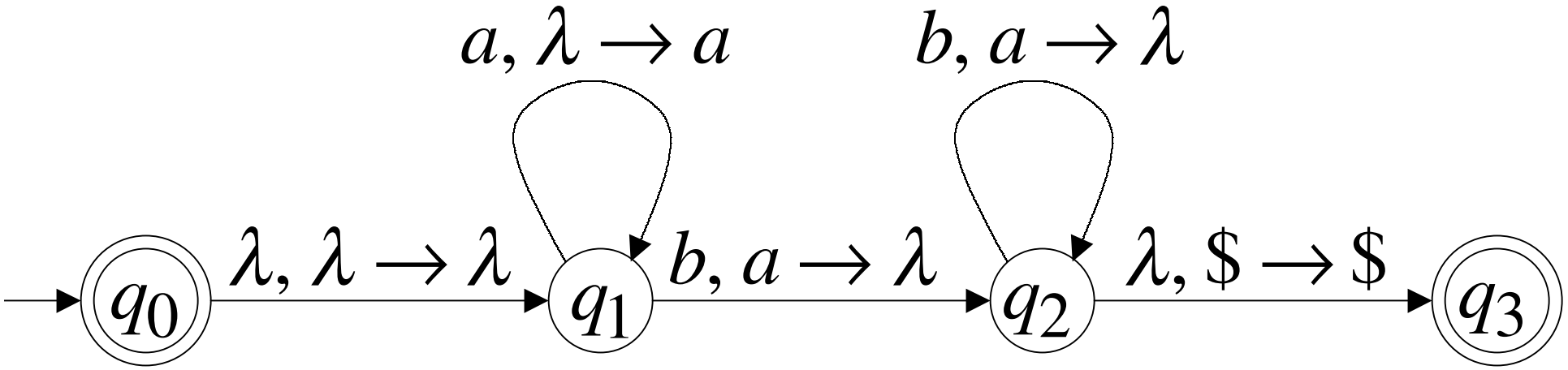


A string is accepted if:

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

We do not care about the stack contents

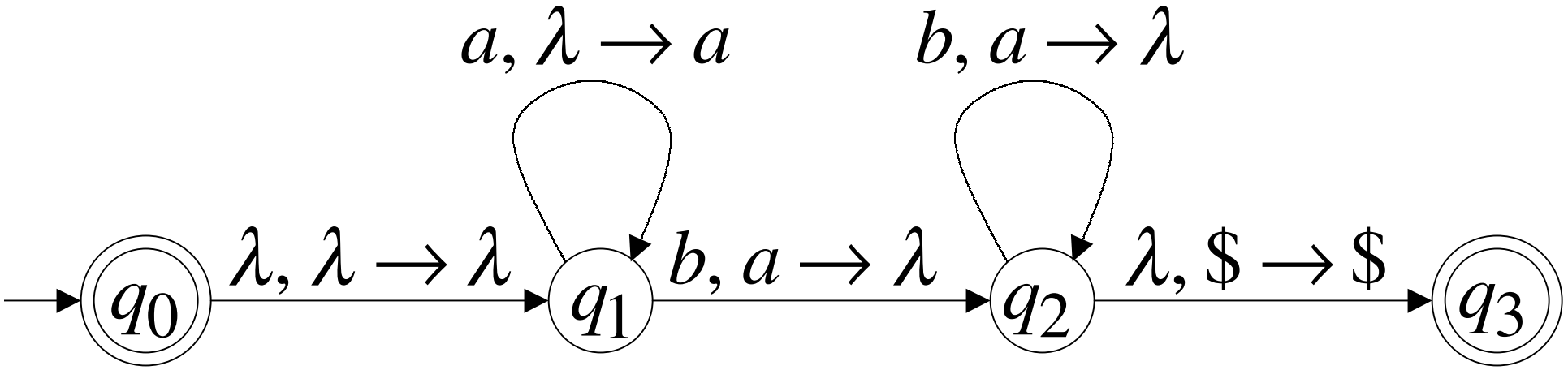
The input string $aaabbb$
is accepted by the NPDA:



In general,

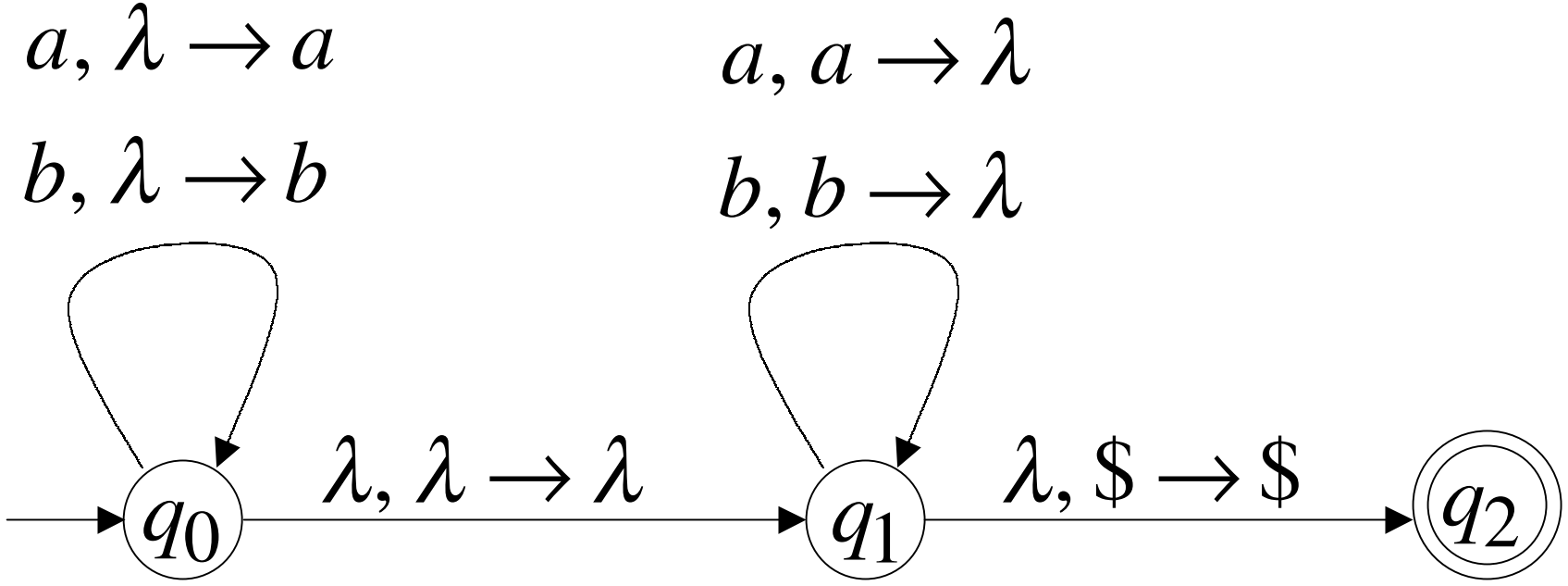
$$L = \{a^n b^n : n \geq 0\}$$

is the language accepted by the NPDA:

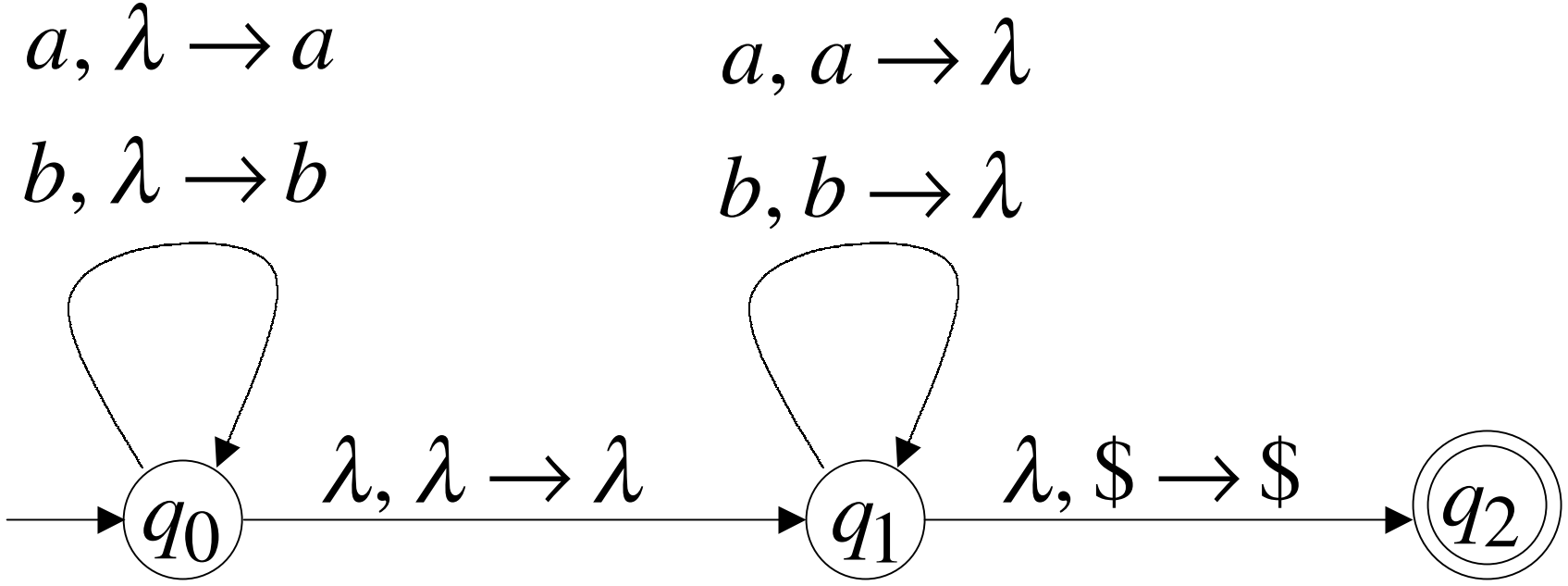


Another NPDA example

NPDA M

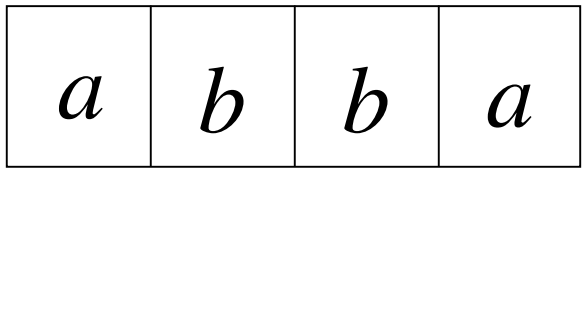


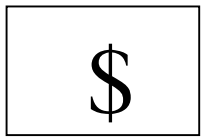
$$L(M) = \{ww^R\}$$


$a, \lambda \to a$
$b, \lambda \to b$
$a, a \to \lambda$
$b, b \to \lambda$
q_0
$\lambda, \lambda \to \lambda$
q_1
$\lambda, \$ \to \$$
q_2

Execution Example: Time 0

Input





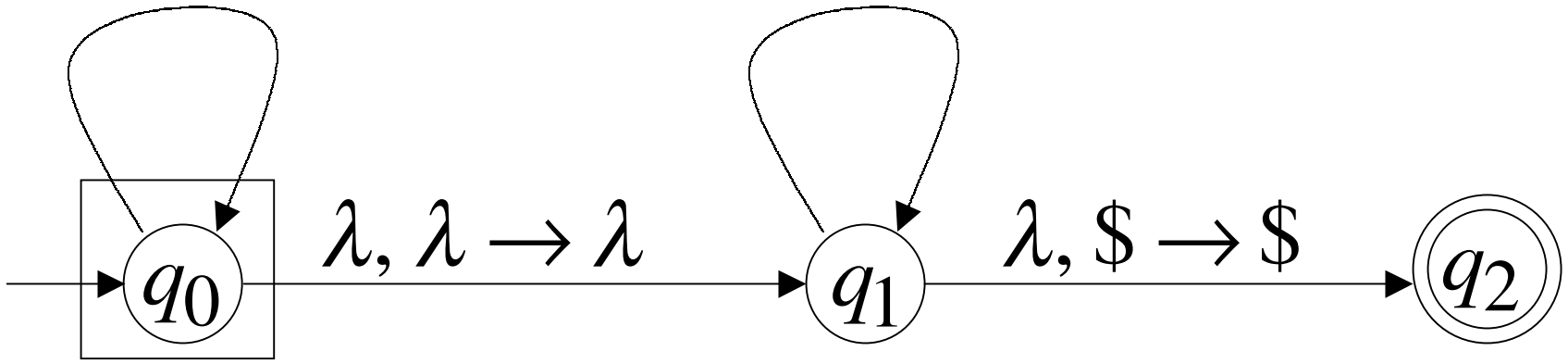
Stack

$a, \lambda \to a$

$b, \lambda \to b$

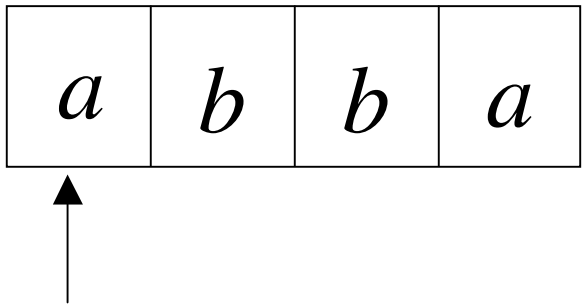
$a, a \to \lambda$

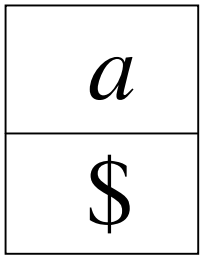
$b, b \to \lambda$



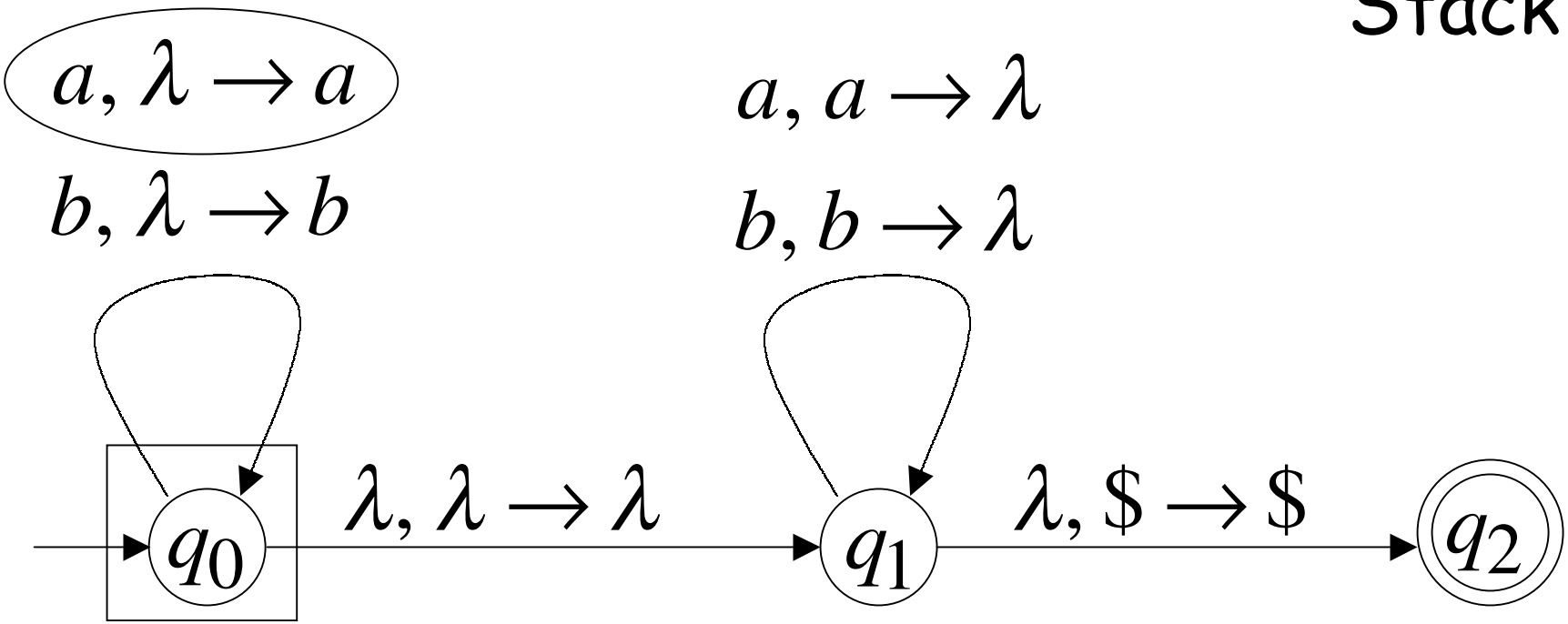
Time 1

Input



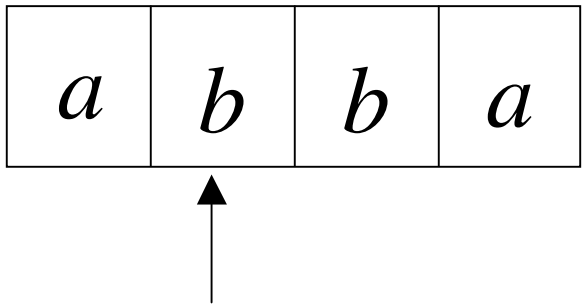


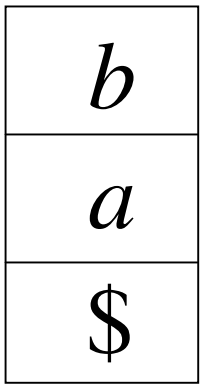
Stack



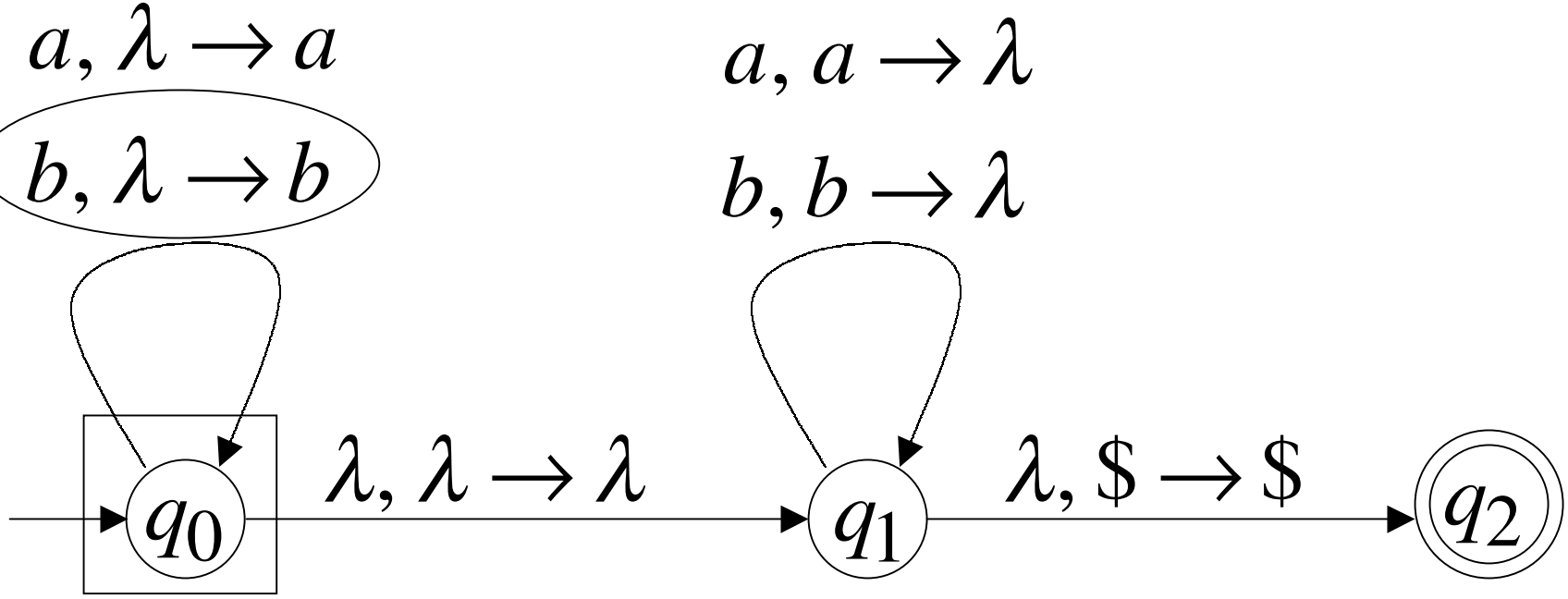
Time 2

Input



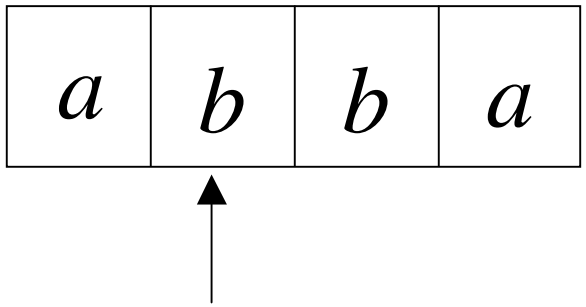


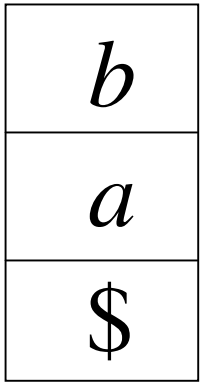
Stack



Time 3

Input





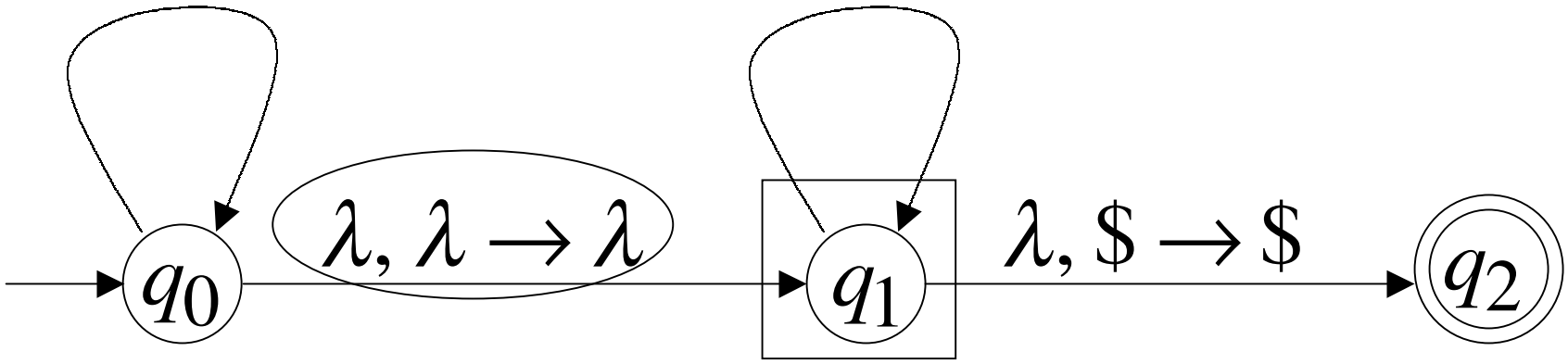
Stack

$a, \lambda \to a$

$b, \lambda \to b$

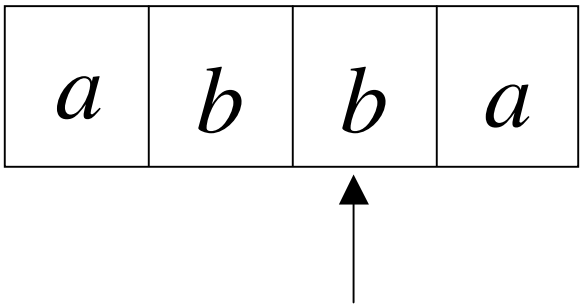
$a, a \to \lambda$

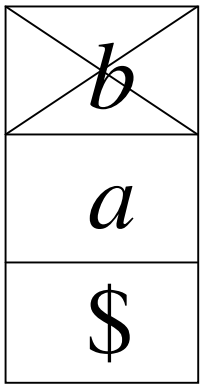
$b, b \to \lambda$



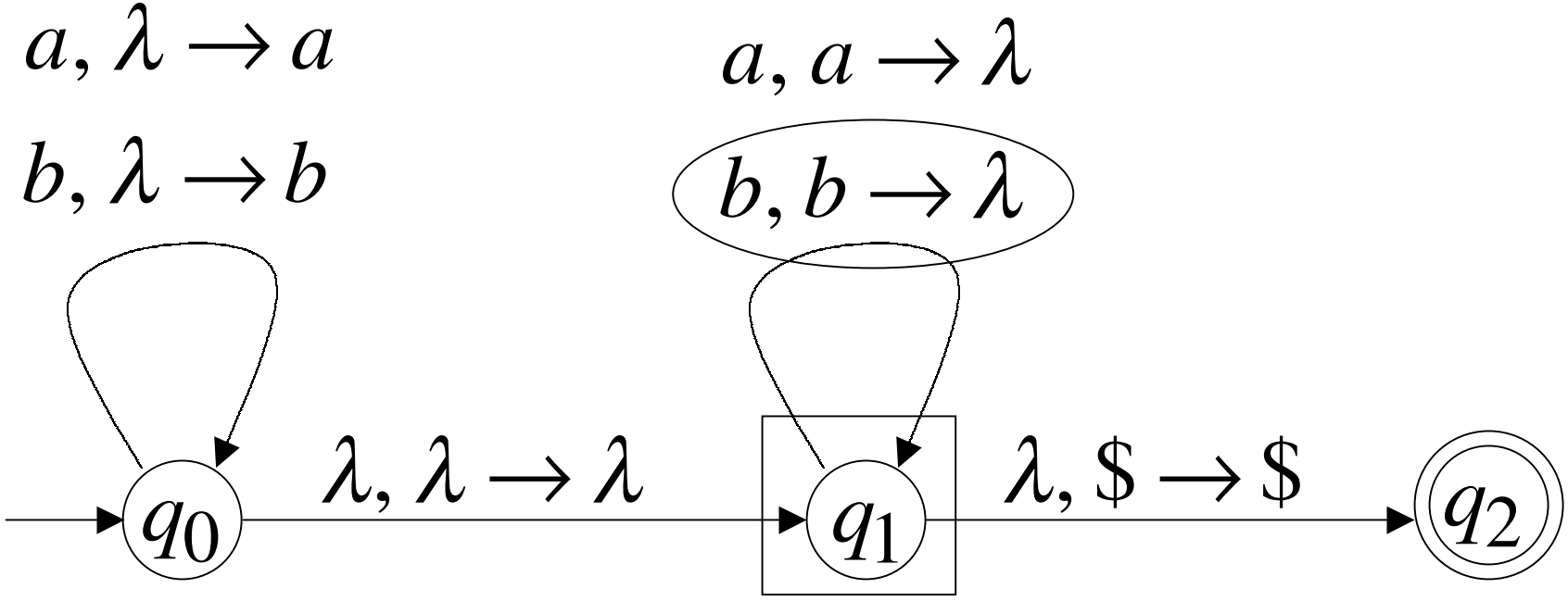
Time 4

Input



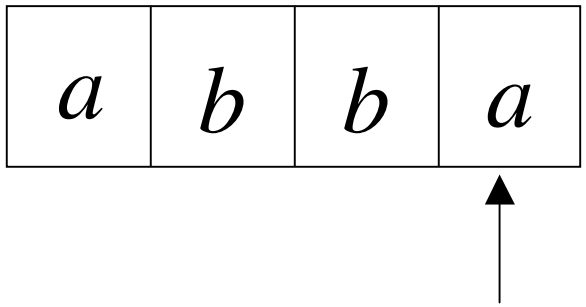


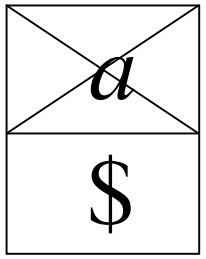
Stack



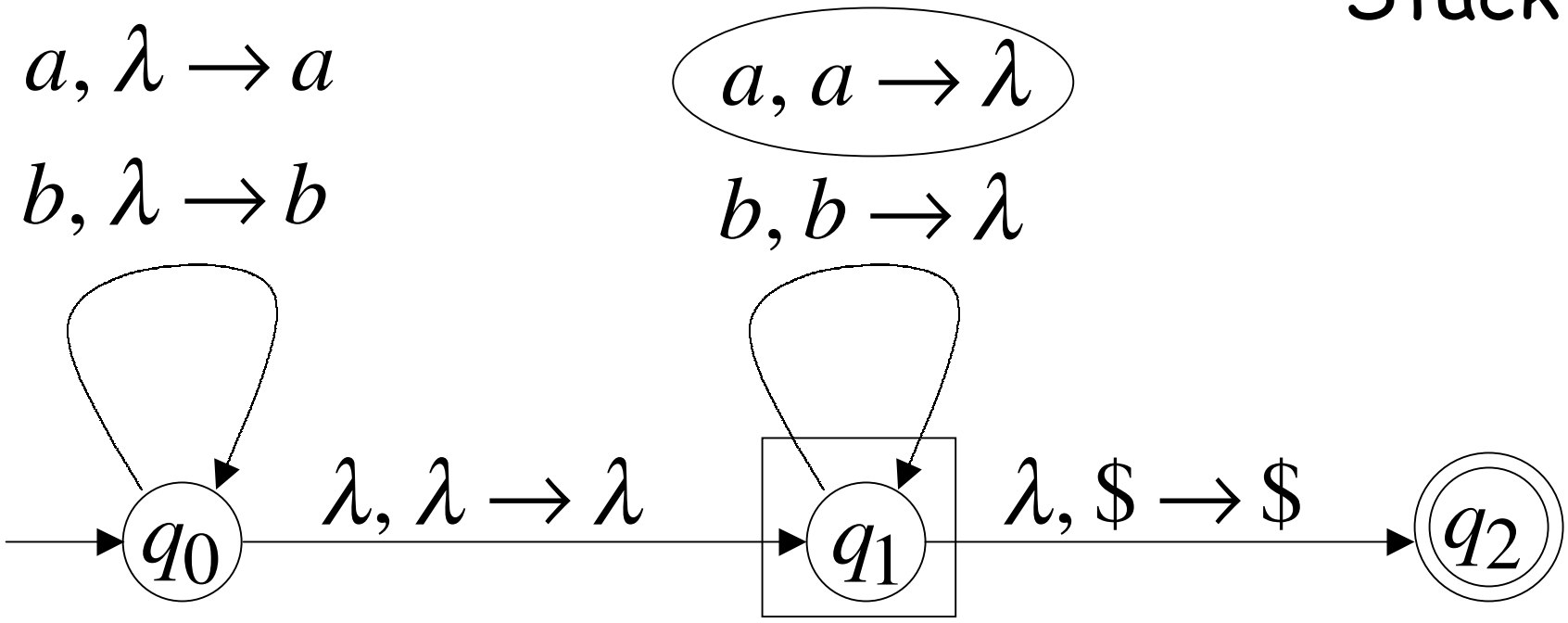
Time 5

Input



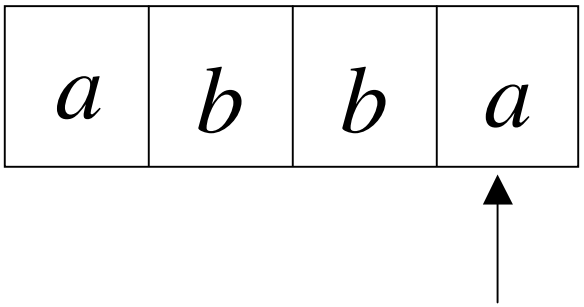


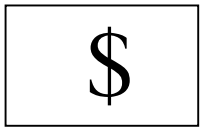
Stack



Time 6

Input





Stack

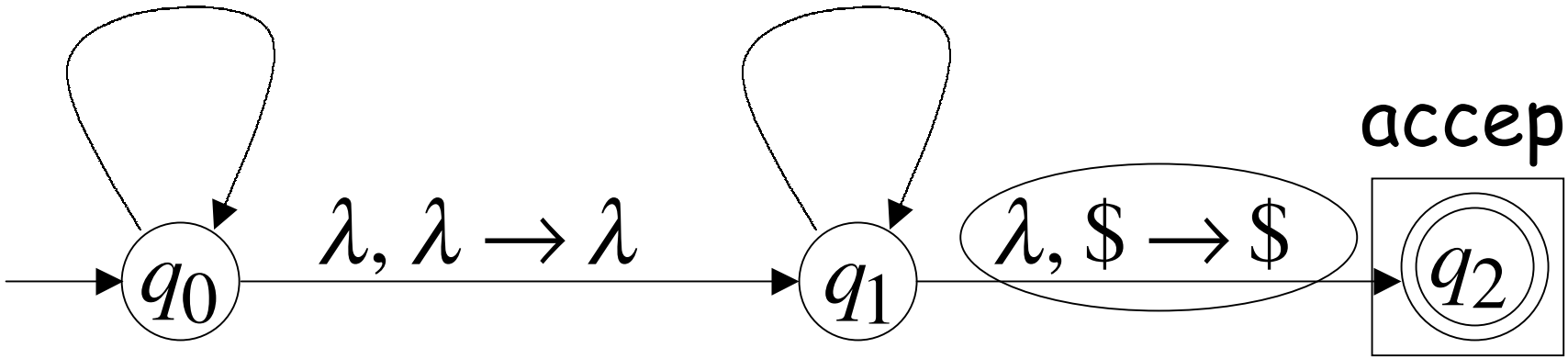
$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

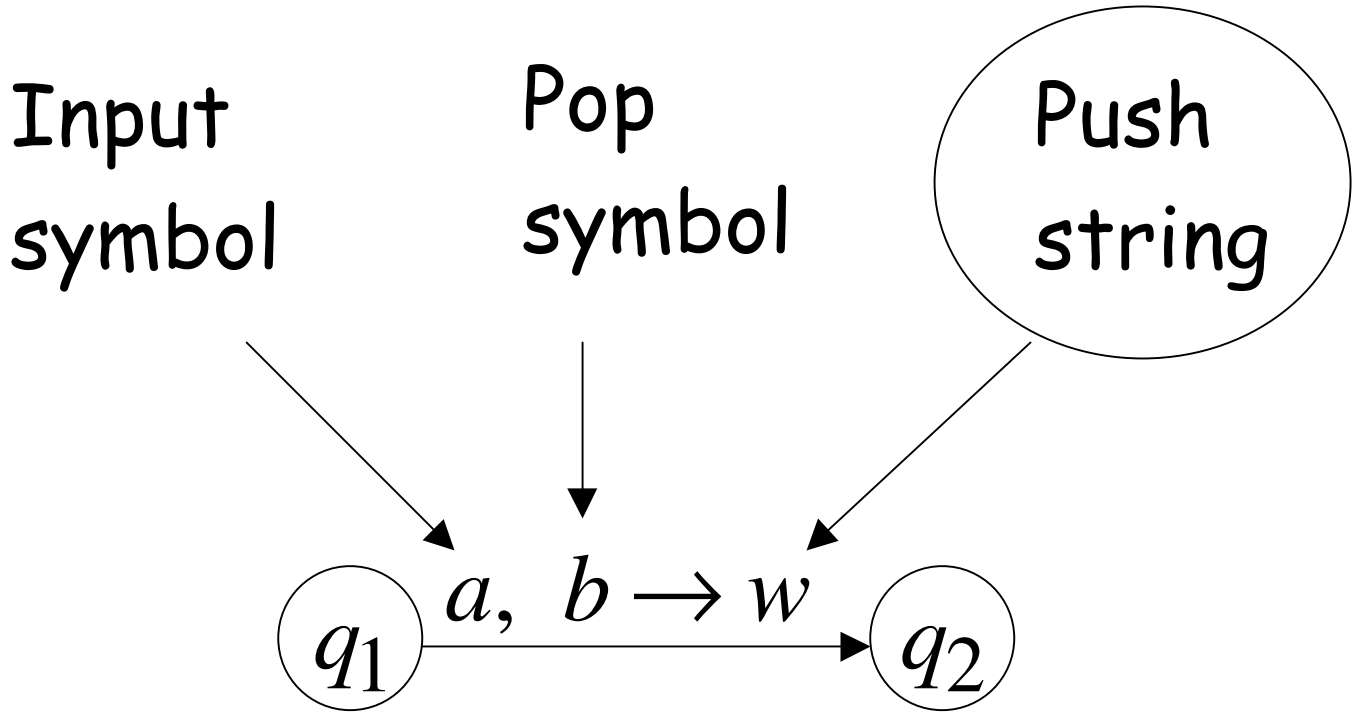
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$

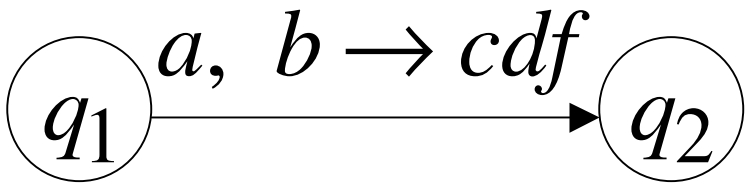
accept



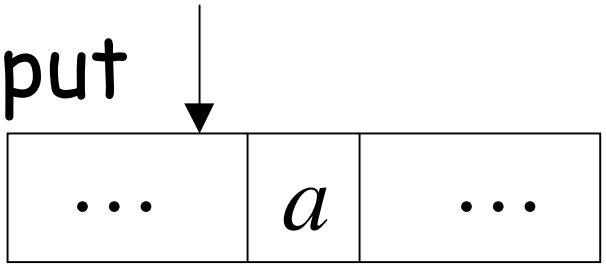
Pushing Strings

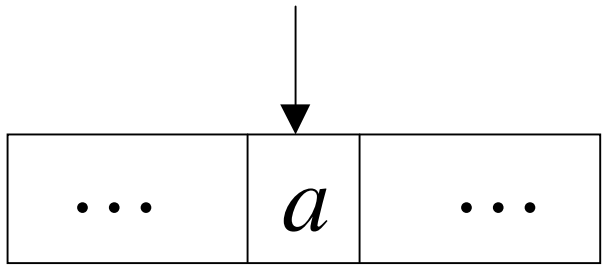


Example:

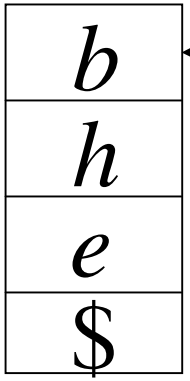


input

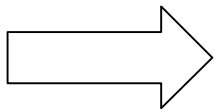


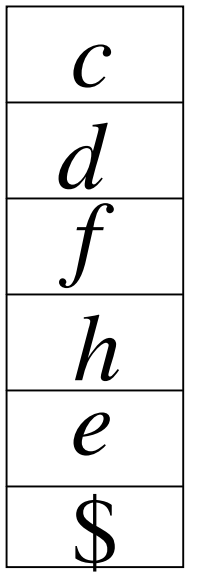


stack



top





pushed string

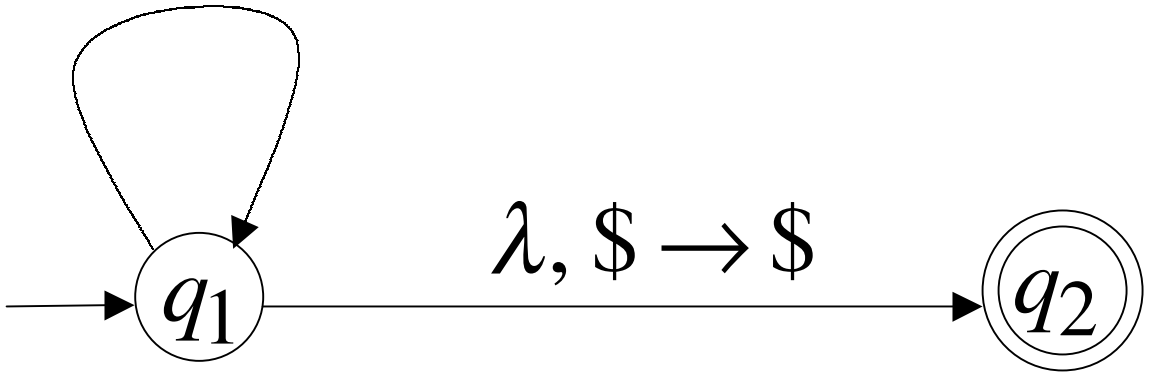
Another NPDA example

NPDA M

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

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

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

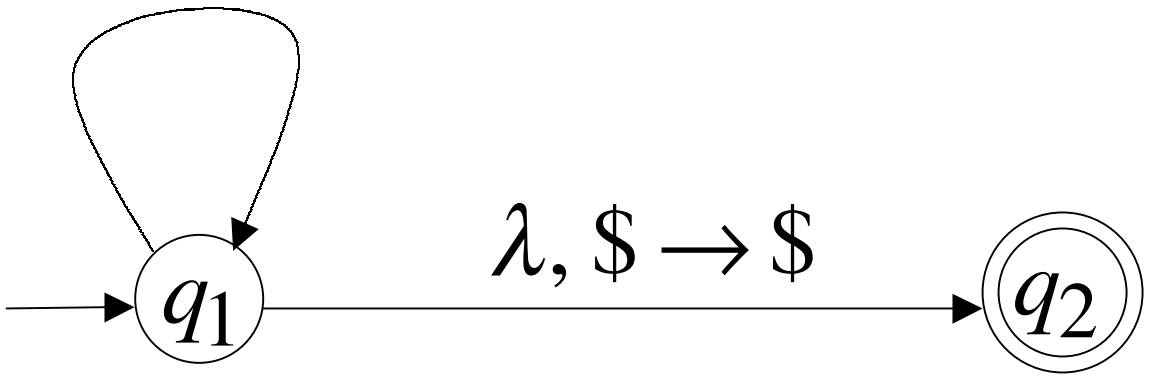


$$L(M) = \{w: \quad n_a = n_b\}$$

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

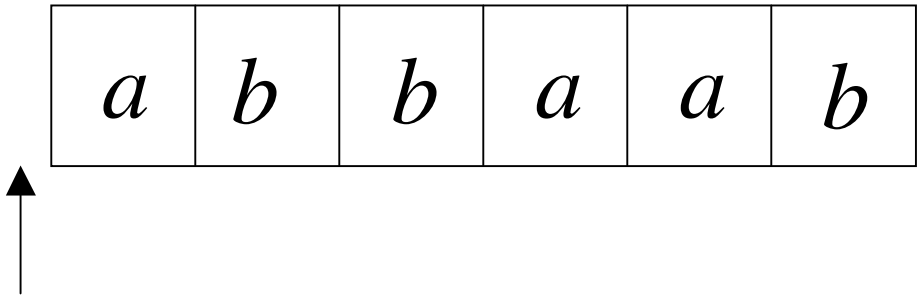
$a, 0 \rightarrow 00 \qquad b, 1 \rightarrow 11$

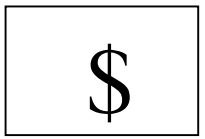
$a, 1 \rightarrow \lambda \qquad b, 0 \rightarrow \lambda$



Execution: Time 0

Input





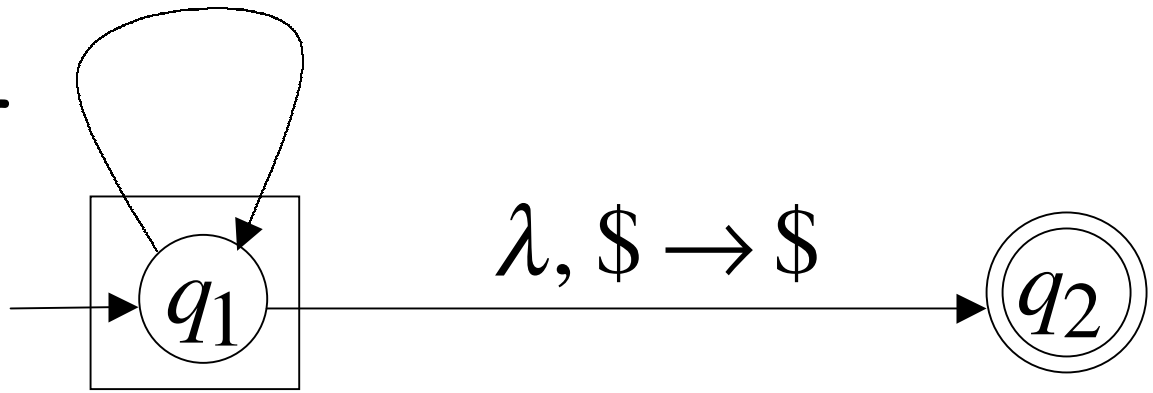
Stack

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

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

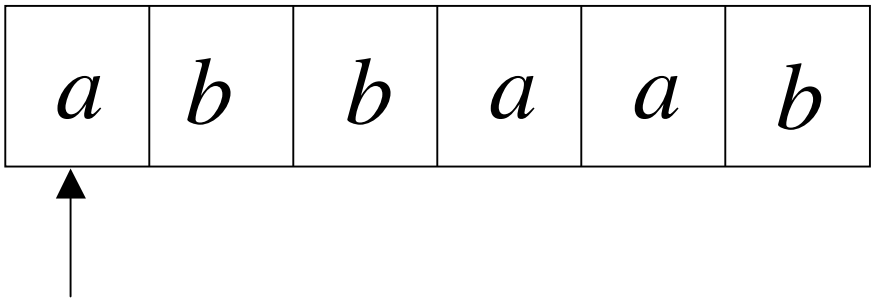
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$

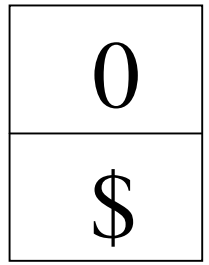
current state



Time 1

Input



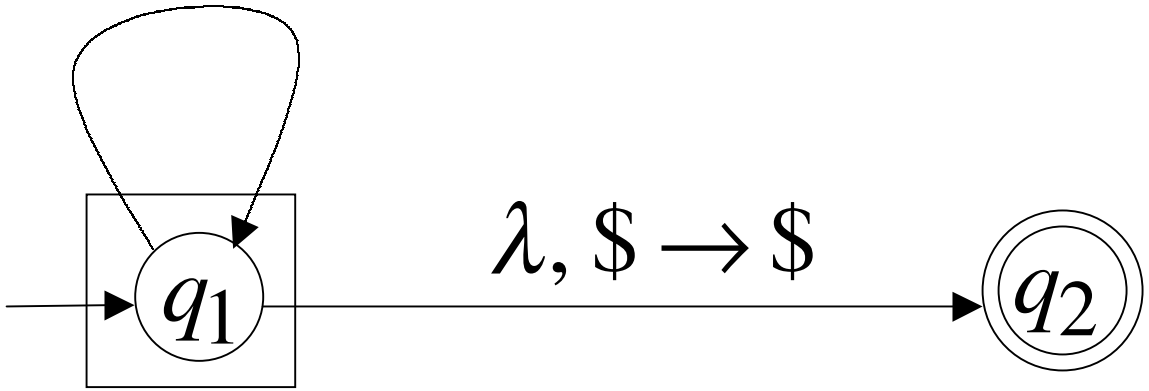


Stack

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

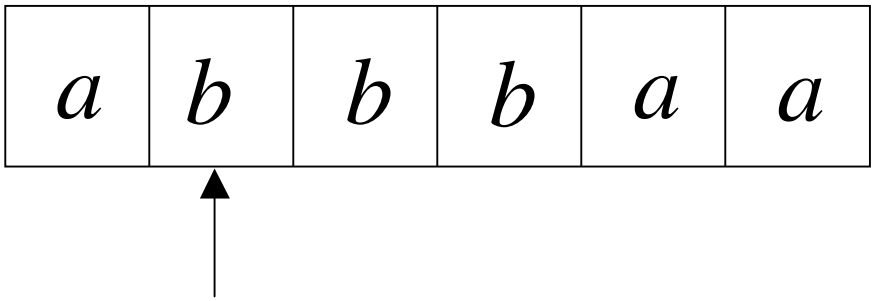
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

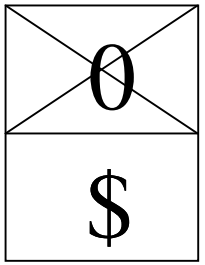
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 3

Input



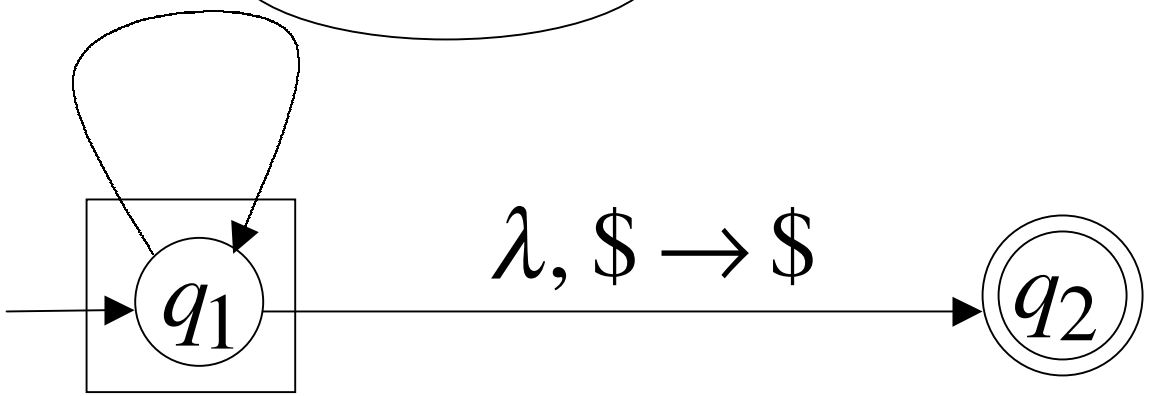


Stack

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

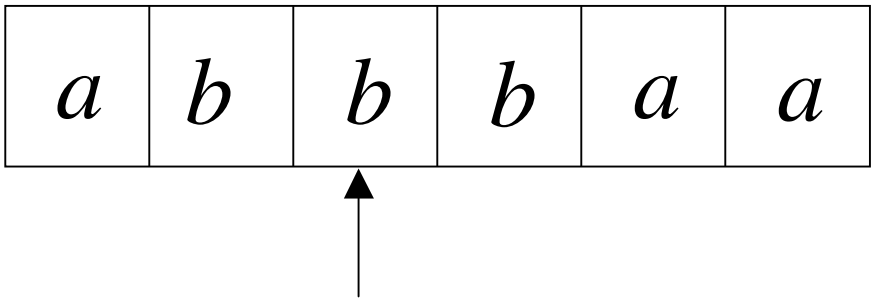
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

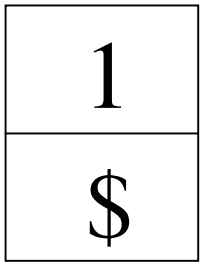
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 4

Input



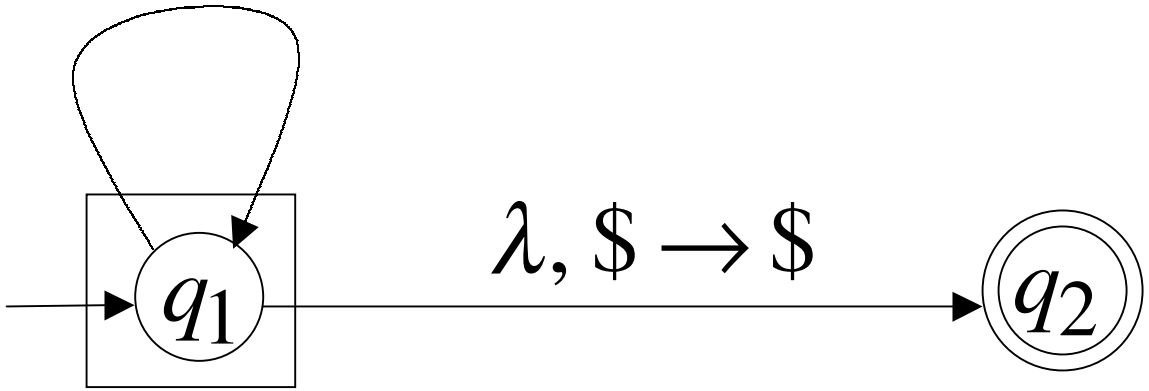


Stack

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

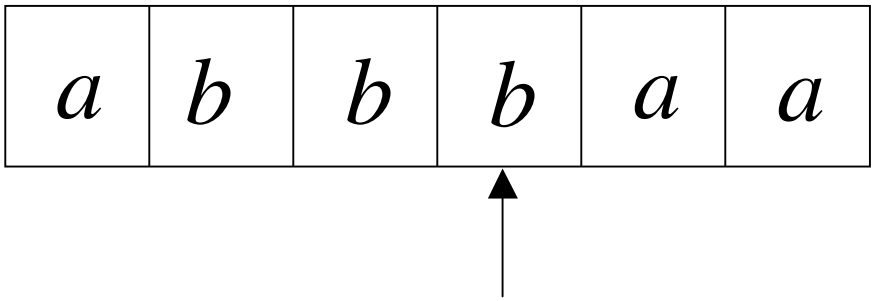
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

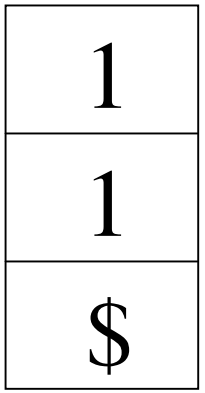
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 5

Input



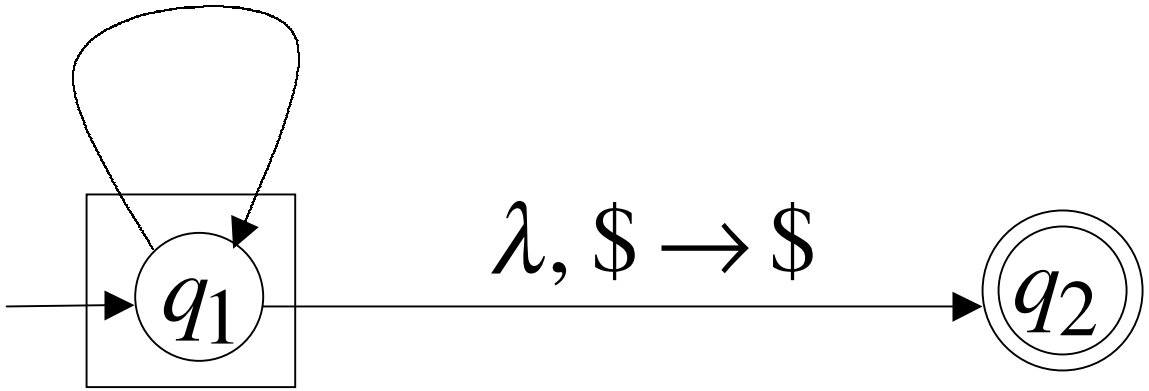


Stack

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

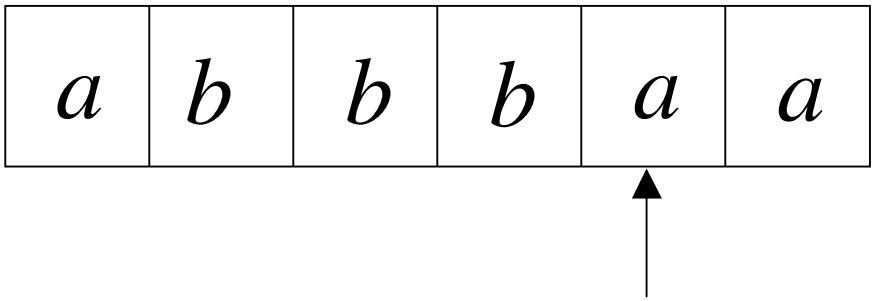
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

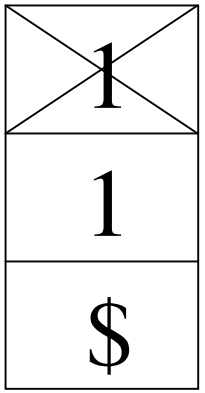
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 6

Input



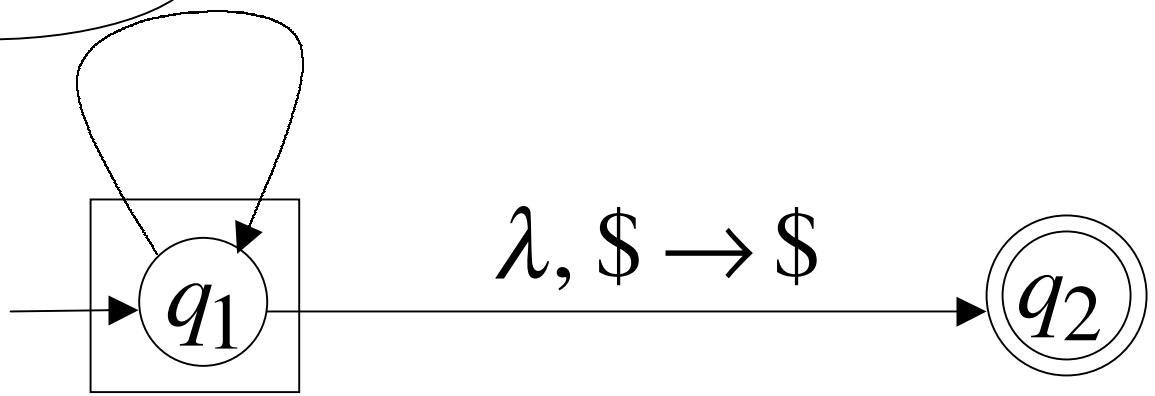


Stack

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

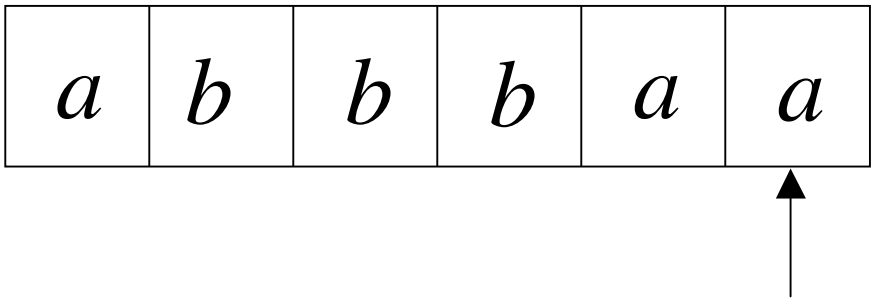
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

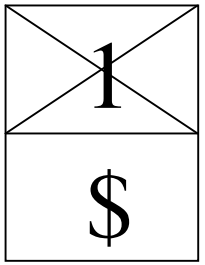
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 7

Input



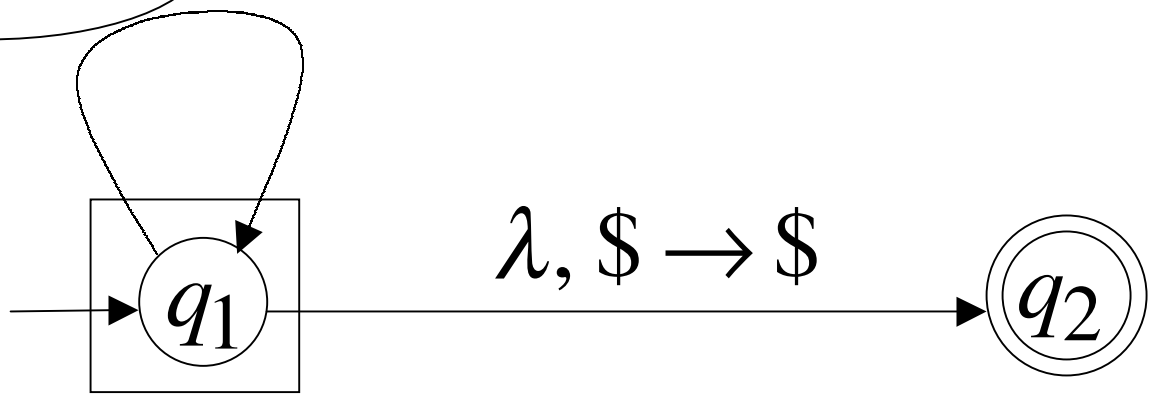


Stack

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

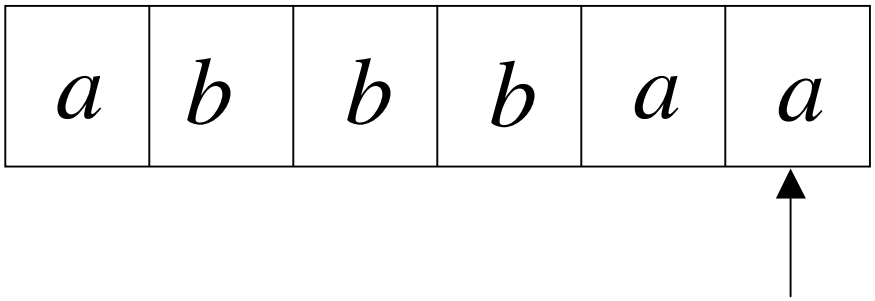
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

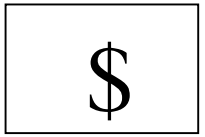
$a, 1 \rightarrow \lambda$ $b, 0 \rightarrow \lambda$



Time 8

Input



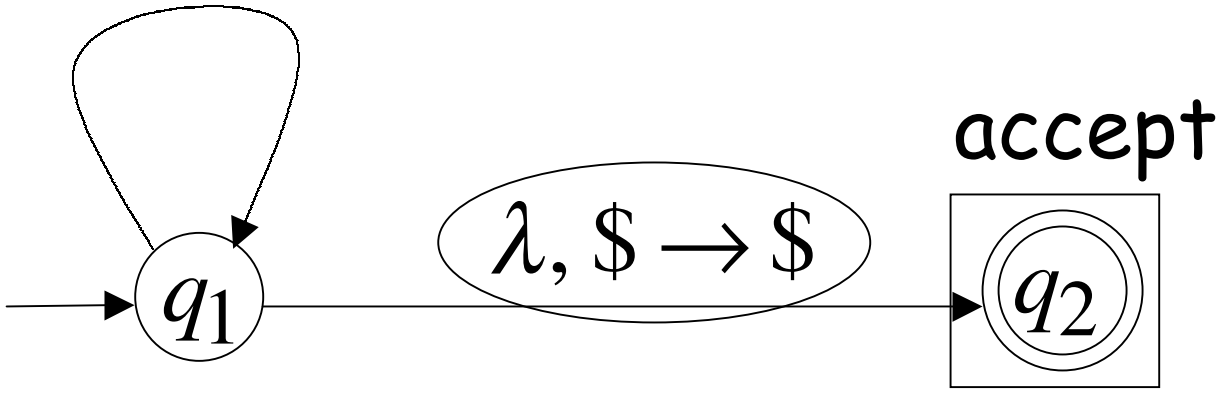


Stack

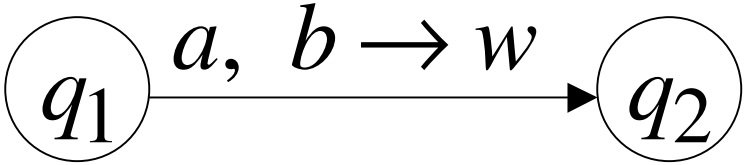
$a, \$ \rightarrow 0\$$ $b, \$ \rightarrow 1\$$

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

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

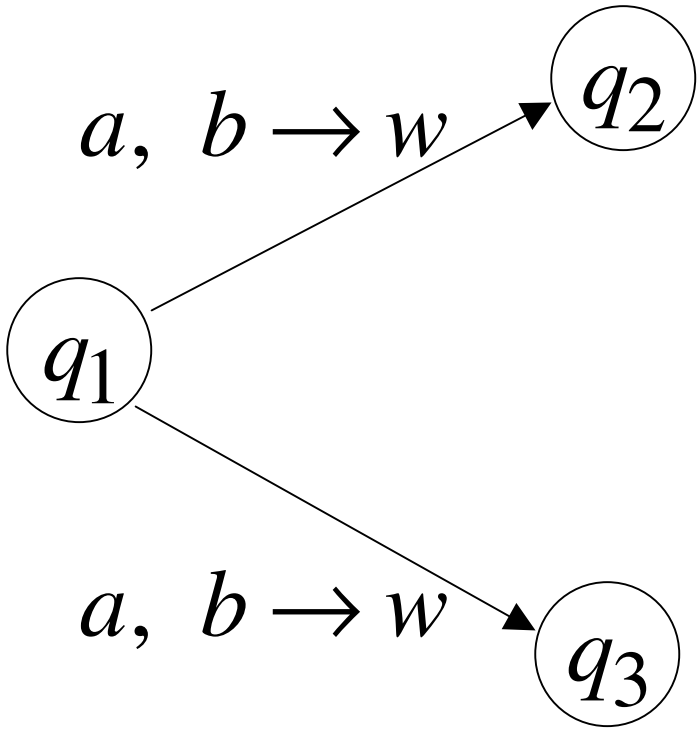


Formalities for NPDAs



Transition function:

$$\delta(q_0, a, b) = \{(q_2, w)\}$$



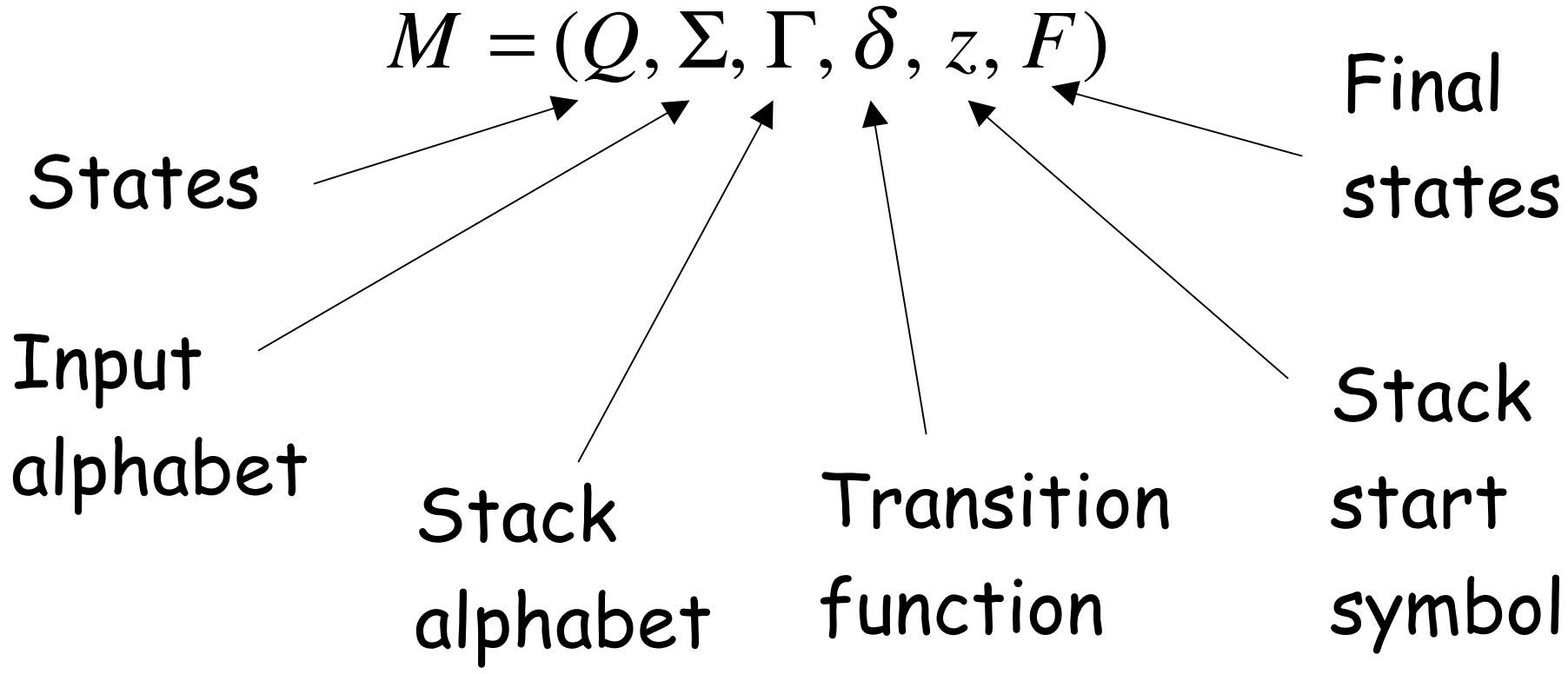
Transition function:

$$\delta(q_0, a, b) = \{(q_2, w),\ (q_3, w)\}$$

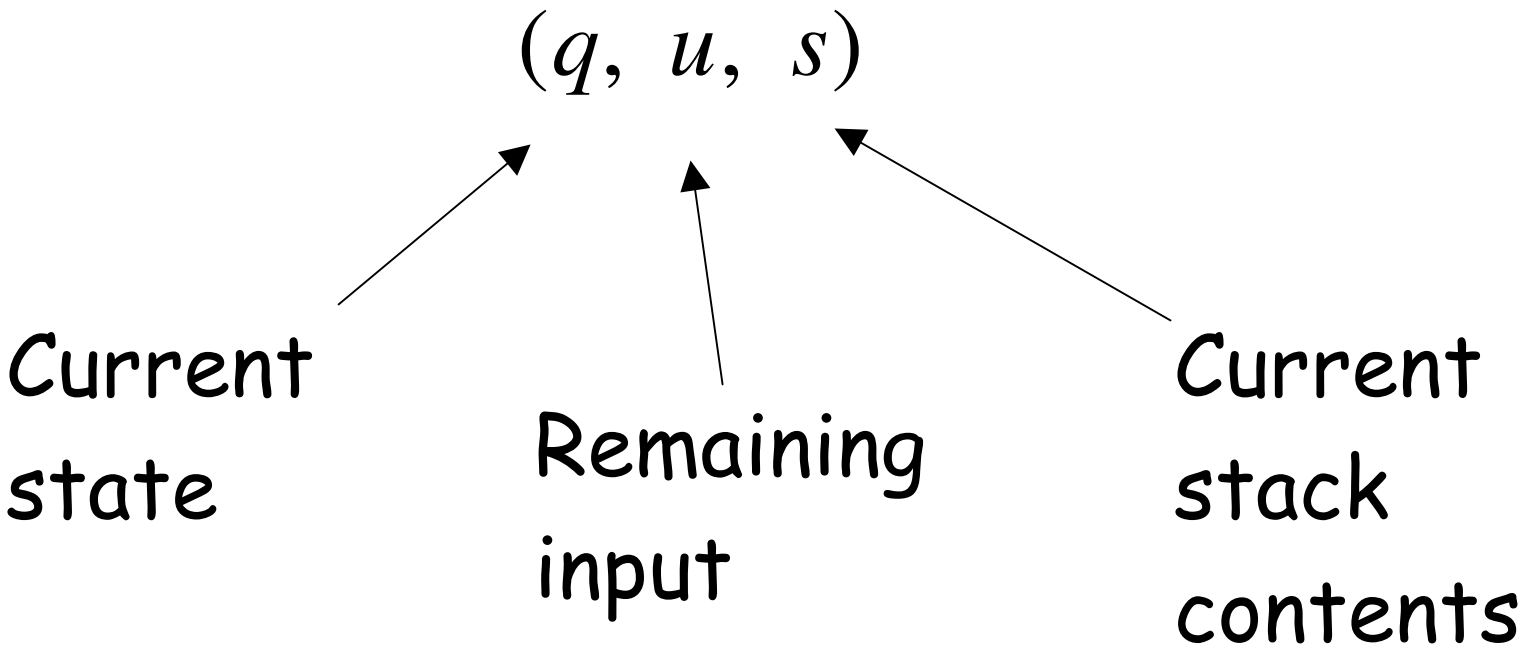
Formal Definition

Non-Deterministic Pushdown Automaton

NPDA



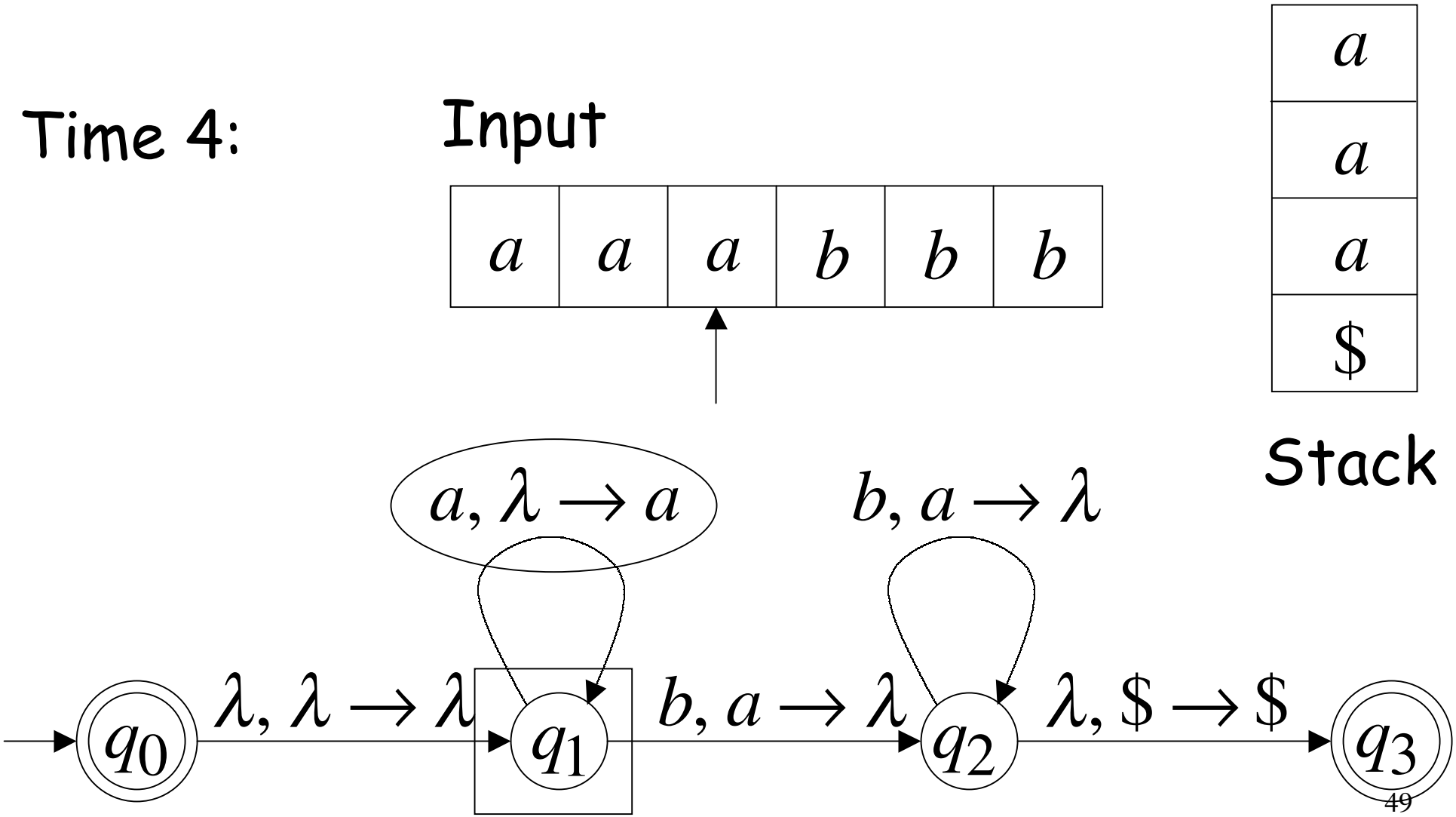
Instantaneous Description



Example: Instantaneous Description

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

Time 4:

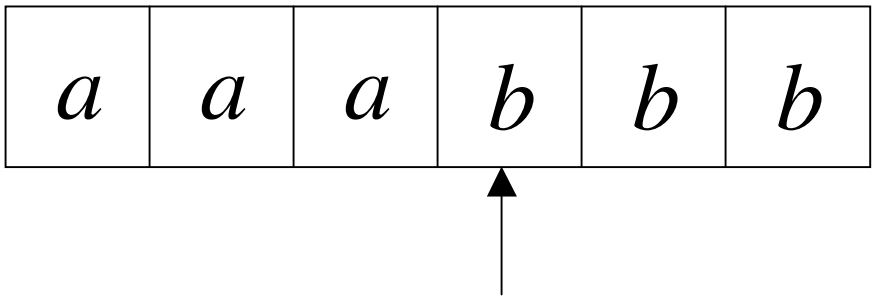


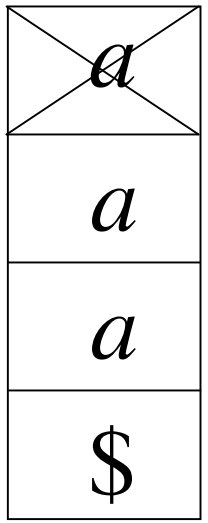
Example: Instantaneous Description

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

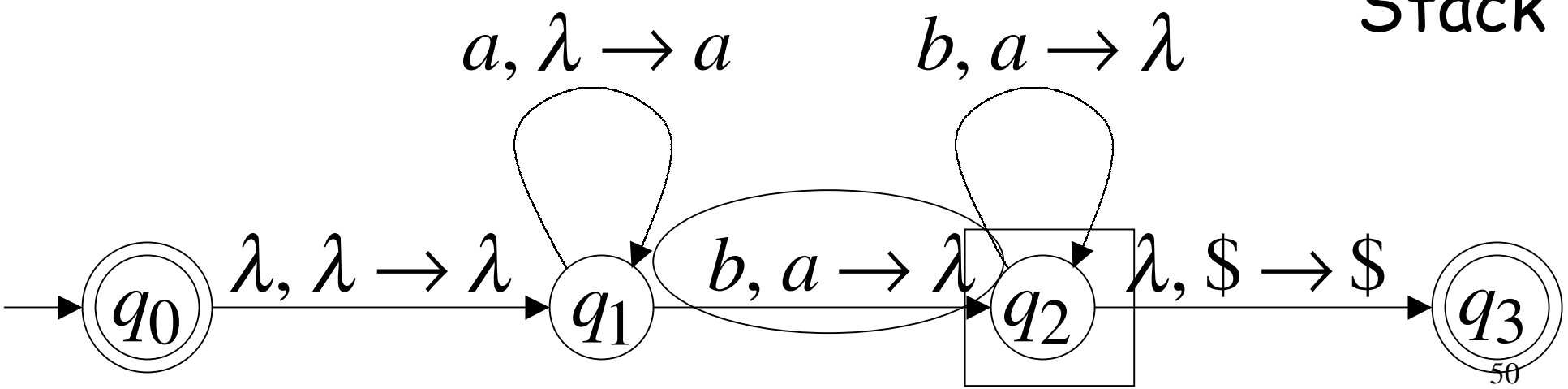
Time 5:

Input





Stack



We write:

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

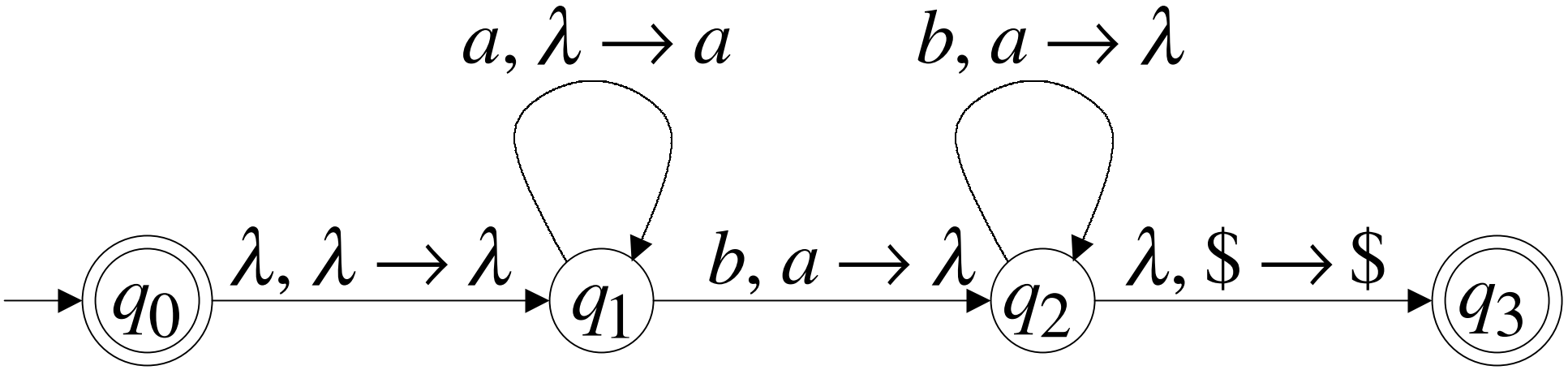
(Time 4) (Time 5)

A computation:

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

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

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



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

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

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

For convenience we write:

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

Formal Definition

Language of NPDA M:

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

Start state

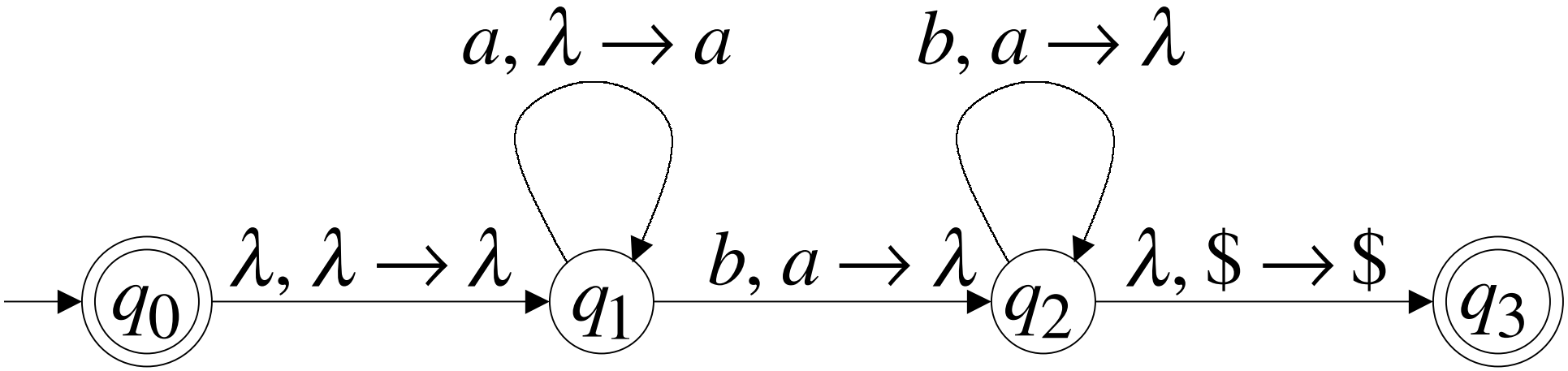
Final state

Example:

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

thus $aaabbb \in L(M)$

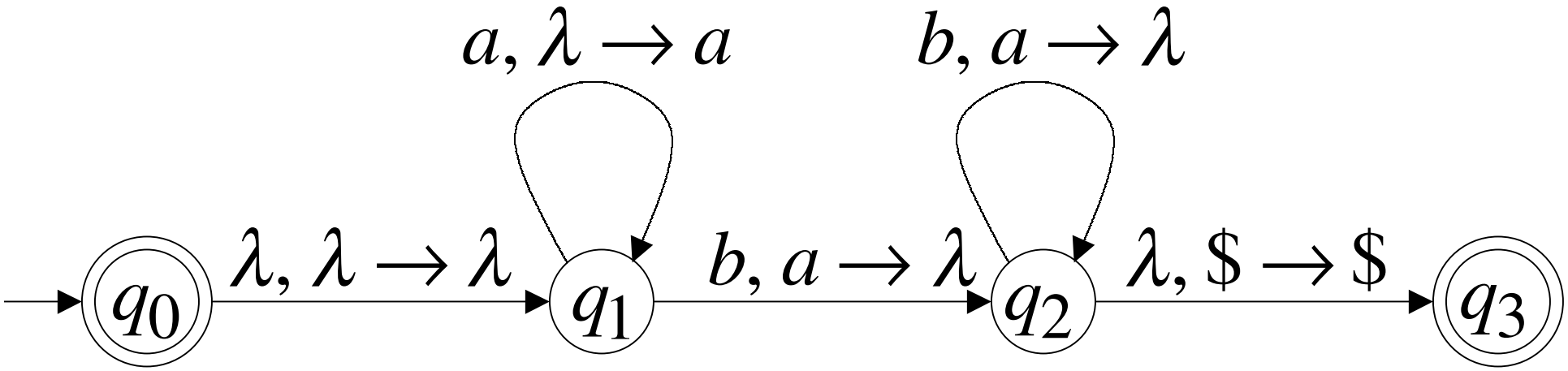
NPDA M :



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

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

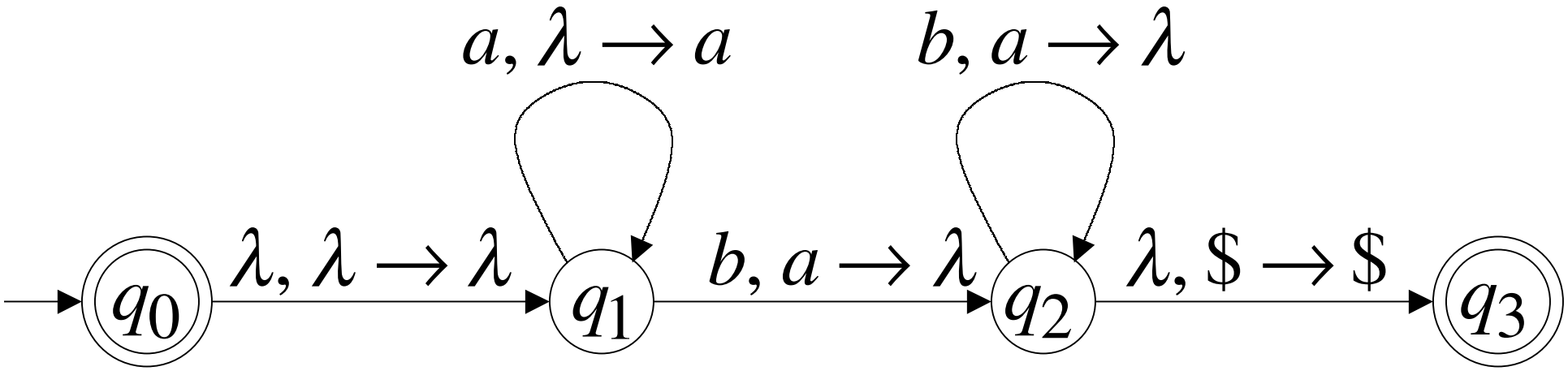
NPDA M :



Therefore:

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

NPDA M :



Converting Context-Free Grammars to NPDAs

An example grammar:

$$S \to aSTb$$

$$S \to b$$

$$T \to Ta$$

$$T \to \lambda$$

What is the equivalent NPDA?

Grammar:

$S \to aSTb$

$S \to b$

$T \to Ta$

$T \to \lambda$

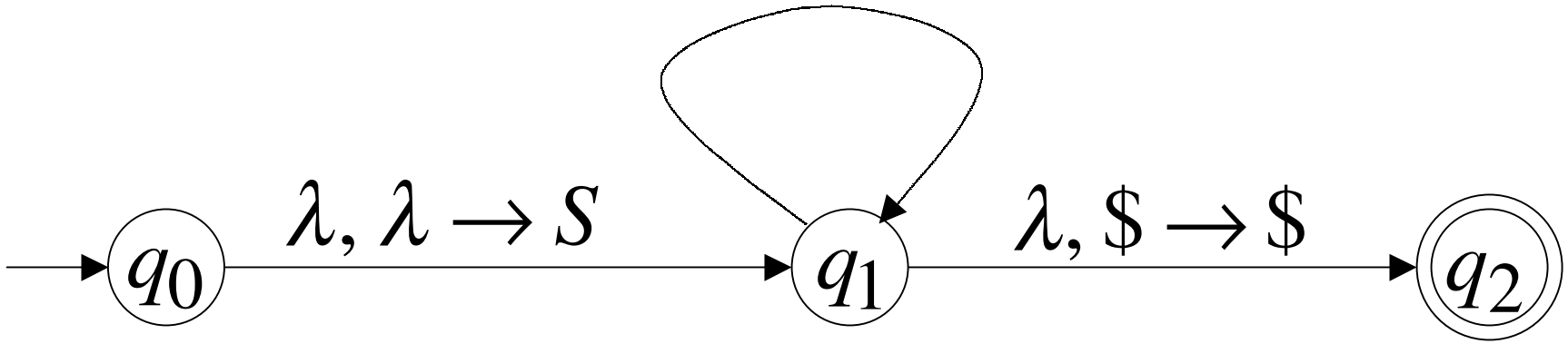
NPDA:

$\lambda, S \to aSTb$

$\lambda, S \to b$

$\lambda, T \to Ta$ $\quad a, a \to \lambda$

$\lambda, T \to \lambda$ $\quad b, b \to \lambda$



The NPDA simulates
leftmost derivations of the grammar

L(grammar) = L(NPDA)

Grammar:

$$S \to aSTb$$

$$S \to b$$

$$T \to Ta$$

$$T \to \lambda$$

A leftmost derivation:

$$S \Rightarrow aSTb \Rightarrow abTb \Rightarrow abTab \Rightarrow abab$$

NPDA execution: Time 0

Input

a	b	a	b

$\lambda, S \to aSTb$

$\lambda, S \to b$

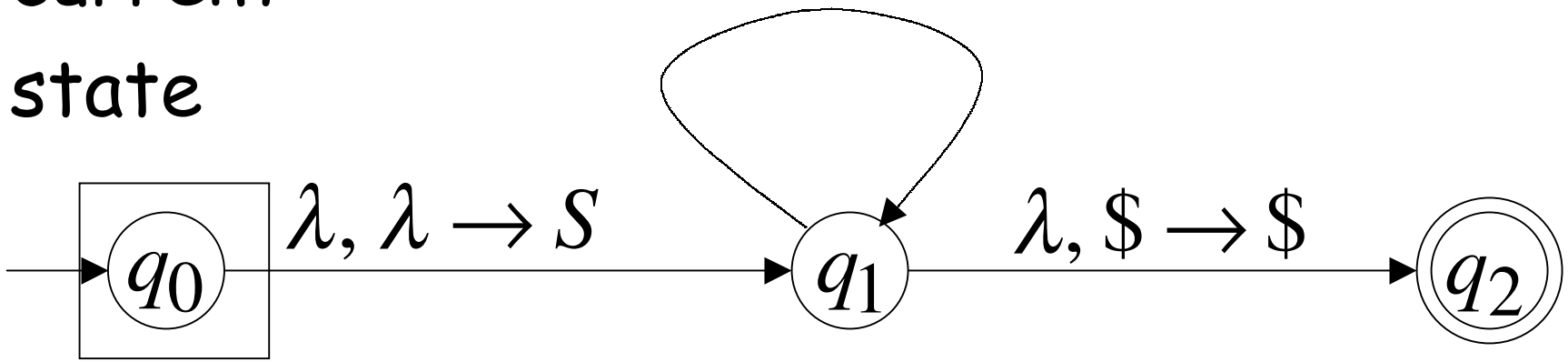
$\lambda, T \to Ta$ $\quad a, a \to \lambda$

$\lambda, T \to \lambda$ $\quad b, b \to \lambda$

$

Stack

current state



Time 1

Input

a	b	a	b

S
\$

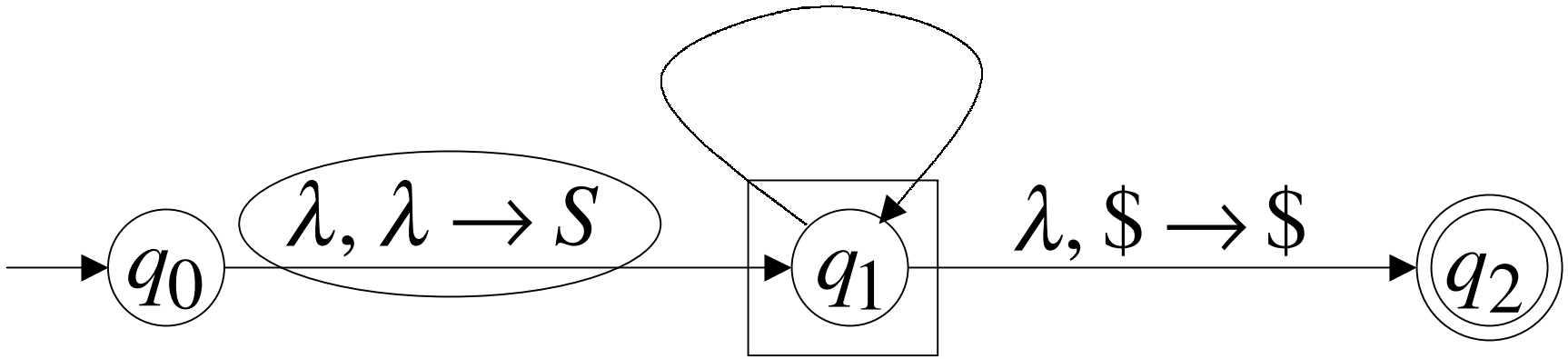
Stack

$\lambda, S \to aSTb$

$\lambda, S \to b$

$\lambda, T \to Ta$ $\quad a, a \to \lambda$

$\lambda, T \to \lambda$ $\quad b, b \to \lambda$



Time 2

Input

a	b	a	b

a
S
T
b
\$

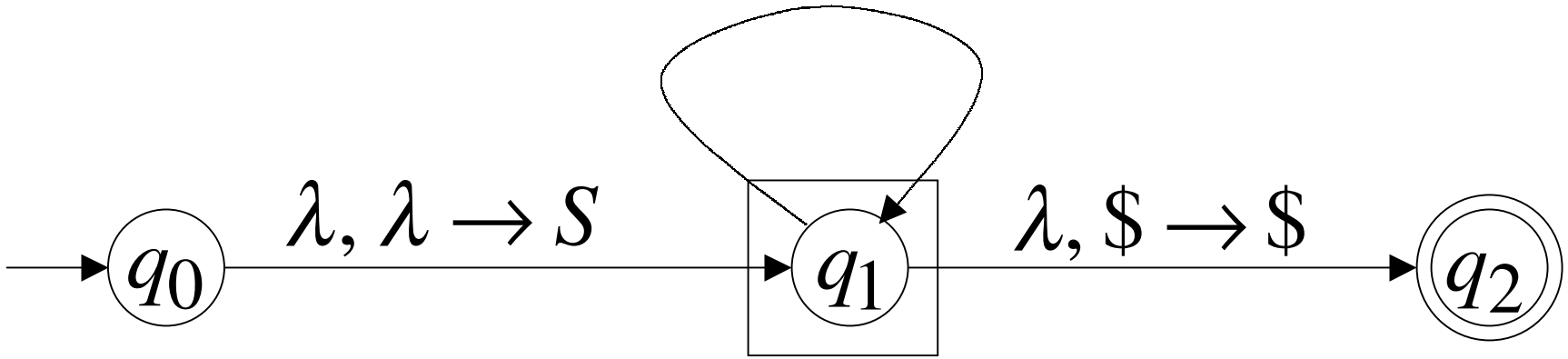
Stack

$\lambda, S \to aSTb$

$\lambda, S \to b$

$\lambda, T \to Ta$ $\quad a, a \to \lambda$

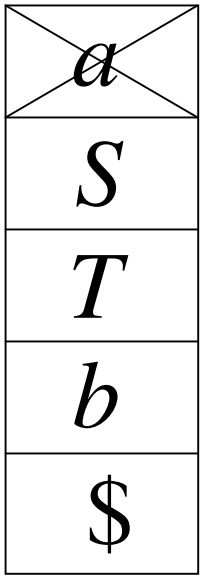
$\lambda, T \to \lambda$ $\quad b, b \to \lambda$



Time 3

Input

a	b	a	b



Stack

$\lambda, S \to aSTb$

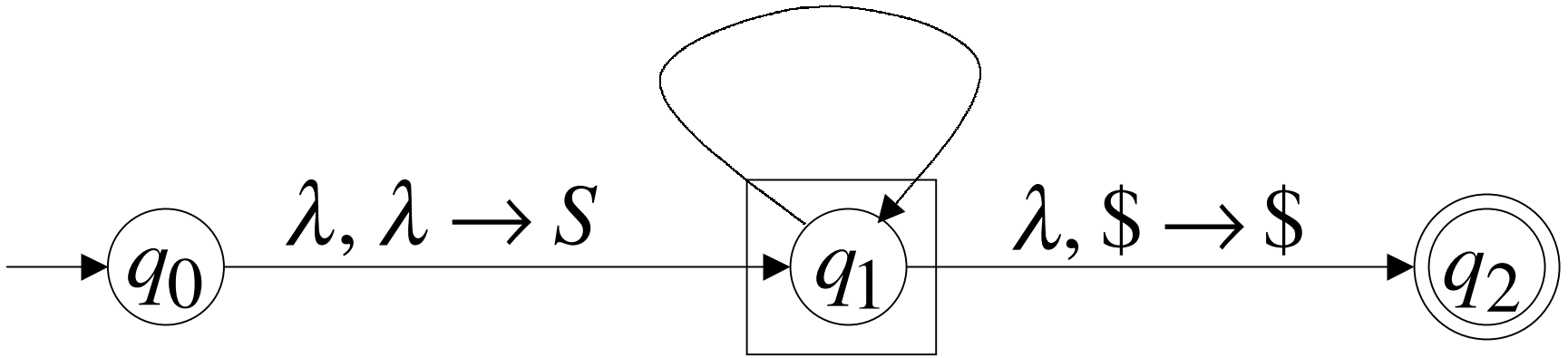
$\lambda, S \to b$

$\lambda, T \to Ta$

$\lambda, T \to \lambda$

$a, a \to \lambda$

$b, b \to \lambda$



Time 4

Input

a	b	a	b

b
T
b
\$

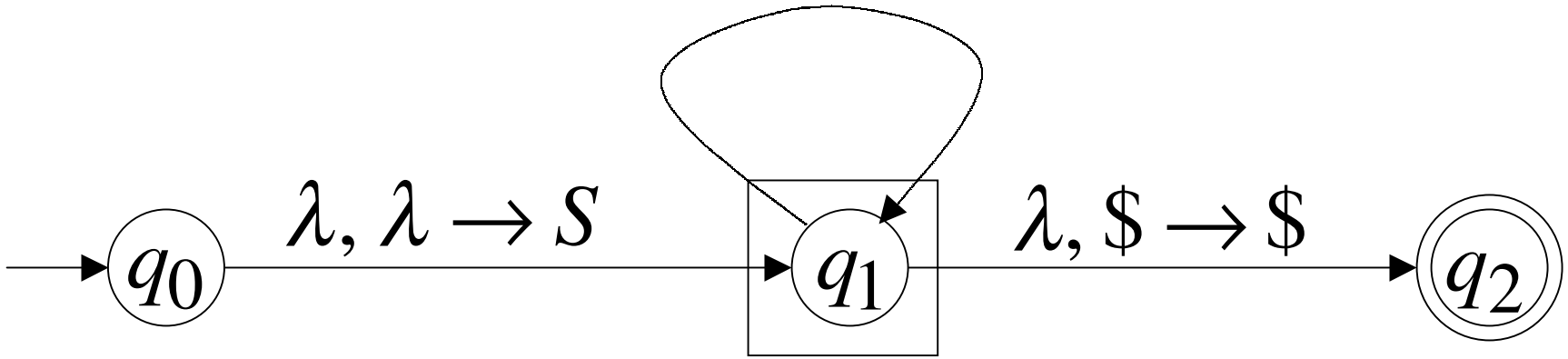
Stack

$\lambda, S \to aSTb$

$\lambda, S \to b$

$\lambda, T \to Ta$ $\quad a, a \to \lambda$

$\lambda, T \to \lambda$ $\quad b, b \to \lambda$



Time 5

Input

a	b	a	b

~~b~~
T
b
\$

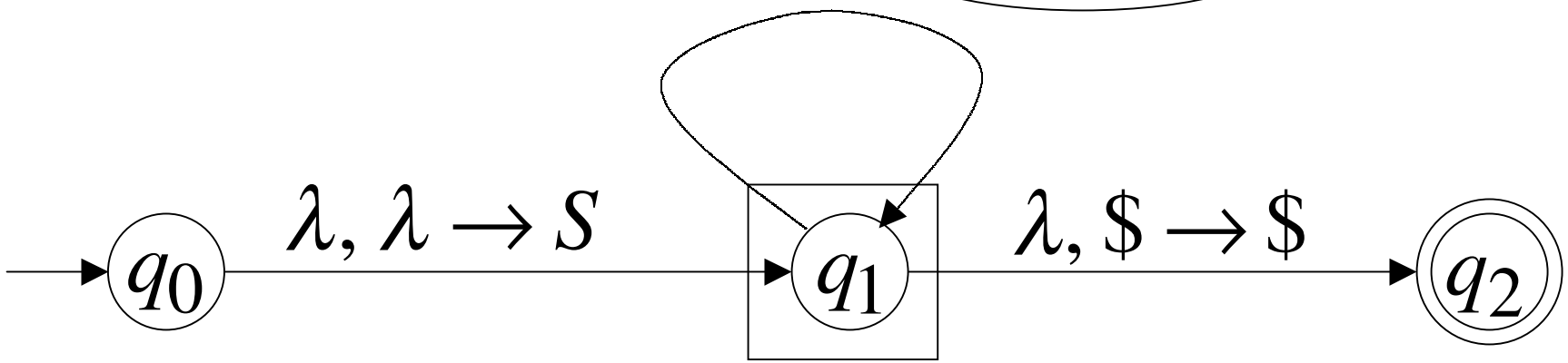
Stack

$\lambda, S \to aSTb$

$\lambda, S \to b$

$\lambda, T \to Ta$ $\quad a, a \to \lambda$

$\lambda, T \to \lambda$ $\quad b, b \to \lambda$



Time 6

Input

a	b	a	b

T
a
b
\$

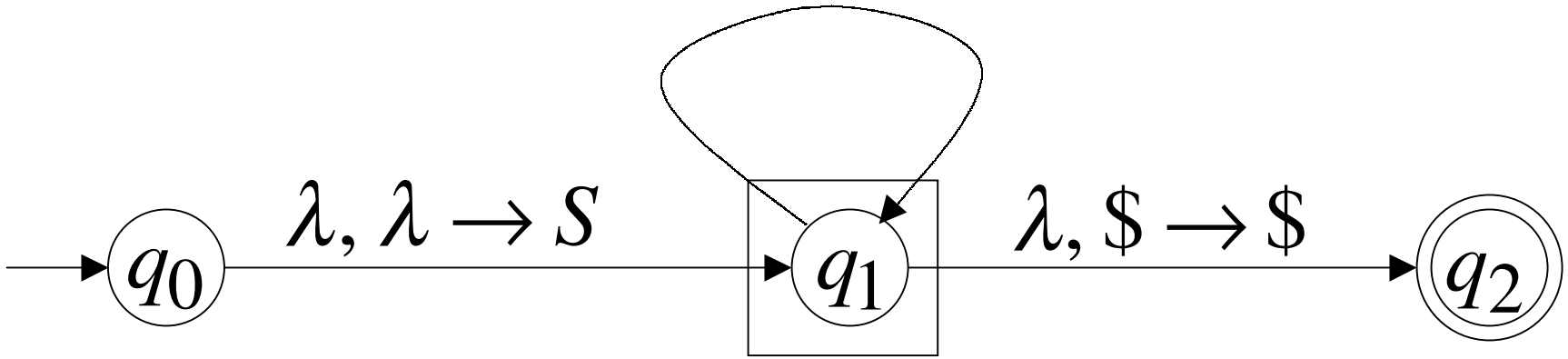
Stack

$\lambda, S \to aSTb$

$\lambda, S \to b$

$\lambda, T \to Ta$ $\quad a, a \to \lambda$

$\lambda, T \to \lambda$ $\quad b, b \to \lambda$



Time 7

Input

a	b	a	b

~~T~~
a
b
\$

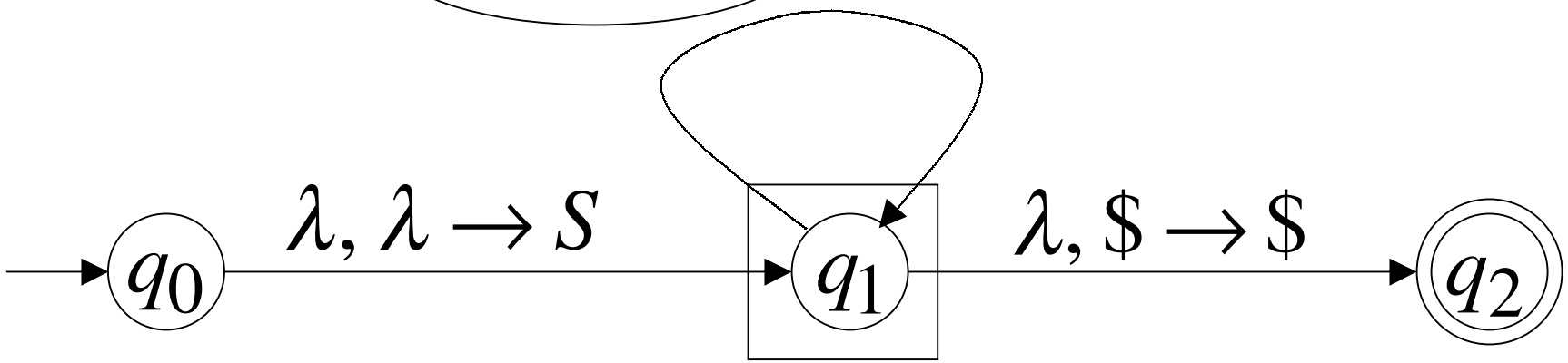
Stack

$\lambda, S \to aSTb$

$\lambda, S \to b$

$\lambda, T \to Ta$ $\quad a, a \to \lambda$

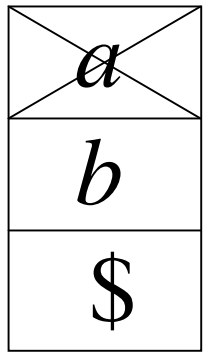
$\lambda, T \to \lambda$ $\quad b, b \to \lambda$



Time 8

Input

a	b	a	b



Stack

$\lambda, S \to aSTb$

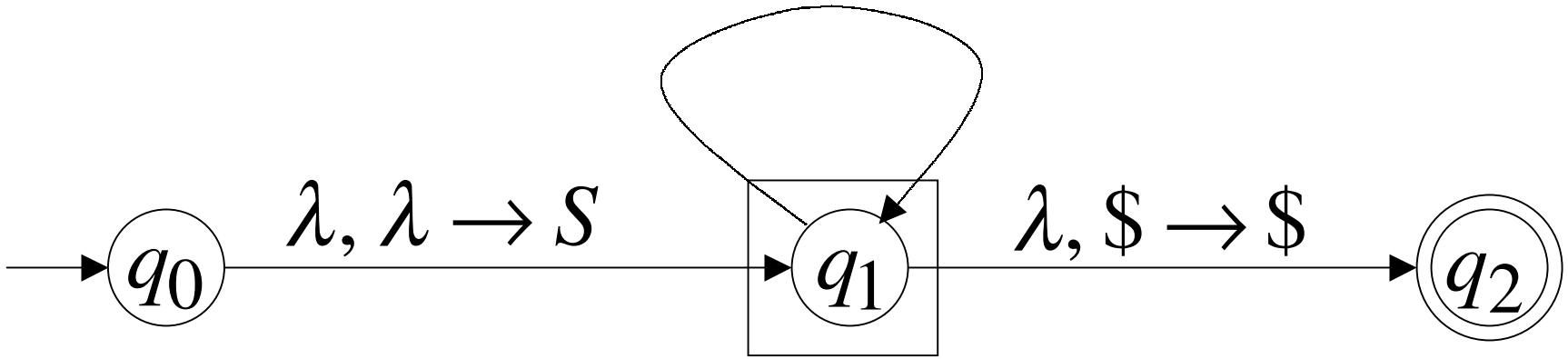
$\lambda, S \to b$

$\lambda, T \to Ta$

$\lambda, T \to \lambda$

$a, a \to \lambda$

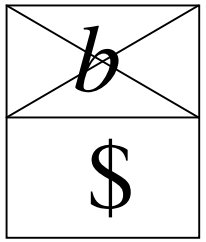
$b, b \to \lambda$



Time 9

Input

a	b	a	b



Stack

$\lambda, S \to aSTb$

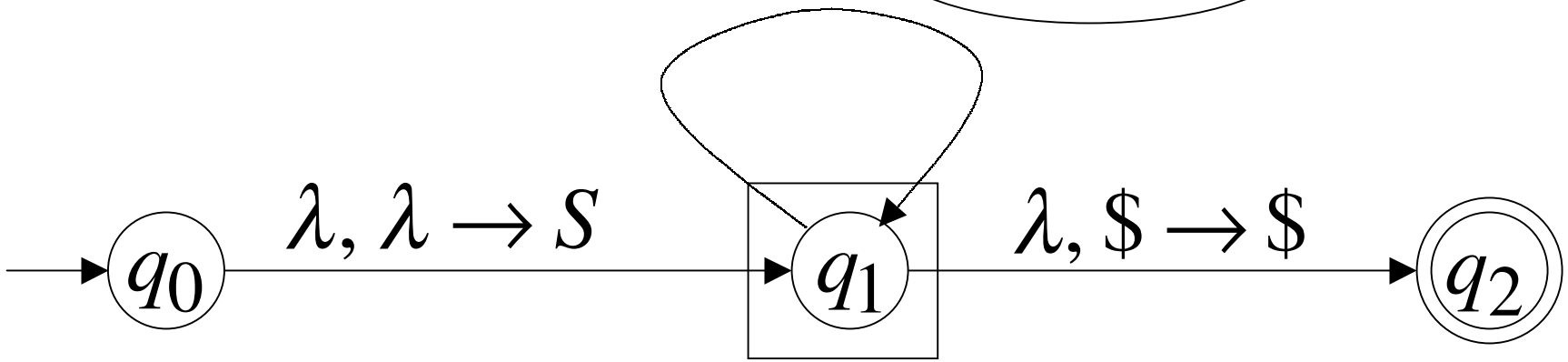
$\lambda, S \to b$

$\lambda, T \to Ta$

$\lambda, T \to \lambda$

$a, a \to \lambda$

$b, b \to \lambda$



Time 10

Input

a	b	a	b

$\lambda, S \to aSTb$

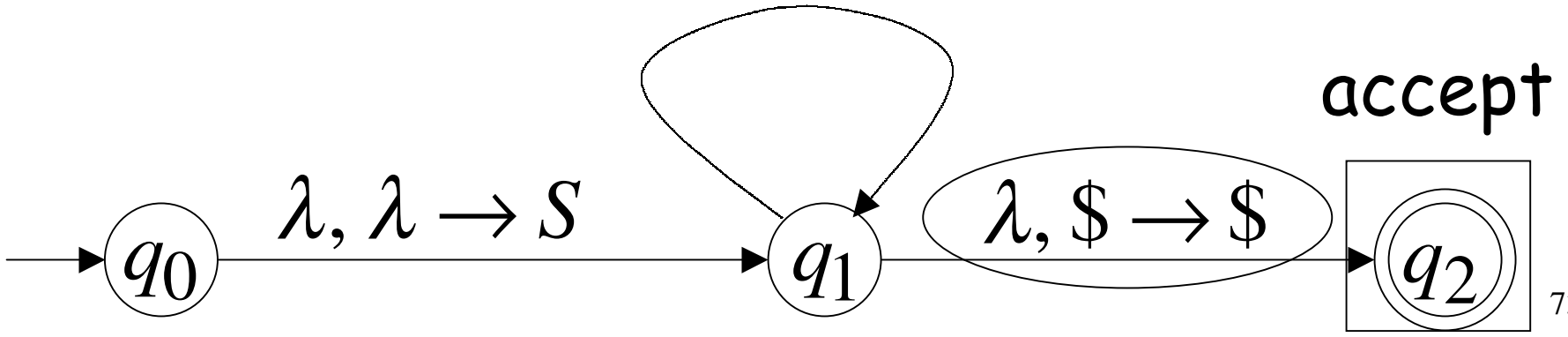
$\lambda, S \to b$

$\lambda, T \to Ta$ $\quad a, a \to \lambda$

$\lambda, T \to \lambda$ $\quad b, b \to \lambda$

$

Stack



In general:

Given any grammar G

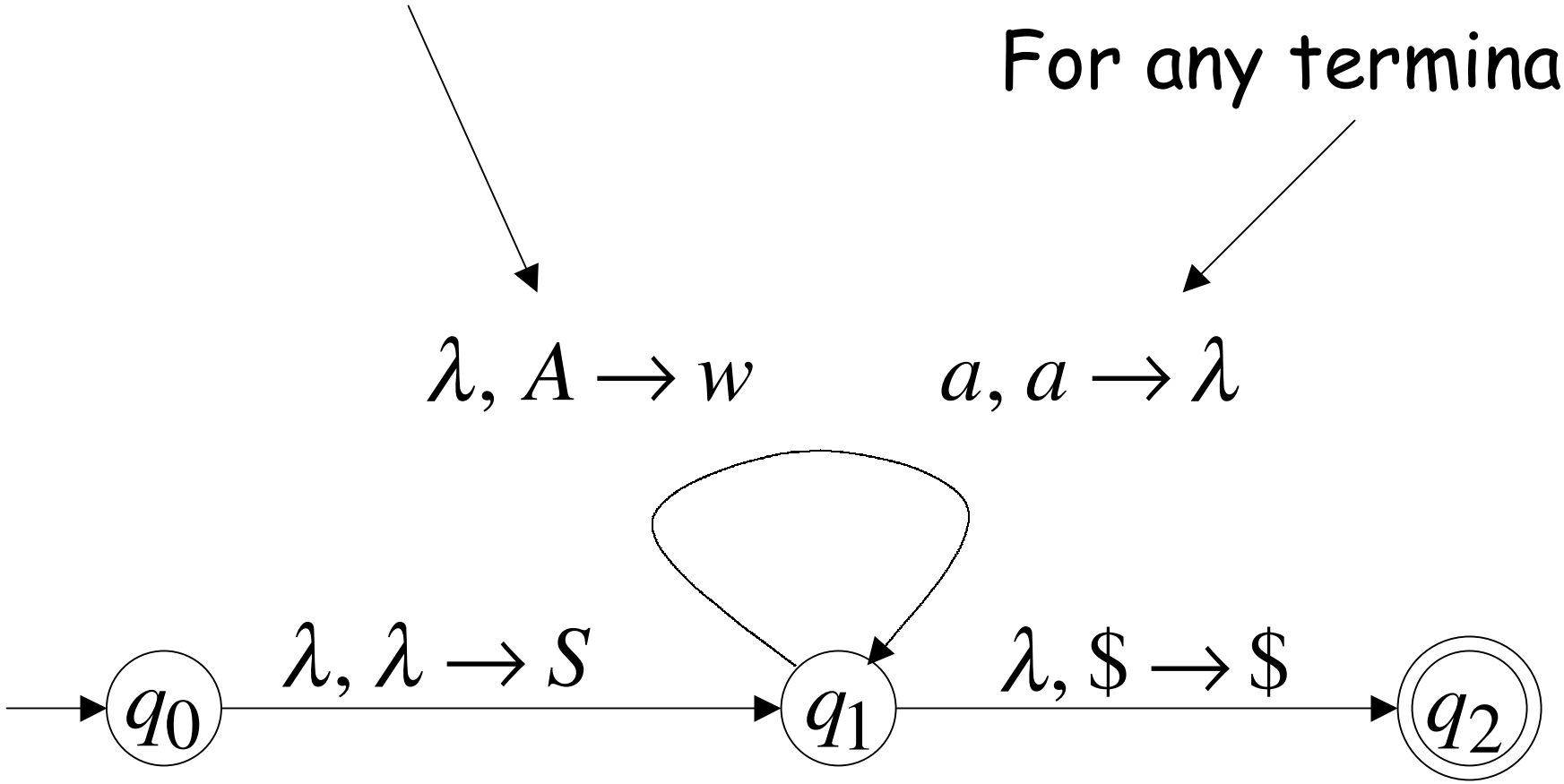
We can construct a NPDA M

With $L(G) = L(M)$

Constructing NPDA M from grammar G :

For any production $A \to w$

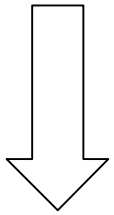
For any terminal a



Grammar G generates string w

if and only if

NPDA M accepts w



$$L(G) = L(M)$$

Therefore:

For any context-free language
there is an NPDA
that accepts the same language