# Formal Languages

# PDAs accept context-free languages

**Theorem:**

$$\left\{\begin{array}{c}\text{Context-Free}\\\text{Languages}\\\text{(Grammars)}\end{array}\right\} = \left\{\begin{array}{c}\text{Languages}\\\text{Accepted by}\\\text{PDAs}\end{array}\right\}$$

# Proof - Step 1:

$$\left\{\begin{array}{c}\text{Context-Free} \\ \text{Languages} \\ \text{(Grammars)}\end{array}\right\} \subseteq \left\{\begin{array}{c}\text{Languages} \\ \text{Accepted by} \\ \text{PDAs}\end{array}\right\}$$

Convert any context-free grammar $G$ to a PDA $M$ with: $L(G) = L(M)$

# Proof - Step 2:

$$\left\{\begin{array}{c}\text{Context-Free}\\\text{Languages}\\\text{(Grammars)}\end{array}\right\} \supseteq \left\{\begin{array}{c}\text{Languages}\\\text{Accepted by}\\\text{PDAs}\end{array}\right\}$$

Convert any PDA $M$ to a context-free grammar $G$ with: $L(G) = L(M)$

# Proof - step 1

*Converting*

Context-Free Grammars
to
PDAs

$$\left\{\begin{array}{c}\text{Context-Free} \\ \text{Languages} \\ \text{(Grammars)}\end{array}\right\} \subseteq \left\{\begin{array}{c}\text{Languages} \\ \text{Accepted by} \\ \text{PDAs}\end{array}\right\}$$

Convert any context-free grammar $G$ to a PDA $M$ with: $L(G) = L(M)$

We will convert grammar $G$

to a PDA $M$ such that:

$M$ simulates leftmost derivations of $G$

# Convert grammar $G$ to PDA $M$

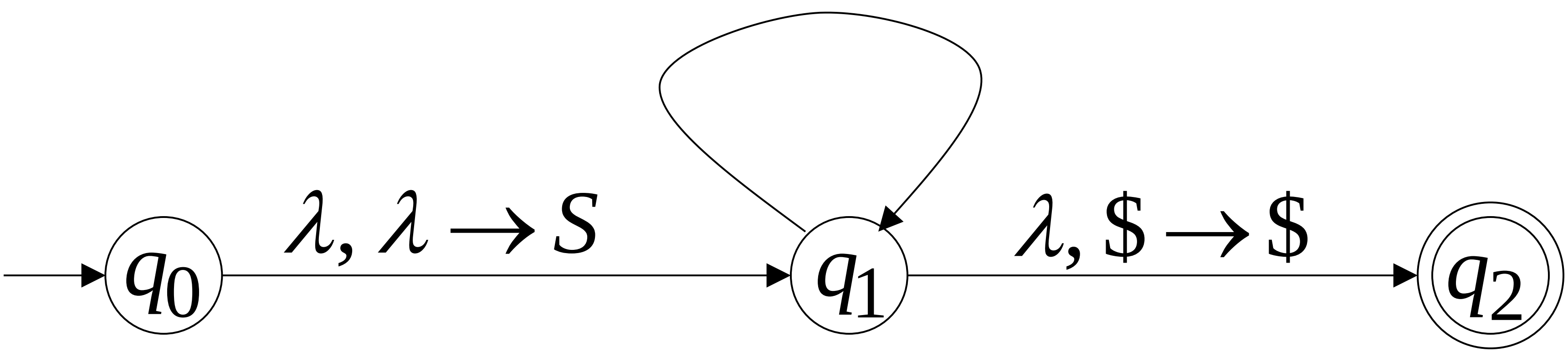
Production in $G$

$A \rightarrow w$

Terminal in $G$

$a$

$\lambda, A \rightarrow w$ $\quad\quad$ $a, a \rightarrow \lambda$



Grammar leftmost derivation

PDA computation
Simulates grammar leftmost derivations

$$(q_0, \sigma_1 \cdots \sigma_k \sigma_{k+1} \cdots \sigma_n, \$)$$

$$S \qquad \succ (q_1, \sigma_1 \cdots \sigma_k \sigma_{k+1} \cdots \sigma_n, S\$)$$

$$\Rightarrow \cdots \qquad \succ \cdots$$

$$\Rightarrow \sigma_1 \cdots \sigma_k X_1 \cdots X_m \qquad \succ (q_1, \sigma_{k+1} \cdots \sigma_n, X_1 \cdots X_m \$)$$

$$\Rightarrow \cdots \qquad \succ \cdots$$

$$\Rightarrow \sigma_1 \cdots \sigma_k \sigma_{k+1} \cdots \sigma_n \qquad \succ (q_2, \lambda, \$)$$

Leftmost variable

# Example

**Grammar**

$S \to aSTb$

$S \to b$

$T \to Ta$

$T \to \lambda$

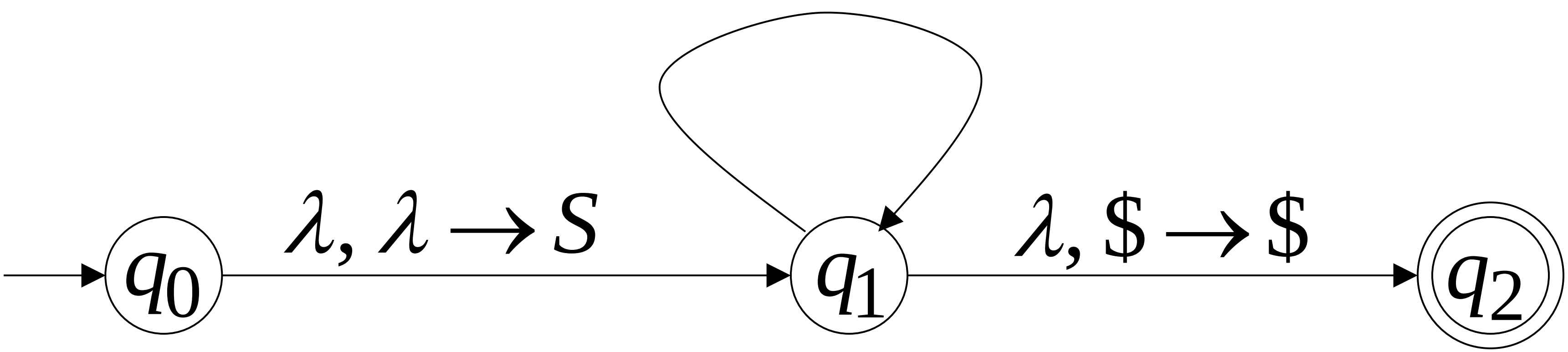
**PDA**

$\lambda, S \to aSTb$

$\lambda, S \to b$

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

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



## Grammar derivation

## PDA computation

$$S$$

$$(q_0, abab, \$)$$

$$\succ (q_1, abab, S\$)$$

$$\Rightarrow aSTb$$

$$\succ (q_1, bab, STb\$)$$

$$\Rightarrow abTb$$

$$\succ (q_1, bab, bTb\$)$$

$$\succ (q_1, ab, Tb\$)$$

$$\Rightarrow abTab$$

$$\succ (q_1, ab, Tab\$)$$

$$\Rightarrow abab$$

$$\succ (q_1, ab, ab\$)$$

$$\succ (q_1, b, b\$)$$

$$\succ (q_1, \lambda, \$)$$

$$\succ (q_2, \lambda, \$)$$

# Derivation:

Input

| | | | |
|---|---|---|---|
| $a$ | $b$ | $a$ | $b$ |

Time 0

$\lambda, S \rightarrow aSTb$

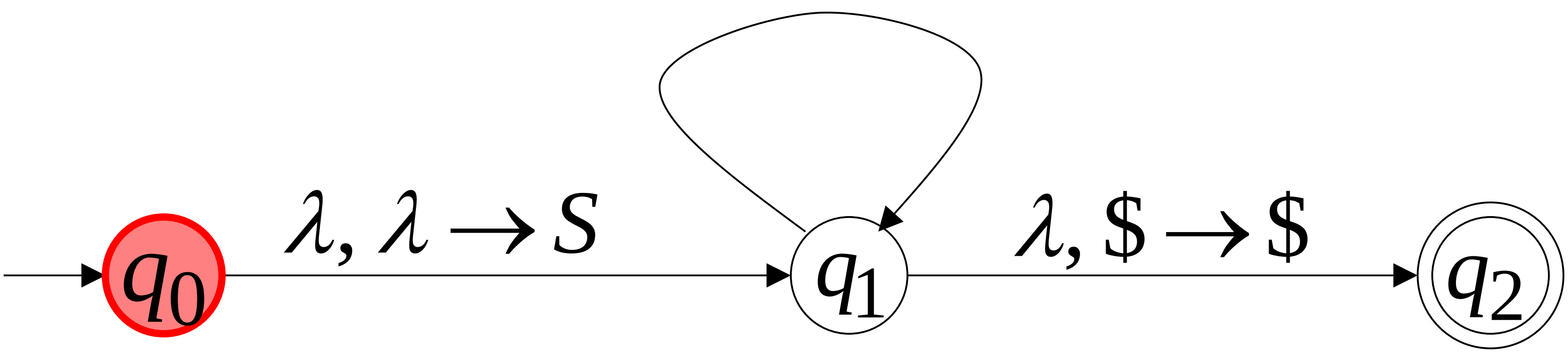
$\lambda, S \rightarrow b$

$\lambda, T \rightarrow Ta$ $\qquad a, a \rightarrow \lambda$

$\lambda, T \rightarrow \lambda$ $\qquad b, b \rightarrow \lambda$

| |
|---|
| \$ |

Stack



Derivation: $S$

Input

| | | | |
|---|---|---|---|
| $a$ | $b$ | $a$ | $b$ |

Time 0

| |
|---|
| $S$ |
| $\$$ |

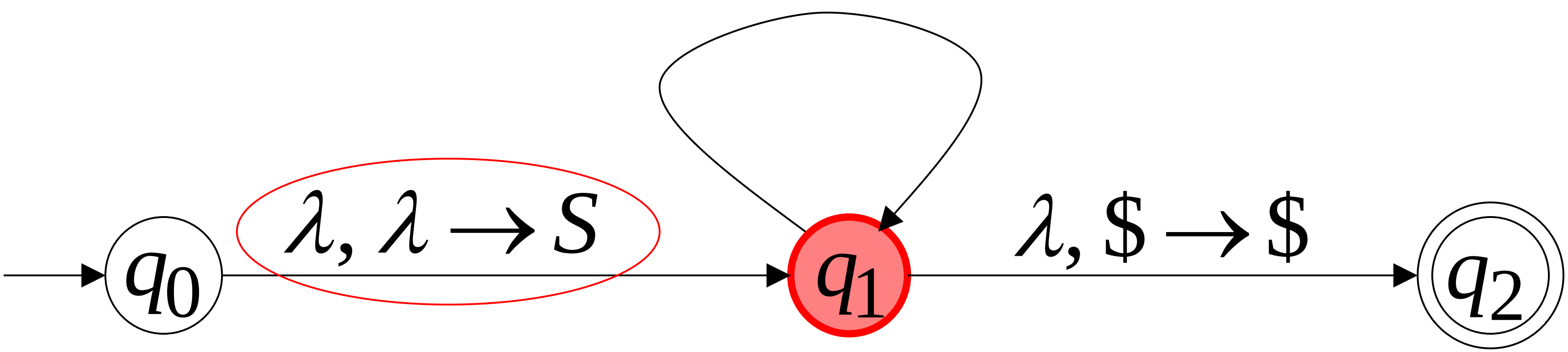
Stack

$$\lambda, S \to aSTb$$

$$\lambda, S \to b$$

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

$$\lambda, T \to \lambda \qquad b, b \to \lambda$$



Derivation: $S \Rightarrow aSTb$

Input

| | | | |
|---|---|---|---|
| $a$ | $b$ | $a$ | $b$ |

Time 1

$\lambda, S \rightarrow aSTb$

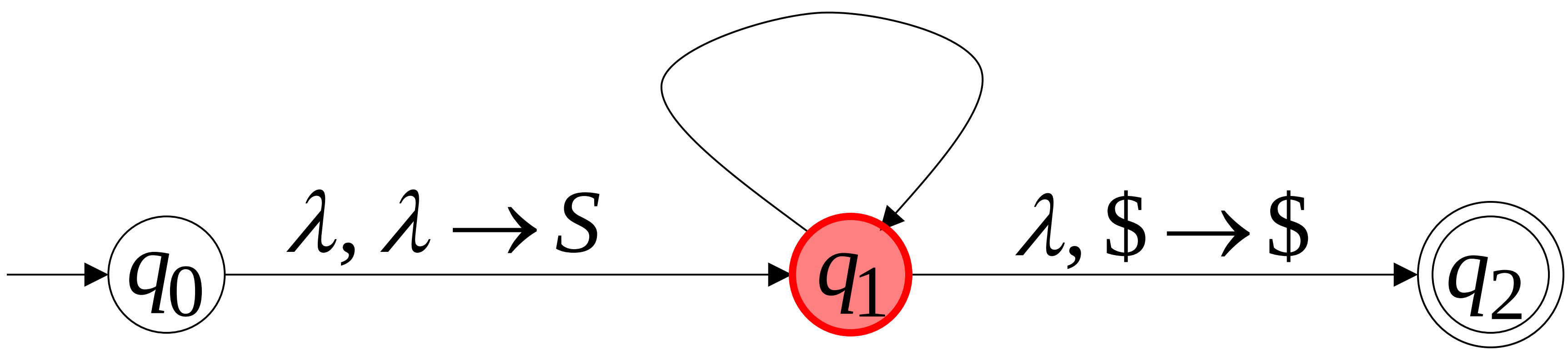
$\lambda, S \rightarrow b$

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

$\lambda, T \rightarrow \lambda$ $\quad b, b \rightarrow \lambda$

| |
|---|
| $a$ |
| $S$ |
| $T$ |
| $b$ |
| $\$$ |

Stack



Derivation: $S \Rightarrow aSTb$

Input

| | | | |
|---|---|---|---|
| $a$ | $b$ | $a$ | $b$ |

Time 2

$\lambda, S \rightarrow aSTb$

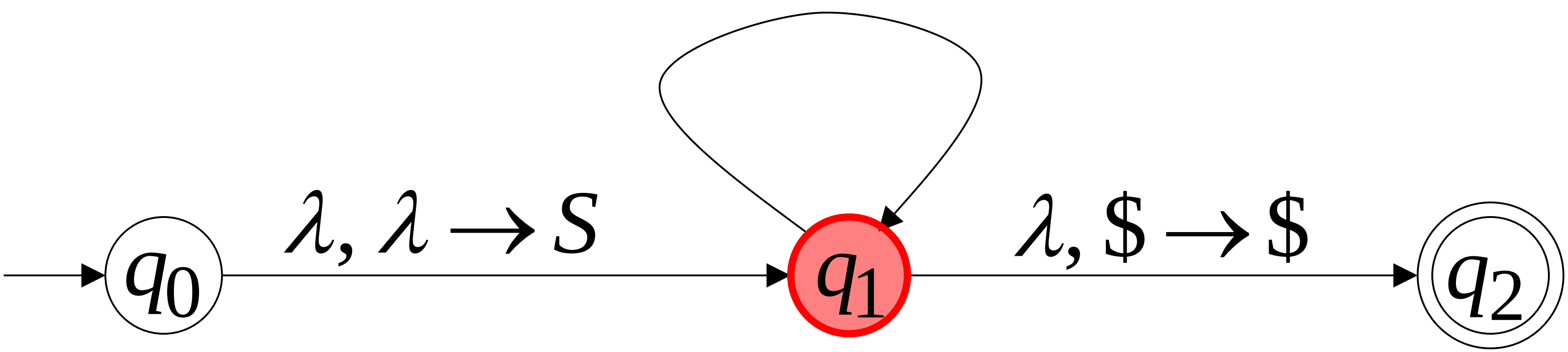
$\lambda, S \rightarrow b$

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

$\lambda, T \rightarrow \lambda$ $\quad b, b \rightarrow \lambda$

| |
|---|
| $\cancel{a}$ |
| $S$ |
| $T$ |
| $b$ |
| $\$$ |

Stack



**Derivation:** $S \Rightarrow aSTb \Rightarrow abTb$

**Input**

| | | | |
|---|---|---|---|
| $a$ | $b$ | $a$ | $b$ |

**Time 3**

$\lambda, S \rightarrow aSTb$

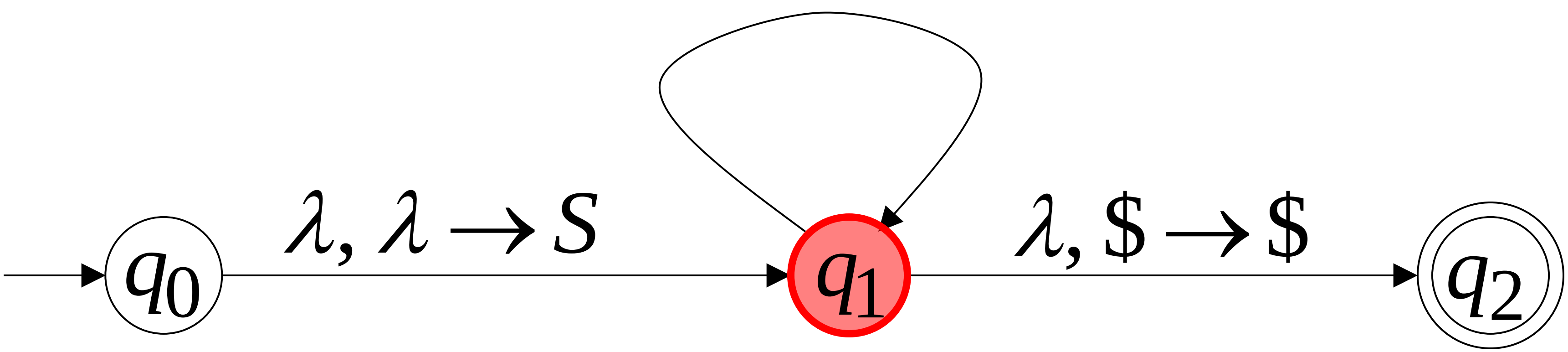
$\lambda, S \rightarrow b$

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

$\lambda, T \rightarrow \lambda$ $\quad b, b \rightarrow \lambda$

| |
|---|
| $b$ |
| $T$ |
| $b$ |
| $\$$ |

**Stack**



Derivation: $S \Rightarrow aSTb \Rightarrow abTb$

Input

| | | | |
|---|---|---|---|
| $a$ | $b$ | $a$ | $b$ |

Time 4

$$\lambda, S \rightarrow aSTb$$

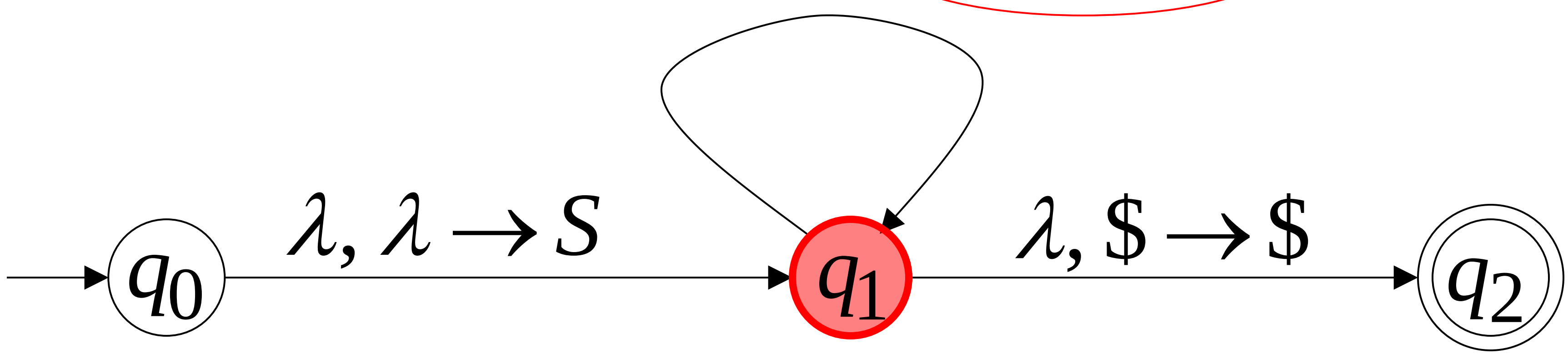
$$\lambda, S \rightarrow b$$

$$\lambda, T \rightarrow Ta \qquad a, a \rightarrow \lambda$$

$$\lambda, T \rightarrow \lambda \qquad b, b \rightarrow \lambda$$

| |
|---|
| $\cancel{b}$ |
| $T$ |
| $b$ |
| $\$$ |

Stack



Derivation: $S \Rightarrow aSTb \Rightarrow abTb \Rightarrow abTab$

Input

| | | | |
|---|---|---|---|
| $a$ | $b$ | $a$ | $b$ |

Time 5

$\lambda, S \rightarrow aSTb$

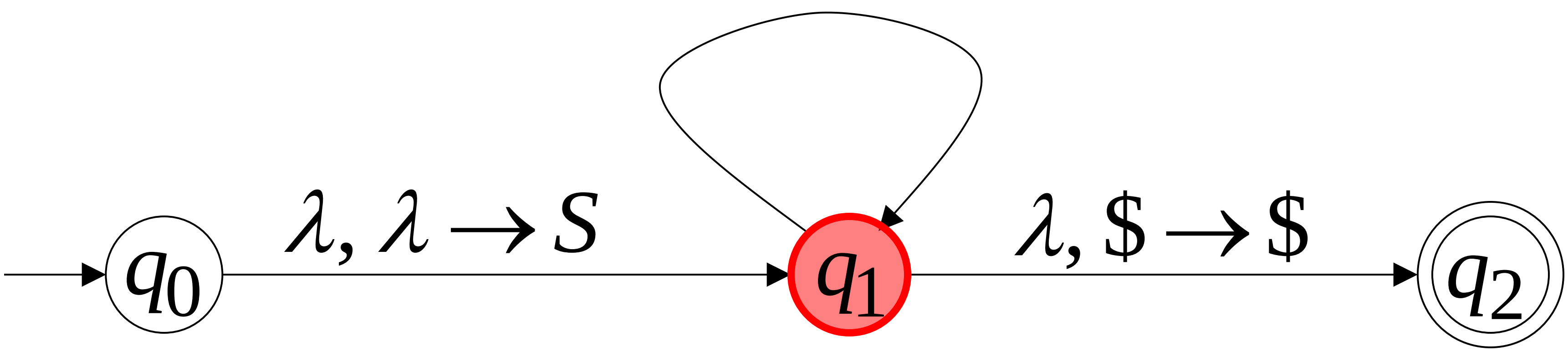
$\lambda, S \rightarrow b$

$\lambda, T \rightarrow Ta$ $\qquad a, a \rightarrow \lambda$

$\lambda, T \rightarrow \lambda$ $\qquad b, b \rightarrow \lambda$

| |
|---|
| $T$ |
| $a$ |
| $b$ |
| $\$$ |

Stack



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

Input

| | | | |
|---|---|---|---|
| $a$ | $b$ | $a$ | $b$ |

Stack

| |
|---|
| $\cancel{T}$ |
| $a$ |
| $b$ |
| $\$$ |

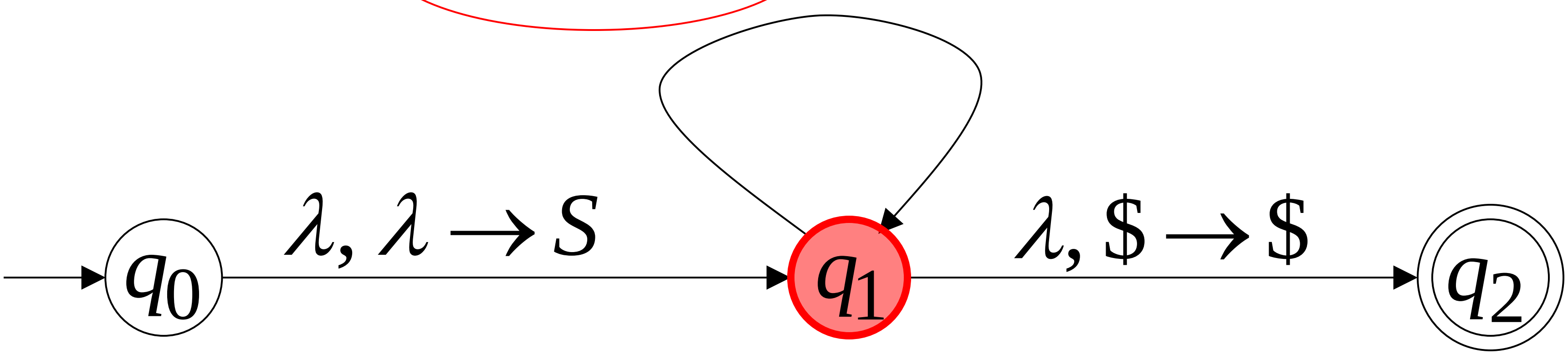
Time 6

$\lambda, S \rightarrow aSTb$

$\lambda, S \rightarrow b$

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

$\lambda, T \rightarrow \lambda$ $\quad b, b \rightarrow \lambda$



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

Input

| | | | |
|---|---|---|---|
| $a$ | $b$ | $a$ | $b$ |

Stack

| |
|---|
| $\cancel{a}$ |
| $b$ |
| $\$$ |

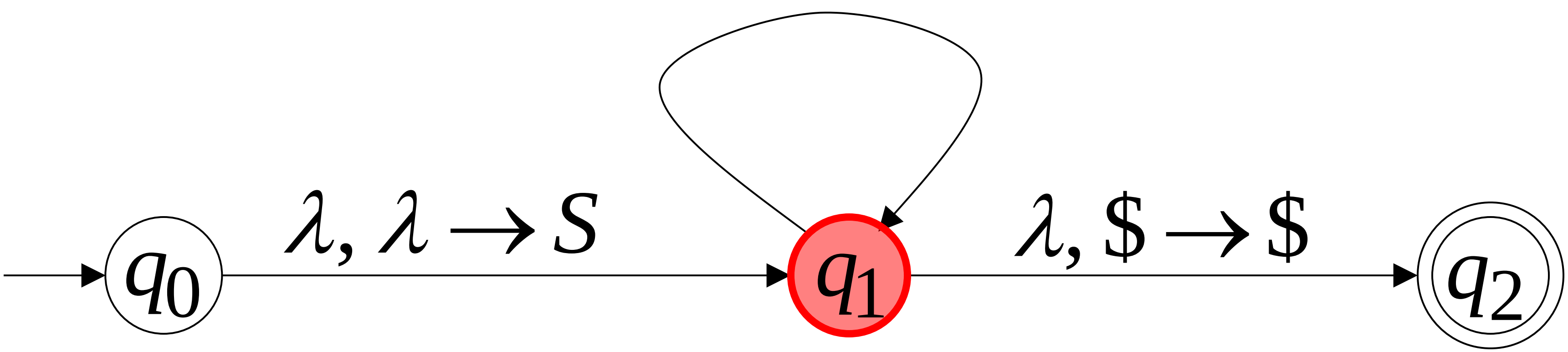
Time 7

$\lambda, S \rightarrow aSTb$

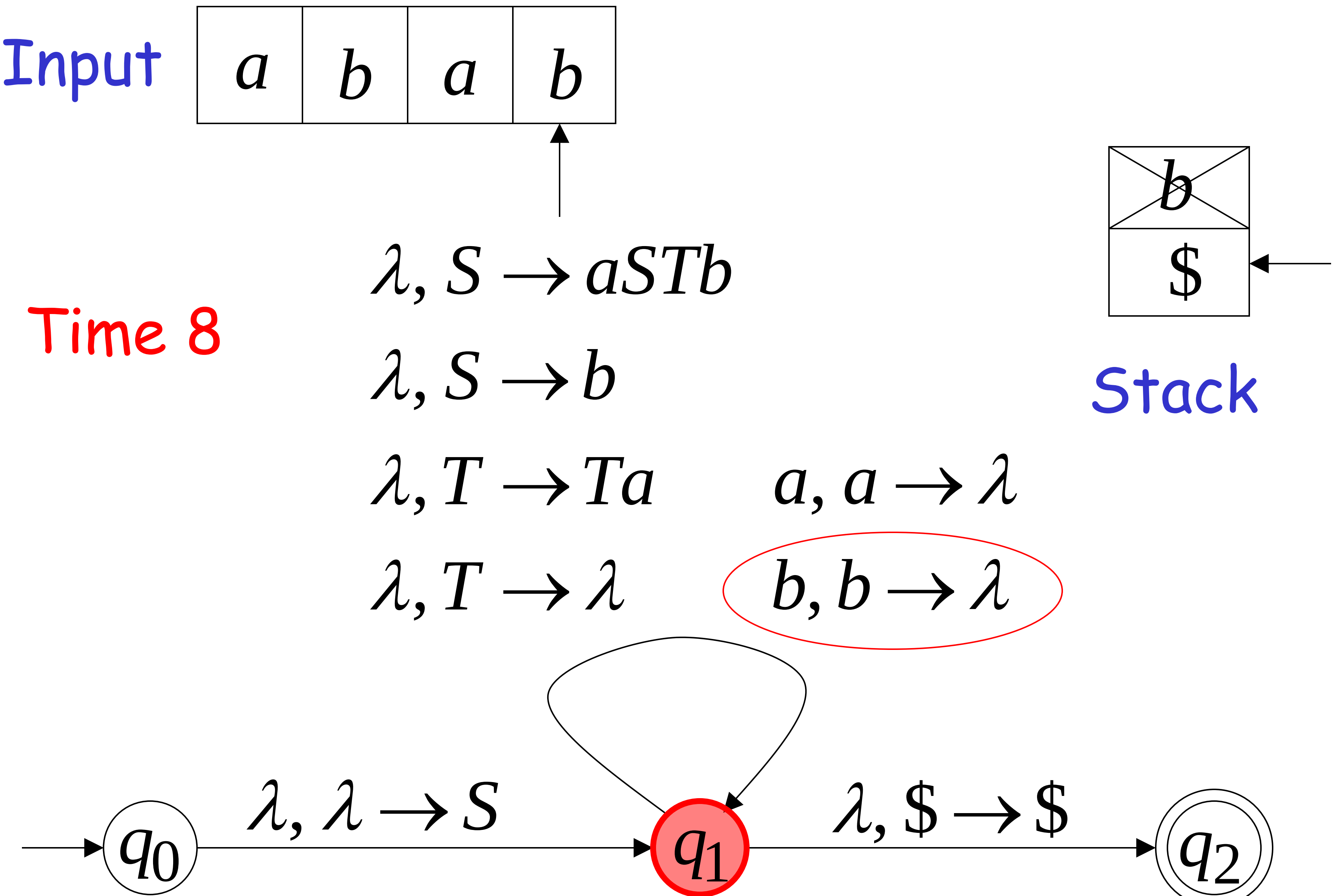
$\lambda, S \rightarrow b$

$\lambda, T \rightarrow Ta$ $\qquad a, a \rightarrow \lambda$

$\lambda, T \rightarrow \lambda$ $\qquad b, b \rightarrow \lambda$



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



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

Input

| | | | |
|---|---|---|---|
| $a$ | $b$ | $a$ | $b$ |

Time 9

$\lambda, S \to aSTb$

$\lambda, S \to b$

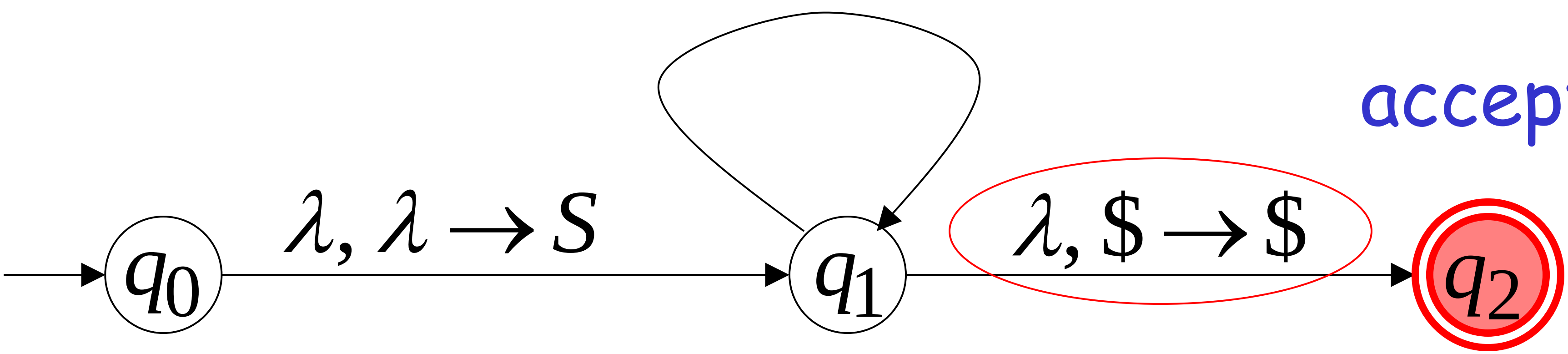
$\lambda, T \to Ta \qquad a, a \to \lambda$

$\lambda, T \to \lambda \qquad b, b \to \lambda$

| |
|---|
| $\$$ |

Stack

accept



In general, it can be shown that:

Grammar $G$
generates
string $w$

$$S \overset{*}{\Rightarrow} w$$

If and Only if

PDA $M$
accepts $w$

$$(q_0, w, \$) \succ (q_2, \lambda, \$)$$

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

## Therefore:

For any context-free language $L$
there is a PDA that accepts $L$

$$\left\{\begin{array}{c}\text{Context-Free}\\ \text{Languages}\\ \text{(Grammars)}\end{array}\right\} \subseteq \left\{\begin{array}{c}\text{Languages}\\ \text{Accepted by}\\ \text{PDAs}\end{array}\right\}$$

# Proof - step 2

Converting

PDAs
to
Context-Free Grammars

$$\left\{\begin{array}{c}\text{Context-Free}\\\text{Languages}\\\text{(Grammars)}\end{array}\right\} \supseteq \left\{\begin{array}{c}\text{Languages}\\\text{Accepted by}\\\text{PDAs}\end{array}\right\}$$

Convert any PDA $M$ to a context-free grammar $G$ with: $L(G) = L(M)$

We will convert PDA $M$ to

a context-free grammar $G$ such that:

$G$ simulates computations of $M$
with leftmost derivations

# Some Necessary Modifications

If necessary, modify the PDA so that:

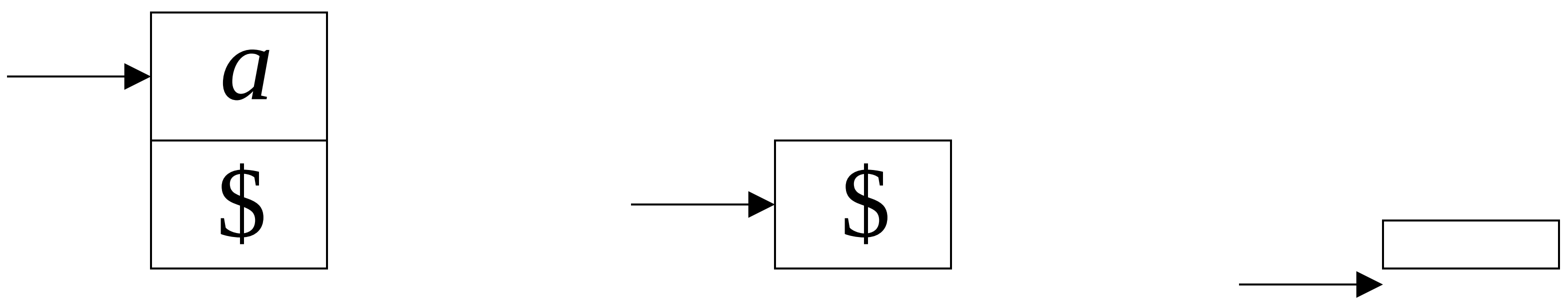
1. The stack is never empty during computation

2. It has a single accept state
   and empties the stack when it accepts a string

3. Has transitions without popping $\lambda$

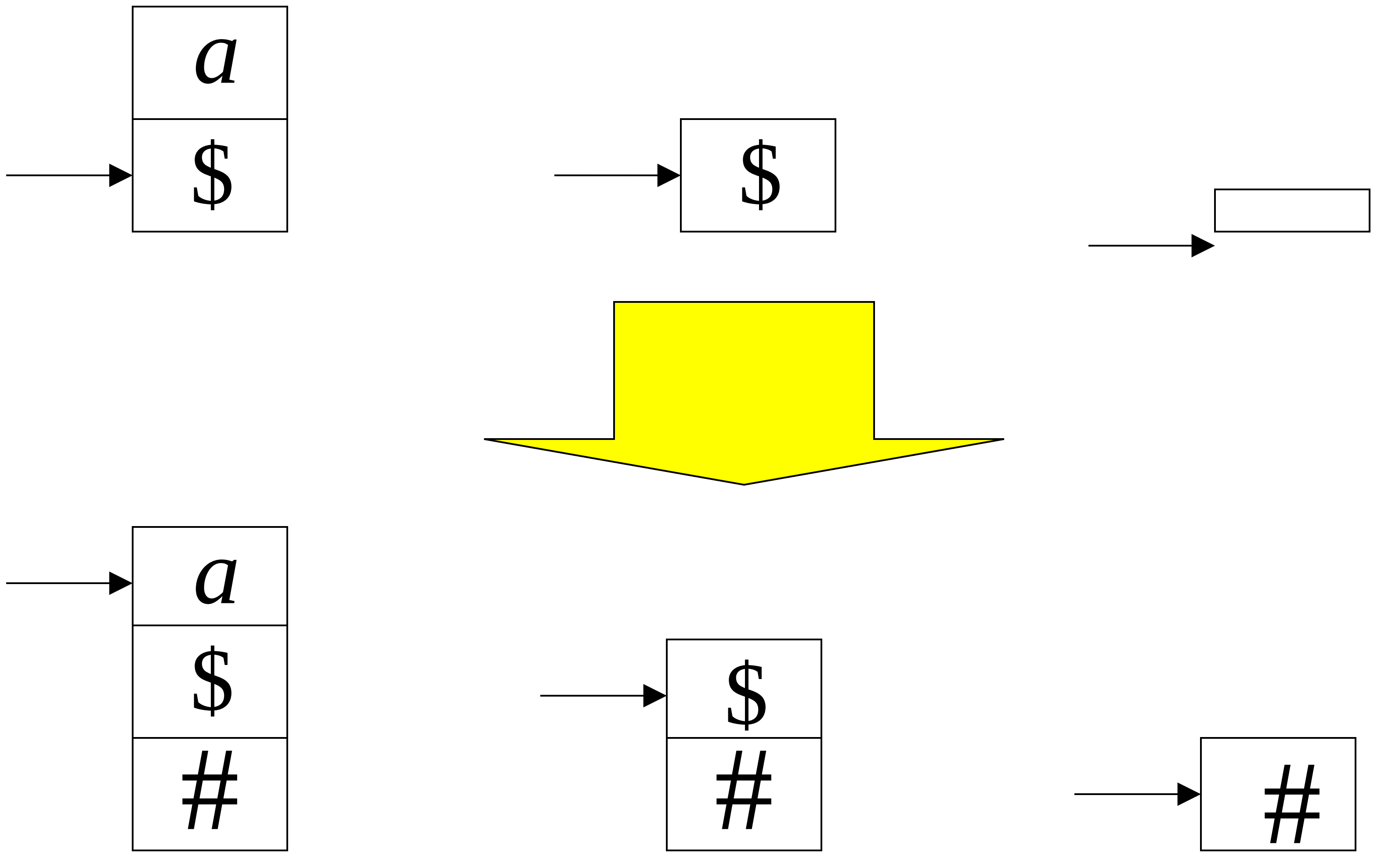
## 1. Modify the PDA so that the stack is never empty during computation

Stack

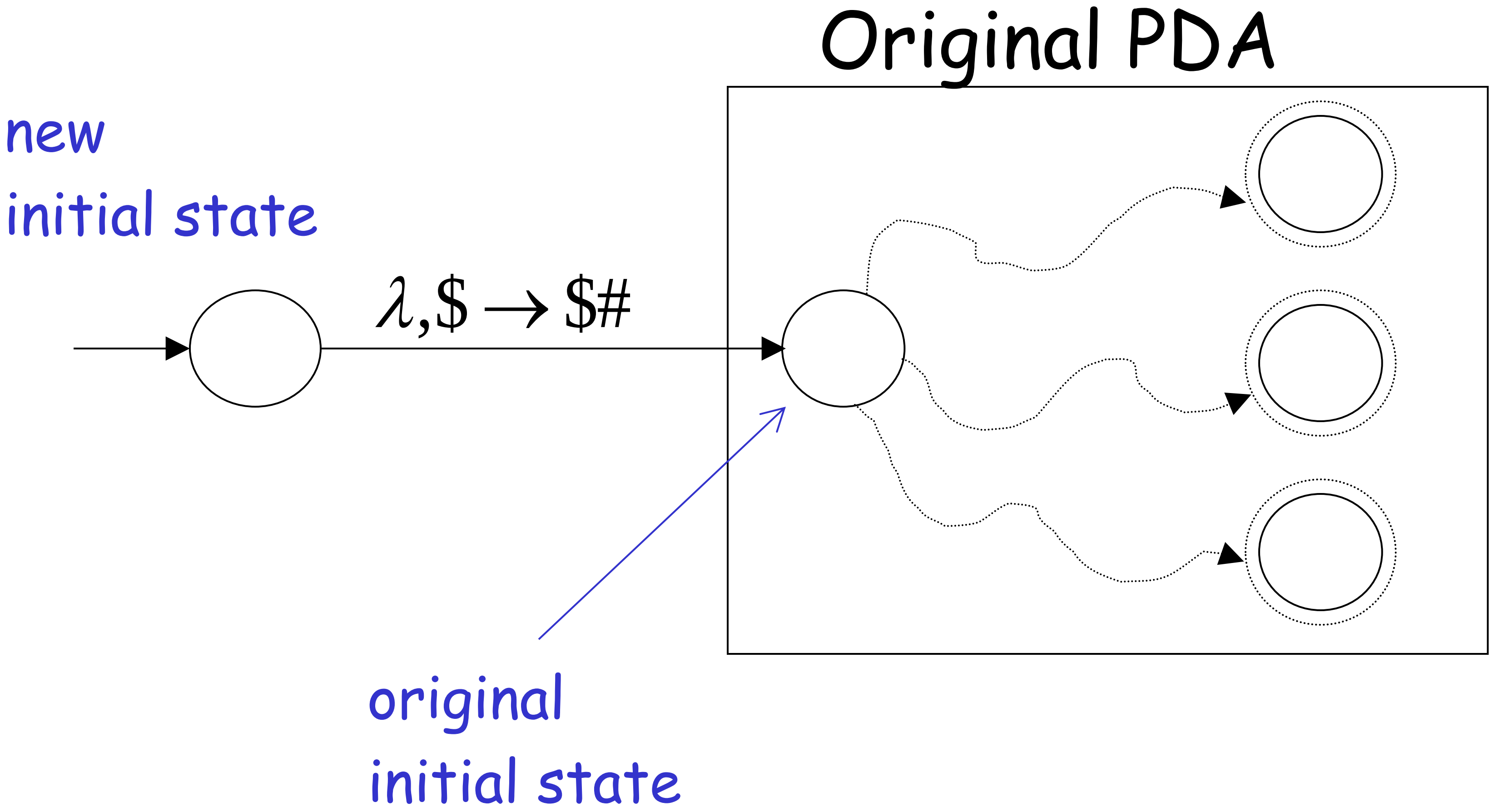



OK OK NOT OK

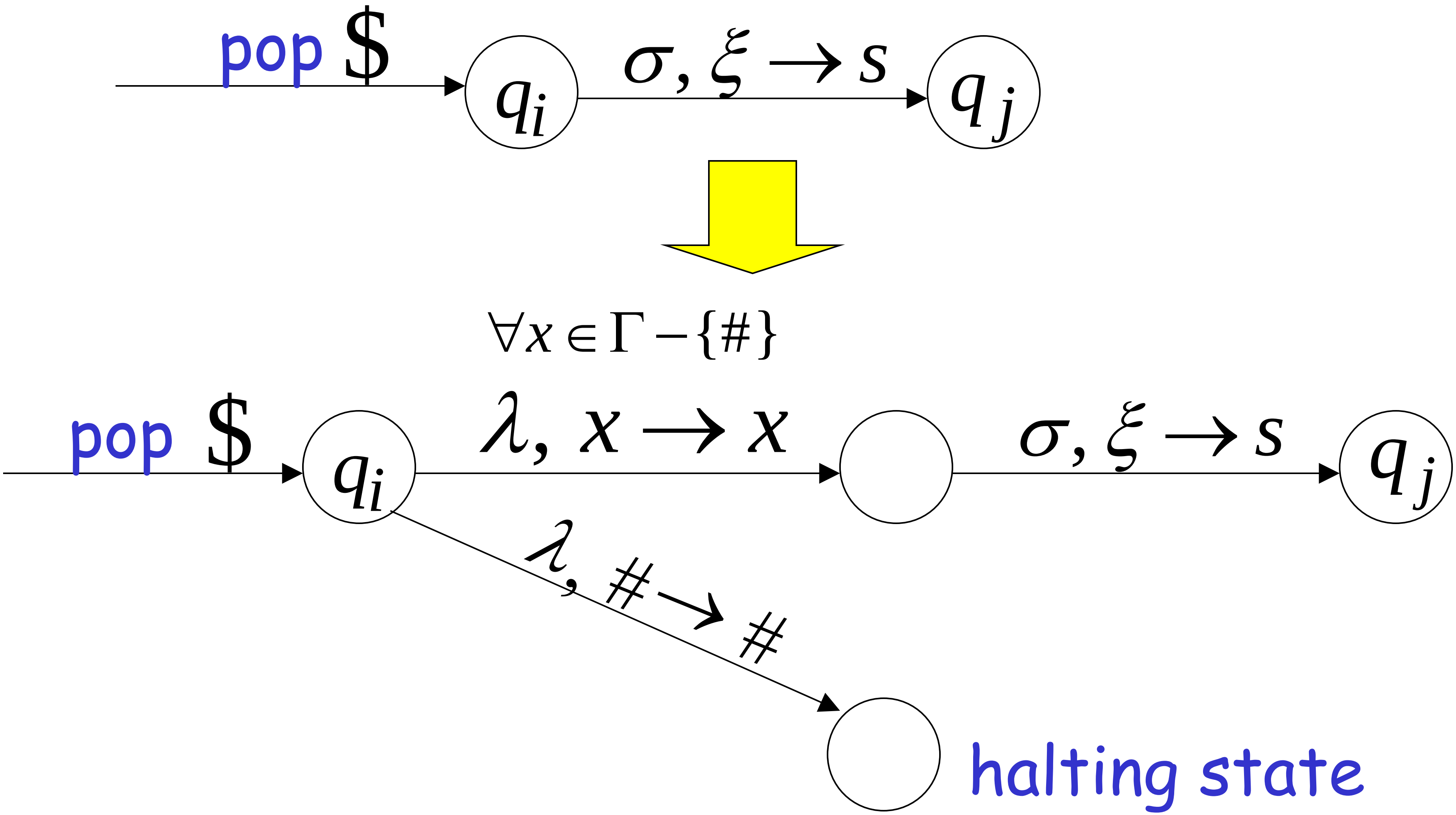
Introduce the new symbol $\#$ to mark the bottom of the stack



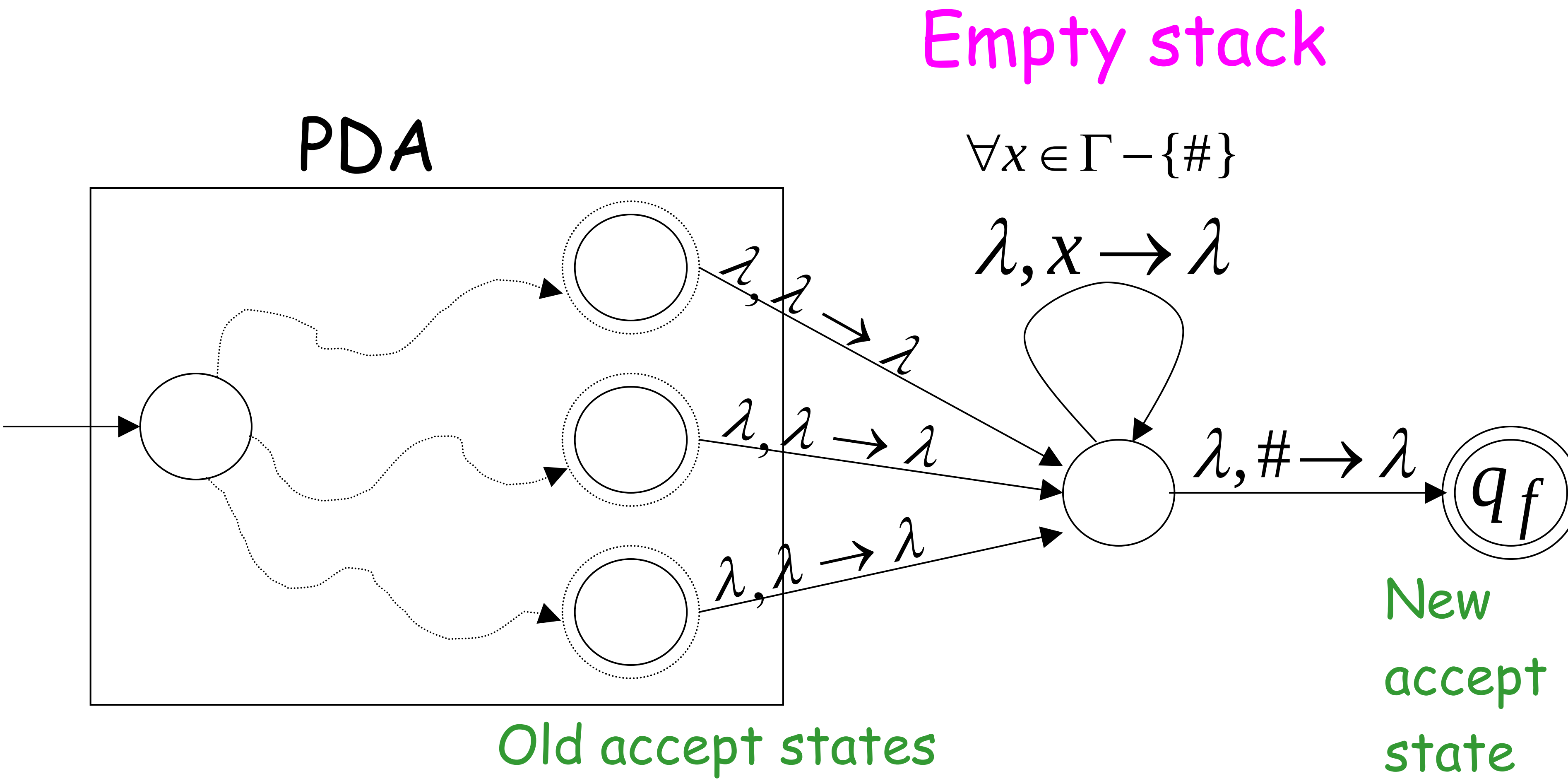
# At the beginning insert # into the stack



Convert all transitions so that after popping $\$$ the automaton halts



# 2. Modify the PDA so that at end it empties stack and has a unique accept state



# 3. Modify the PDA so that it has no transitions popping $\lambda$ :

$$q_i \xrightarrow{\sigma,\, \lambda \rightarrow y} q_j$$

$$\Downarrow$$

$$q_i \xrightarrow{\sigma,\, \tau \rightarrow y\tau} q_j$$

$$\forall\, \tau \in \Gamma - \{\#\}$$

# Example of a PDA in correct form:

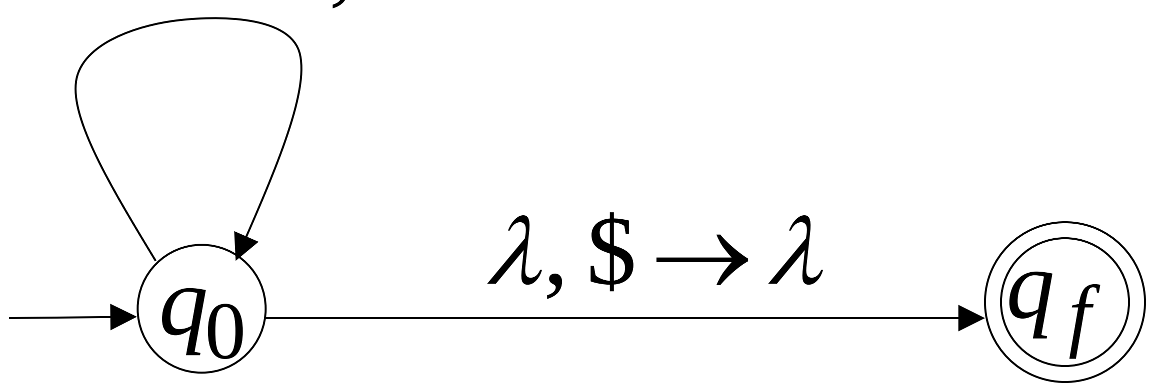
## (modifications are not necessary)

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

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

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

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



# Grammar Construction

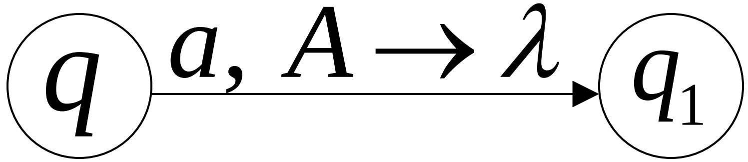
In grammar $G$:

Variables: $A$ PDA stack symbols

Terminals: $a$ PDA input symbols

Start Variable: $\$$ or $\#$ Stack bottom symbol

PDA transition




Grammar production

$$A \to a$$

PDA transition

$$q \xrightarrow{a,\ A \to B_1 B_2 \cdots B_m} q_1$$

Grammar production

$$A \to a B_1 B_2 \cdots B_m$$

Grammar leftmost derivation

PDA computation

| | |
|---|---|
| $S$ | $\succ (q_1, \sigma_1 \cdots \sigma_k \sigma_{k+1} \cdots \sigma_n, \$)$ |
| $\Rightarrow \cdots$ | $\succ \cdots$ |
| $\Rightarrow \sigma_1 \cdots \sigma_k X_1 \cdots X_m$ | $\succ (q_1, \sigma_{k+1} \cdots \sigma_n, X_1 \cdots X_m)$ |
| $\Rightarrow \cdots$ | $\succ \cdots$ |
| $\Rightarrow \sigma_1 \cdots \sigma_k \sigma_{k+1} \cdots \sigma_n$ | $\succ (q_2, \lambda, \lambda)$ |

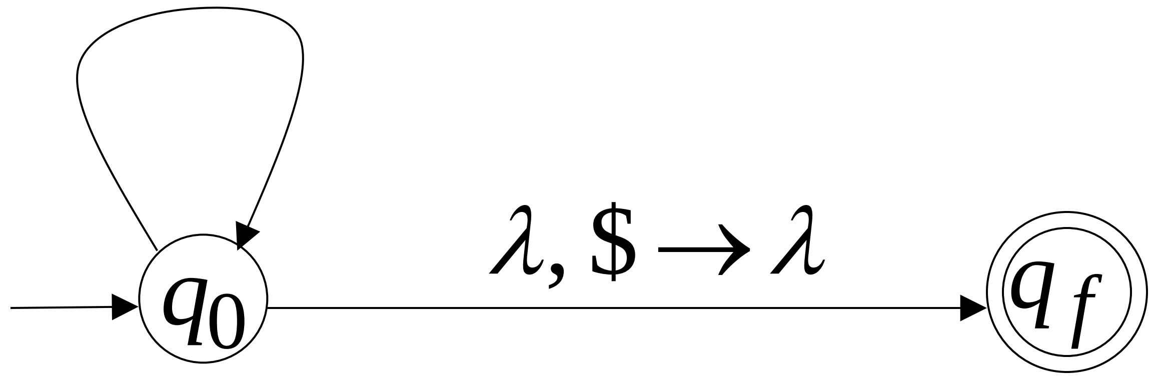
Leftmost variable

Example PDA:

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

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

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




Grammar:

$\$ \rightarrow a0\$ \quad \$ \rightarrow b1\$$

$0 \rightarrow a00 \quad 1 \rightarrow b11$

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

Grammar Leftmost derivation:

$$
\begin{aligned}
&\$ \\
&\Rightarrow a0\$ \\
&\Rightarrow ab\$ \\
&\Rightarrow abb1\$ \\
&\Rightarrow abba\$ \\
&\Rightarrow abba
\end{aligned}
$$

PDA Computation:

$$
\begin{aligned}
&(q_0, abba, \$) \\
&\succ (q_0, bba, 0\$) \\
&\succ (q_0, ba, \$) \\
&\succ (q_0, a, 1\$) \\
&\succ (q_0, \lambda, \$) \\
&\succ (q_f, \lambda, \lambda)
\end{aligned}
$$

Derivation: $\$$

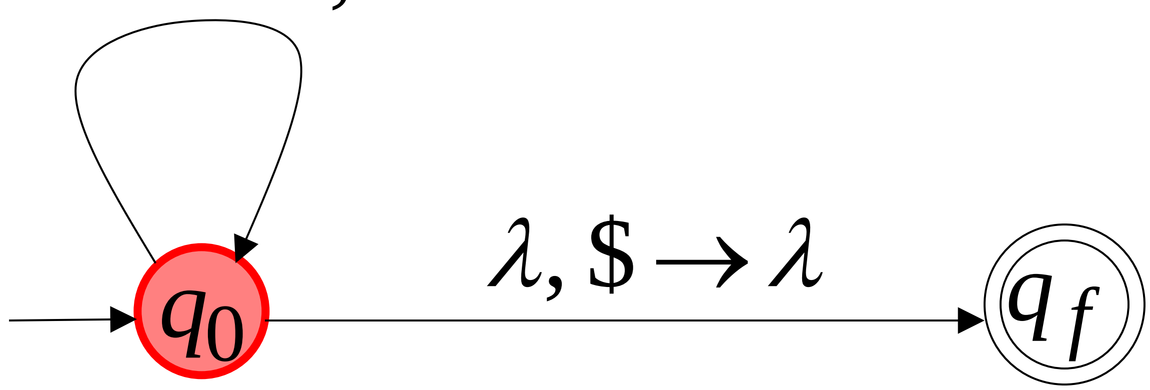
| | | | |
|---|---|---|---|
| $a$ | $b$ | $b$ | $a$ |

Time 0

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

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

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



**Derivation:** $\$ \Longrightarrow a0\$$

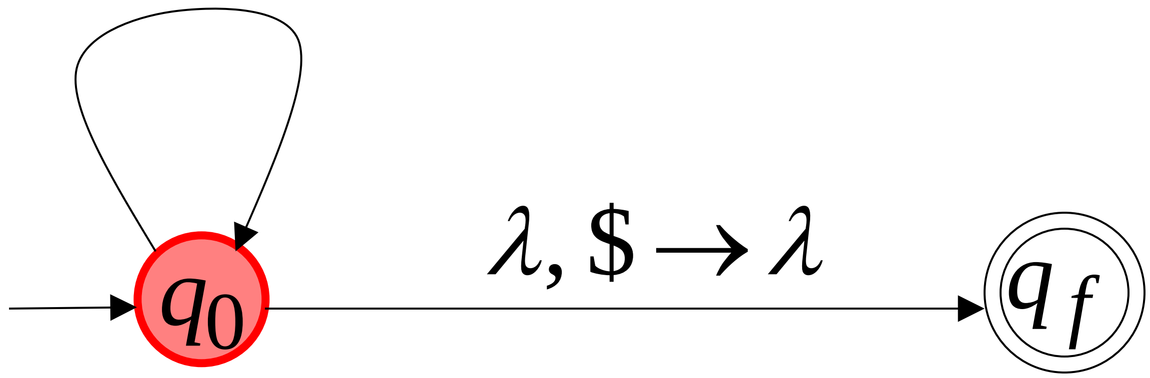
| | | | |
|---|---|---|---|
| $a$ | $b$ | $b$ | $a$ |

**Time 1**

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

| |
|---|
| 0 |
| $ |

**Stack**



Derivation: $\$ \Rightarrow a0\$ \Rightarrow ab\$$

| | | | |
|---|---|---|---|
| $a$ | $b$ | $b$ | $a$ |

Time 2

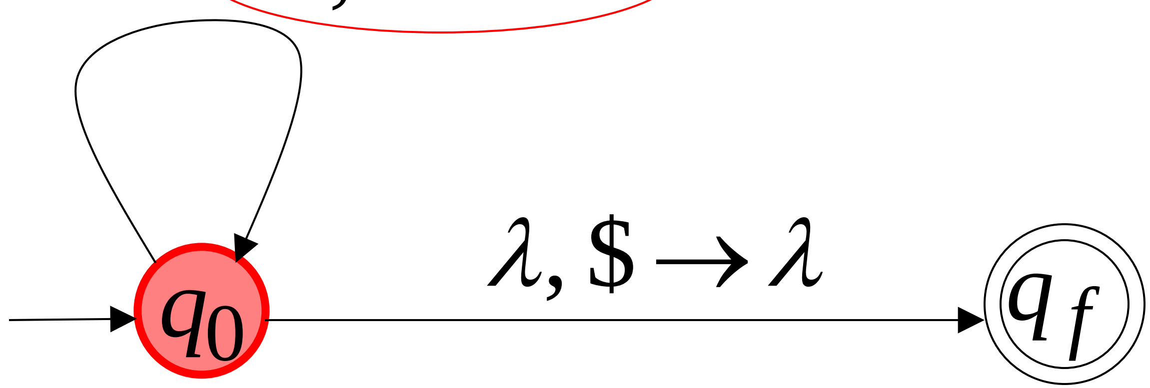
$a, \$ \rightarrow 0\$$ $\quad b, \$ \rightarrow 1\$$

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

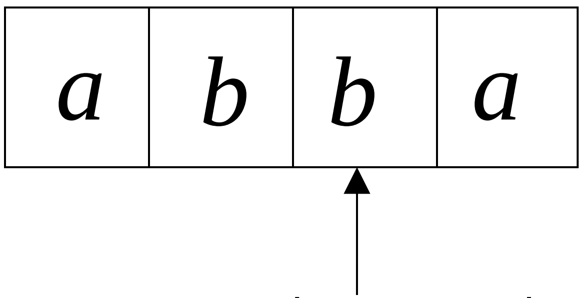
$a, 1 \rightarrow \lambda$ $\quad b, 0 \rightarrow \lambda$

| |
|---|
| $\$$ |

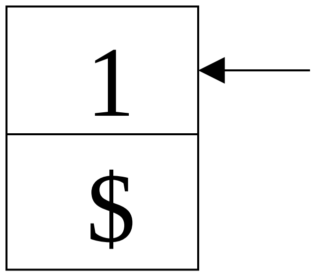
Stack



**Derivation:** $\$ \Rightarrow a0\$ \Rightarrow ab\$ \Rightarrow abb1\$$




**Time 3**

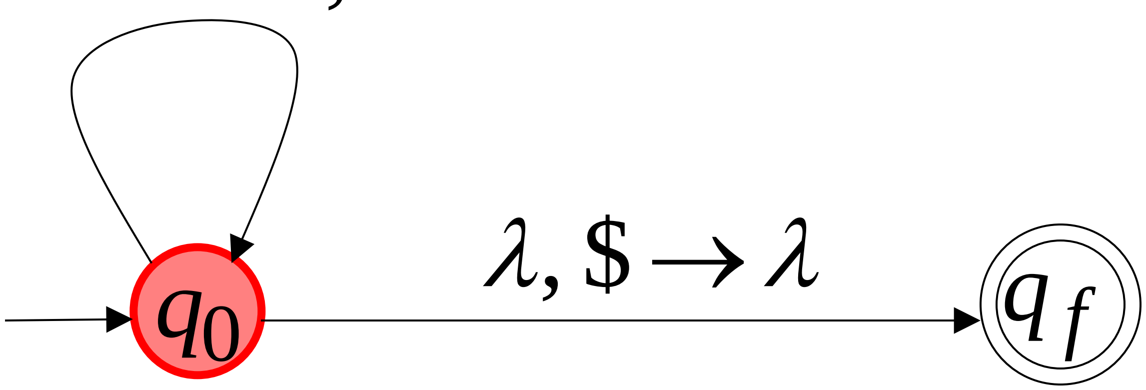



**Stack**

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

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

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



Derivation: $\$ \Rightarrow a0\$ \Rightarrow ab\$ \Rightarrow abb1\$$

$abba\$$

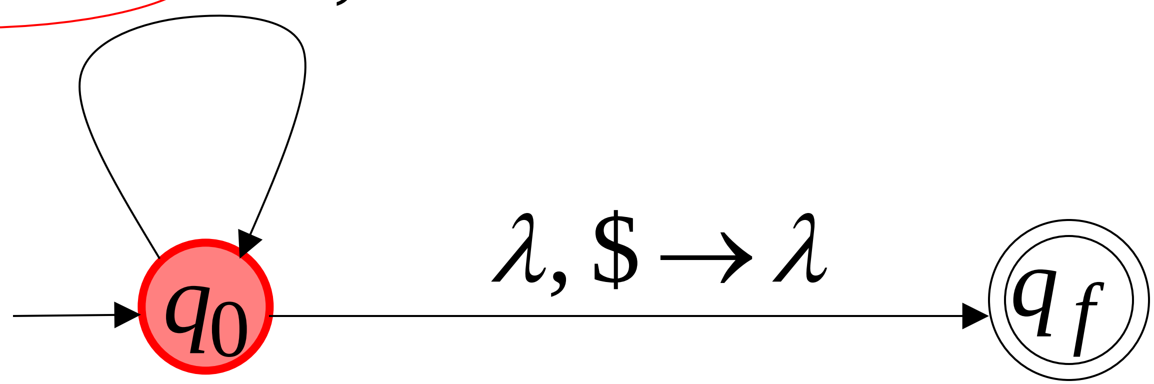
| | | | |
|---|---|---|---|
| $a$ | $b$ | $b$ | $a$ |

Time 4

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

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

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




| |
|---|
| $\$$ |

Stack

Derivation: $\$ \Rightarrow a0\$ \Rightarrow ab\$ \Rightarrow abb1\$$

$abba\$ \Rightarrow abba$

| | | | |
|---|---|---|---|
| $a$ | $b$ | $b$ | $a$ |

Time 5

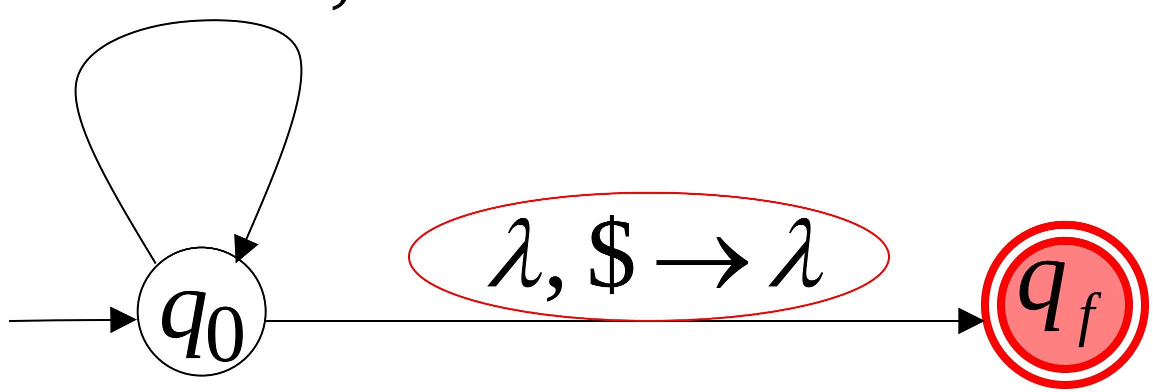
$a, \$ \rightarrow 0\$$ $\quad b, \$ \rightarrow 1\$$

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

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

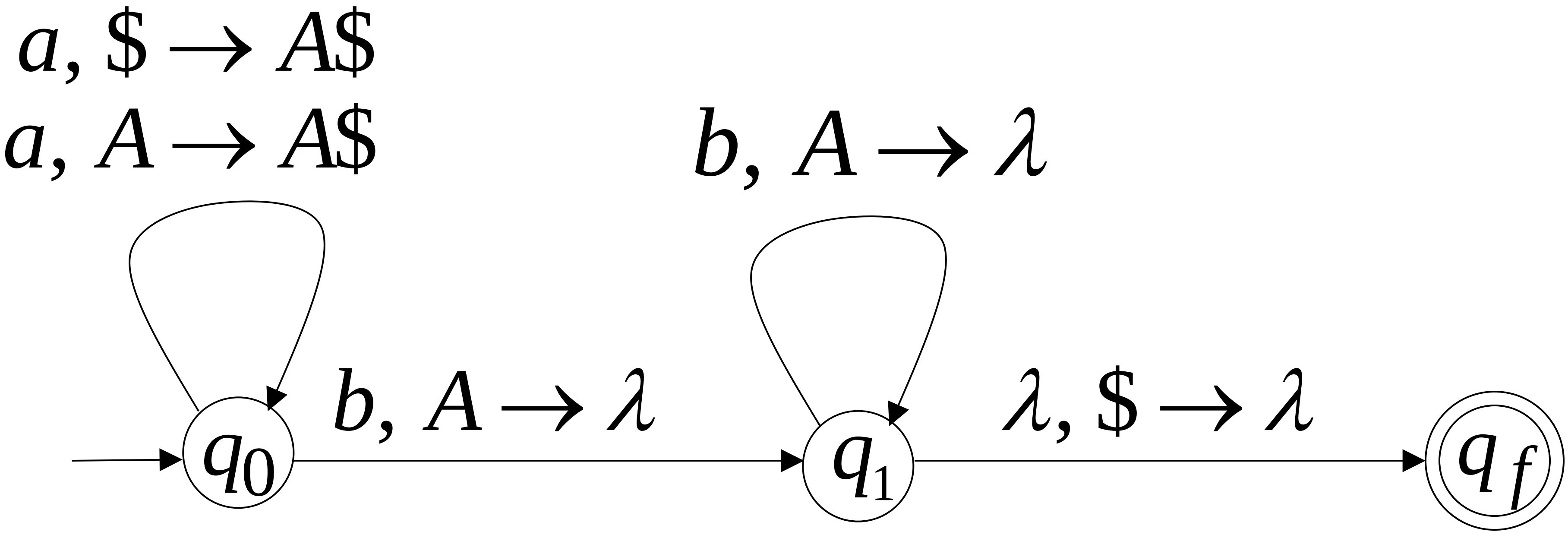
empty

Stack

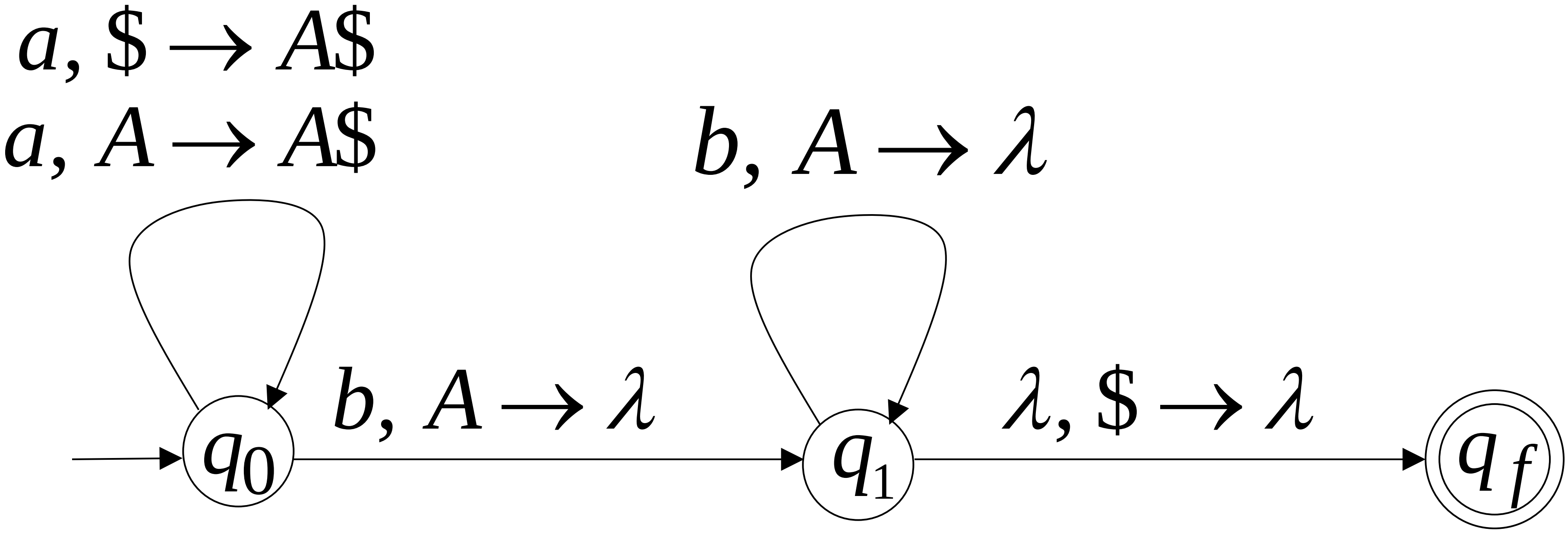


# Exercise□

However, this grammar conversion does not work for all PDAs:




$$L(M) = \{a^n b^n : n \geq 1\}$$




Grammar:

$\$ \to aA\$$

$A \to aA\$$

$\$ \to \lambda$

$A \to b$

Bad Derivation:

$$S \Rightarrow aA\$ \Rightarrow aaA\$ \Rightarrow aab\$ \Rightarrow aab \notin L(M)$$

Grammar:

$$\$ \rightarrow aA\$ \qquad A \rightarrow aA\$$$

$$\$ \rightarrow \lambda \qquad A \rightarrow b$$

# What went wrong?

# The Correct Grammar Construction

In grammar $G$ :

PDA stack symbol

Variables: $(q_i A q_j)$

PDA states

Terminals: Input symbols of PDA

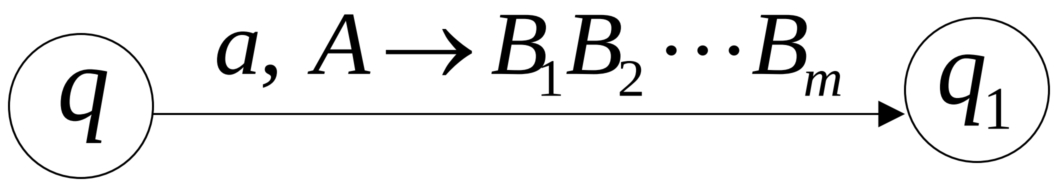
PDA transition

$$q \xrightarrow{a,\ A \to \lambda} q_1$$

Grammar production

$$(qAq_1) \to a$$

# PDA transition




# Grammar production

$$(qAq_{m+1}) \rightarrow a(q_1 B_1 q_2)(q_2 B_2 q_3) \cdots (q_m B_m q_{m+1})$$

For all possible states $q_2, \ldots, q_{m+1}$ in PDA

Stack bottom symbol

$\$ \text{ or } \#$

Start Variable: $(q_o Z q_f)$

Start state

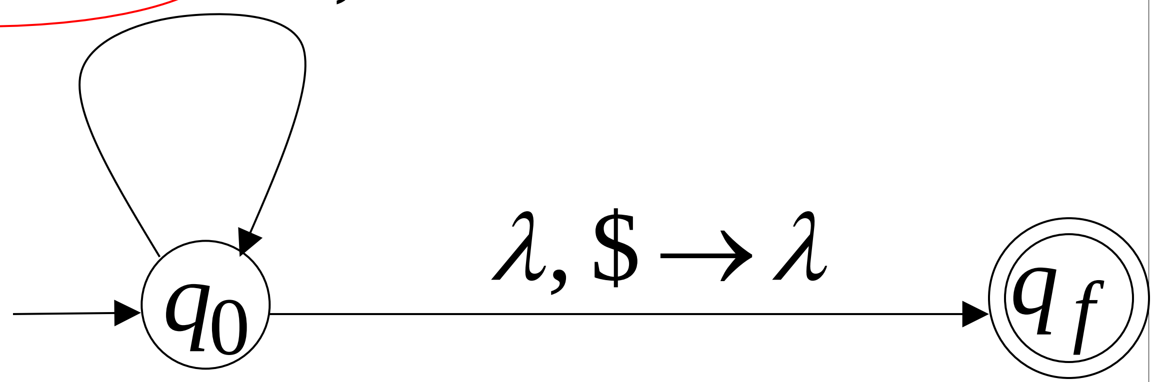
accept state

# Example:

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

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

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



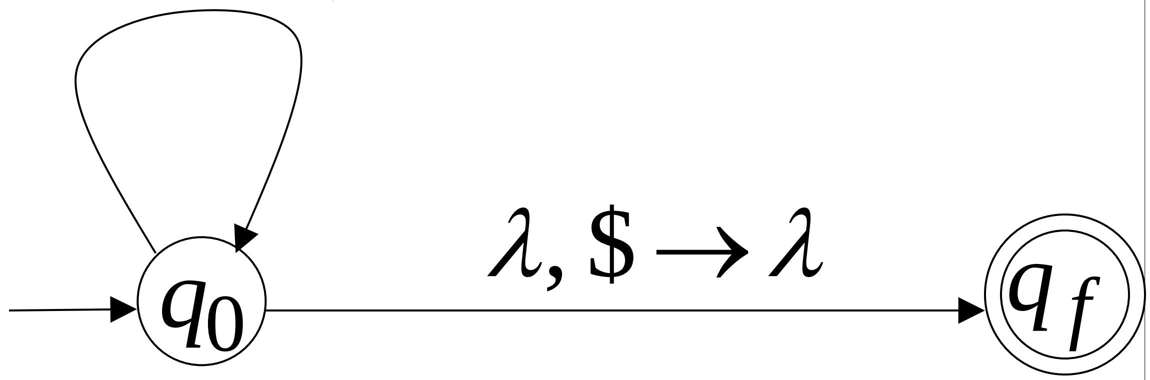

Grammar production: $(q_0 1 q_0) \rightarrow a$

# Example:

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

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

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



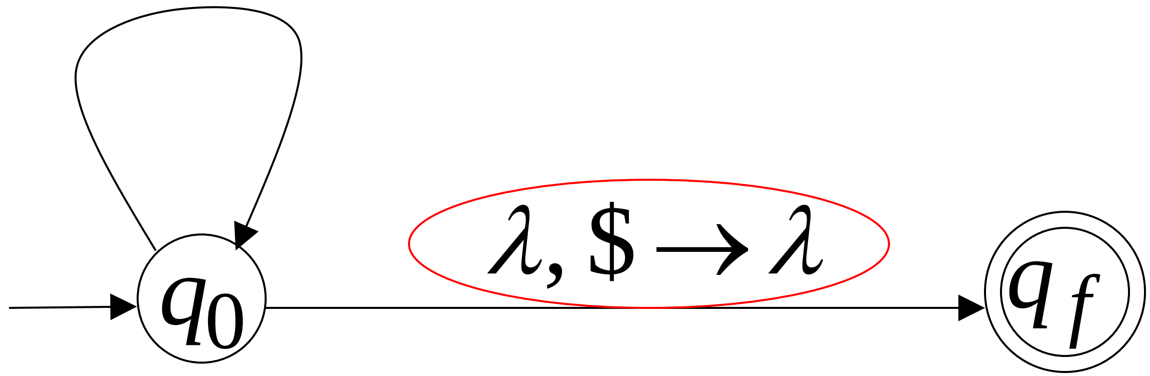

## Grammar productions:

$$(q_0\$q_0) \rightarrow b(q_0 1 q_0)(q_0\$q_0) \mid b(q_0 1 q_f)(q_f\$q_0)$$

$$(q_0\$q_f) \rightarrow b(q_0 1 q_0)(q_0\$q_f) \mid b(q_0 1 q_f)(q_f\$q_f)$$

# Example:

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




Grammar production: $(q_0 \$ q_f) \rightarrow \lambda$

## Resulting Grammar: $(q_0\$q_f)$ : start variable

$$(q_0\$q_0) \to b(q_01q_0)(q_0\$q_0) \mid b(q_01q_f)(q_f\$q_0)$$

$$(q_0\$q_f) \to b(q_01q_0)(q_0\$q_f) \mid b(q_01q_f)(q_f\$q_f)$$

$$(q_01q_0) \to b(q_01q_0)(q_01q_0) \mid b(q_01q_f)(q_f1q_0)$$

$$(q_01q_f) \to b(q_01q_0)(q_01q_f) \mid b(q_01q_f)(q_f1q_f)$$

$$(q_0\$q_0) \to a(q_00q_0)(q_0\$q_0) \mid a(q_00q_f)(q_f\$q_0)$$

$$(q_0\$q_f) \to a(q_00q_0)(q_0\$q_f) \mid a(q_00q_f)(q_f\$q_f)$$

$$(q_0 0 q_0) \to a(q_0 0 q_0)(q_0 0 q_0) \mid a(q_0 0 q_f)(q_f 0 q_0)$$

$$(q_0 0 q_f) \to a(q_0 0 q_0)(q_0 0 q_f) \mid a(q_0 0 q_f)(q_f 0 q_f)$$

$$(q_0 1 q_0) \to a$$

$$(q_0 0 q_0) \to b$$

$$(q_0 \$ q_f) \to \lambda$$

Grammar
Leftmost
derivation

PDA
computation

$$(q_0 \$ q_f)$$
$$\Rightarrow a(q_0 0 q_0)(q_0 \$ q_f)$$
$$\Rightarrow ab(q_0 \$ q_f)$$
$$\Rightarrow abb(q_0 1 q_0)(q_0 \$ q_f)$$
$$\Rightarrow abba(q_0 \$ q_f)$$
$$\Rightarrow abba$$

$$(q_0, abba, \$)$$
$$\succ (q_0, bba, 0\$)$$
$$\succ (q_0, ba, \$)$$
$$\succ (q_0, a, 1\$)$$
$$\succ (q_0, \lambda, \$)$$
$$\succ (q_f, \lambda, \lambda)$$

# Derivation: $(q_0 \$ q_f)$

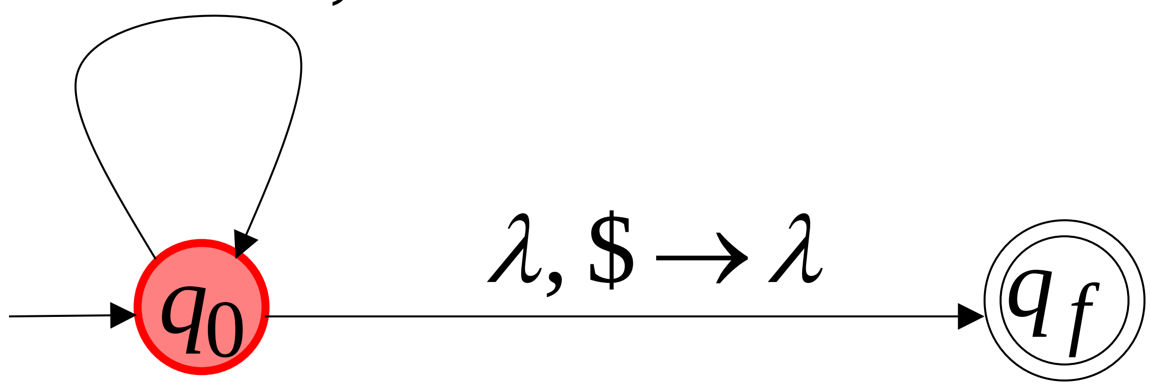
| | | | |
|---|---|---|---|
| $a$ | $b$ | $b$ | $a$ |

Time 0

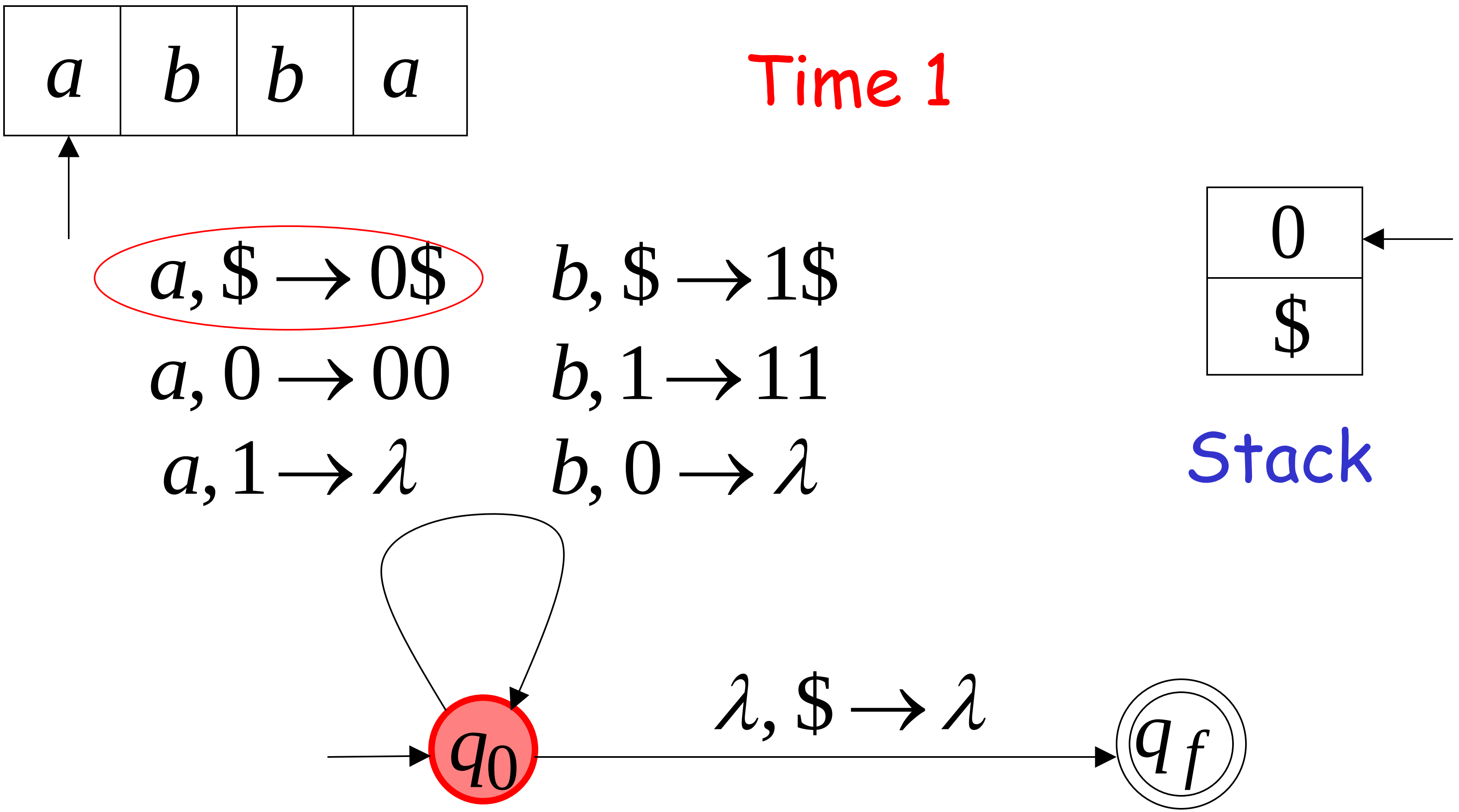
$a, \$ \rightarrow 0\$$ $\quad$ $b, \$ \rightarrow 1\$$

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

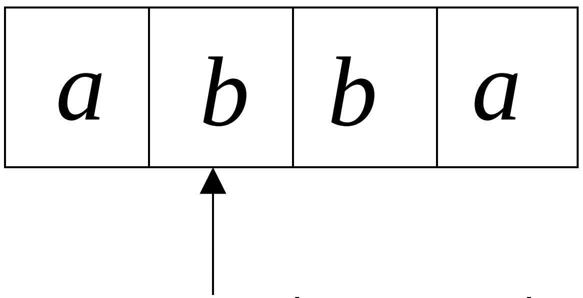
$a, 1 \rightarrow \lambda$ $\quad$ $b, 0 \rightarrow \lambda$



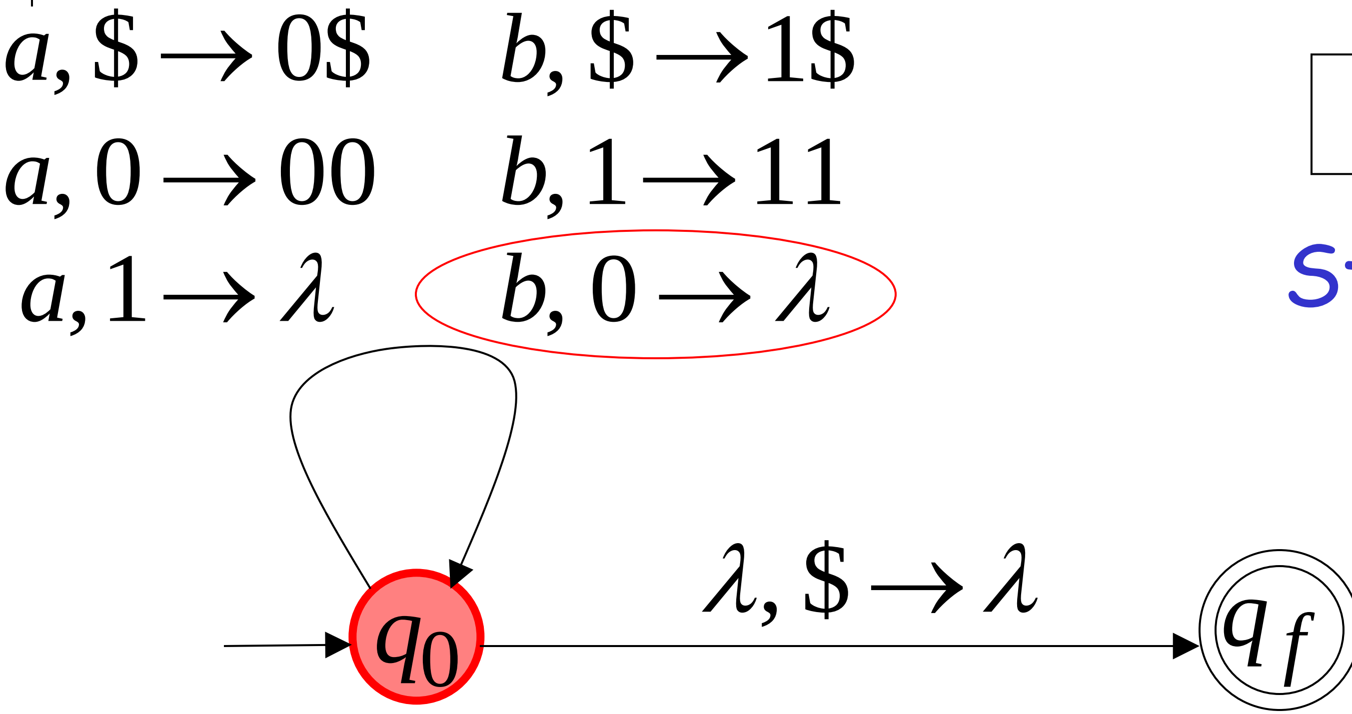
**Derivation:** $(q_0 \$ q_f) \Rightarrow a(q_0 0 q_0)(q_0 \$ q_f)$

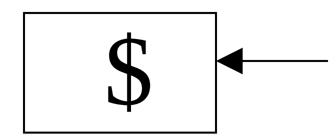


Derivation: $(q_0 \$ q_f) \Rightarrow a(q_0 0 q_0)(q_0 \$ q_f) \Rightarrow ab(q_0 \$ q_f)$




Time 2

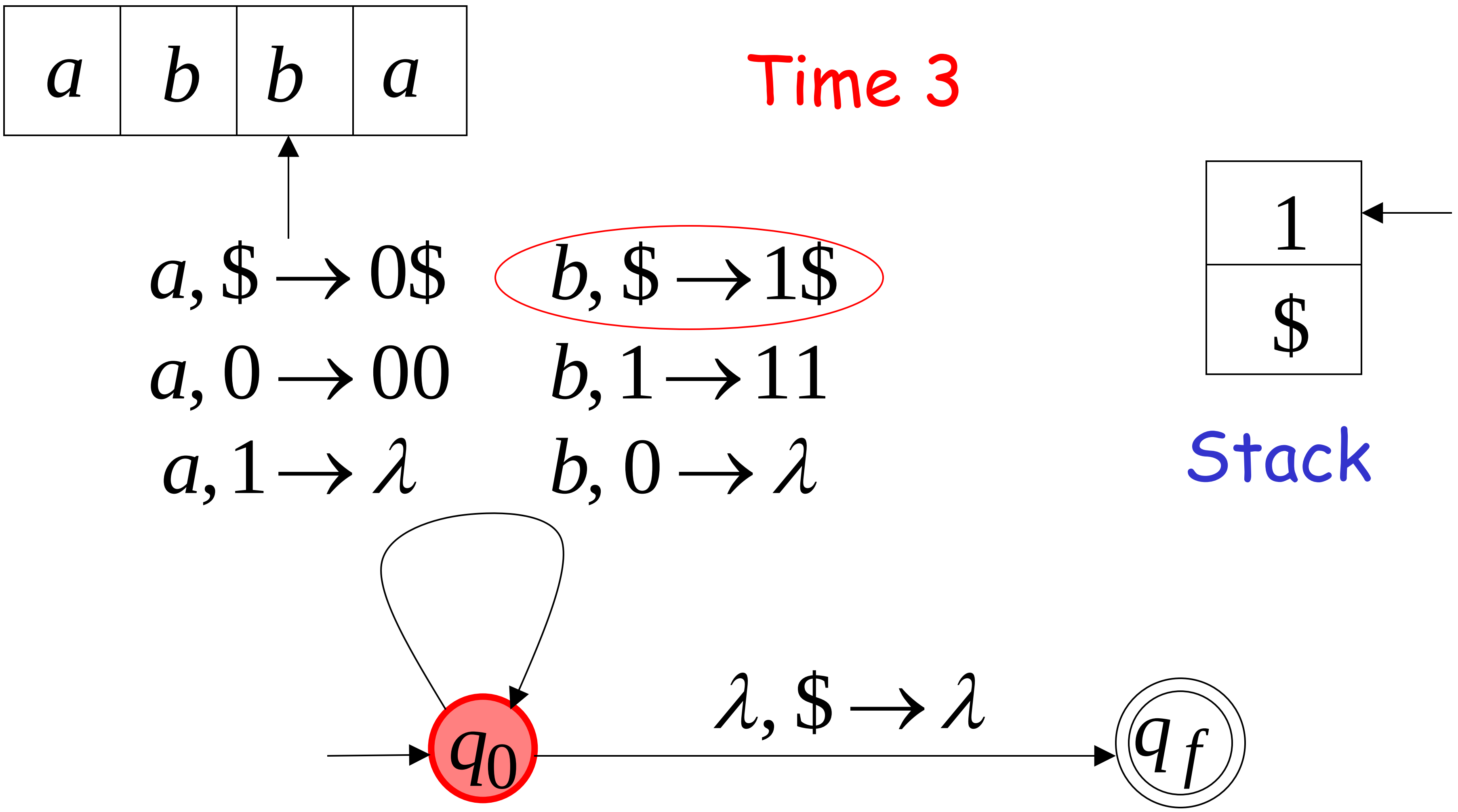



Stack

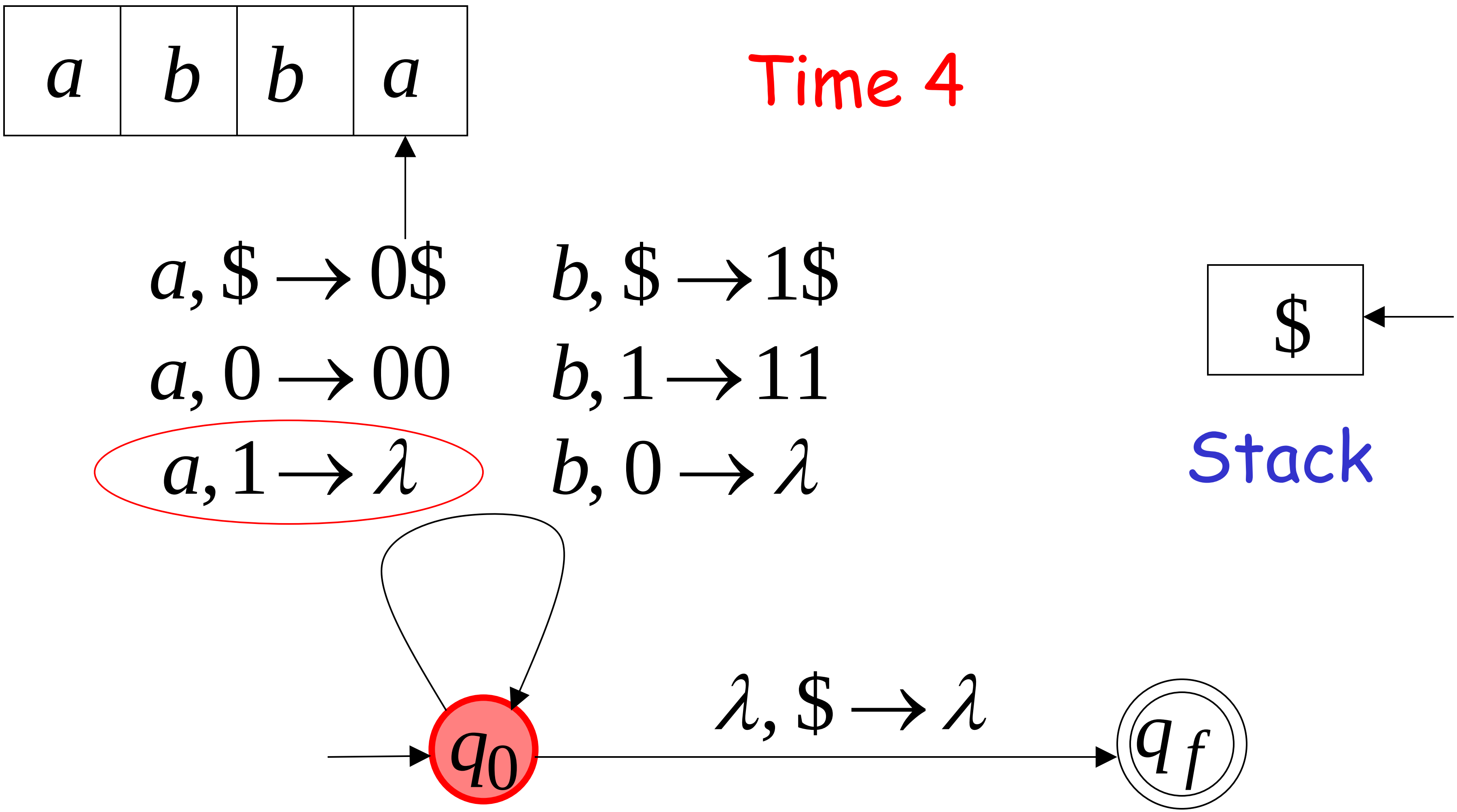
**Derivation:** $(q_0\$q_f) \Longrightarrow a(q_0 0 q_0)(q_0\$q_f) \Longrightarrow ab(q_0\$q_f)$

$\Longrightarrow abb(q_0 1 q_0)(q_0\$q_f)$



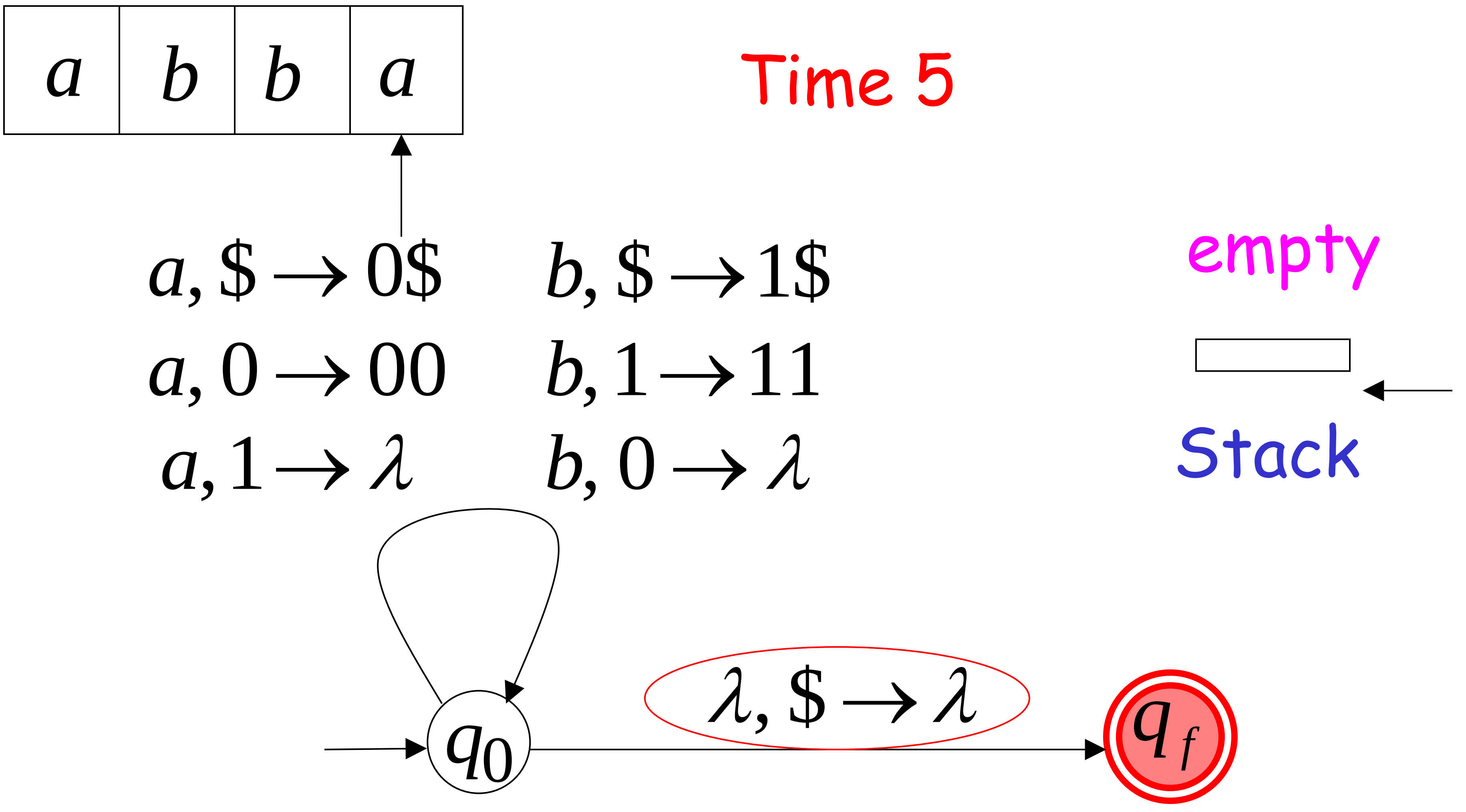
**Derivation:** $(q_0\$q_f) \Longrightarrow a(q_0 0 q_0)(q_0\$q_f) \Longrightarrow ab(q_0\$q_f)$

$$\Longrightarrow abb(q_0 1 q_0)(q_0\$q_f) \Longrightarrow abba(q_0\$q_f)$$



**Derivation:** $(q_0\$q_f) \Rightarrow a(q_0 0 q_0)(q_0\$q_f) \Rightarrow ab(q_0\$q_f)$

$\Rightarrow abb(q_0 1 q_0)(q_0\$q_f) \Rightarrow abba(q_0\$q_f) \Rightarrow abba$



# In general:

Grammar

$$(q_i A q_j) \overset{*}{\Rightarrow} wB$$

If and Only if

PDA

$$(q_i, w, A) \overset{*}{\succ} (q_j, \lambda, B)$$

## Thus:

Grammar generates $w$

$$(q_0 \$ q_f) \overset{*}{\Rightarrow} w$$

If and Only if

PDA accepts $w$

$$(q_0, w, \$) \overset{*}{\succ} (q_f, \lambda, \lambda)$$

# Therefore:

For any PDA
there is a context-free grammar
that accepts the same language

$$\left\{\begin{array}{c}\text{Context-Free}\\ \text{Languages}\\ \text{(Grammars)}\end{array}\right\} \supseteq \left\{\begin{array}{c}\text{Languages}\\ \text{Accepted by}\\ \text{PDAs}\end{array}\right\}$$