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

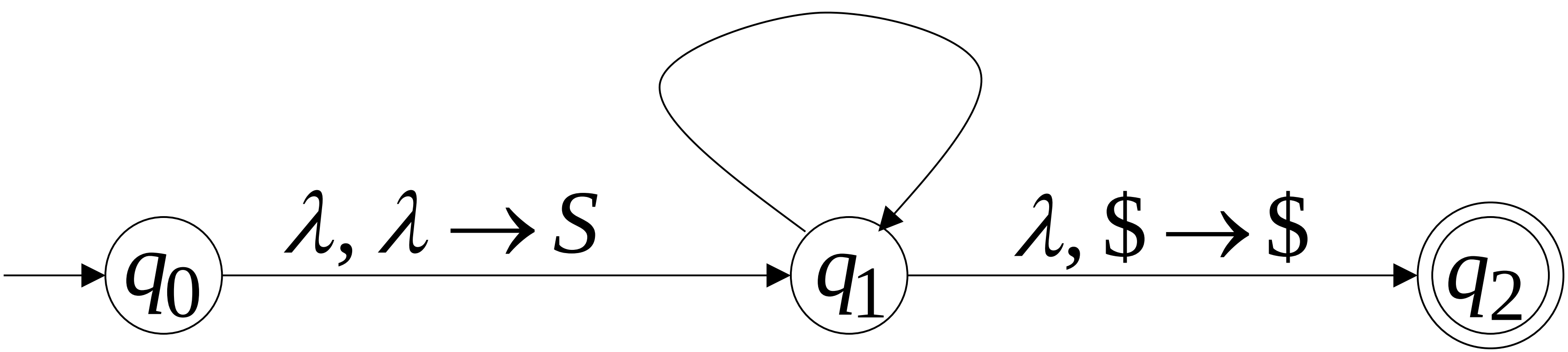
Production in G

$$A \to w$$

Terminal in G

$$a$$

$$\lambda, A \to w \qquad a, a \to \lambda$$



Grammar leftmost derivation

PDA computation

Simulates grammar leftmost derivations

$$
\begin{array}{ll}
 & (q_0, \sigma_1 \cdots \sigma_k \sigma_{k+1} \cdots \sigma_n, \$) \\
S & \succ (q_1, \sigma_1 \cdots \sigma_k \sigma_{k+1} \cdots \sigma_n, S\$) \\
\Rightarrow \cdots & \succ \cdots \\
\Rightarrow \sigma_1 \cdots \sigma_k X_1 \cdots X_m & \succ (q_1, \sigma_{k+1} \cdots \sigma_n, X_1 \cdots X_m \$) \\
\Rightarrow \cdots & \succ \cdots \\
\Rightarrow \sigma_1 \cdots \sigma_k \sigma_{k+1} \cdots \sigma_n & \succ (q_2, \lambda, \$)
\end{array}
$$

Leftmost variable

Example

Grammar

$S \to aSTb$

$S \to b$

$T \to Ta$

$T \to \lambda$

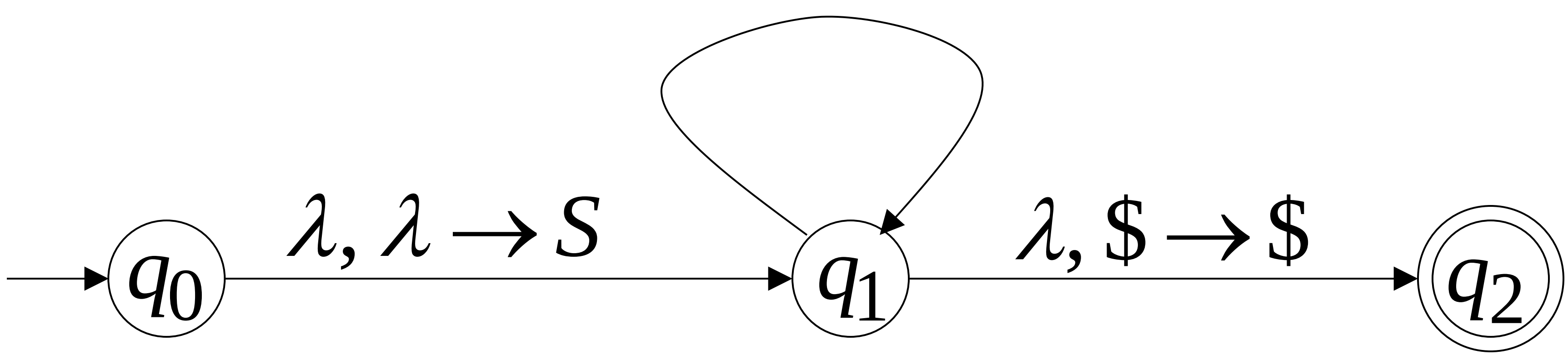
PDA

$\lambda, S \to aSTb$

$\lambda, S \to b$

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

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



Grammar derivation

$$S$$

$$\Rightarrow aSTb$$

$$\Rightarrow abTb$$

$$\Rightarrow abTab$$

$$\Rightarrow abab$$

PDA computation

$$\left.\begin{array}{l}(q_0, abab, \$) \\ \succ (q_1, abab, S\$)\end{array}\right.$$

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

$$\left.\begin{array}{l}\succ (q_1, bab, bTb\$) \\ \succ (q_1, ab, Tb\$)\end{array}\right.$$

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

$$\left.\begin{array}{l}\succ (q_1, ab, ab\$) \\ \succ (q_1, b, b\$) \\ \succ (q_1, \lambda, \$) \\ \succ (q_2, \lambda, \$)\end{array}\right.$$

Derivation:

Input

a	b	a	b

Time 0

$$\lambda, S \to aSTb$$

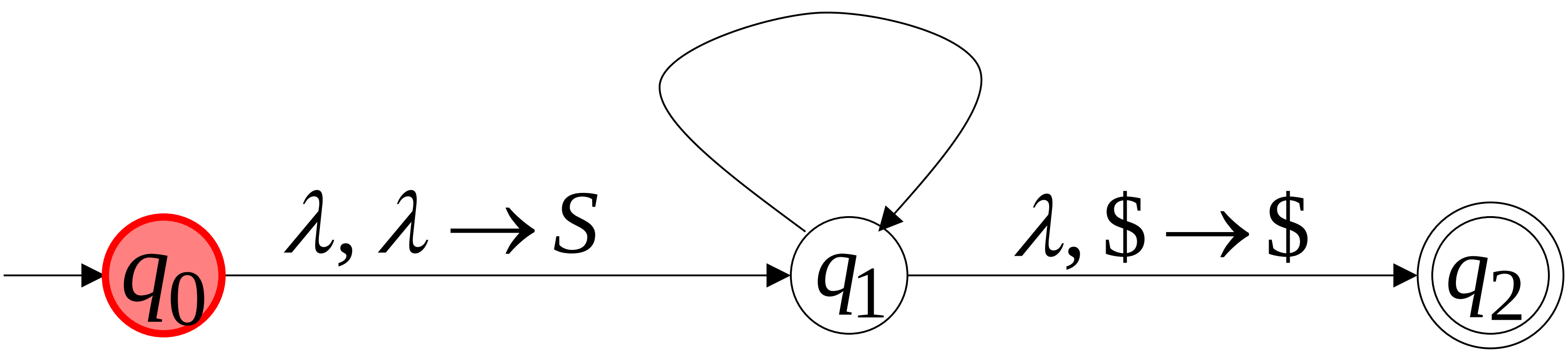
$$\lambda, S \to b$$

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

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

$\$$

Stack



Derivation: S

Input

a	b	a	b

Time 0

$\lambda, S \rightarrow aSTb$

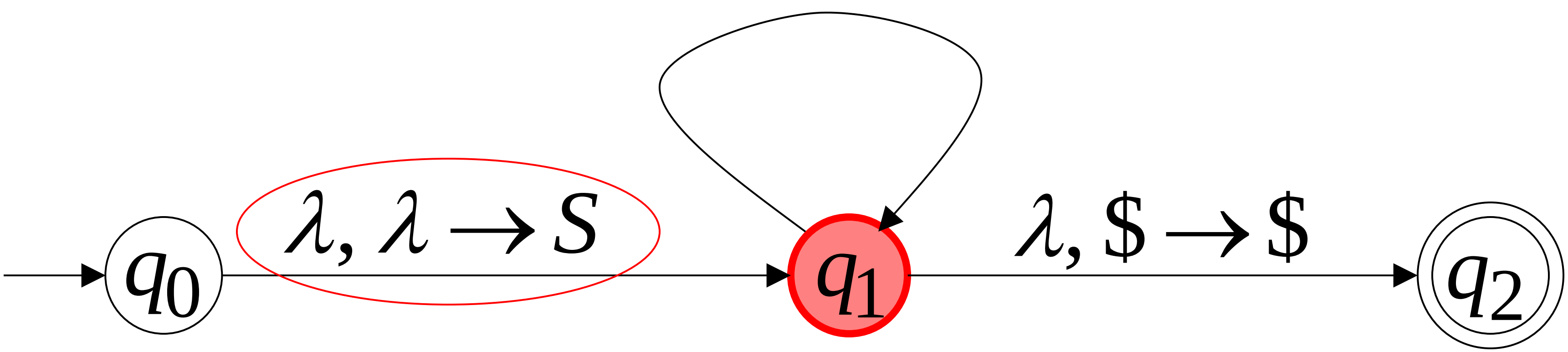
$\lambda, S \rightarrow b$

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

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

S
$\$$

Stack



Derivation: $S \Rightarrow aSTb$

Input

a	b	a	b

Time 1

$\lambda, S \rightarrow aSTb$

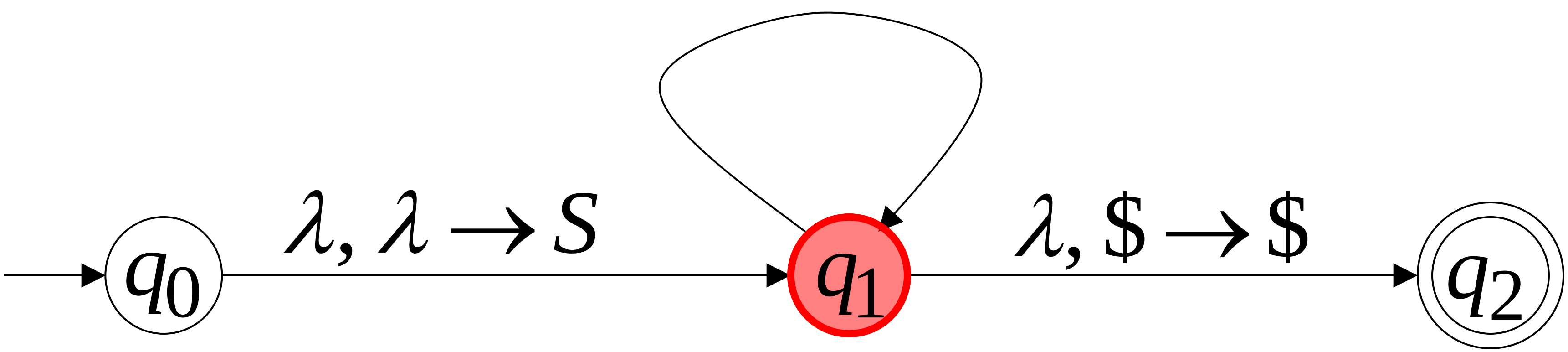
$\lambda, S \rightarrow b$

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

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

a
S
T
b
$\$$

Stack



Derivation: $S \Rightarrow aSTb$

Input

a	b	a	b

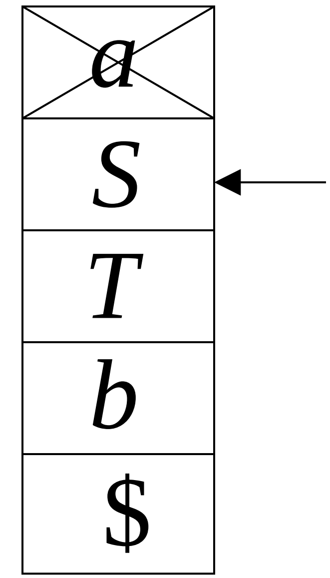
Time 2

$\lambda, S \rightarrow aSTb$

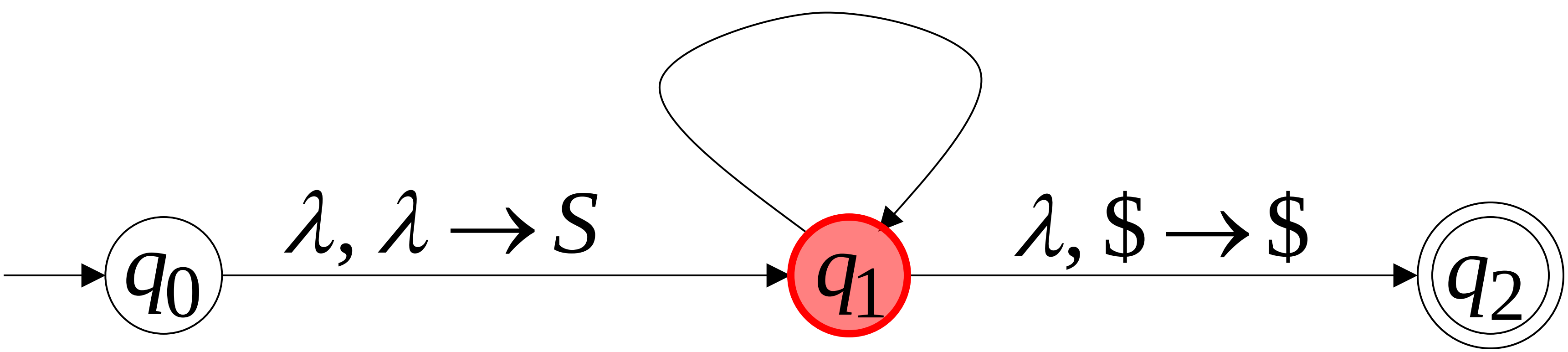
$\lambda, S \rightarrow b$

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

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



Stack



Derivation: $S \Rightarrow aSTb \Rightarrow abTb$

Input

a	b	a	b

Time 3

$\lambda, S \rightarrow aSTb$

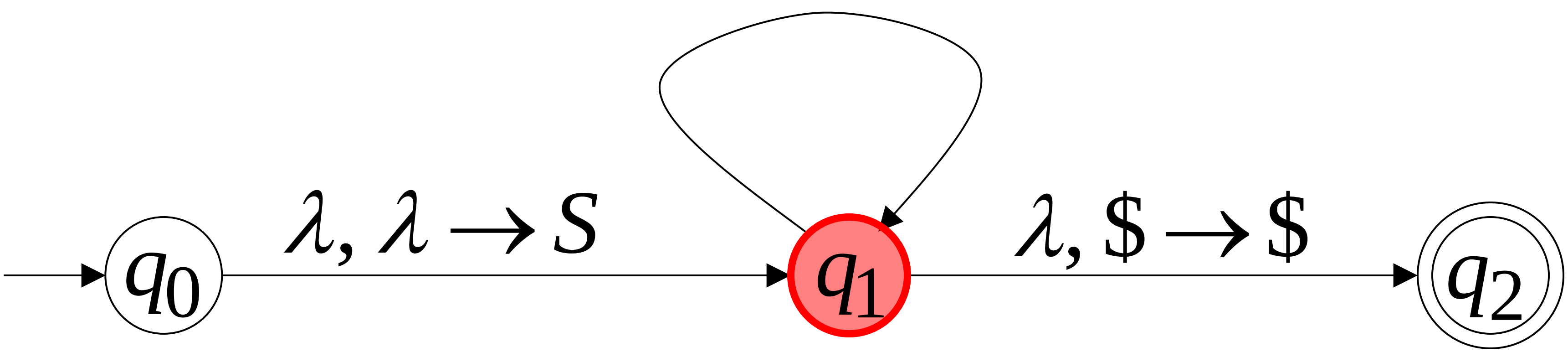
$\lambda, S \rightarrow b$

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

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

b
T
b
$\$$

Stack



Derivation: $S \Rightarrow aSTb \Rightarrow abTb$

Input

a	b	a	b

Stack

$\cancel{b}$
T
b
$\$$

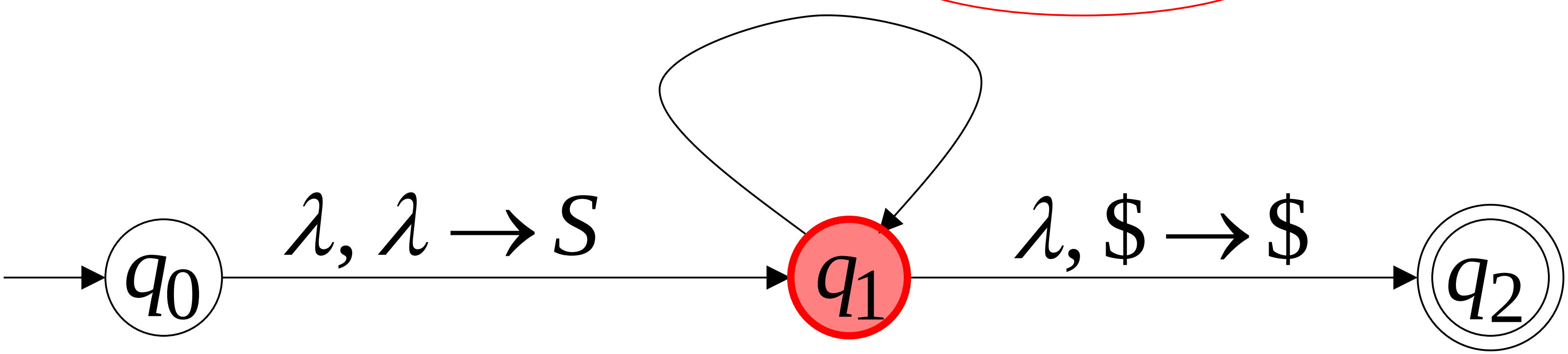
Time 4

$\lambda, S \rightarrow aSTb$

$\lambda, S \rightarrow b$

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

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



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

Input

a	b	a	b

Time 5

$\lambda, S \rightarrow aSTb$

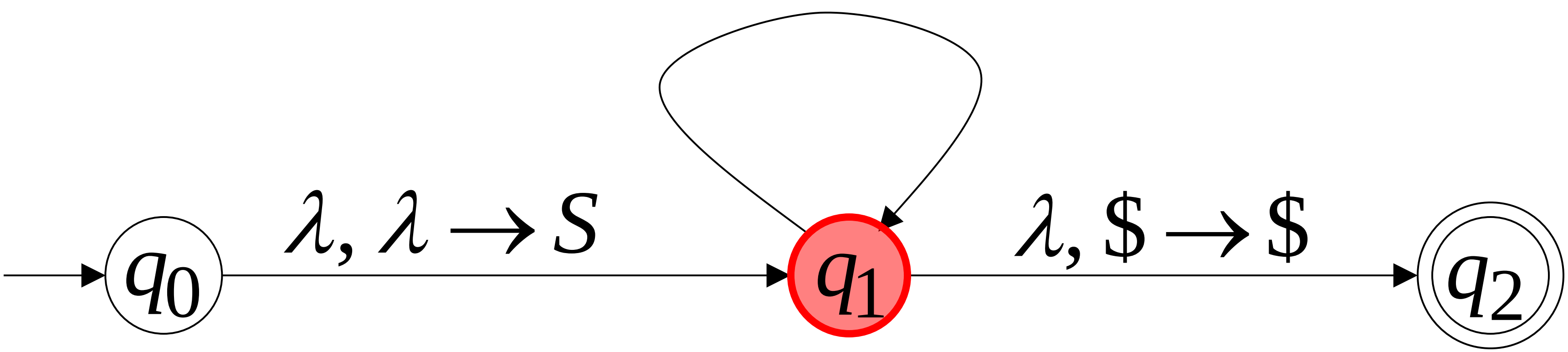
$\lambda, S \rightarrow b$

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

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

T
a
b
$\$$

Stack



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

Input

a	b	a	b

Stack

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

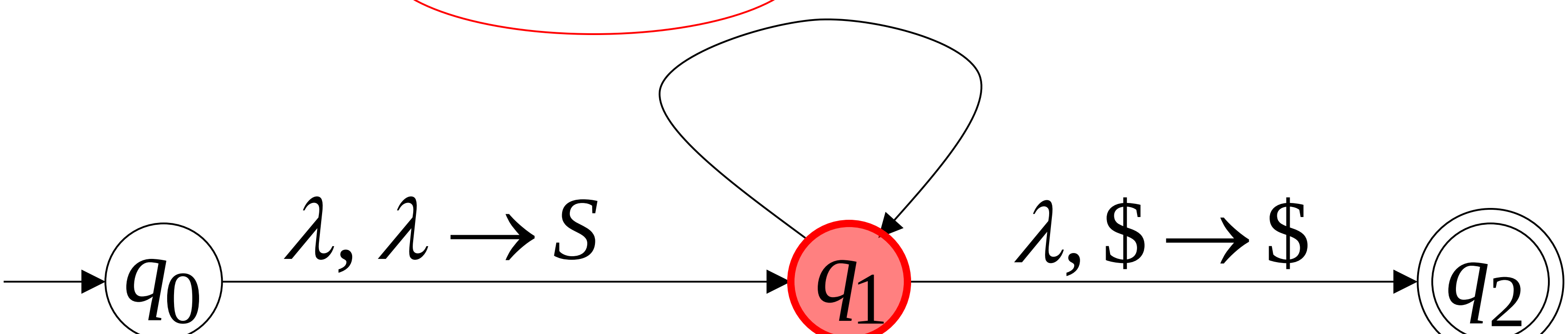
Time 6

$\lambda, S \rightarrow aSTb$

$\lambda, S \rightarrow b$

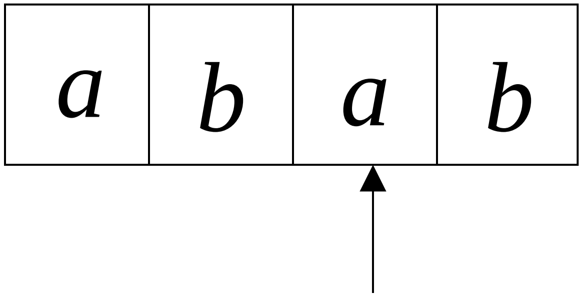
$\lambda, T \rightarrow Ta$ $\qquad a, a \rightarrow \lambda$

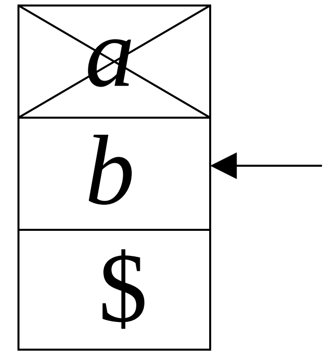
$\lambda, T \rightarrow \lambda$ $\qquad b, b \rightarrow \lambda$



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

Input





Stack

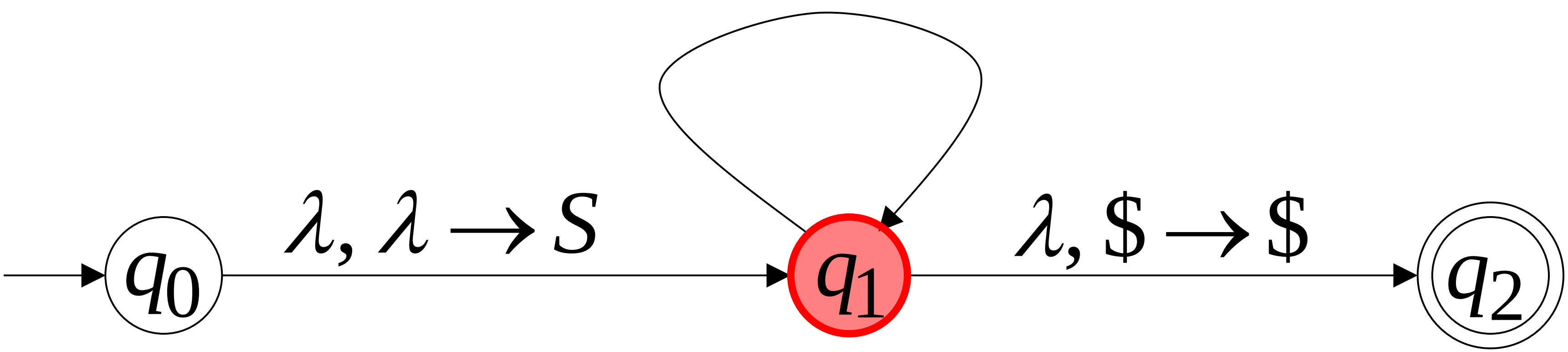
Time 7

$\lambda, S \rightarrow aSTb$

$\lambda, S \rightarrow b$

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

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



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

Input

a	b	a	b

Time 8

$\lambda, S \rightarrow aSTb$

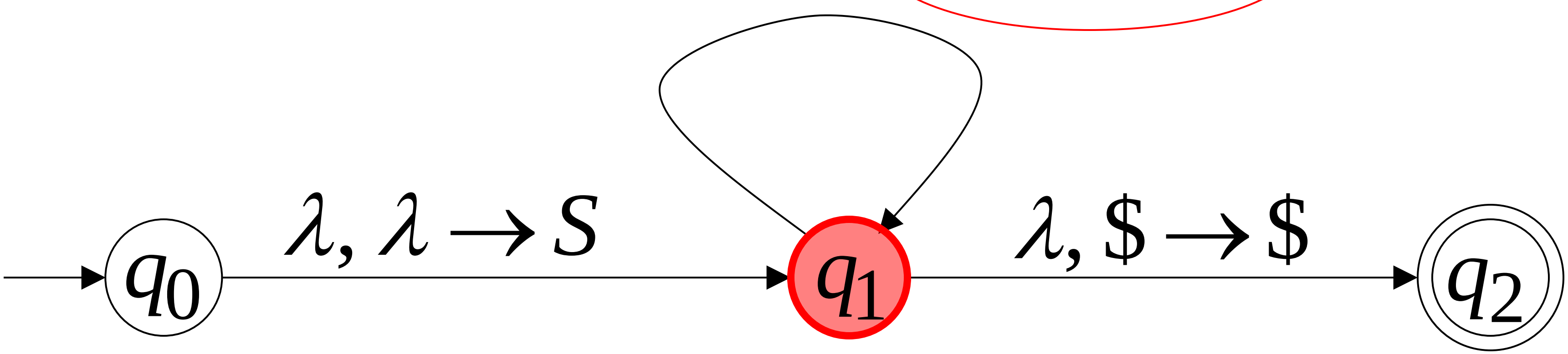
$\lambda, S \rightarrow b$

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

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

$\cancel{b}$
$\$$

Stack



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

Input

a	b	a	b

Time 9

$\lambda, S \rightarrow aSTb$

$\lambda, S \rightarrow b$

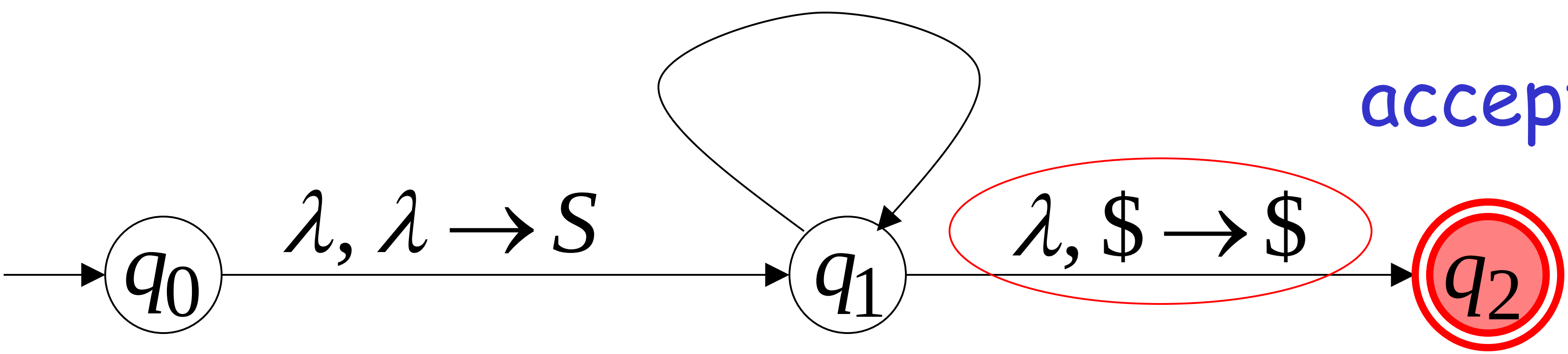
$\lambda, T \rightarrow Ta$ $\quad a, a \rightarrow \lambda$

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

\$

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

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