

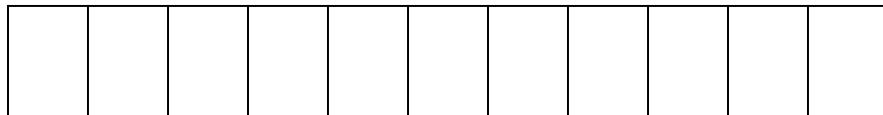
CS116-Automata Theory and Formal Languages

Lecture 9 Pushdown Automata (PDAs)

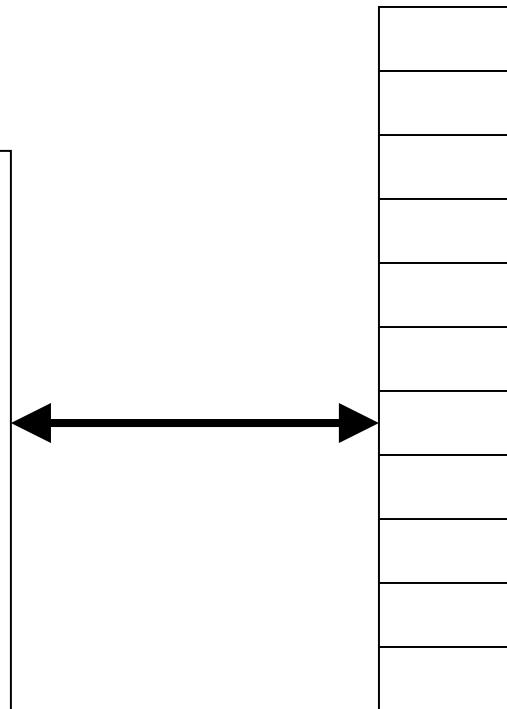
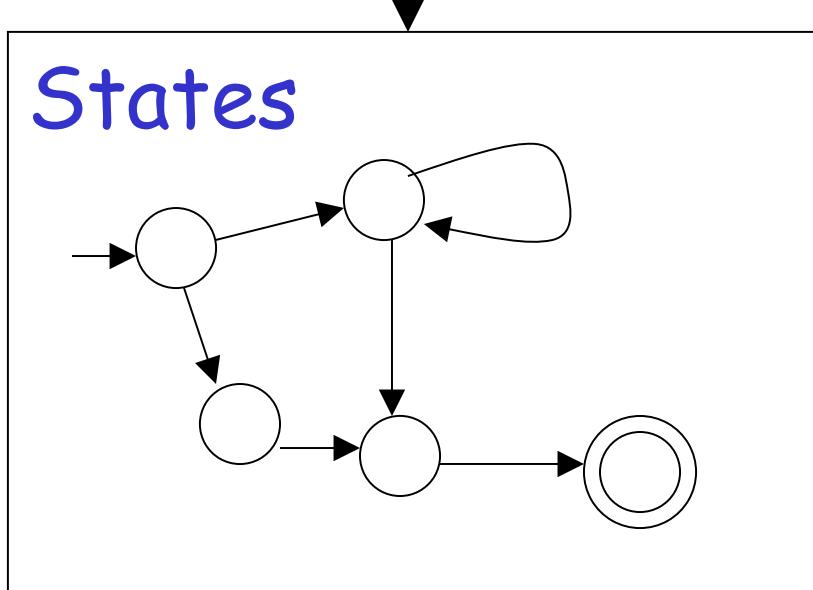
Computer Science Department
1st Semester 2025-2026

Pushdown Automaton -- PDA

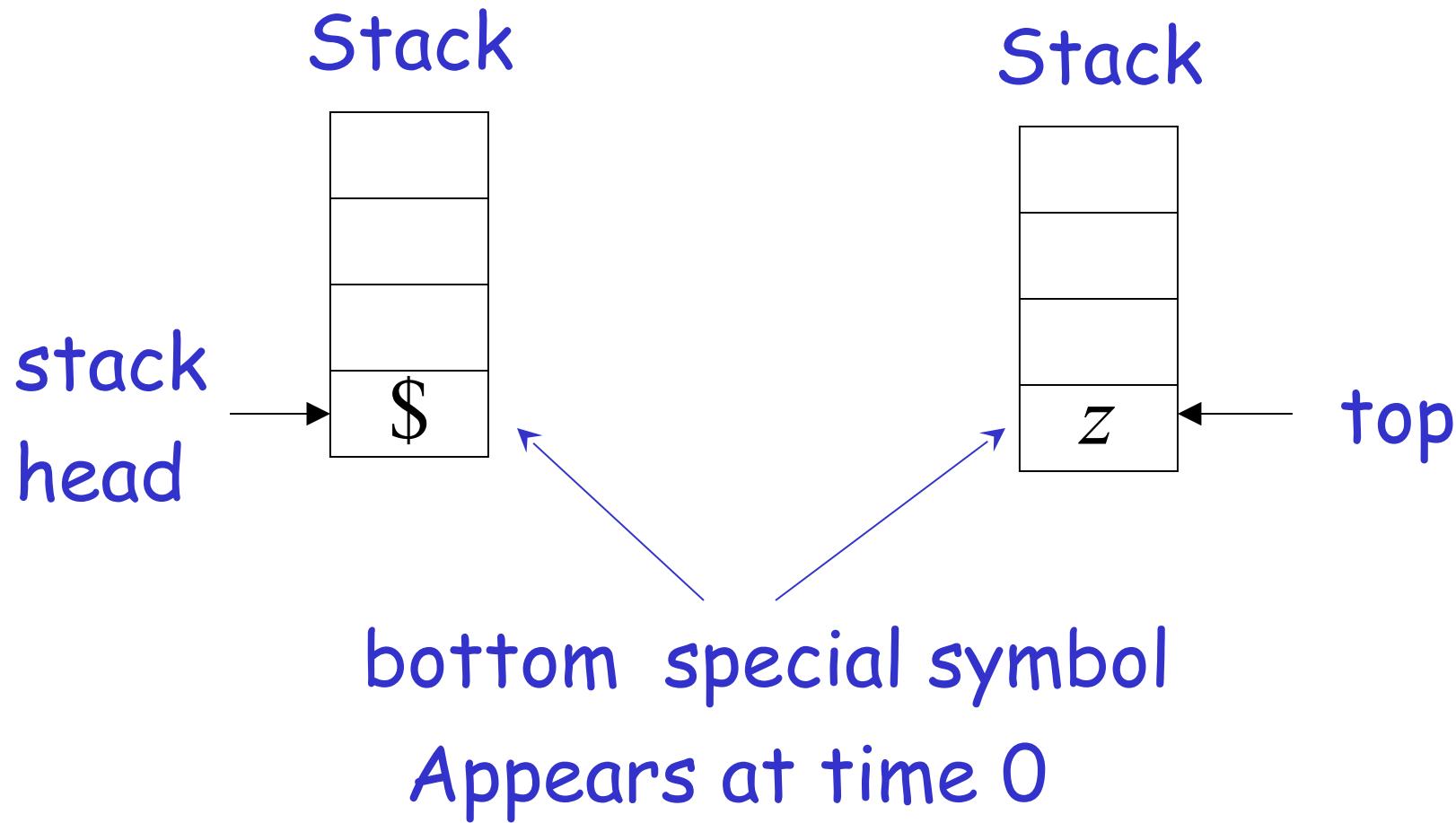
Input String



Stack



Initial Stack Symbol

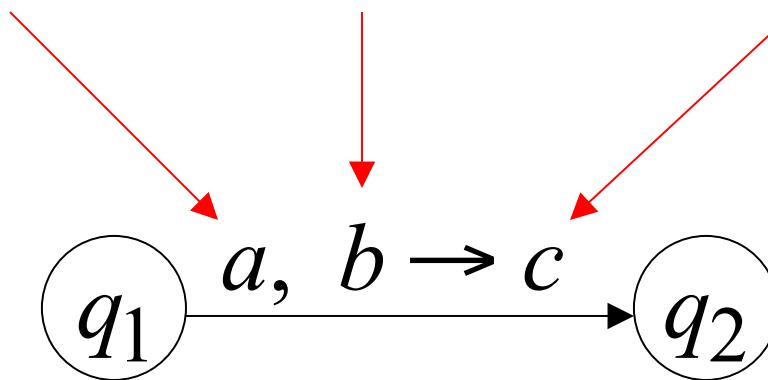


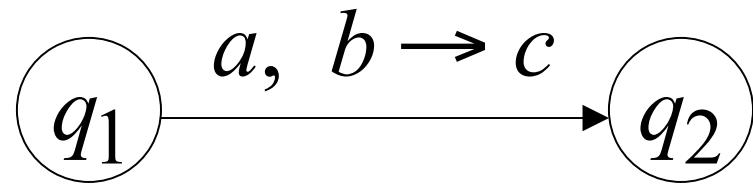
The States

Input
symbol

Pop
symbol

Push
symbol



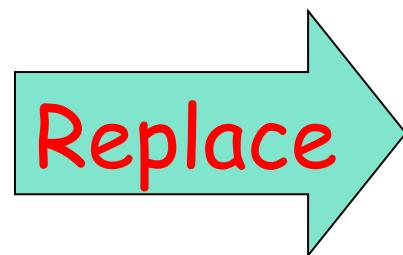


stack

b
h
e
$\$$

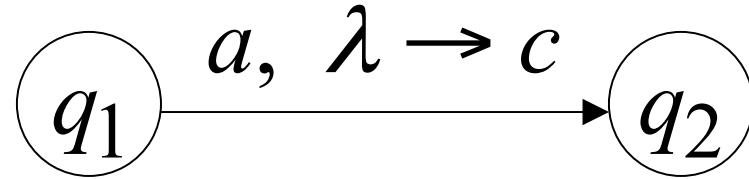
\leftarrow

top



c
h
e
$\$$

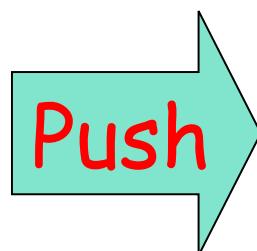
\leftarrow



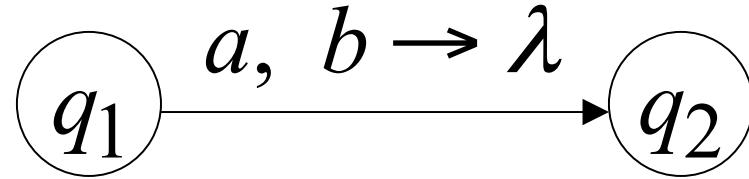
stack

b
h
e
\$

top



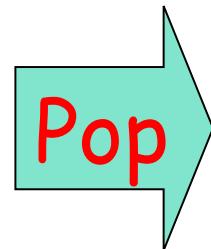
c
b
h
e
\$



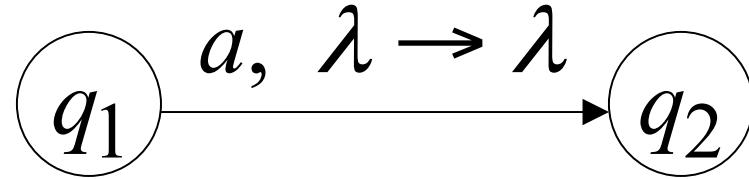
stack

b
h
e
\$

top



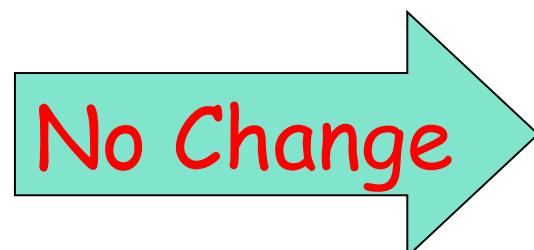
h
e
\$



stack

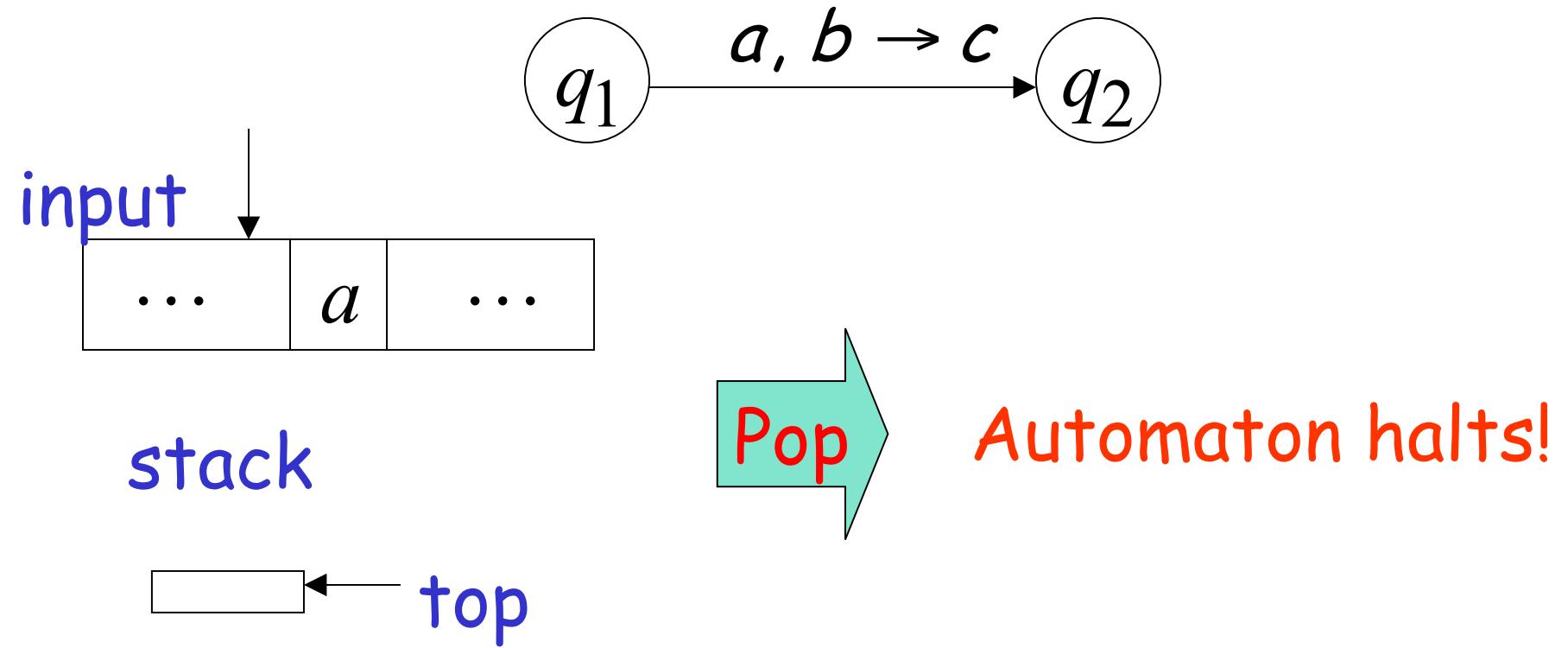
b
h
e
$\$$

top



b
h
e
$\$$

Pop from Empty Stack

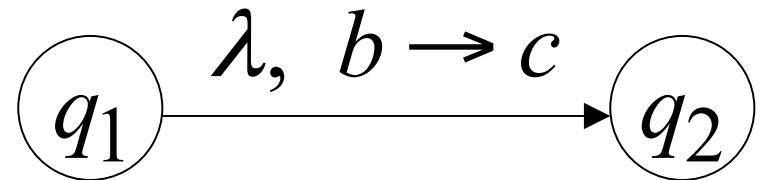
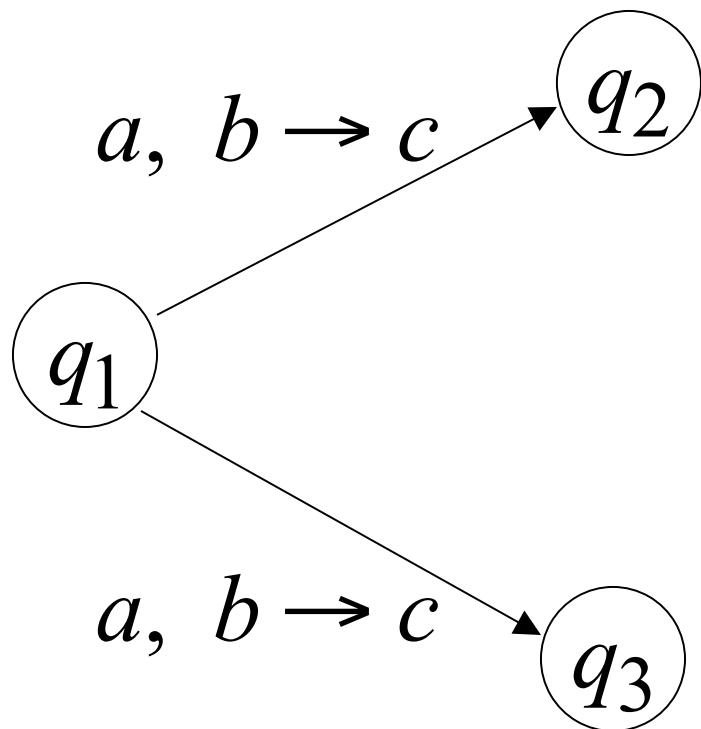


If the automaton attempts to pop from empty stack then it halts and rejects input

Non-Determinism

PDAs are non-deterministic

Allowed non-deterministic transitions

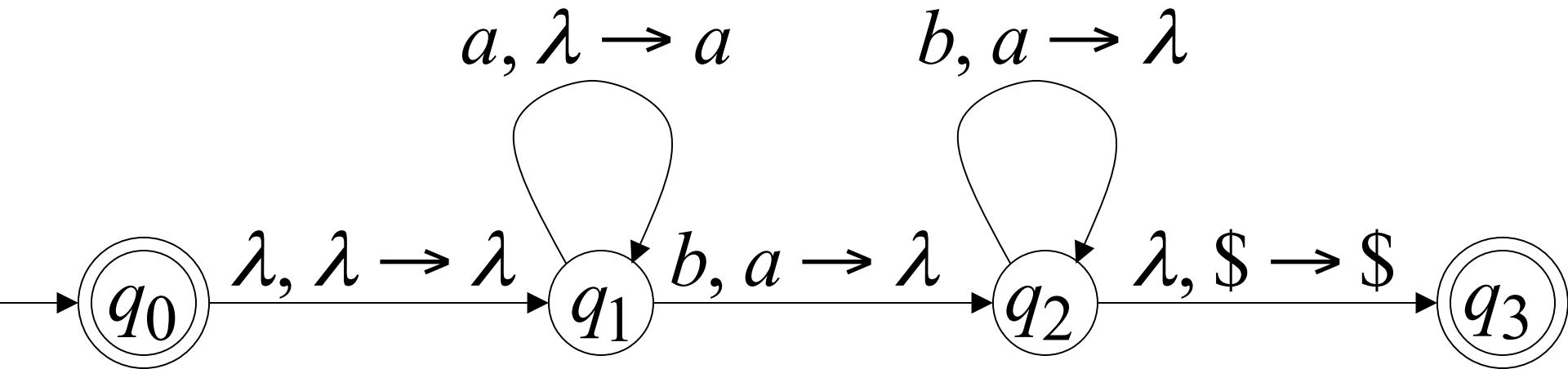


λ - transition

Example PDA

PDA M :

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



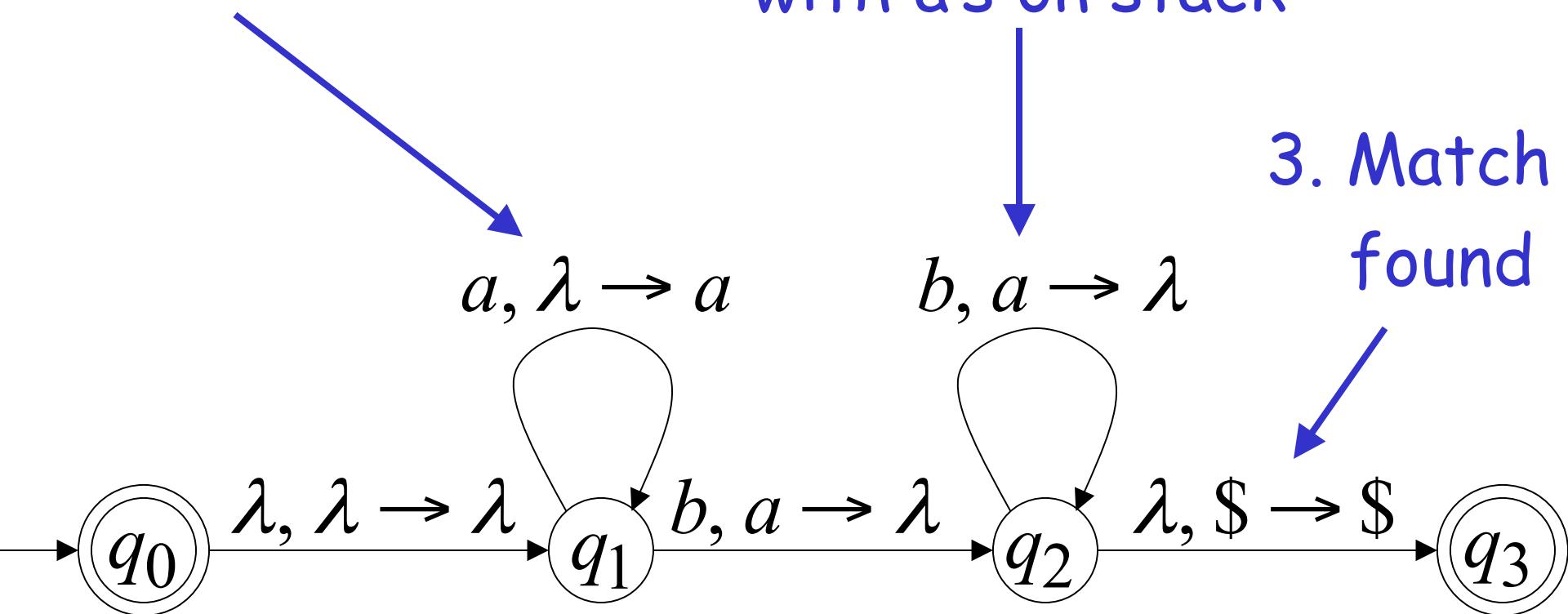
$$L(M) = \{a^n b^n : n \geq 0\}$$

Basic Idea:

Push the a's
on the stack

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

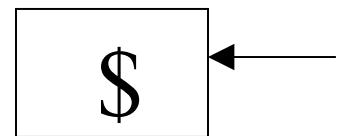
3. Match
found



Execution Example: Time 0

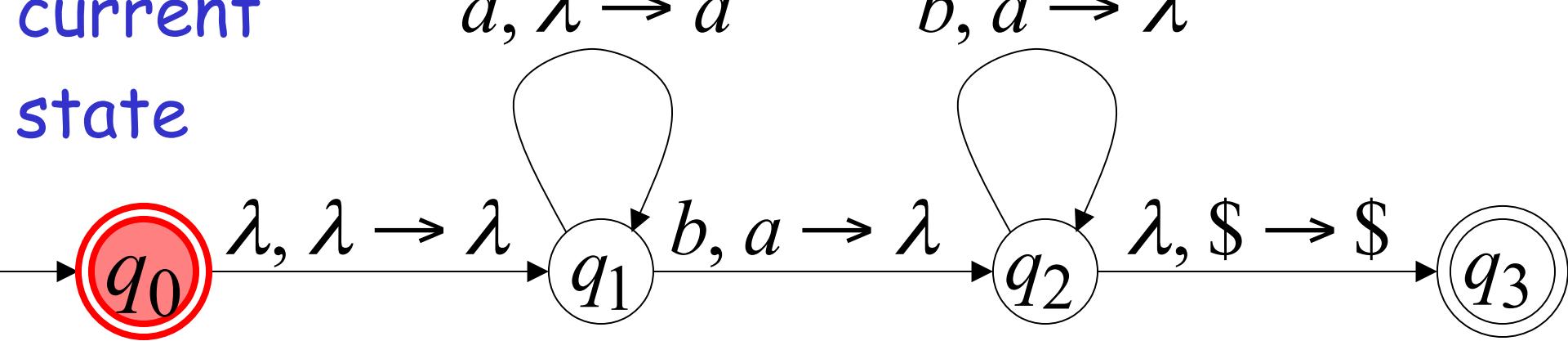
Input

a	a	a	b	b	b
-----	-----	-----	-----	-----	-----



Stack

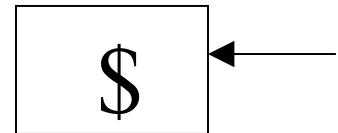
current
state



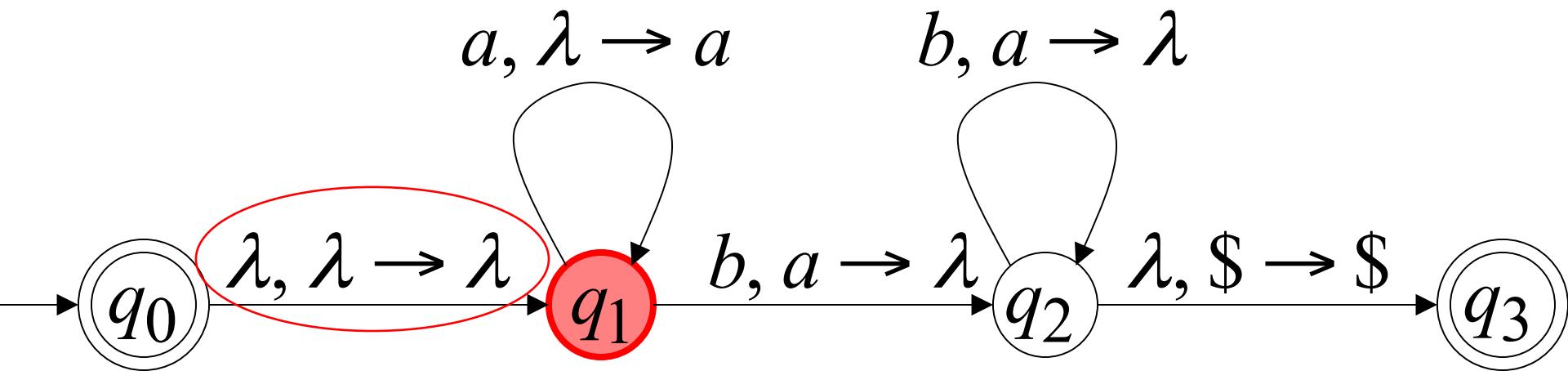
Time 1

Input

a	a	a	b	b	b
-----	-----	-----	-----	-----	-----



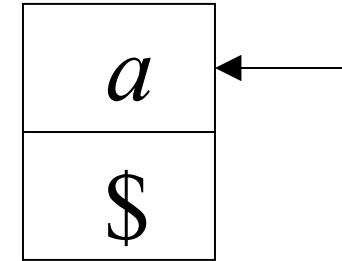
Stack



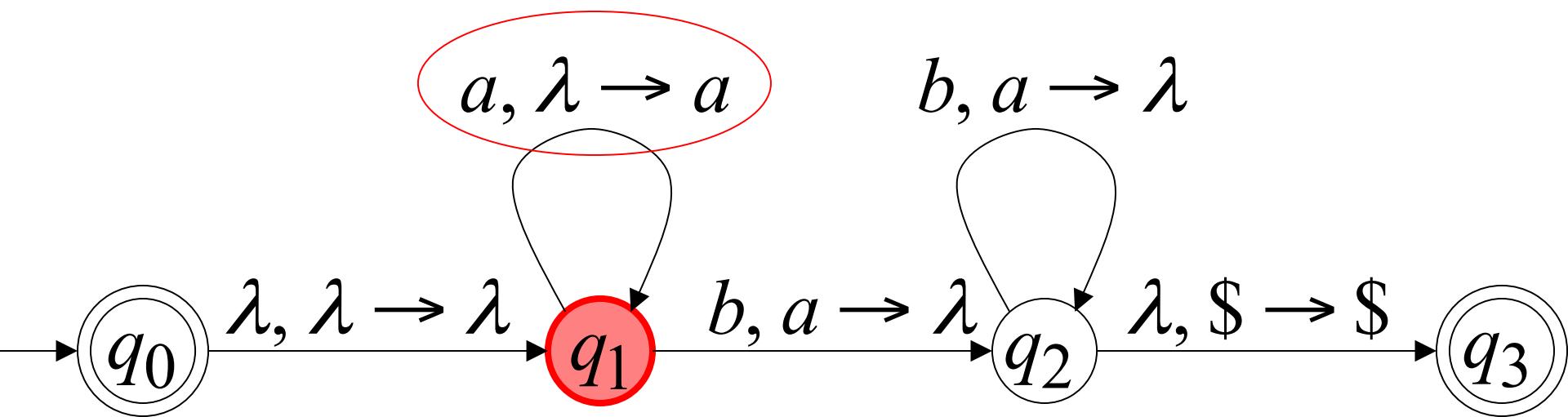
Time 2

Input

a	a	a	b	b	b
-----	-----	-----	-----	-----	-----



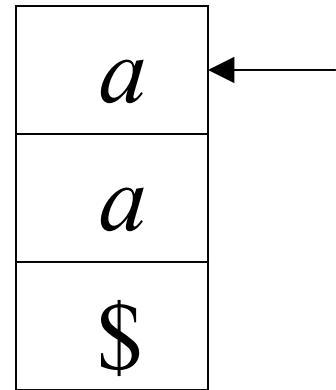
Stack



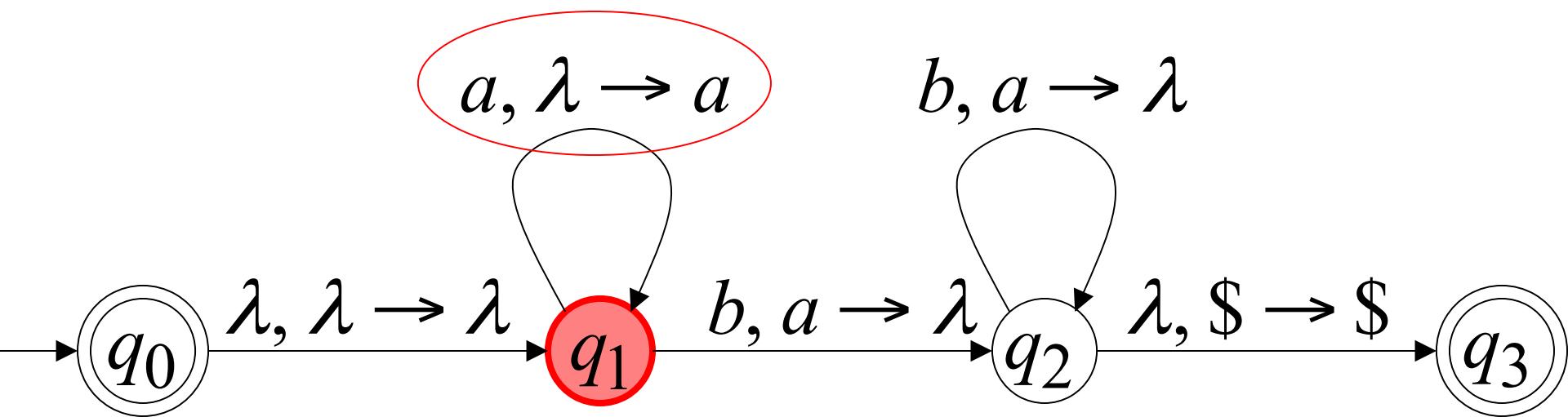
Time 3

Input

a	a	a	b	b	b
---	---	---	---	---	---



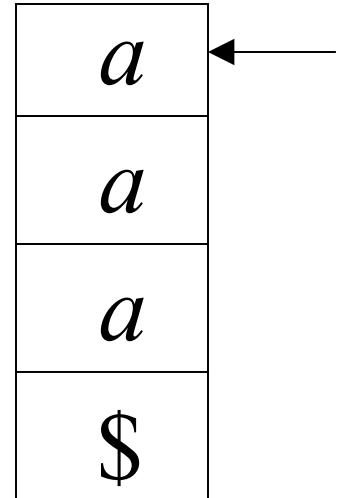
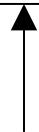
Stack



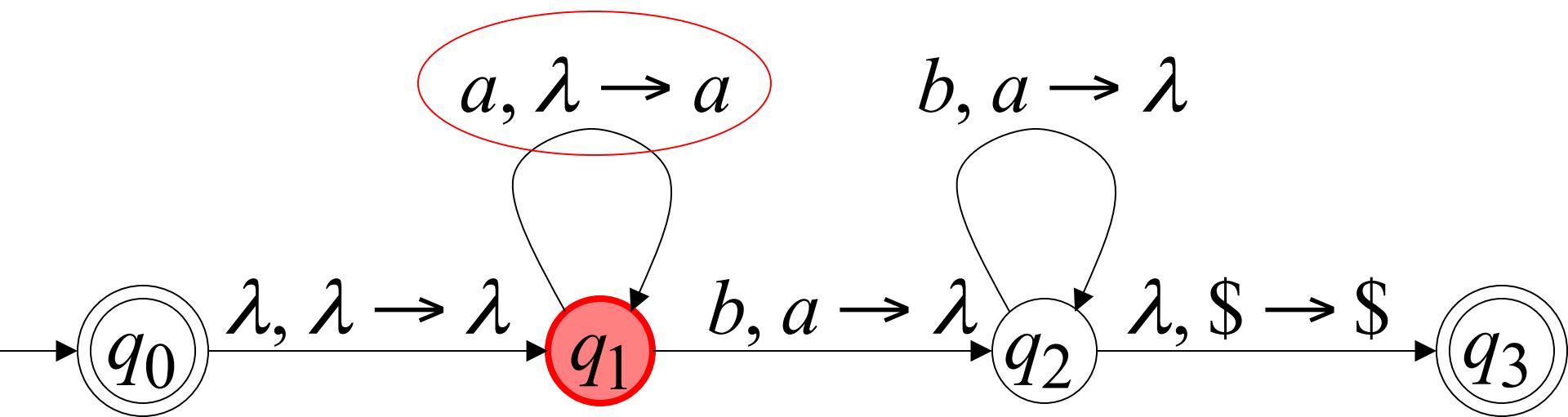
Time 4

Input

a	a	a	b	b	b
---	---	---	---	---	---



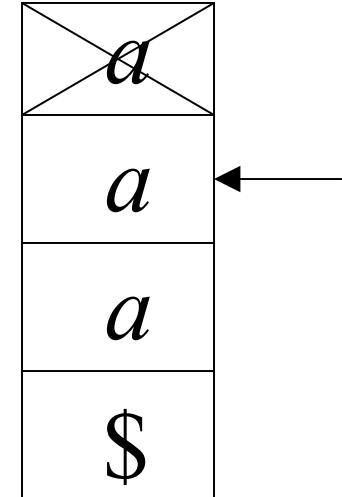
Stack



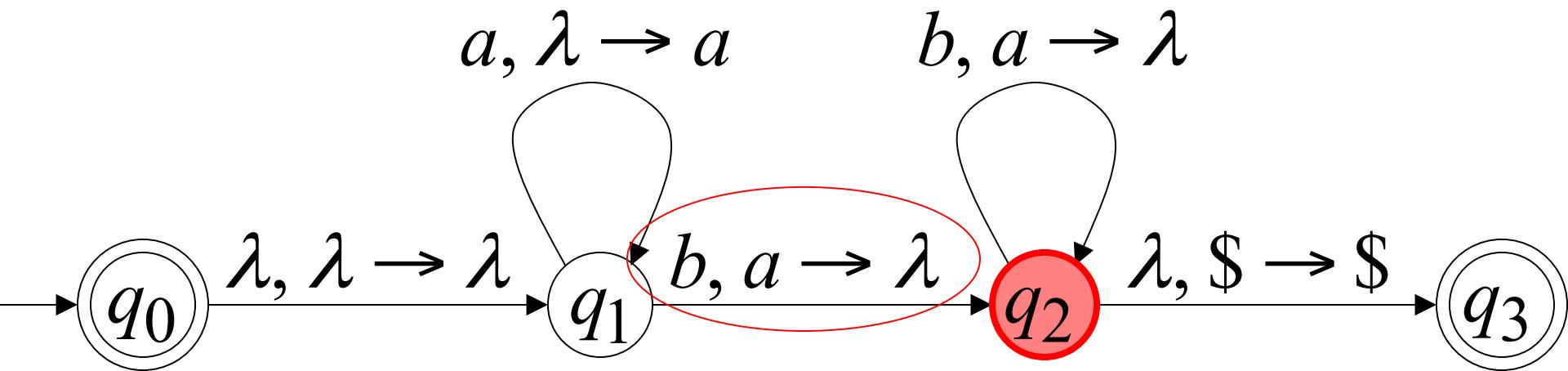
Time 5

Input

a	a	a	b	b	b
-----	-----	-----	-----	-----	-----



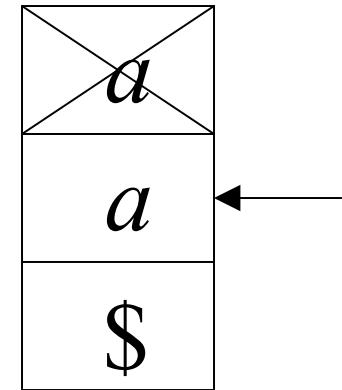
Stack



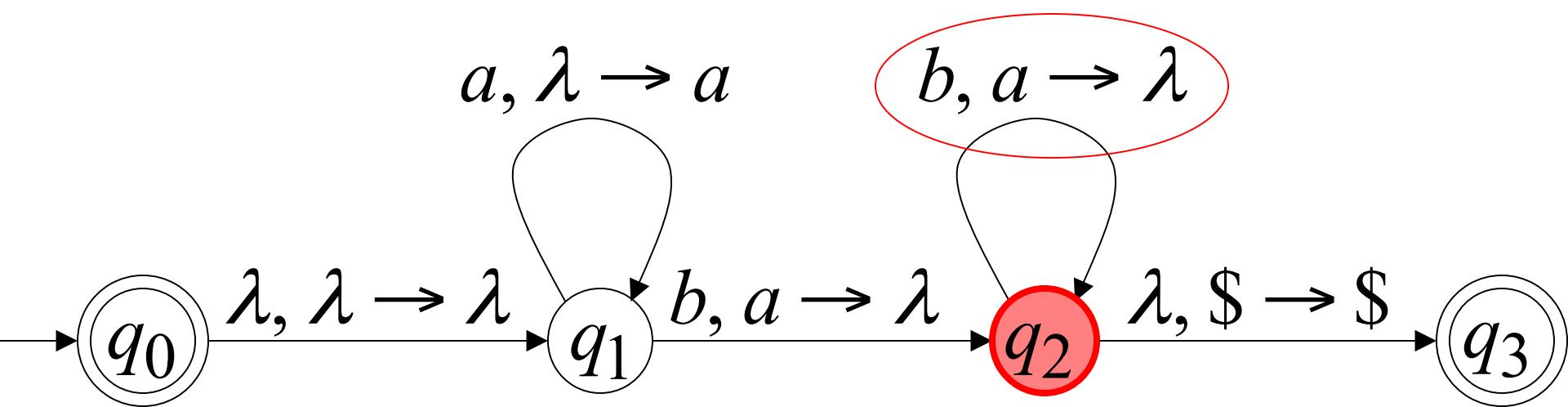
Time 6

Input

a	a	a	b	b	b
-----	-----	-----	-----	-----	-----



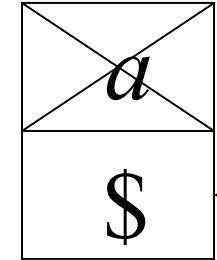
Stack



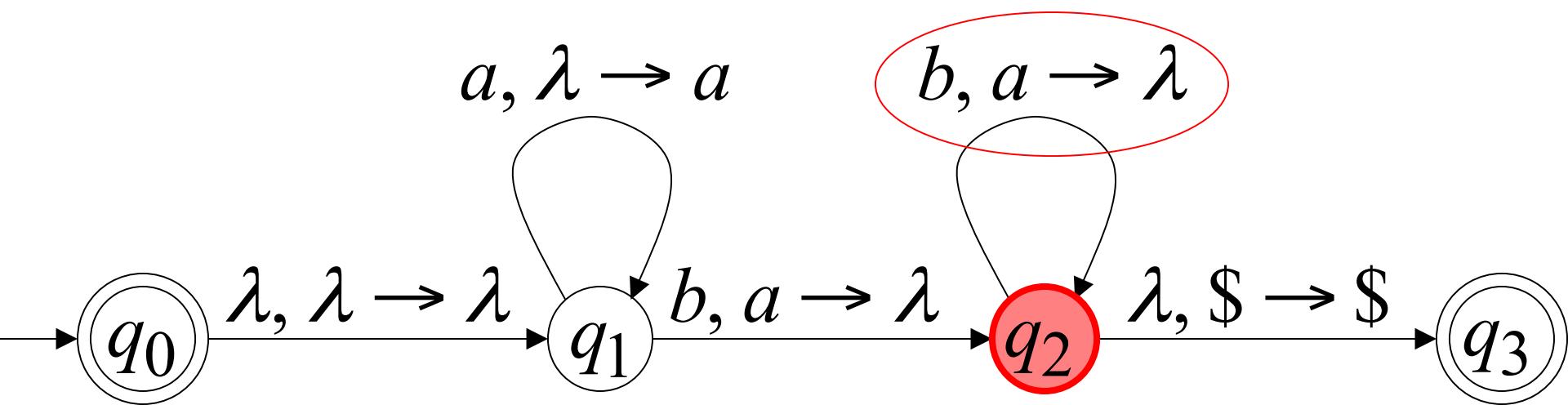
Time 7

Input

a	a	a	b	b	b
-----	-----	-----	-----	-----	-----



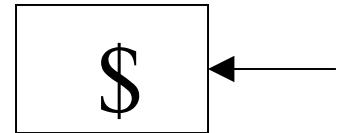
Stack



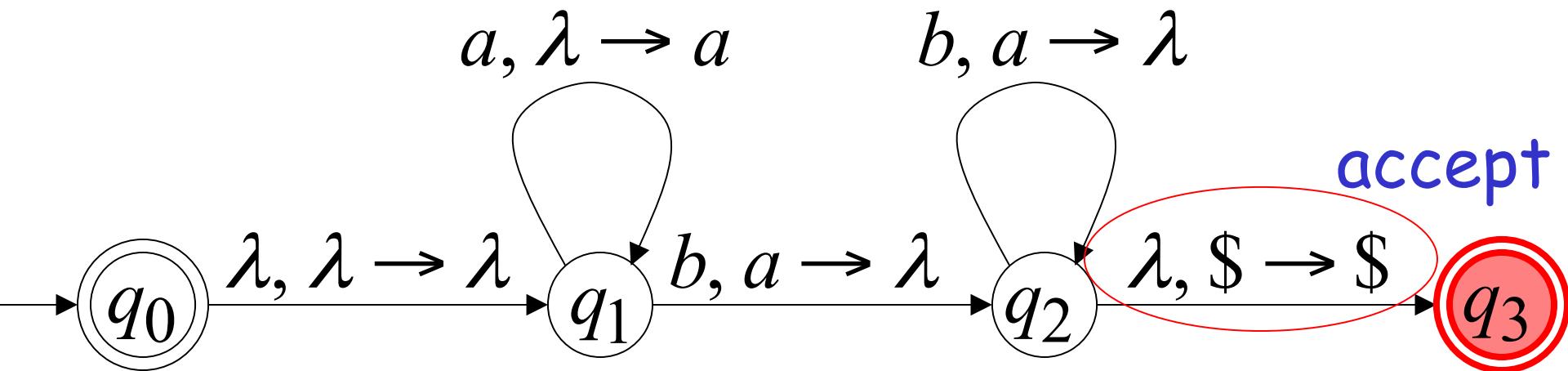
Time 8

Input

a	a	a	b	b	b
-----	-----	-----	-----	-----	-----



Stack



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

All the input is consumed

AND

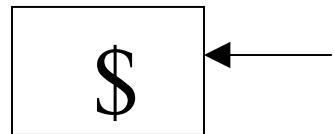
The last state is an accepting state

we do not care about the stack contents
at the end of the accepting computation

Rejection Example: Time 0

Input

a	a	b
-----	-----	-----

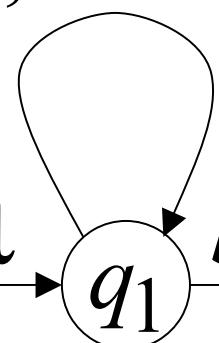


Stack

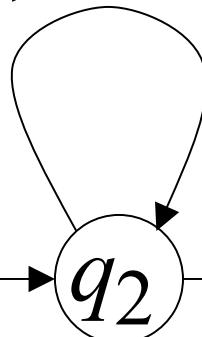
current
state



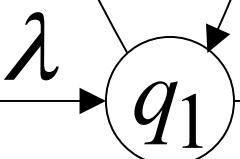
$$a, \lambda \rightarrow a$$



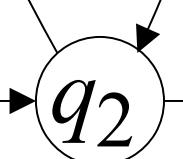
$$b, a \rightarrow \lambda$$



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



$$b, a \rightarrow \lambda$$



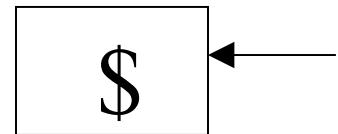
$$\lambda, \$ \rightarrow \$$$



Rejection Example: Time 1

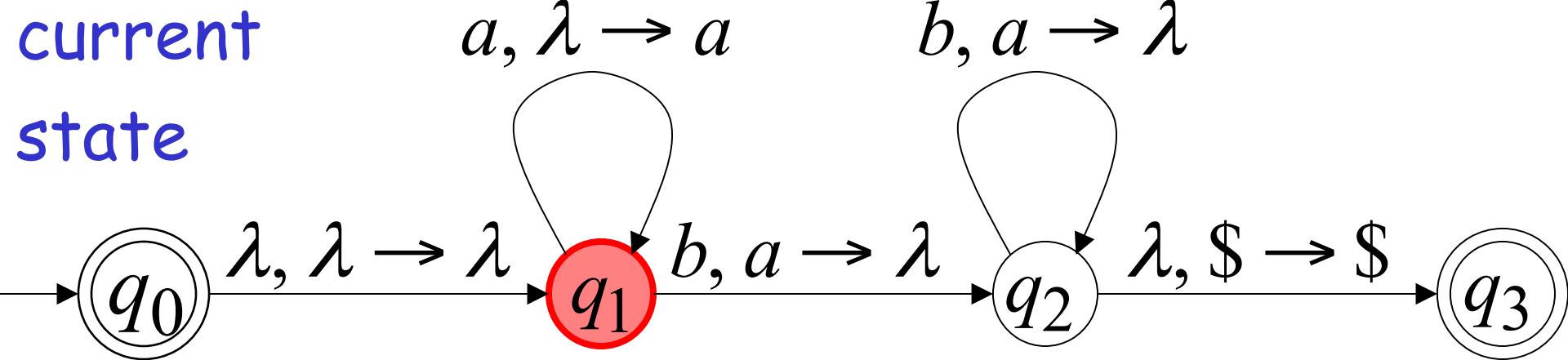
Input

a	a	b
-----	-----	-----



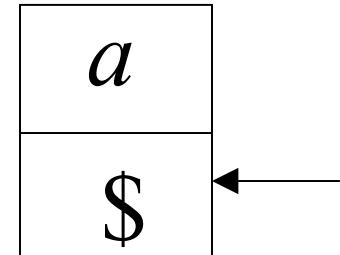
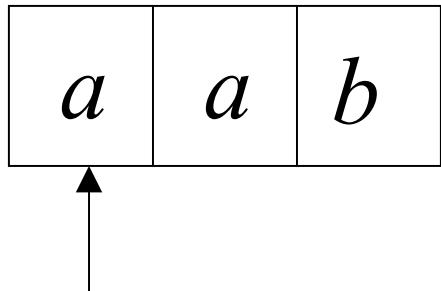
Stack

current
state



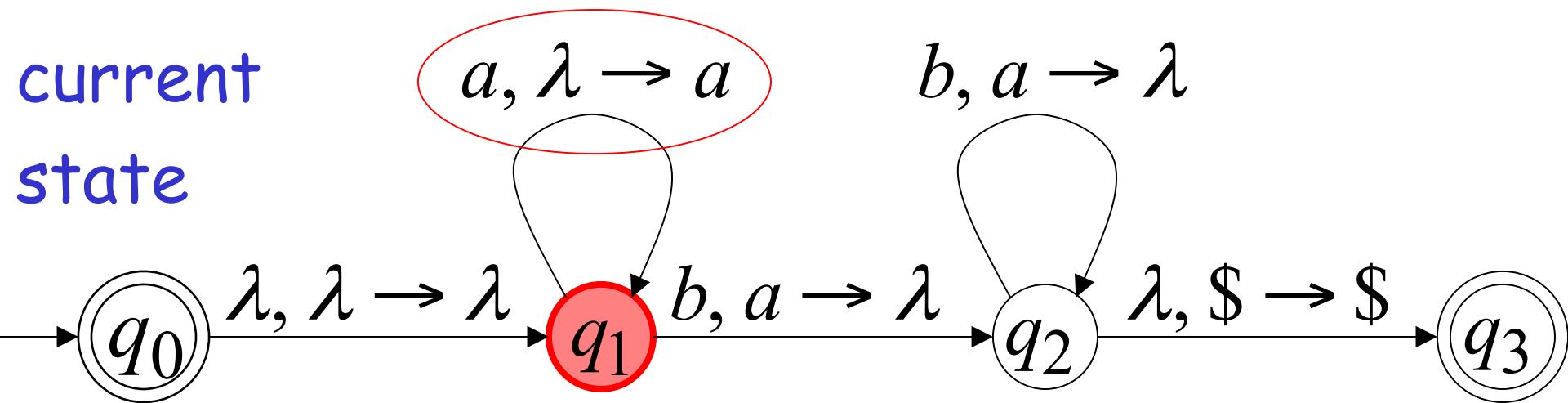
Rejection Example: Time 2

Input



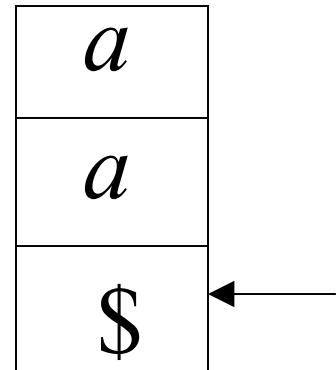
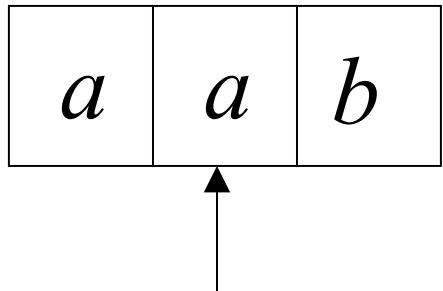
Stack

current
state



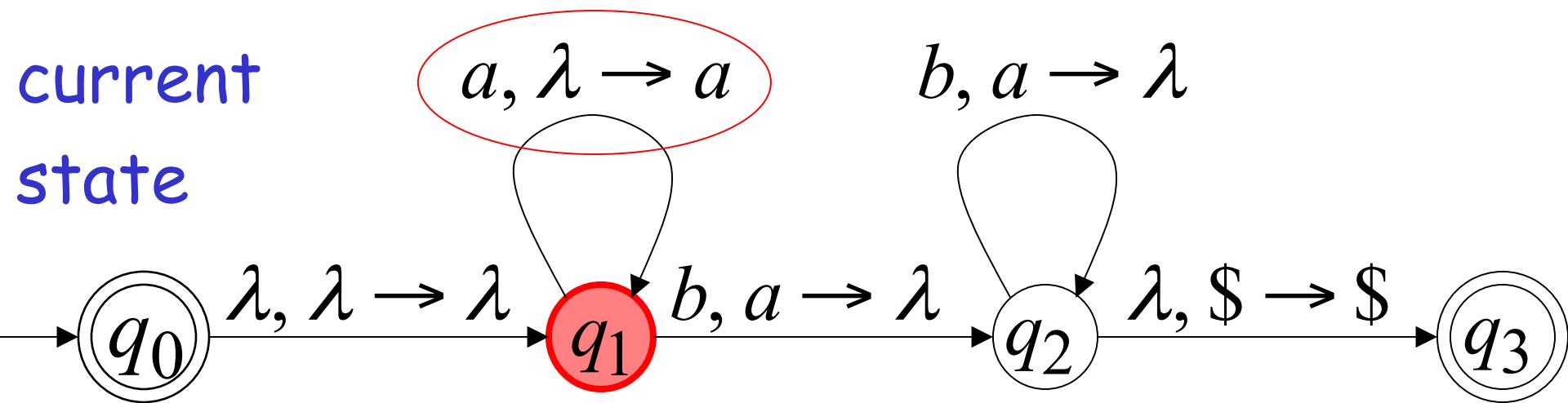
Rejection Example: Time 3

Input



Stack

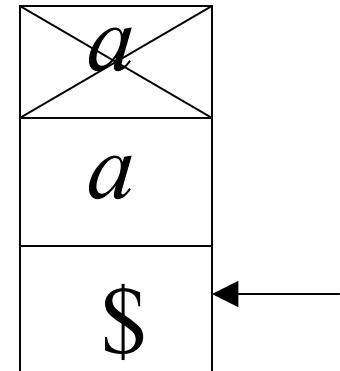
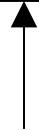
current
state



Rejection Example: Time 4

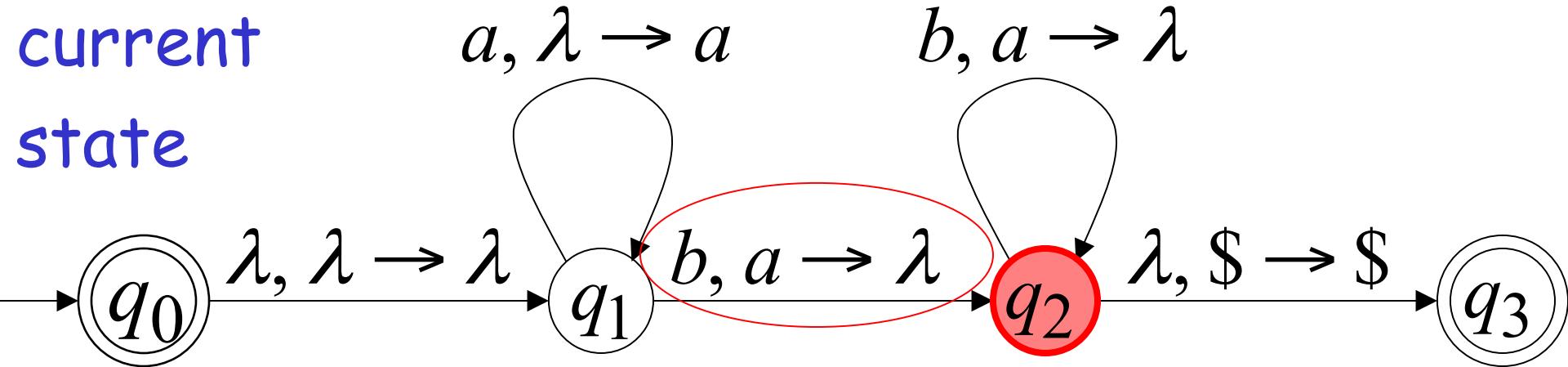
Input

a	a	b
-----	-----	-----



Stack

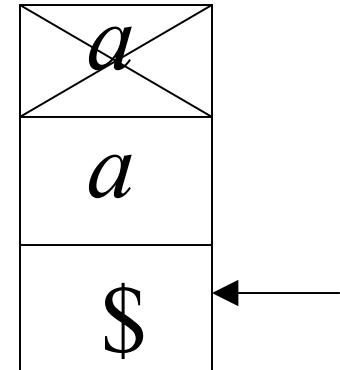
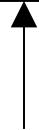
current
state



Rejection Example: Time 4

Input

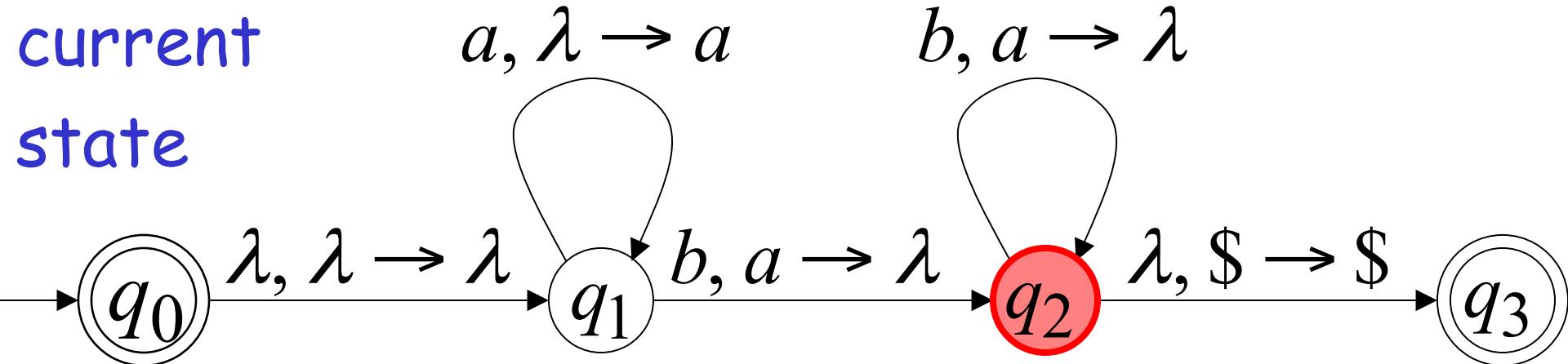
a	a	b
-----	-----	-----



Stack

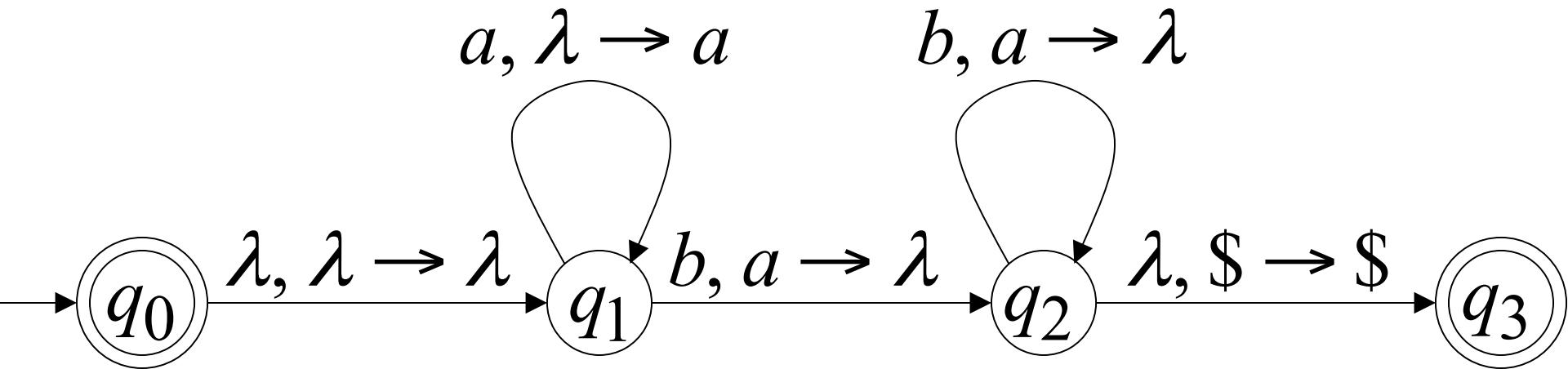
reject

current
state



There is no accepting computation for aab

The string aab is rejected by the PDA



Another PDA example

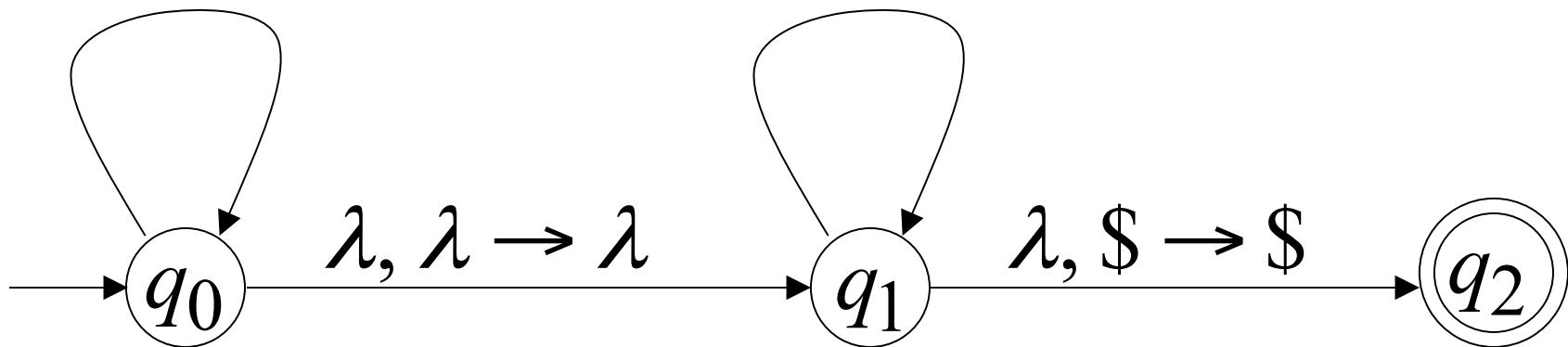
PDA M : $L(M) = \{vv^R : v \in \{a, b\}^*\}$

$$a, \lambda \rightarrow a$$

$$b, \lambda \rightarrow b$$

$$a, a \rightarrow \lambda$$

$$b, b \rightarrow \lambda$$



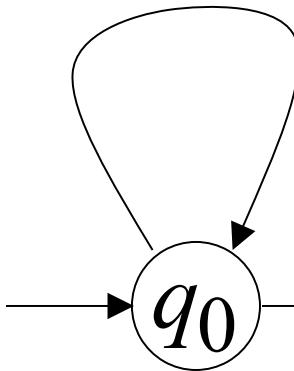
Basic Idea:

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

Push v
on stack



$$\begin{array}{l} a, \lambda \rightarrow a \\ b, \lambda \rightarrow b \end{array}$$

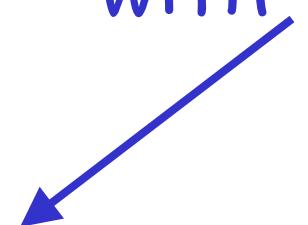


2. Guess
middle
of input

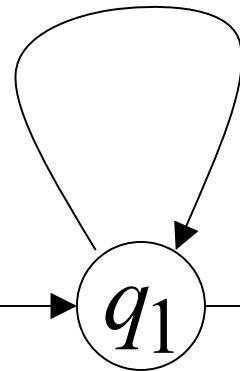


$$\lambda, \lambda \rightarrow \lambda$$

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



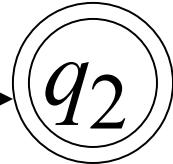
$$\begin{array}{l} a, a \rightarrow \lambda \\ b, b \rightarrow \lambda \end{array}$$



4. Match
found



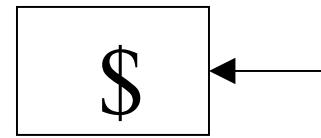
$$\lambda, \$ \rightarrow \$$$



Execution Example: Time 0

Input

a	b	b	a
-----	-----	-----	-----

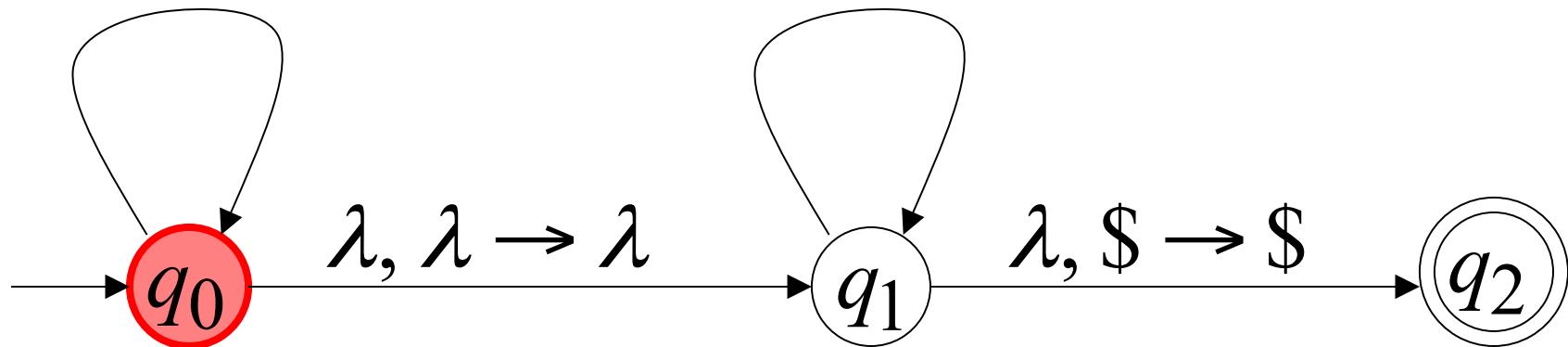


$$a, \lambda \rightarrow a$$

$$a, a \rightarrow \lambda$$

$$b, \lambda \rightarrow b$$

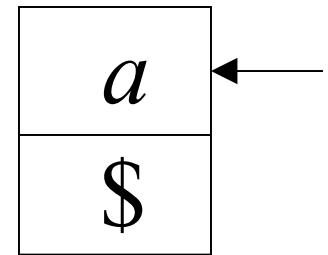
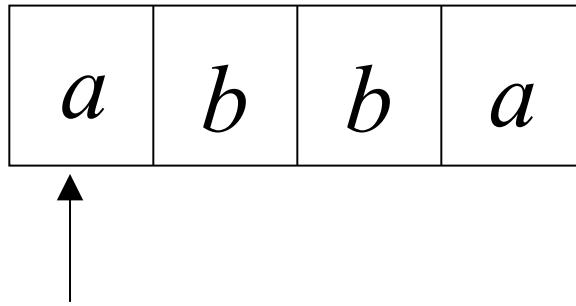
$$b, b \rightarrow \lambda$$



Stack

Time 1

Input



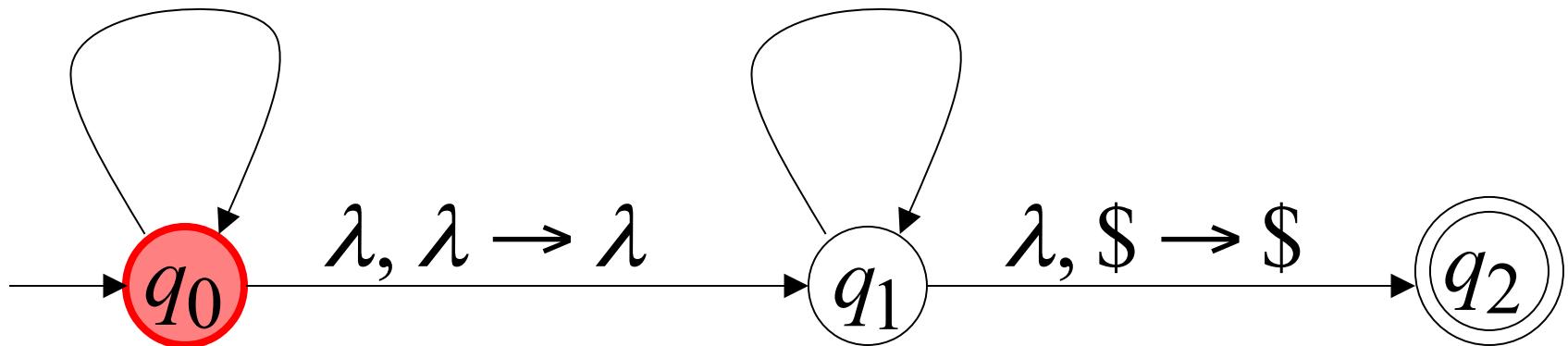
$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$

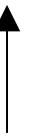
Stack



Time 2

Input

a	b	b	a
-----	-----	-----	-----

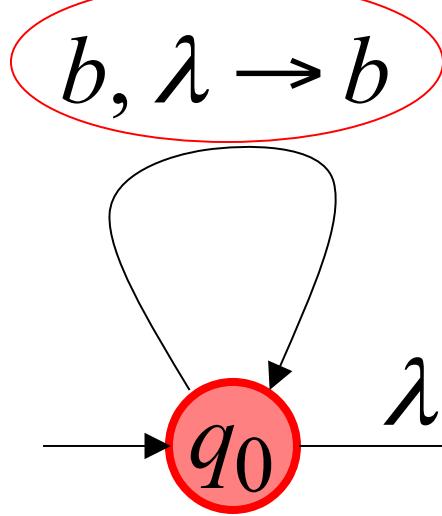


b
a
$\$$

Stack

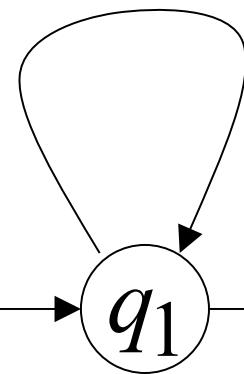
$$a, \lambda \rightarrow a$$

$$b, \lambda \rightarrow b$$



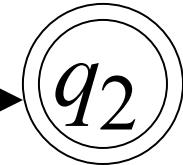
$a, a \rightarrow \lambda$

$$b, b \rightarrow \lambda$$



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

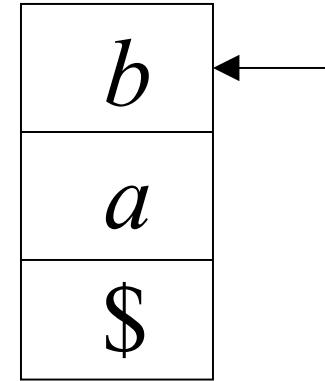
λ , \$ → \$



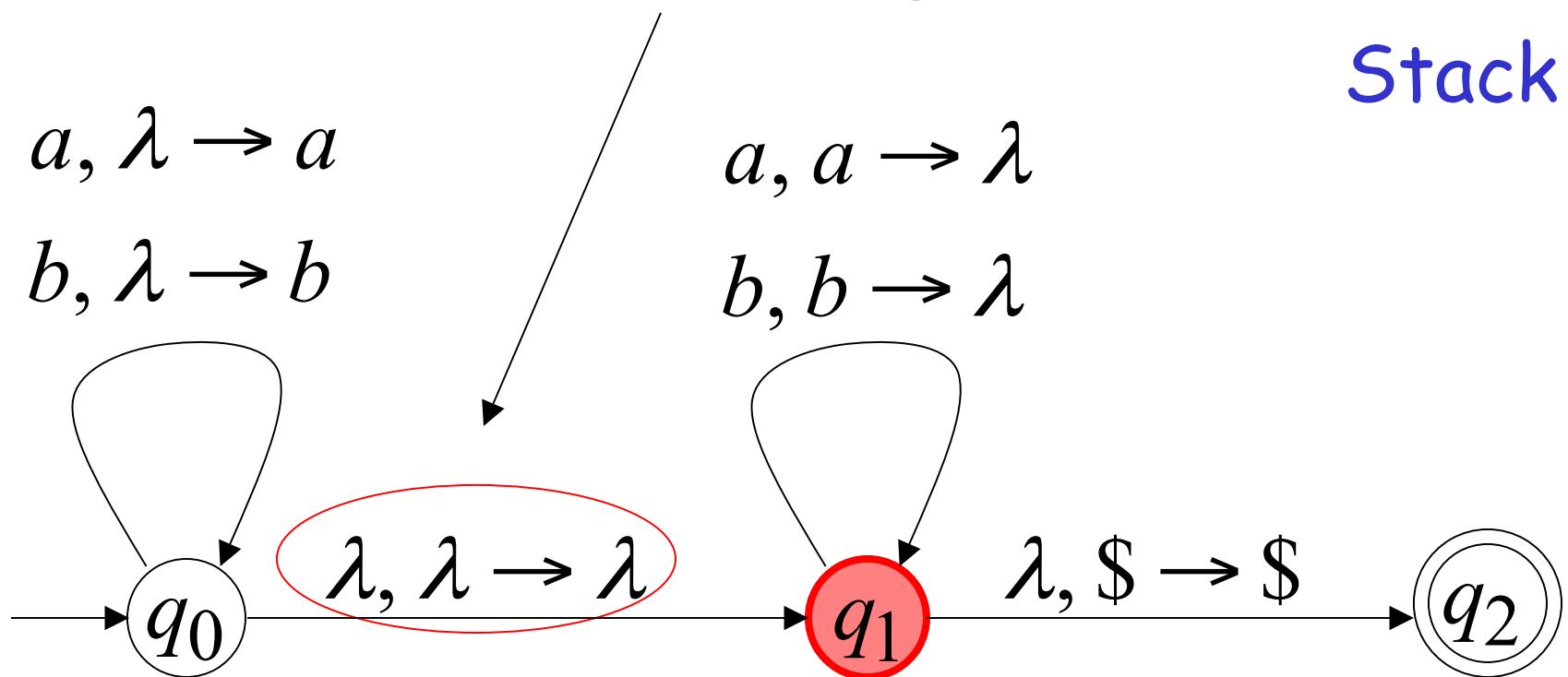
Time 3

Input

a	b	b	a
---	---	---	---



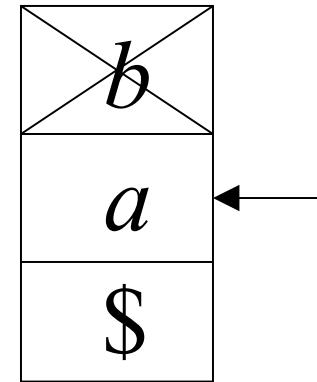
Guess the middle
of string



Time 4

Input

a	b	b	a
-----	-----	-----	-----



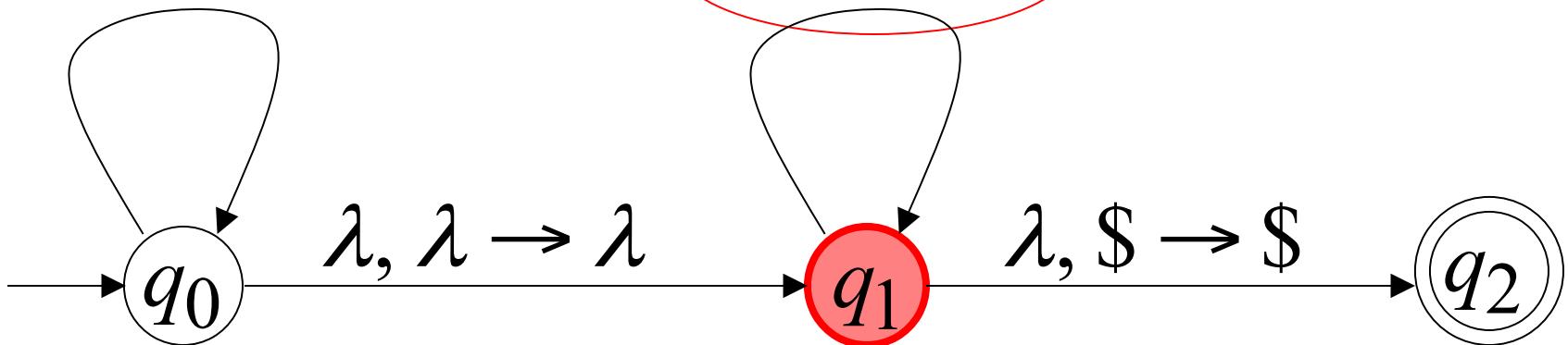
Stack

$$a, \lambda \rightarrow a$$

$$a, a \rightarrow \lambda$$

$$b, \lambda \rightarrow b$$

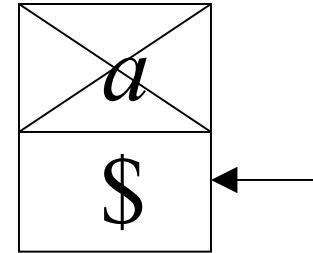
$$b, b \rightarrow \lambda$$



Time 5

Input

a	b	b	a
-----	-----	-----	-----

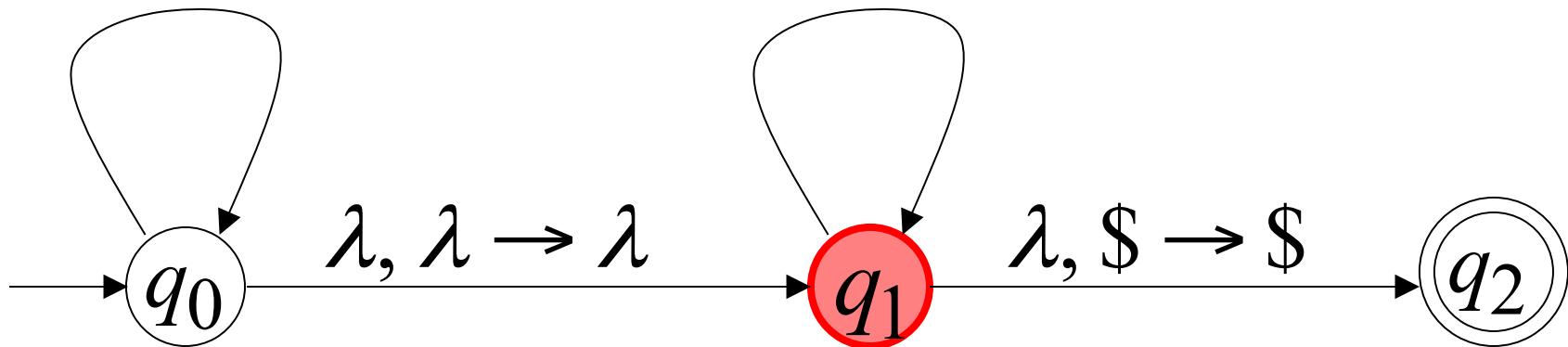


$$a, \lambda \rightarrow a$$

$$b, \lambda \rightarrow b$$

$$a, a \rightarrow \lambda$$

$$b, b \rightarrow \lambda$$

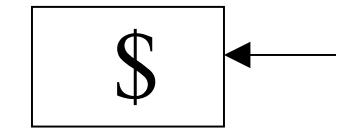


Stack

Time 6

Input

a	b	b	a
-----	-----	-----	-----

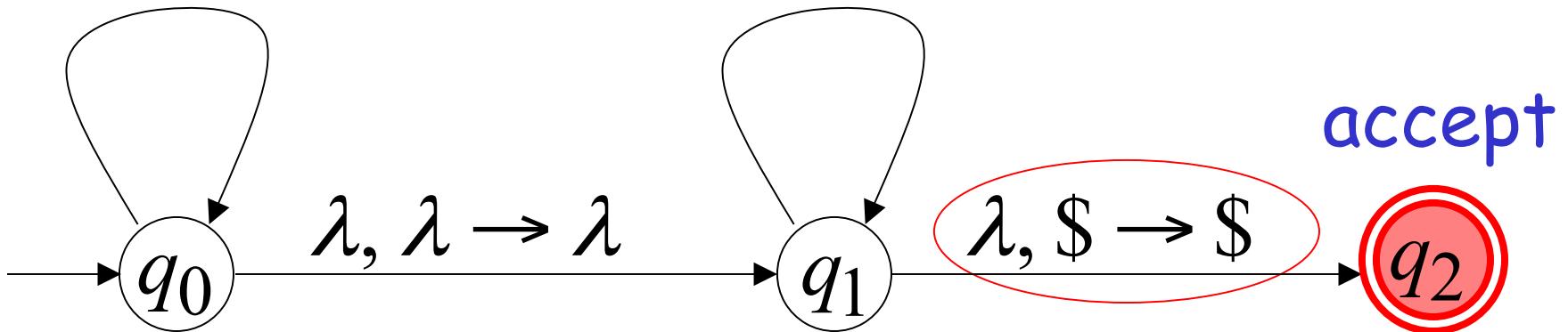


$$a, \lambda \rightarrow a$$

$$a, a \rightarrow \lambda$$

$$b, \lambda \rightarrow b$$

$$b, b \rightarrow \lambda$$



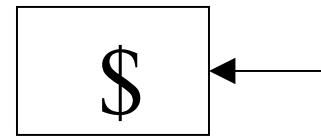
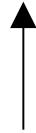
accept

Rejection Example:

Time 0

Input

a	b	b	b
-----	-----	-----	-----

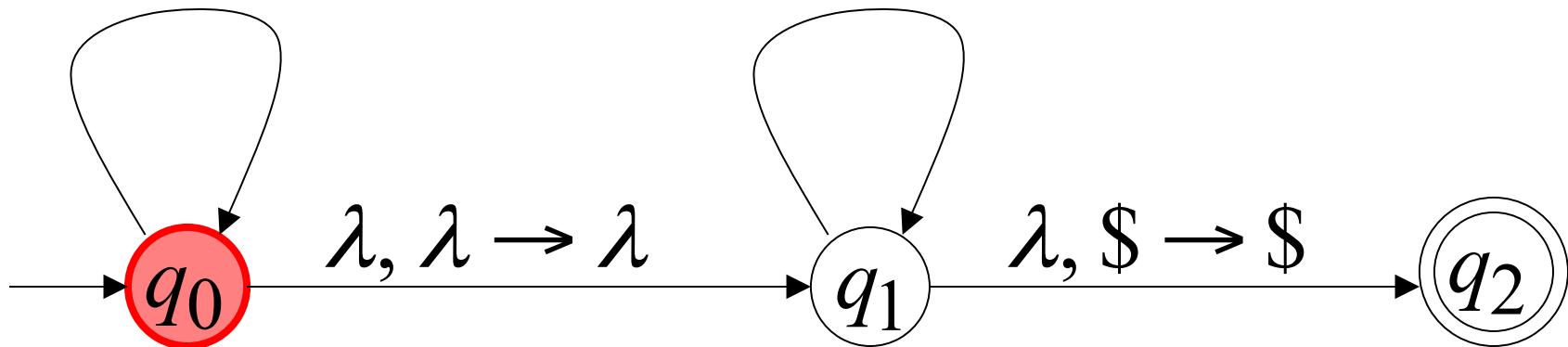


$$a, \lambda \rightarrow a$$

$$a, a \rightarrow \lambda$$

$$b, \lambda \rightarrow b$$

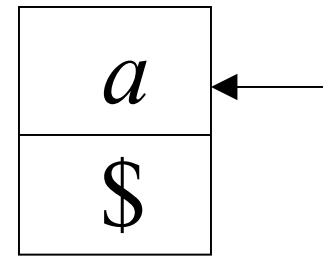
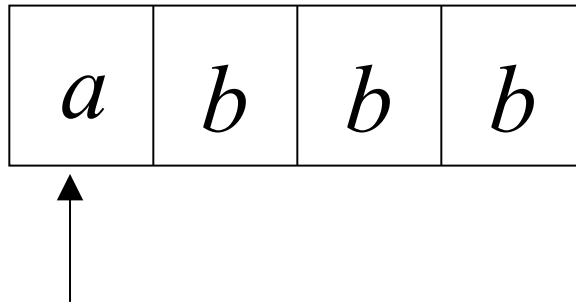
$$b, b \rightarrow \lambda$$



Stack

Time 1

Input



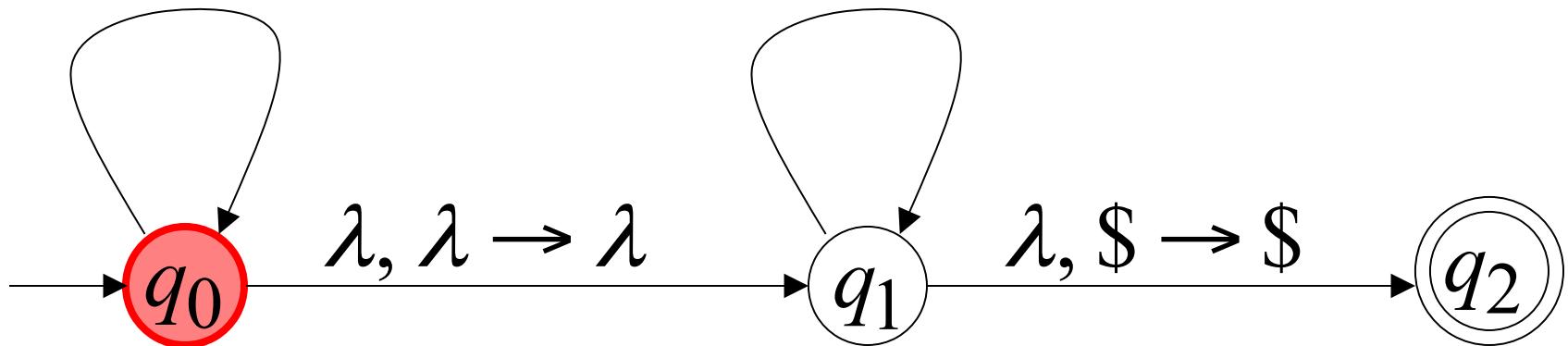
$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$

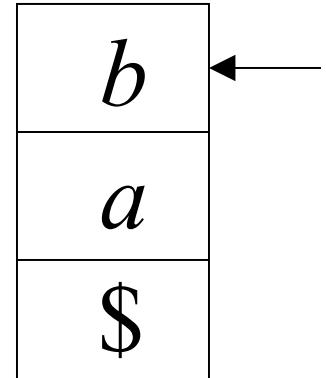
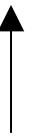
Stack



Time 2

Input

a	b	b	b
---	---	---	---



Stack

$$a, \lambda \rightarrow a$$

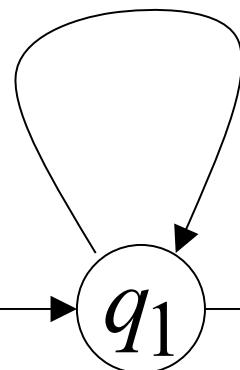
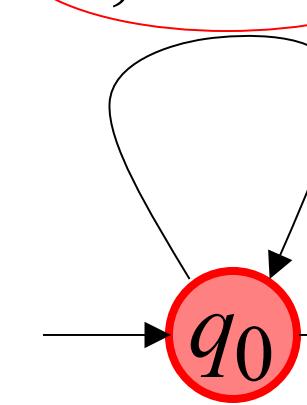
$$b, \lambda \rightarrow b$$

$$a, a \rightarrow \lambda$$

$$b, b \rightarrow \lambda$$

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

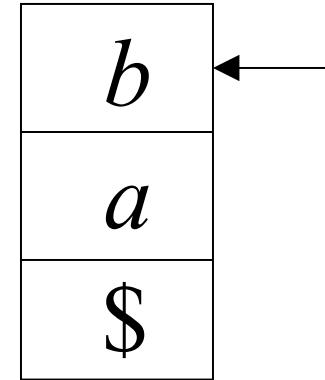
$$\lambda, \$ \rightarrow \$$$



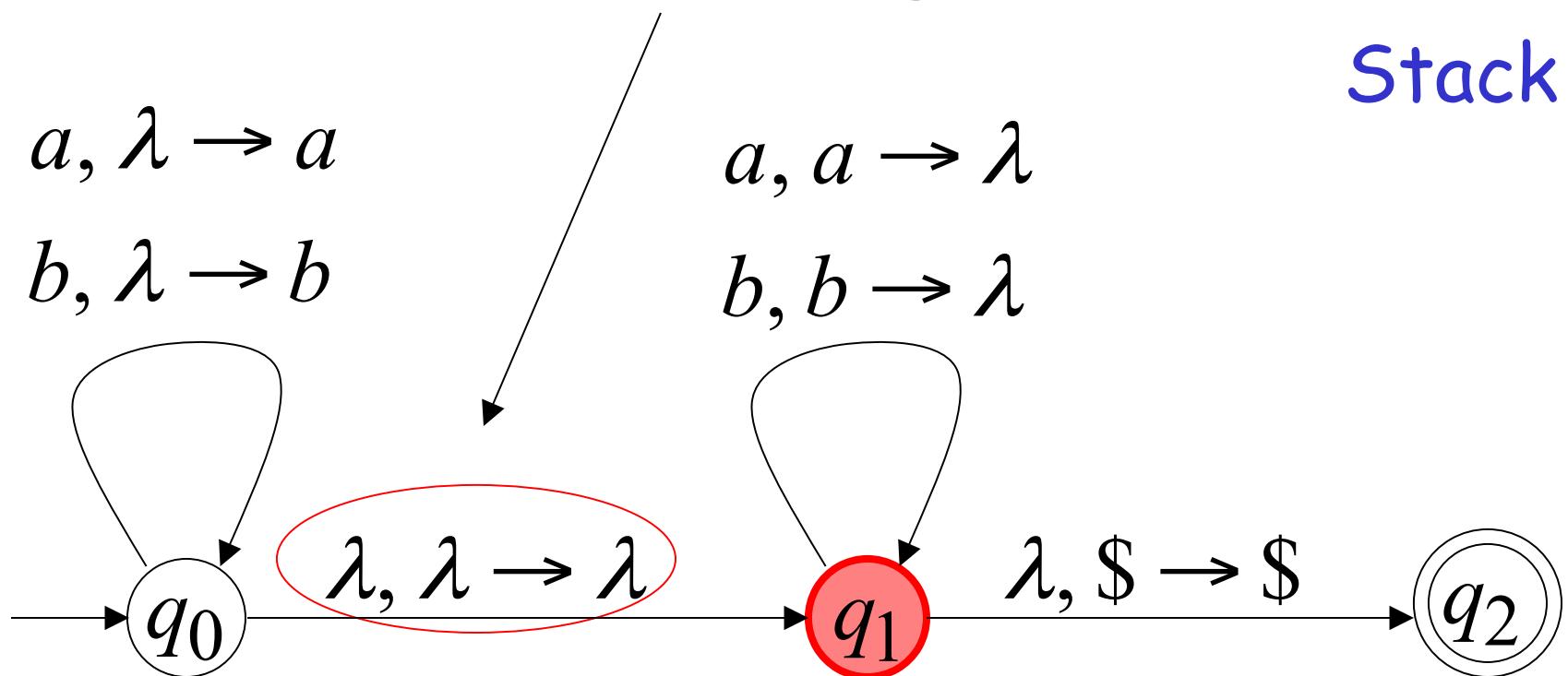
Time 3

Input

a	b	b	b
---	---	---	---



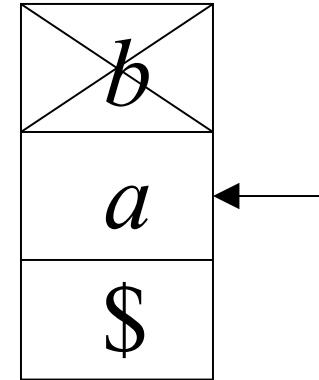
Guess the middle
of string



Time 4

Input

a	b	b	b
-----	-----	-----	-----

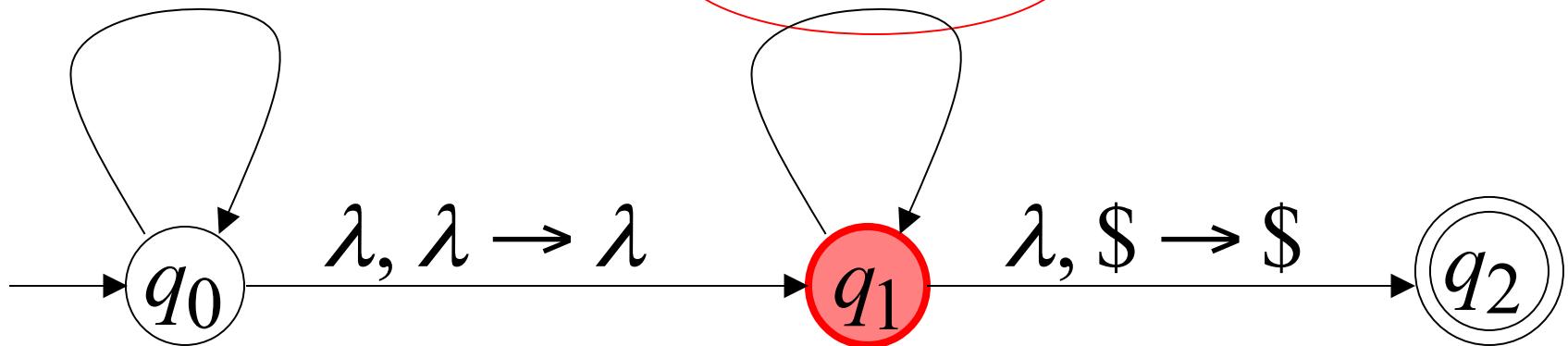


$$a, \lambda \rightarrow a$$

$$b, \lambda \rightarrow b$$

$$a, a \rightarrow \lambda$$

$$b, b \rightarrow \lambda$$

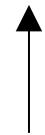


Stack

Time 5

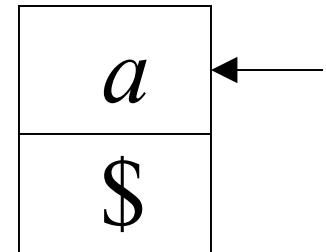
Input

a	b	b	b
-----	-----	-----	-----



There is no possible transition.

Input is not
consumed

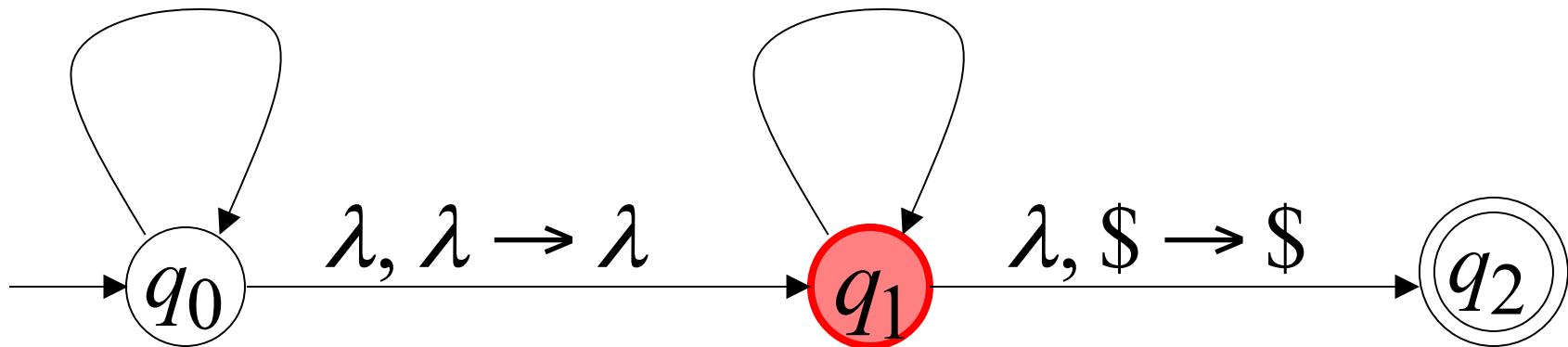


$$a, \lambda \rightarrow a$$

$$a, a \rightarrow \lambda$$

$$b, \lambda \rightarrow b$$

$$b, b \rightarrow \lambda$$



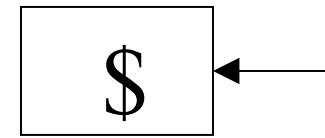
Stack

Another computation on same string:

Input

a	b	b	b
---	---	---	---

Time 0



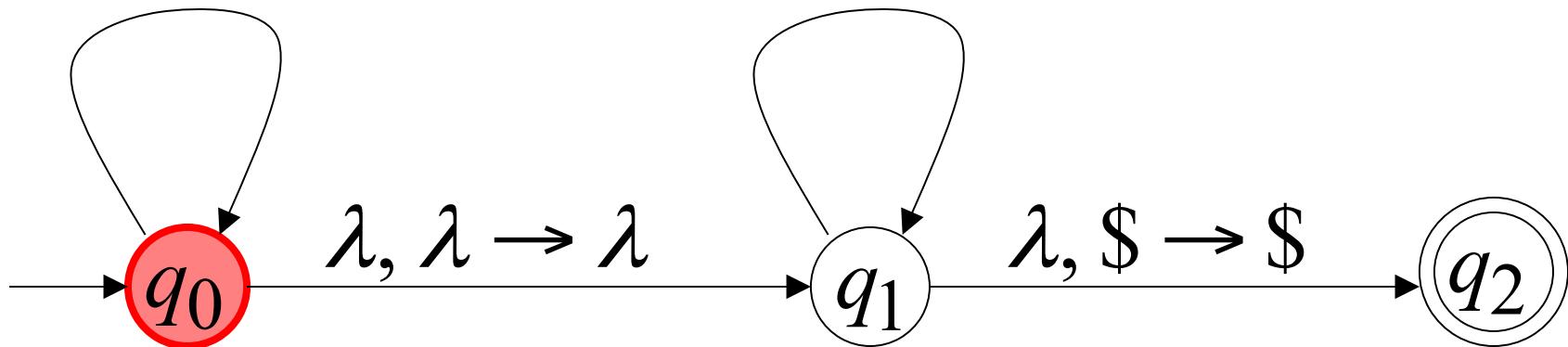
$$a, \lambda \rightarrow a$$

$$a, a \rightarrow \lambda$$

$$b, \lambda \rightarrow b$$

$$b, b \rightarrow \lambda$$

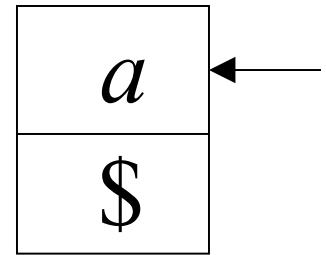
Stack



Time 1

Input

a	b	b	b
-----	-----	-----	-----



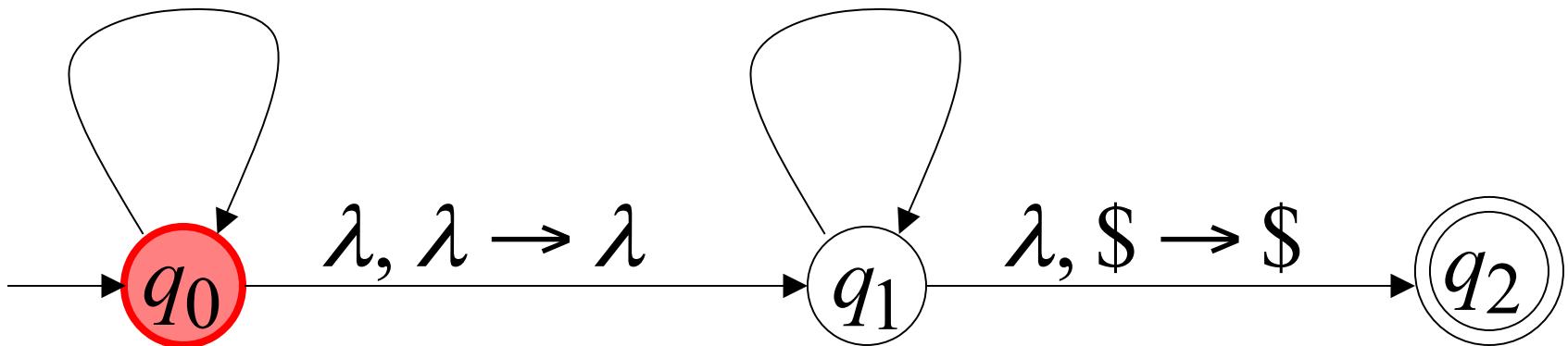
Stack

$$a, \lambda \rightarrow a$$

$$b, \lambda \rightarrow b$$

$$a, a \rightarrow \lambda$$

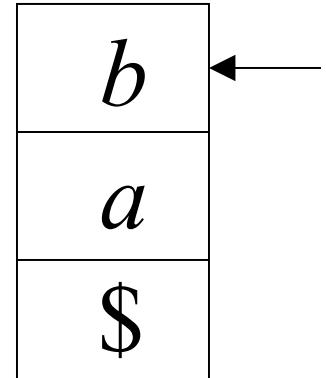
$$b, b \rightarrow \lambda$$



Time 2

Input

a	b	b	b
---	---	---	---



Stack

$$a, \lambda \rightarrow a$$

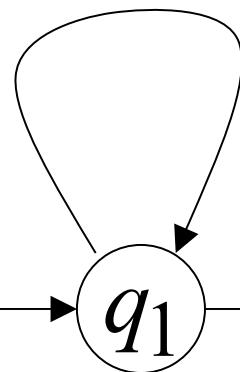
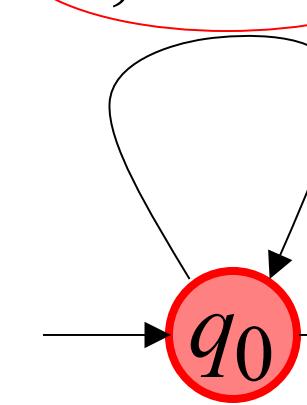
$$b, \lambda \rightarrow b$$

$$a, a \rightarrow \lambda$$

$$b, b \rightarrow \lambda$$

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

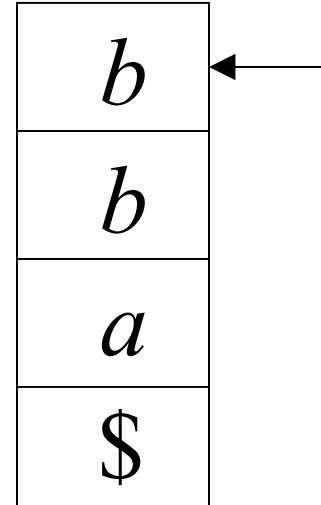
$$\lambda, \$ \rightarrow \$$$



Time 3

Input

a	b	b	b
-----	-----	-----	-----



Stack

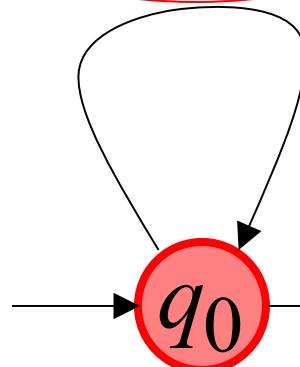
$$a, \lambda \rightarrow a$$

$$b, \lambda \rightarrow b$$

$$a, a \rightarrow \lambda$$

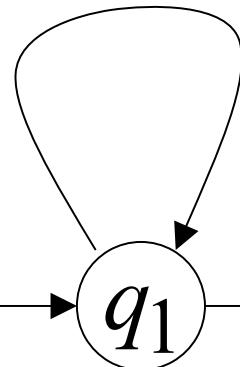
$$b, b \rightarrow \lambda$$

q_0



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

q_1



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

q_2

Input

a	b	b	b
-----	-----	-----	-----

Time 4

b
b
b
a
\$

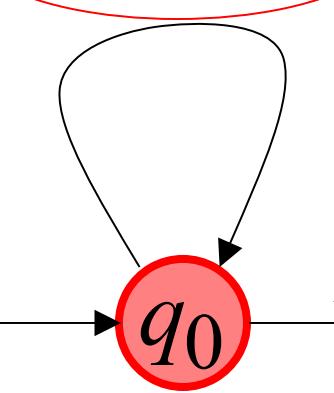
Stack

$$a, \lambda \rightarrow a$$

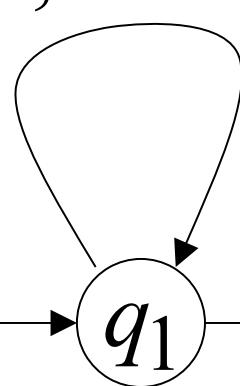
$$b, \lambda \rightarrow b$$

$$a, a \rightarrow \lambda$$

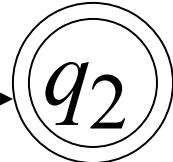
$$b, b \rightarrow \lambda$$



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



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



Input

a	b	b	b
---	---	---	---

Time 5

No accept state
is reached

b
b
b
a
\$

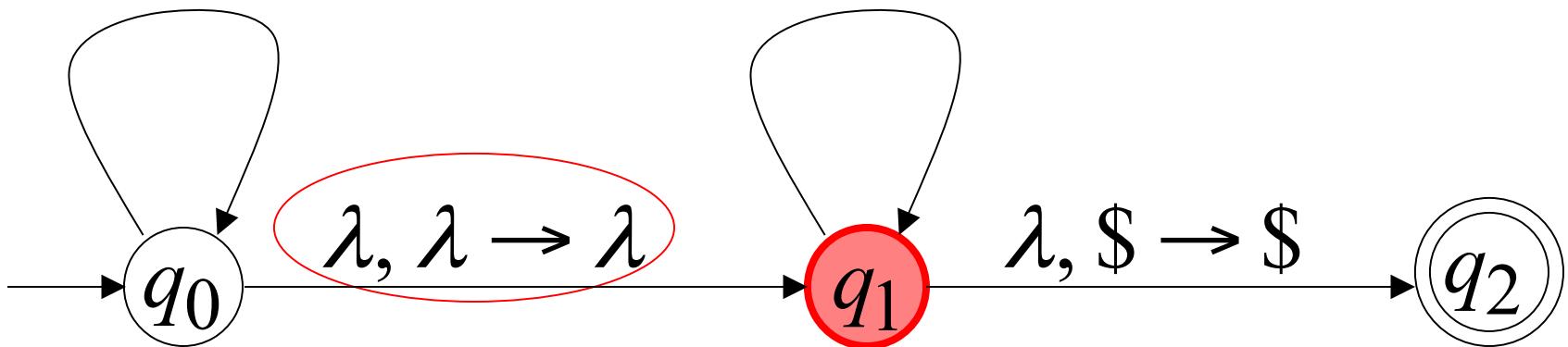
Stack

$$a, \lambda \rightarrow a$$

$$a, a \rightarrow \lambda$$

$$b, \lambda \rightarrow b$$

$$b, b \rightarrow \lambda$$



There is no computation
that accepts string $abbb$

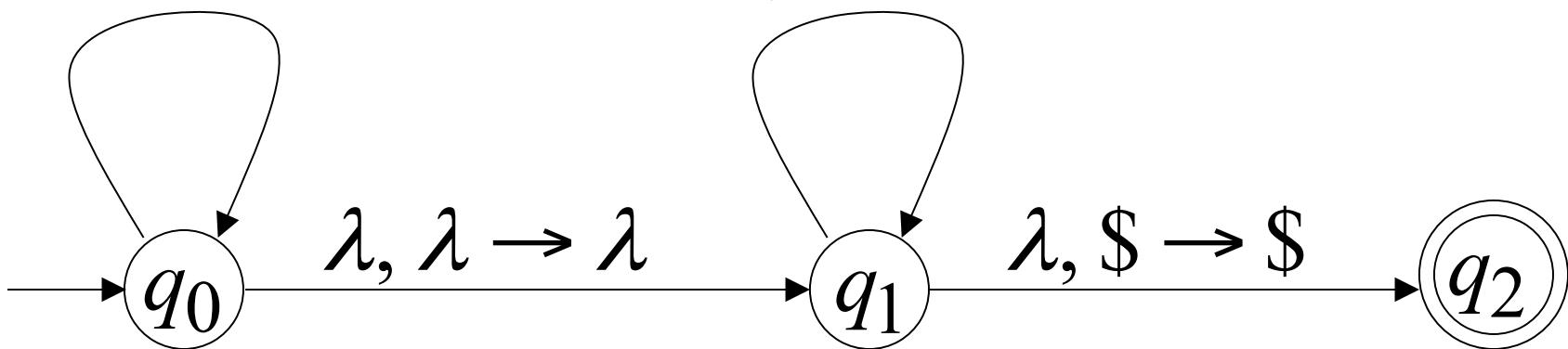
$$abbb \notin L(M)$$

$$a, \lambda \rightarrow a$$

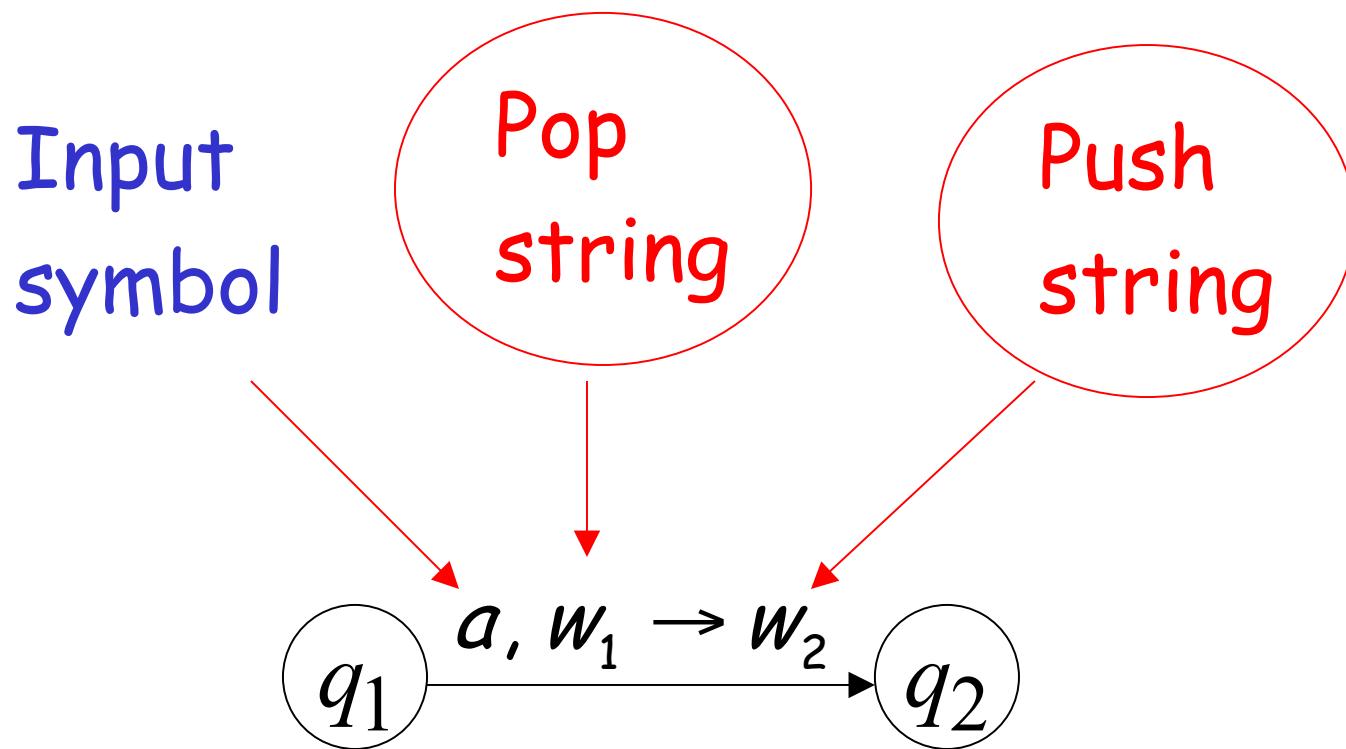
$$a, a \rightarrow \lambda$$

$$b, \lambda \rightarrow b$$

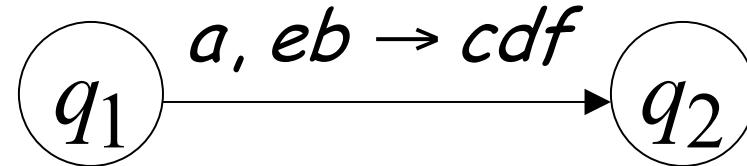
$$b, b \rightarrow \lambda$$



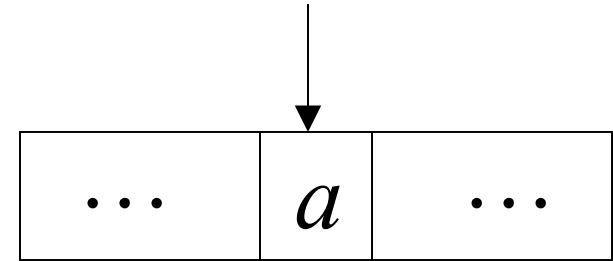
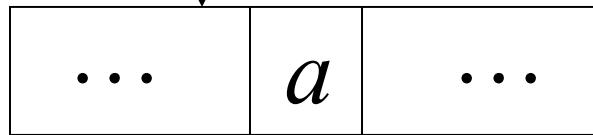
Pushing & Popping Strings



Example:

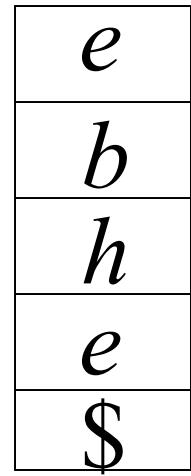


input



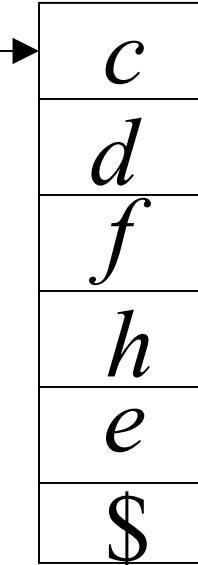
stack

pop
string

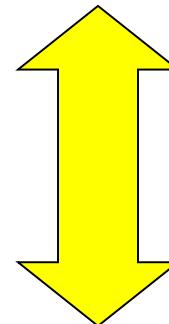
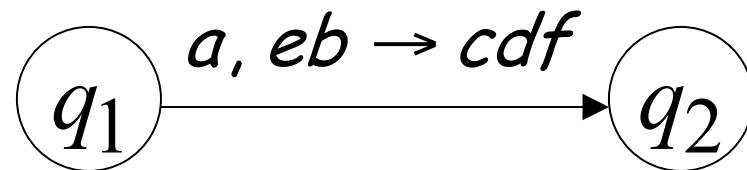


Replace

top

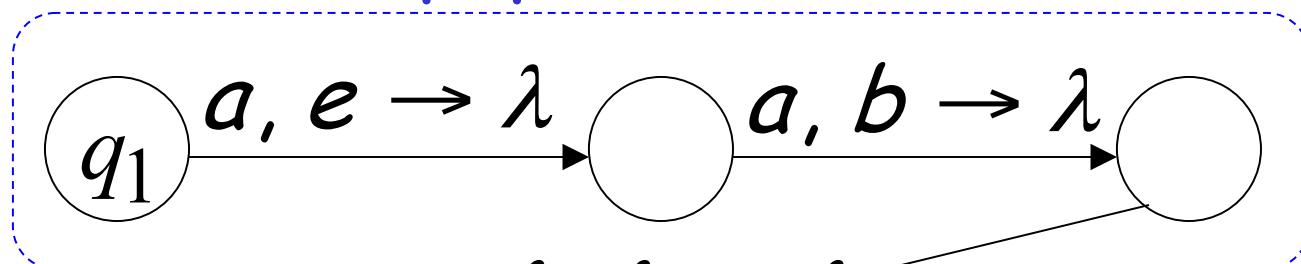


push
string

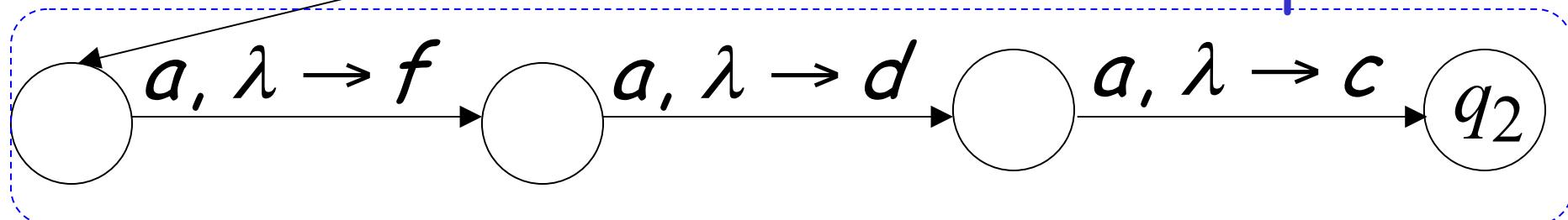


Equivalent
transitions

pop



push



Another PDA example

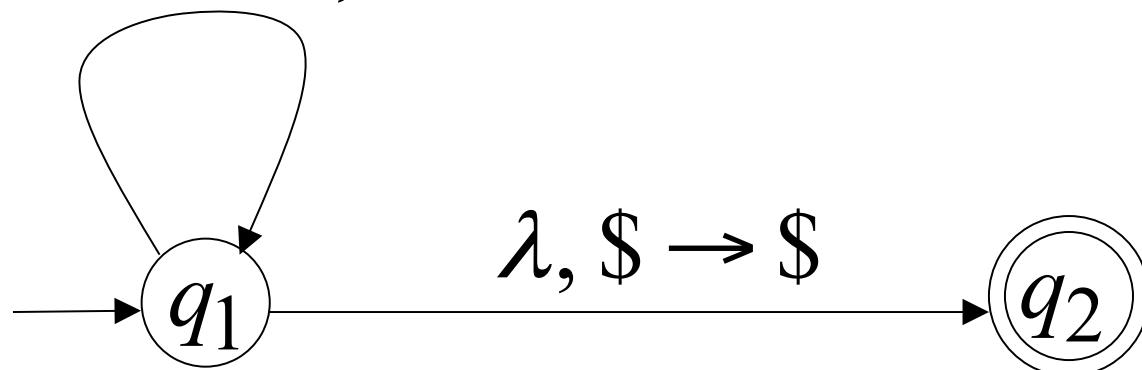
$$L(M) = \{w \in \{a, b\}^*: n_a(w) = n_b(w)\}$$

PDA M

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

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

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



Execution Example:

Time 0

Input

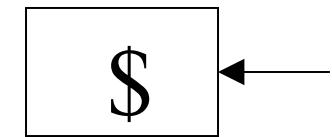
a	b	b	b	a	a
-----	-----	-----	-----	-----	-----



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

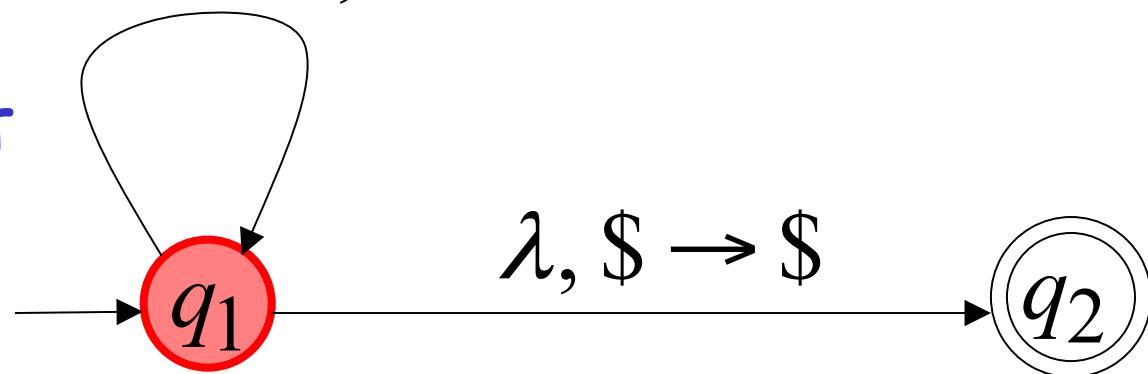
$a, 0 \rightarrow 00$ $b, 1 \rightarrow 11$

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



Stack

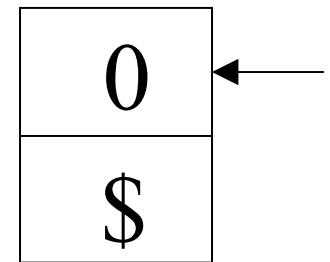
current
state



Time 1

Input

a	b	b	b	a	a
-----	-----	-----	-----	-----	-----



Stack

$a, \$ \rightarrow 0\$$

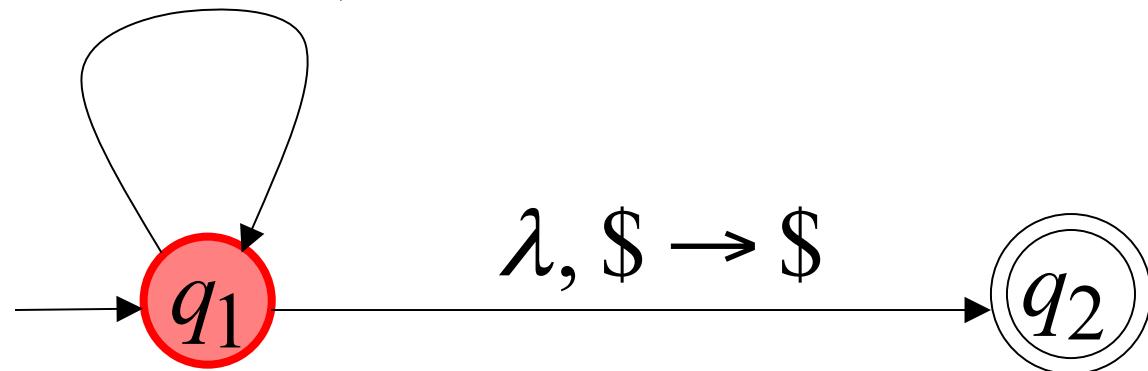
$b, \$ \rightarrow 1\$$

$a, 0 \rightarrow 00$

$b, 1 \rightarrow 11$

$a, 1 \rightarrow \lambda$

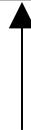
$b, 0 \rightarrow \lambda$



Time 3

Input

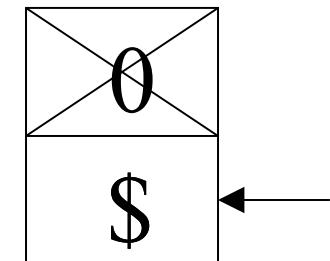
a	b	b	b	a	a
-----	-----	-----	-----	-----	-----



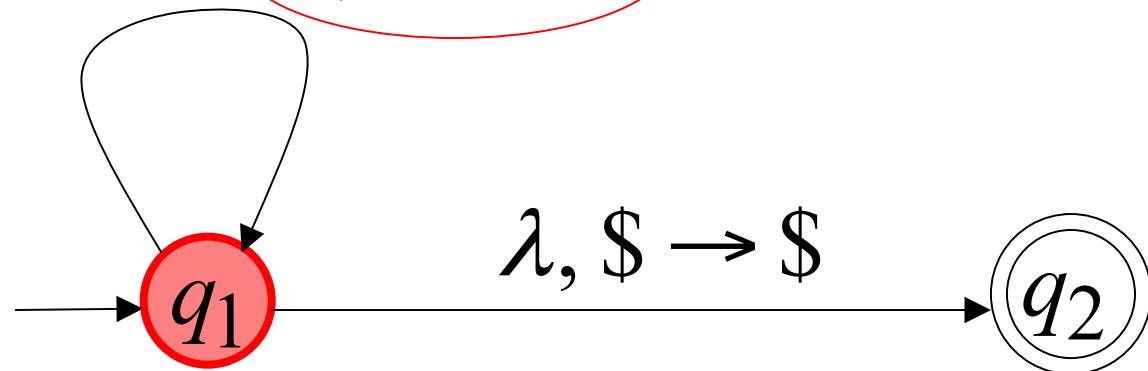
$$a, \$ \rightarrow 0\$ \quad b, \$ \rightarrow 1\$$$

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

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



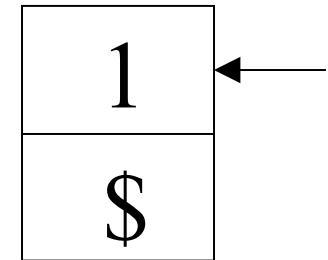
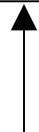
Stack



Time 4

Input

a	b	b	b	a	a
-----	-----	-----	-----	-----	-----



Stack

$$a, \$ \rightarrow 0\$$$

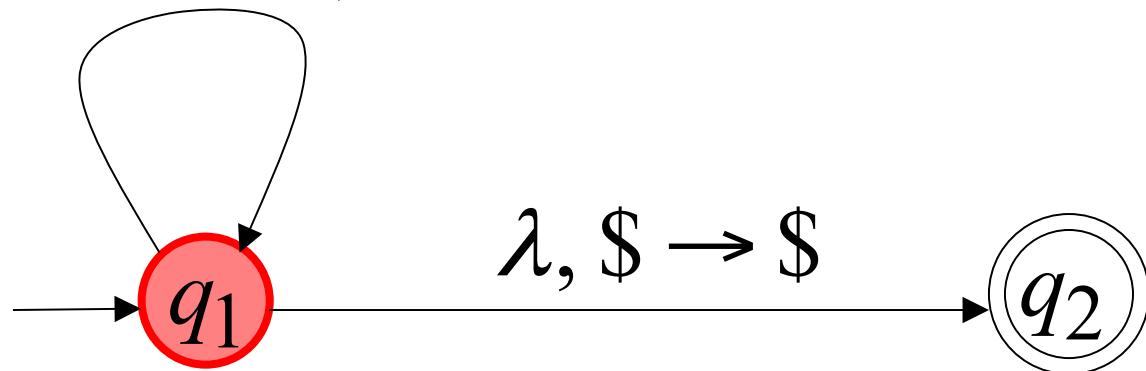
$$a, 0 \rightarrow 00$$

$$a, 1 \rightarrow \lambda$$

$$b, \$ \rightarrow 1\$$$

$$b, 1 \rightarrow 11$$

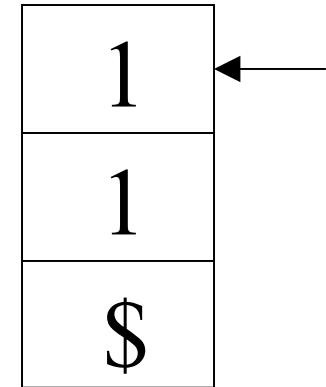
$$b, 0 \rightarrow \lambda$$



Time 5

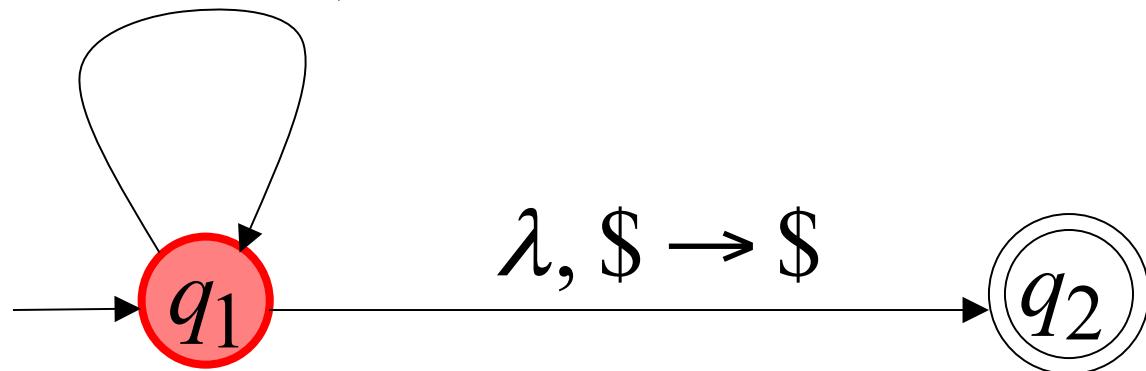
Input

a	b	b	b	a	a
-----	-----	-----	-----	-----	-----



Stack

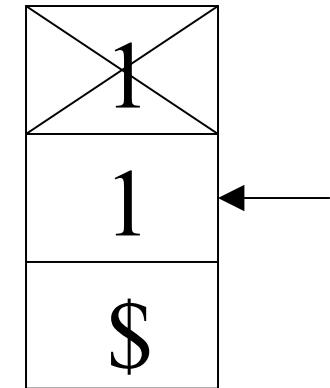
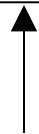
$$\begin{array}{ll} a, \$ \rightarrow 0\$ & b, \$ \rightarrow 1\$ \\ a, 0 \rightarrow 00 & b, 1 \rightarrow 11 \\ a, 1 \rightarrow \lambda & b, 0 \rightarrow \lambda \end{array}$$



Time 6

Input

a	b	b	b	a	a
-----	-----	-----	-----	-----	-----

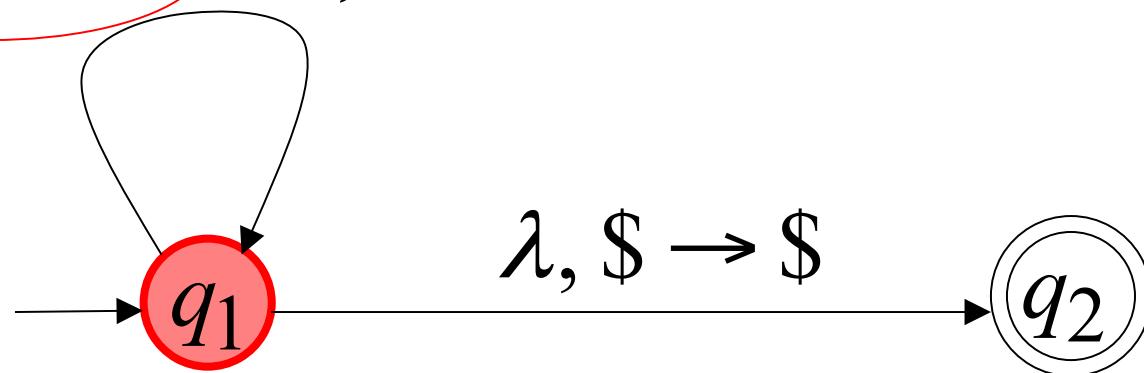


Stack

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

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

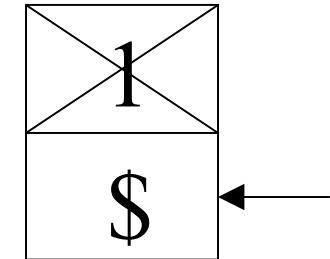
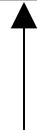
$$a, 1 \rightarrow \lambda \quad b, 0 \rightarrow \lambda$$



Time 7

Input

a	b	b	b	a	a
-----	-----	-----	-----	-----	-----

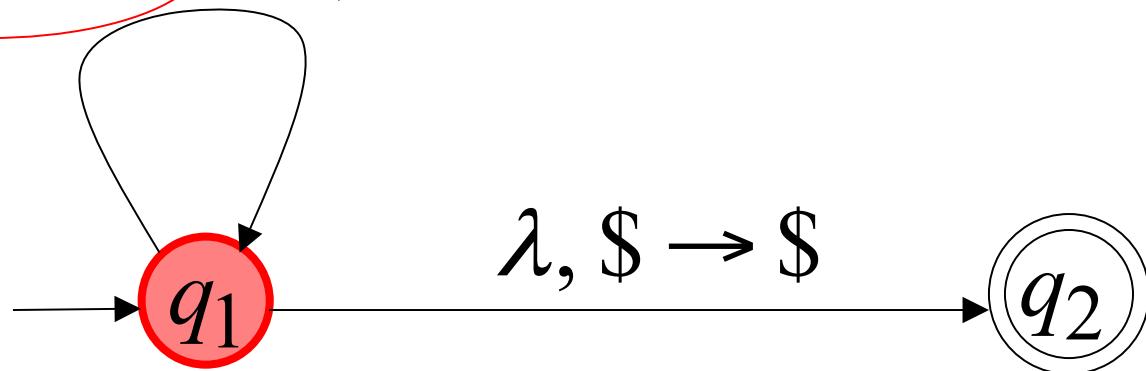


Stack

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

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

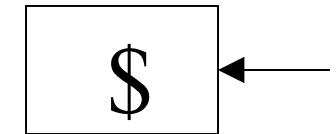
$$a, 1 \rightarrow \lambda \quad b, 0 \rightarrow \lambda$$



Time 8

Input

a	b	b	b	a	a
-----	-----	-----	-----	-----	-----

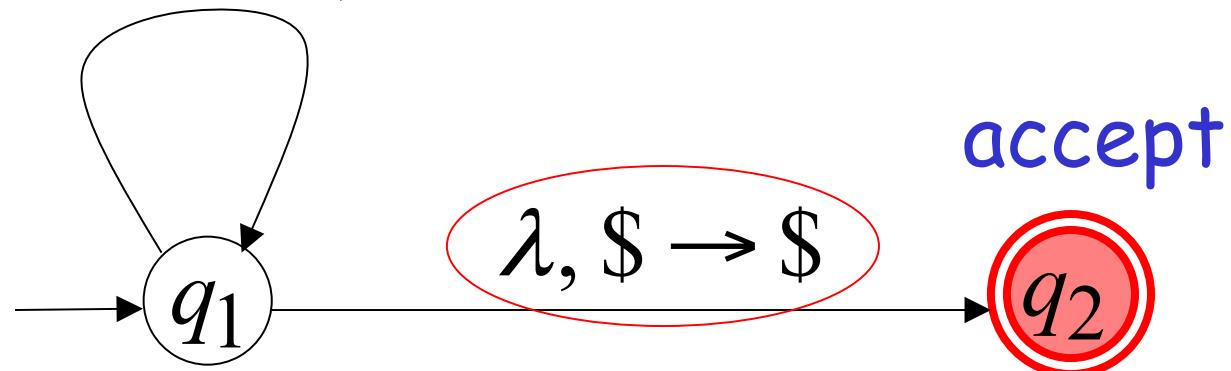


$$a, \$ \rightarrow 0\$ \quad b, \$ \rightarrow 1\$$$

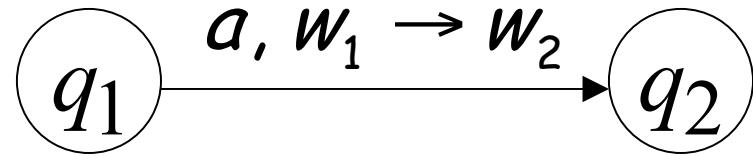
$$a, 0 \rightarrow 00 \quad b, 1 \rightarrow 11$$

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

Stack

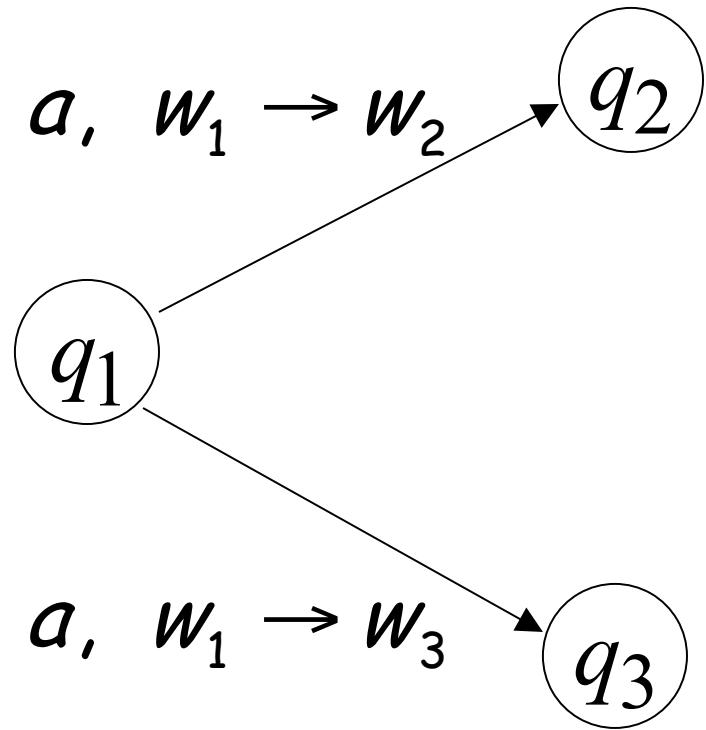


Formalities for PDAs



Transition function:

$$\delta(q_1, a, w_1) = \{(q_2, w_2)\}$$

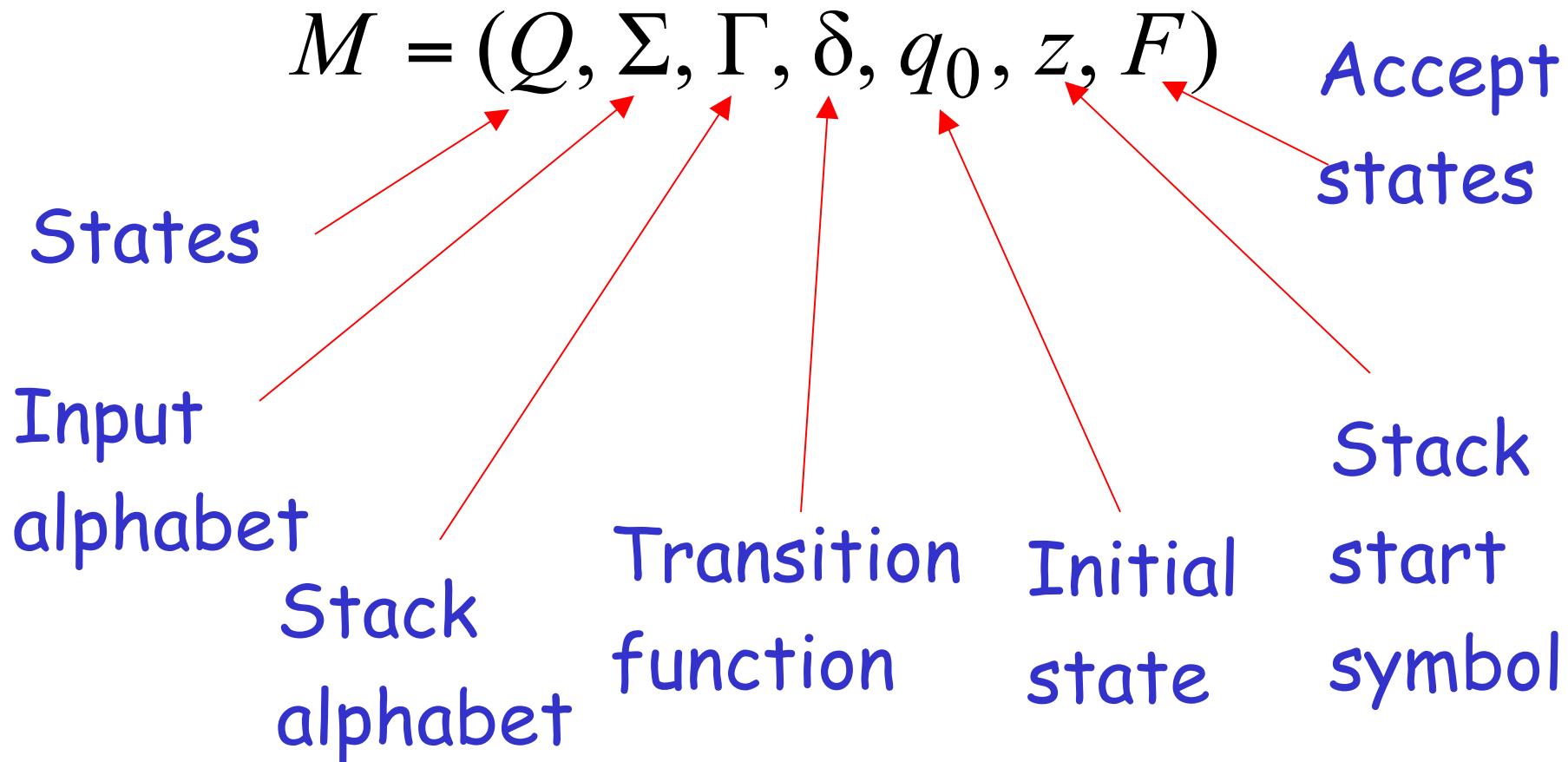


Transition function:

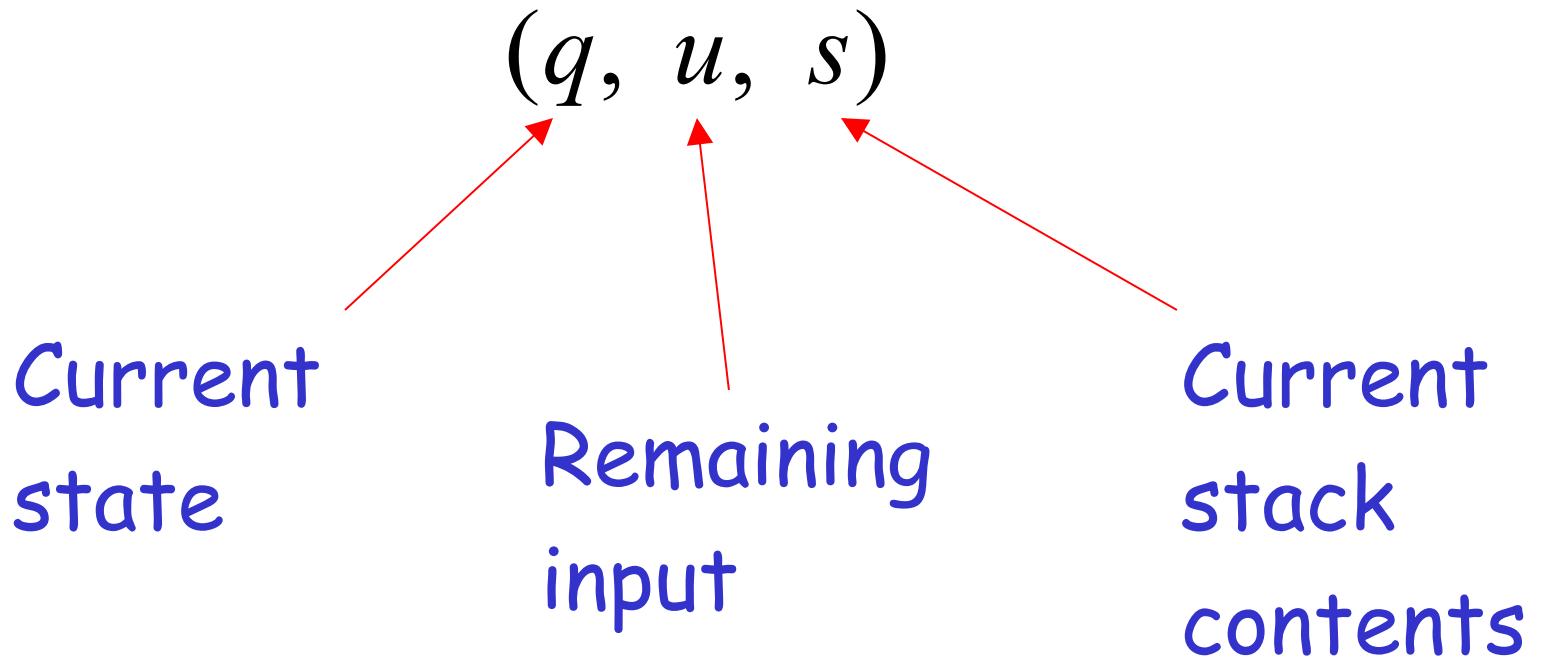
$$\delta(q_1, a, w_1) = \{(q_2, w_2), (q_3, w_3)\}$$

Formal Definition

Pushdown Automaton (PDA)



Instantaneous Description



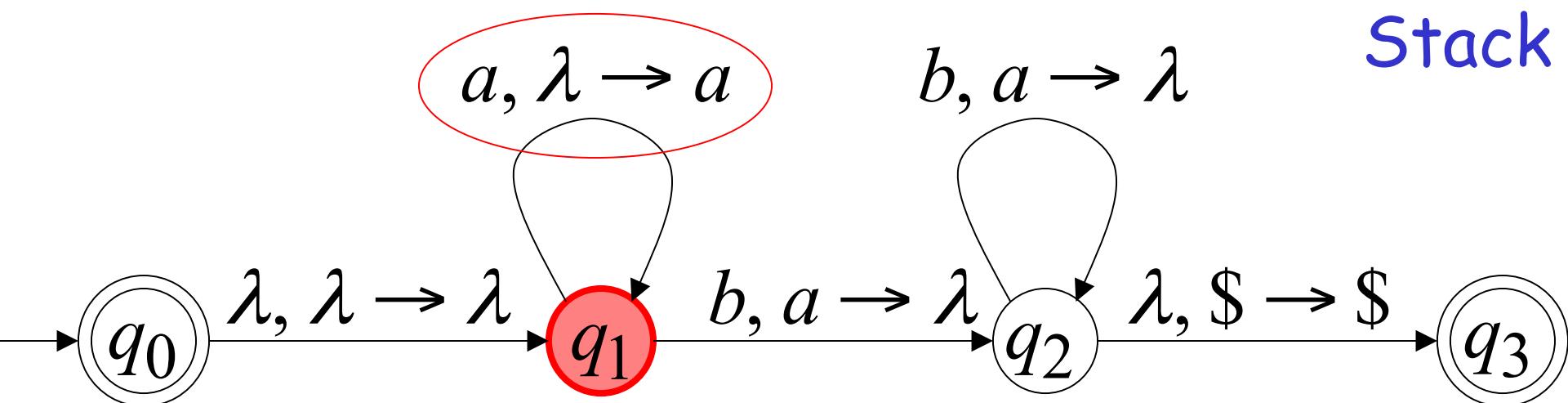
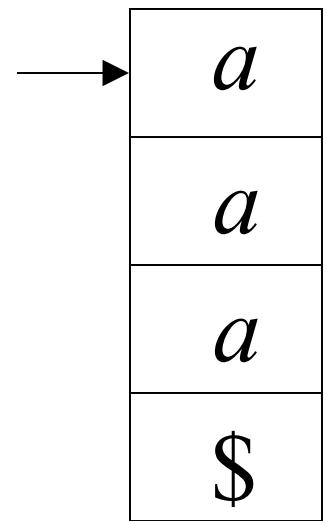
Example: Instantaneous Description

$(q_1, bbb, aaa\$)$

Time 4:

Input

a	a	a	b	b	b
---	---	---	---	---	---

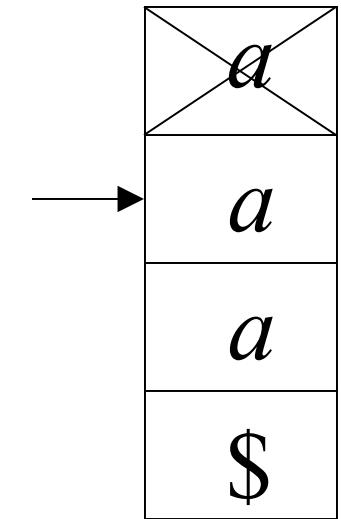
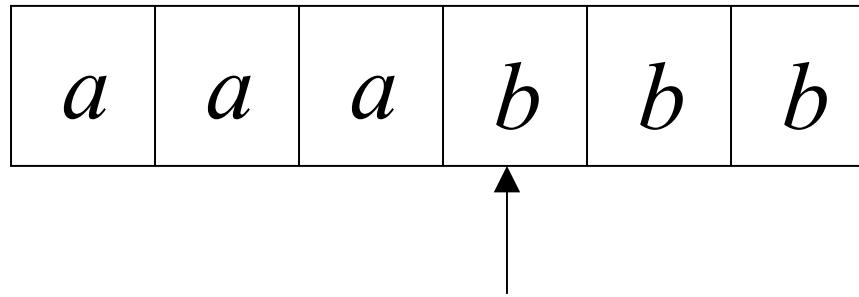


Example: Instantaneous Description

$(q_2, bb, aa\$)$

Time 5:

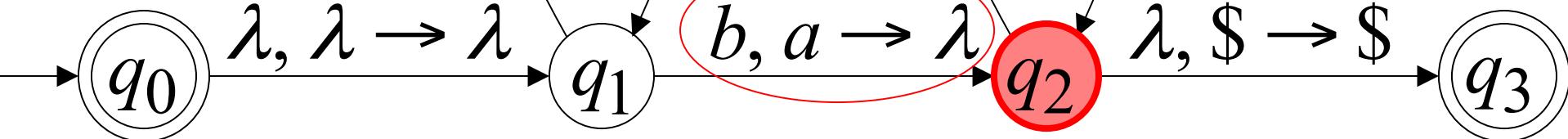
Input



Stack

$a, \lambda \rightarrow a$

$b, a \rightarrow \lambda$



We write:

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

Time 4

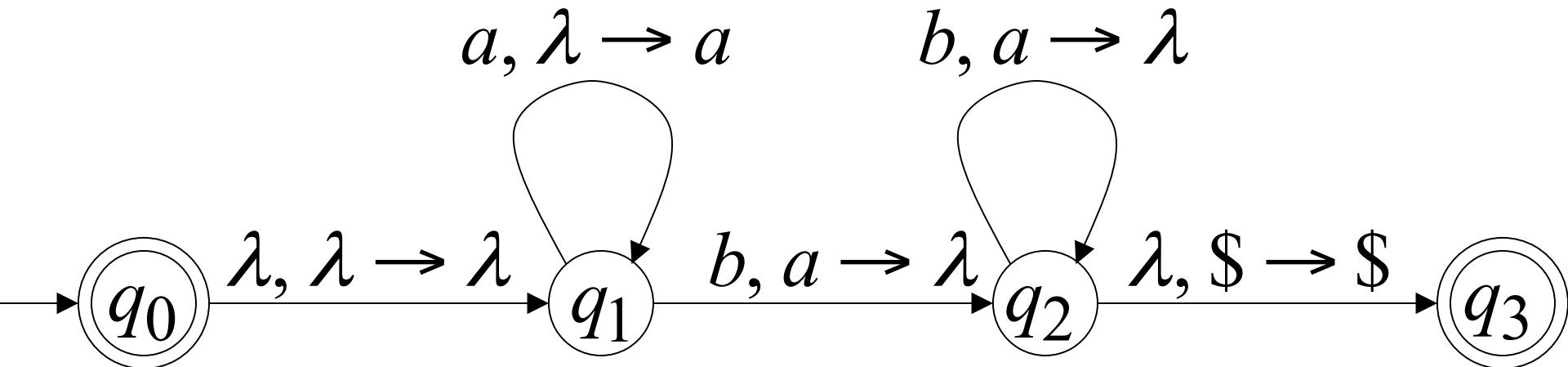
Time 5

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:

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

Language of PDA

Language $L(M)$ accepted by PDA M :

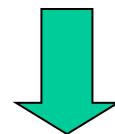
$$L(M) = \{w : (q_0, w, z) \xrightarrow{*} (q_f, \lambda, s)\}$$

Initial state

Accept state

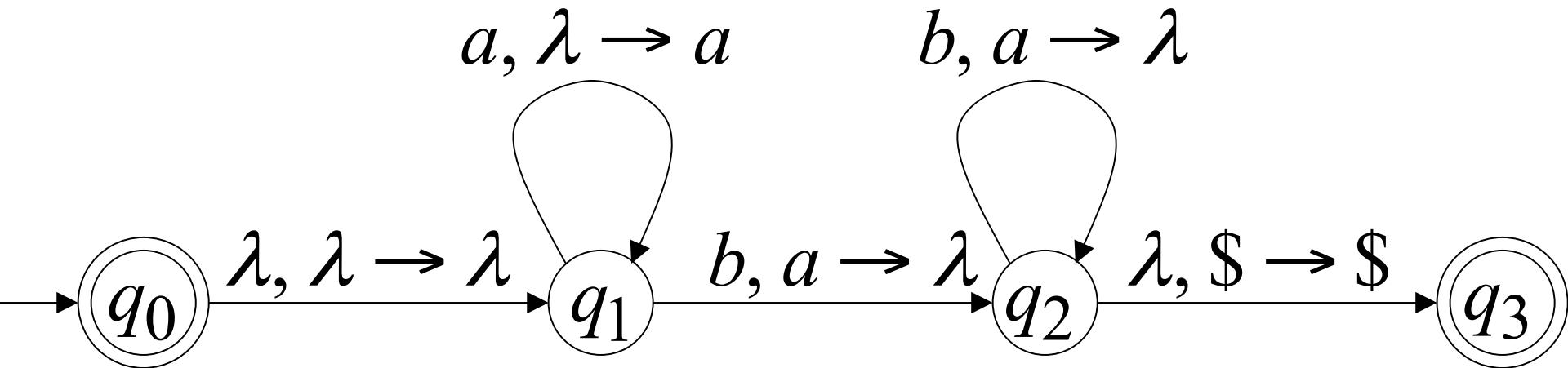
Example:

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

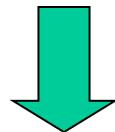


$$aaabbb \in L(M)$$

PDA M :

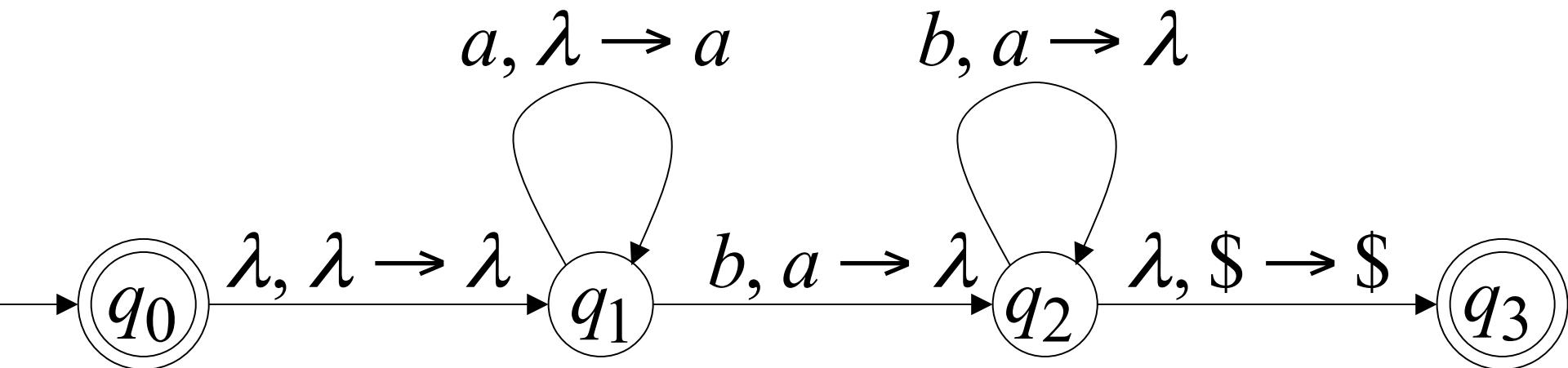


$$(q_0, a^n b^n, \$) \xrightarrow{*} (q_3, \lambda, \$)$$



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

PDA M :



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

PDA M :

