NPDAs Accept
Context-Free Languages

Theorem:

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

Proof - Step 1:

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

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

Proof - Step 2:

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

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

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:

$$S \to aSTb$$

$$S \to b$$

$$T \to Ta$$

$$T \to \lambda$$

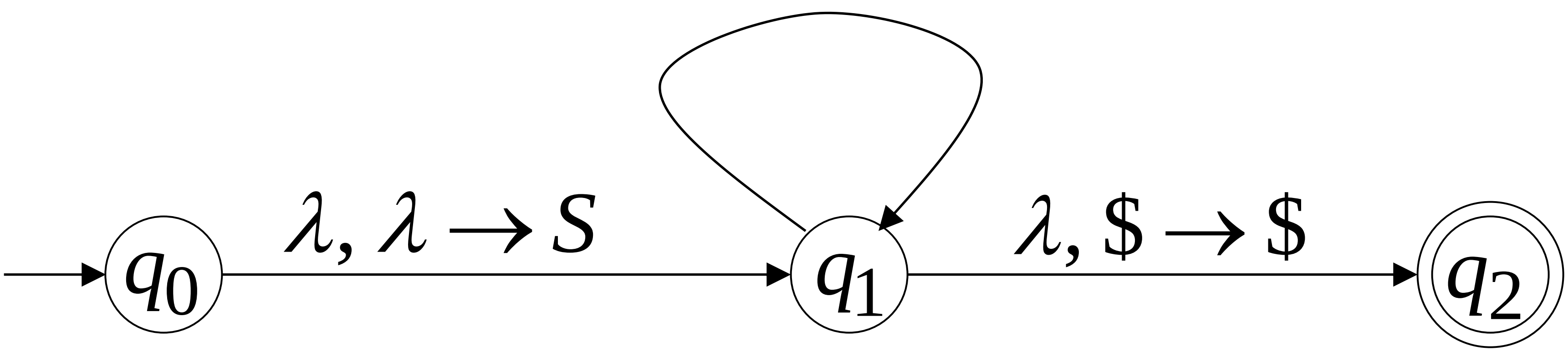
NPDA:

$$\lambda, S \to aSTb$$

$$\lambda, S \to b$$

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

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



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

$$\lambda, S \rightarrow aSTb$$

$$\lambda, S \rightarrow b$$

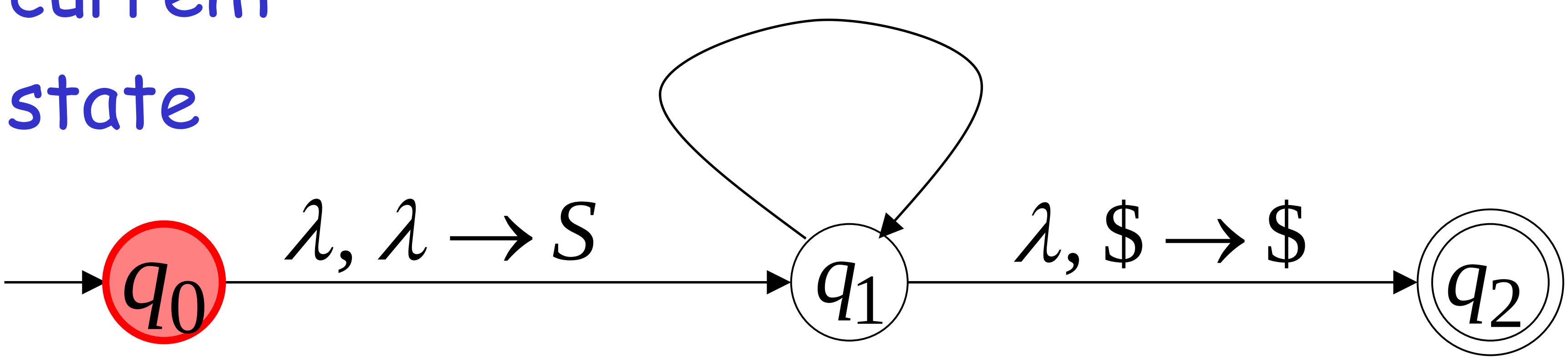
$$\lambda, T \rightarrow Ta \qquad a, a \rightarrow \lambda$$

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

$

Stack

current state



Time 1

Input

a	b	a	b

Stack

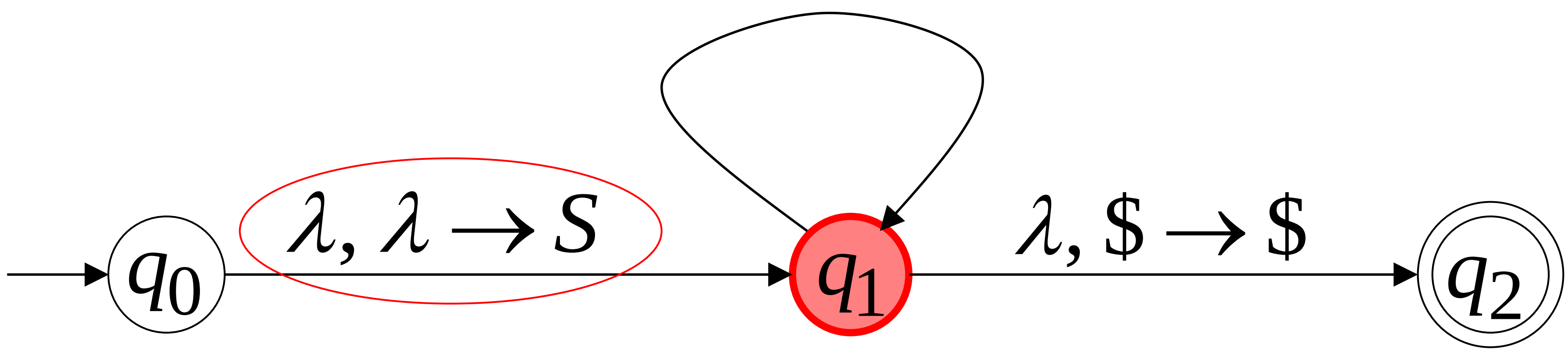
S
$\$$

$$\lambda, S \rightarrow aSTb$$

$$\lambda, S \rightarrow b$$

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

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



Time 2

Input

a	b	a	b

Stack

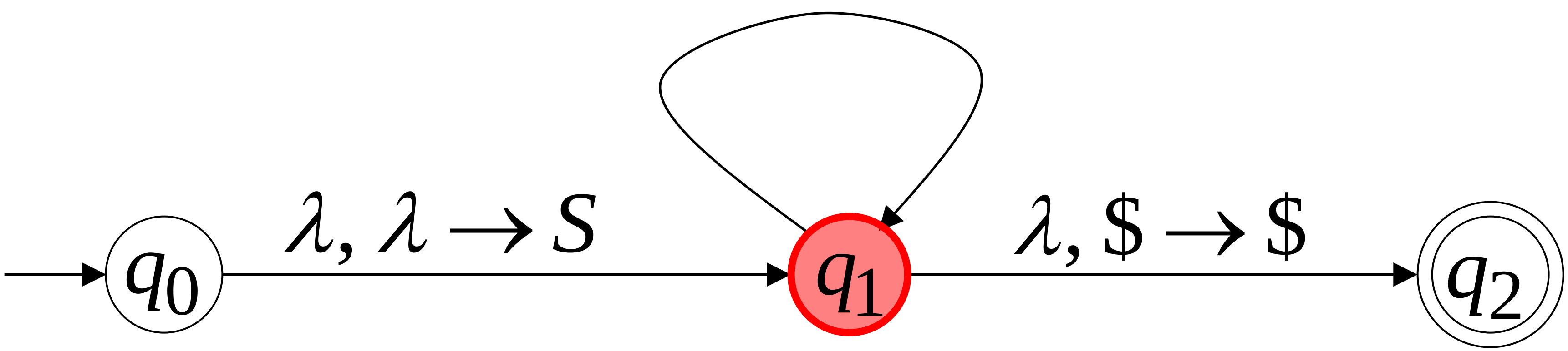
a
S
T
b
$\$$

$$\lambda, S \rightarrow aSTb$$

$$\lambda, S \rightarrow b$$

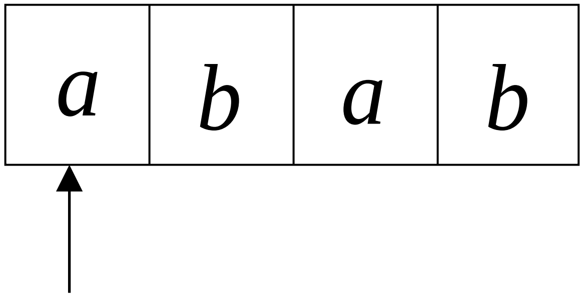
$$\lambda, T \rightarrow Ta \qquad a, a \rightarrow \lambda$$

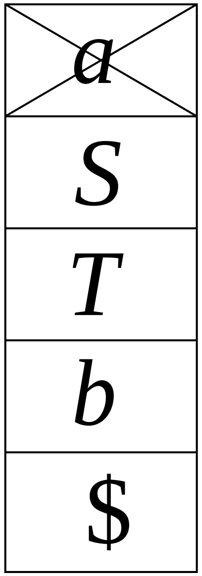
$$\lambda, T \rightarrow \lambda \qquad b, b \rightarrow \lambda$$



Time 3

Input





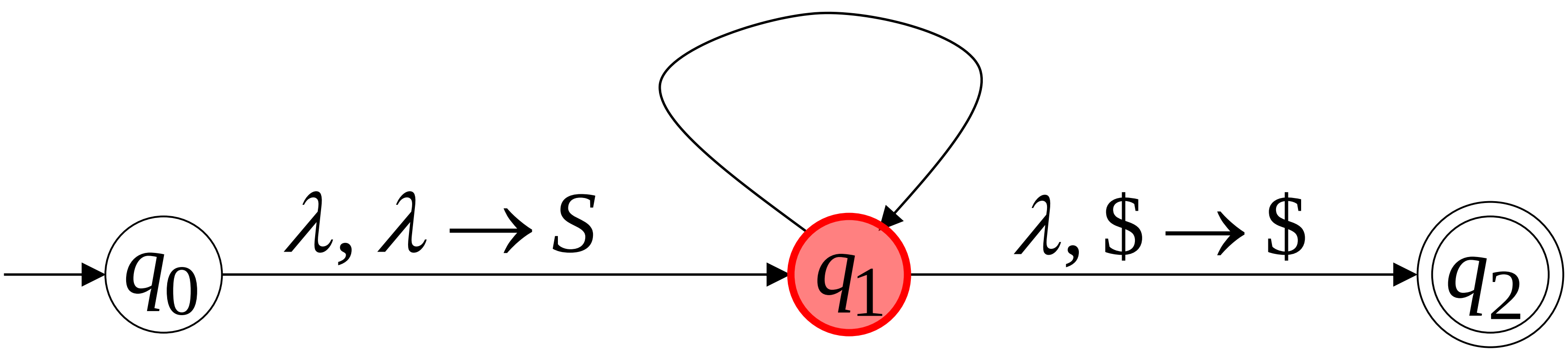
Stack

$$\lambda, S \rightarrow aSTb$$

$$\lambda, S \rightarrow b$$

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

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



Time 4

Input

a	b	a	b

Stack

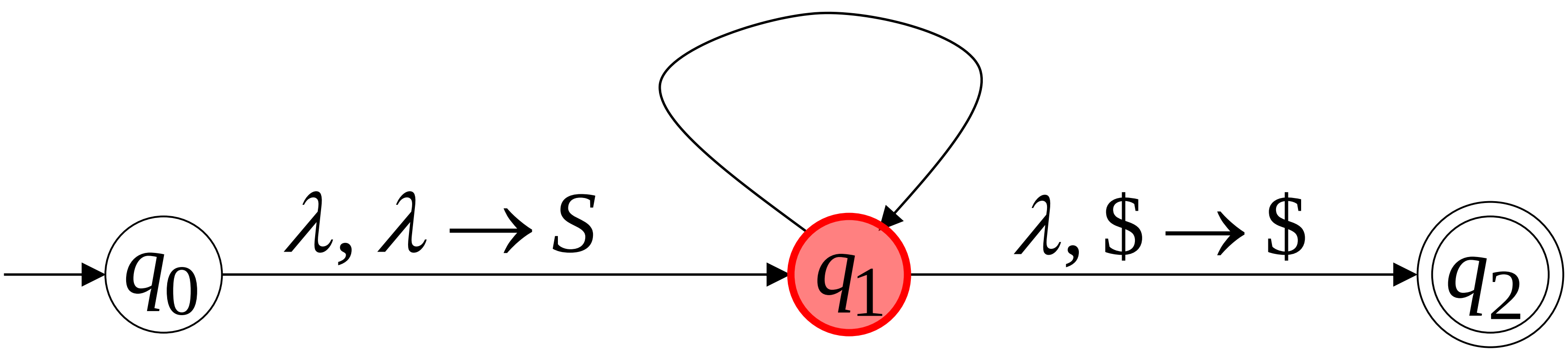
b
T
b
$\$$

$$\lambda, S \rightarrow aSTb$$

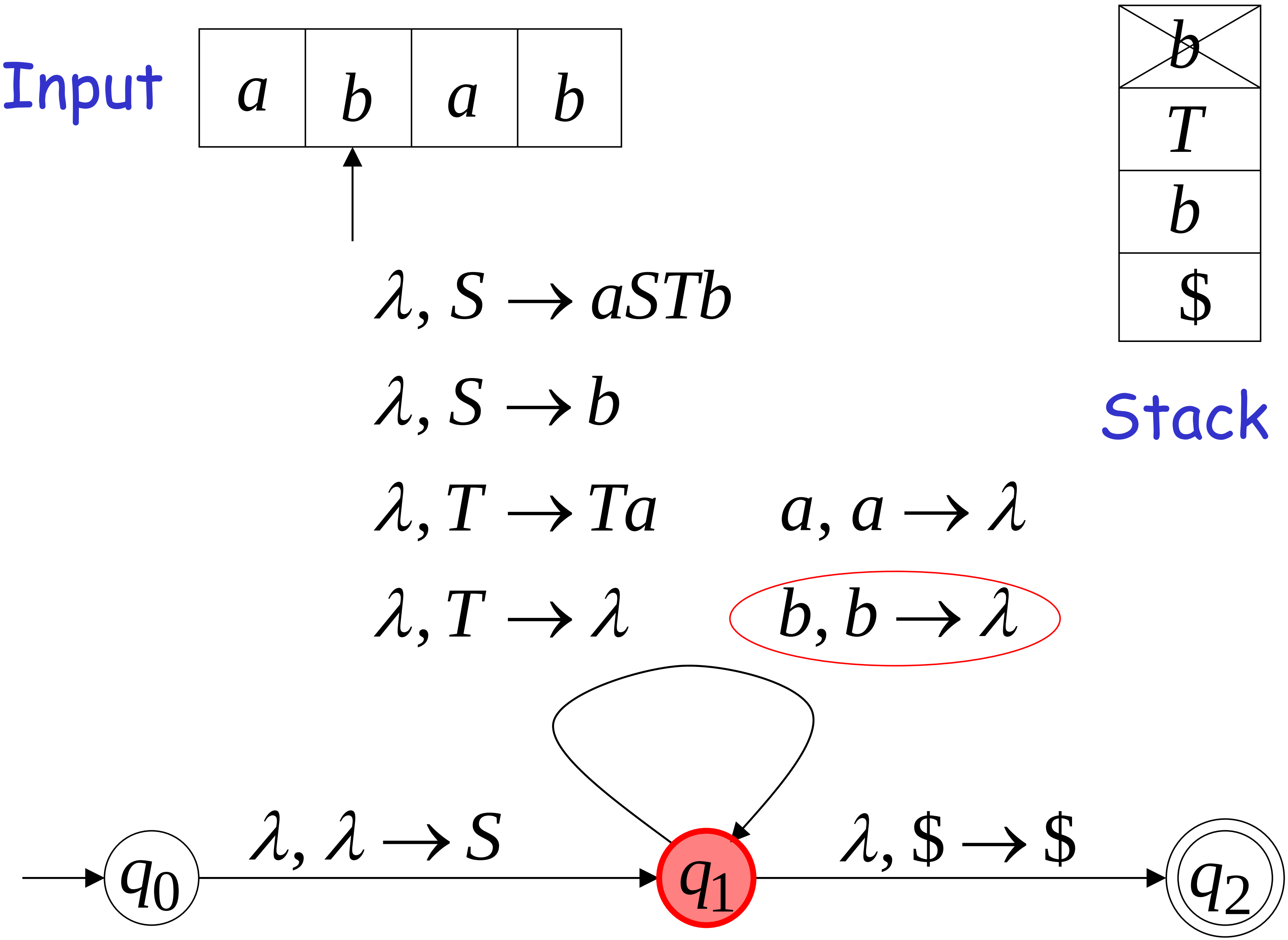
$$\lambda, S \rightarrow b$$

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

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



Time 5



Time 6

Input

a	b	a	b

Stack

T
a
b
$\$$

$\lambda, S \rightarrow aSTb$

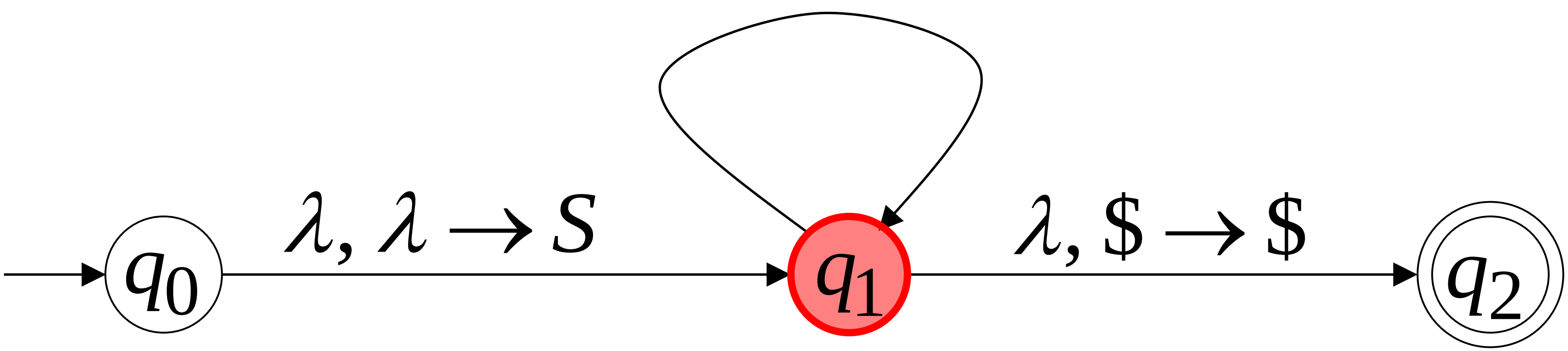
$\lambda, S \rightarrow b$

$\lambda, T \rightarrow Ta$

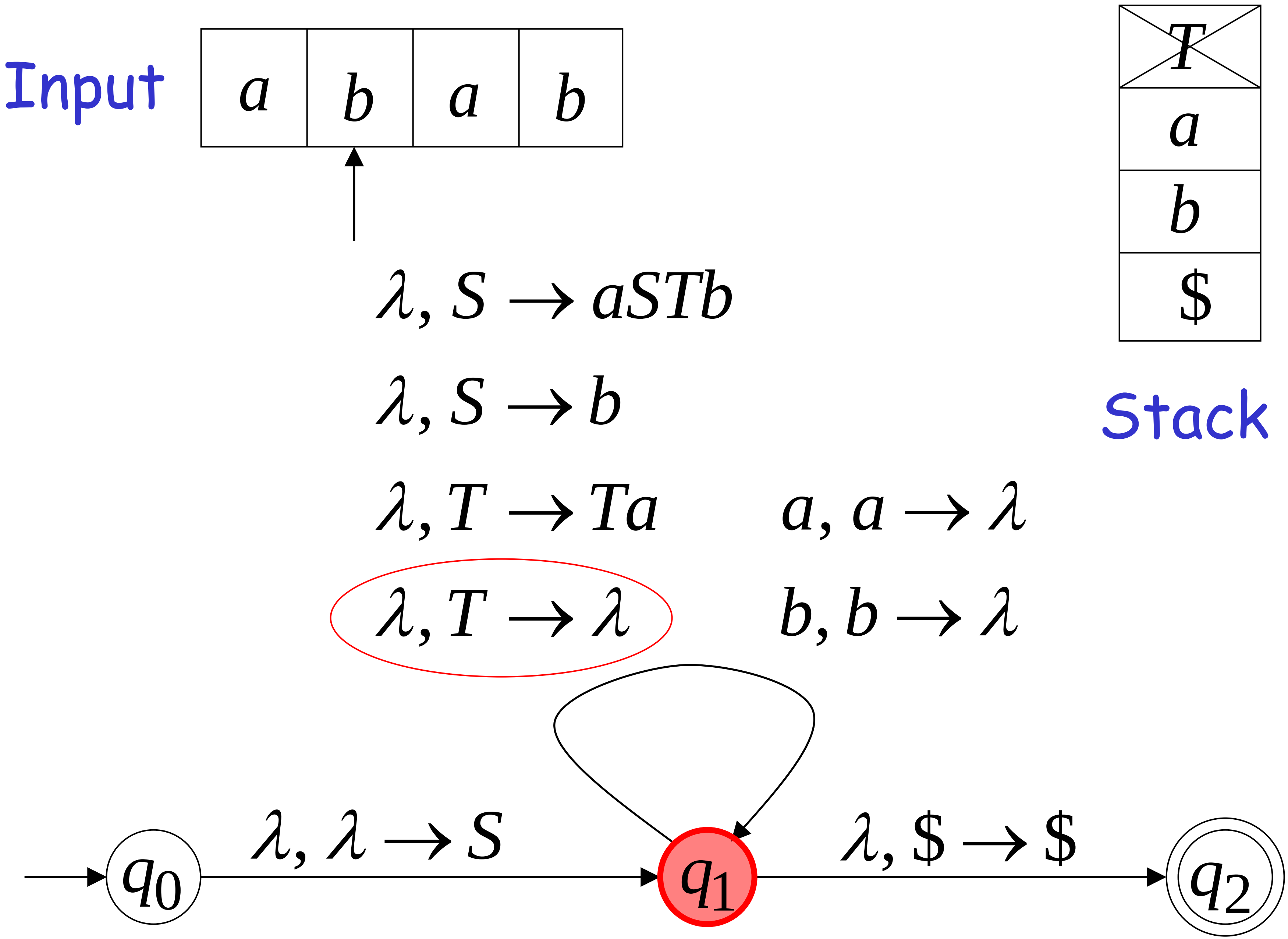
$\lambda, T \rightarrow \lambda$

$a, a \rightarrow \lambda$

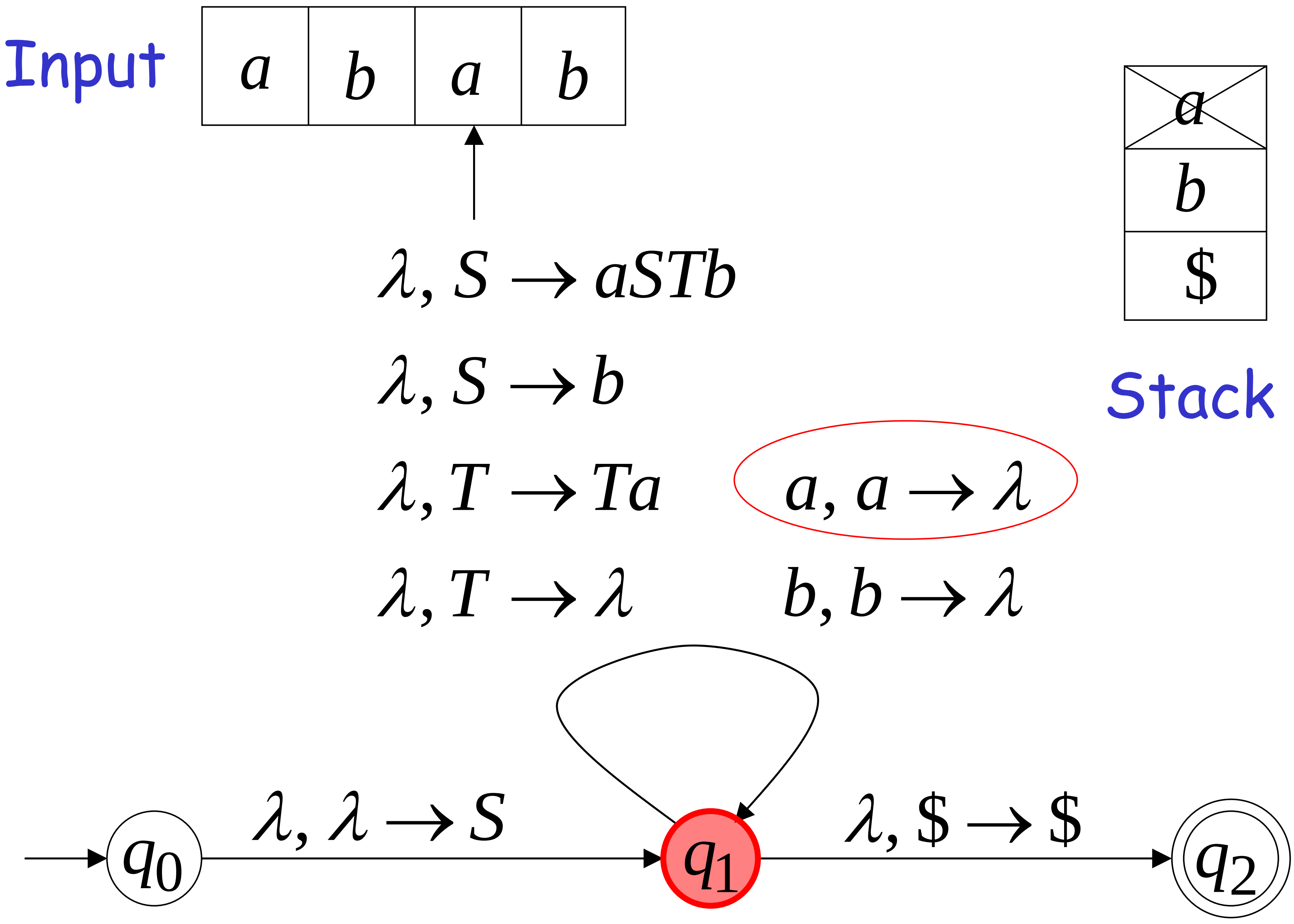
$b, b \rightarrow \lambda$



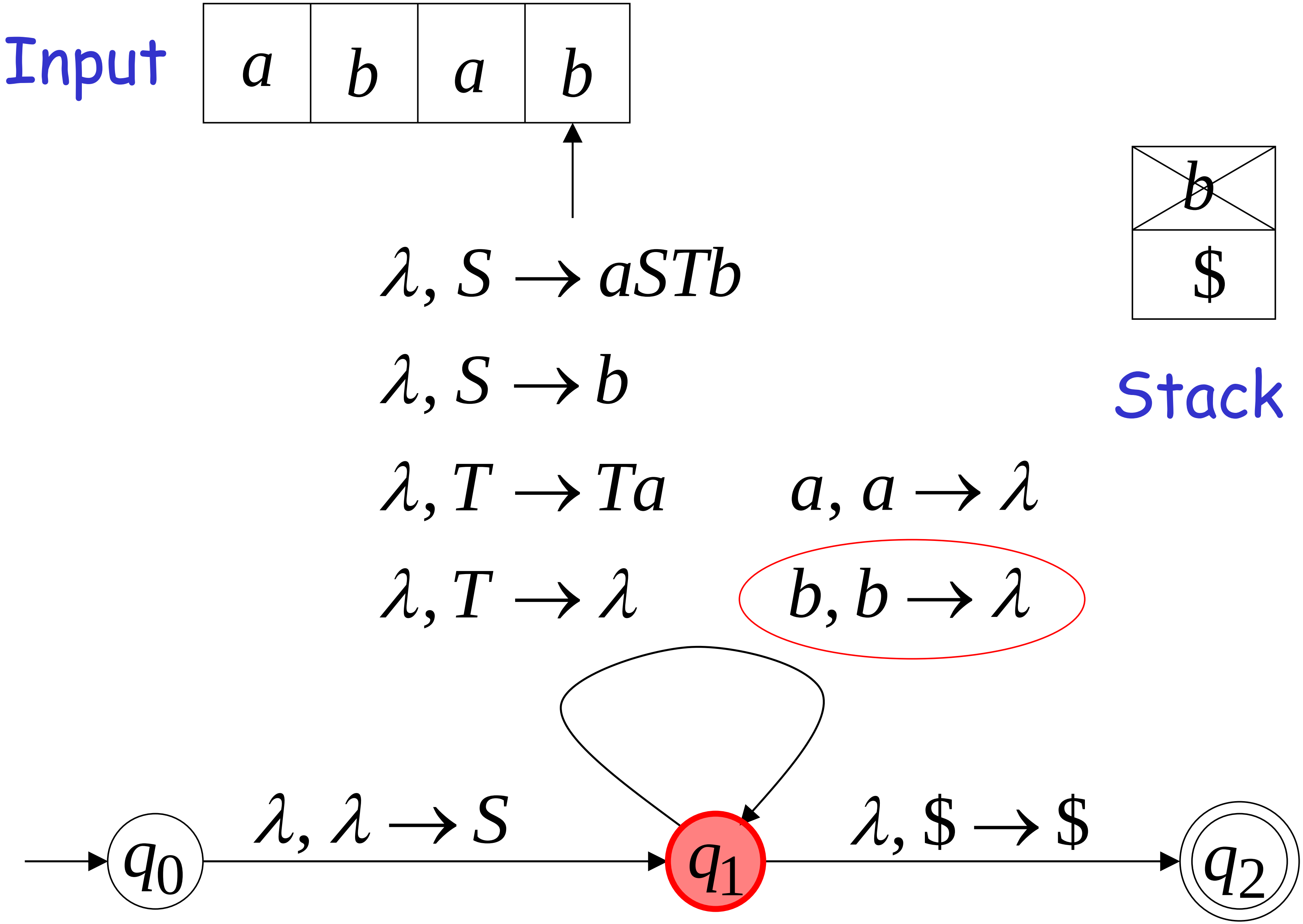
Time 7



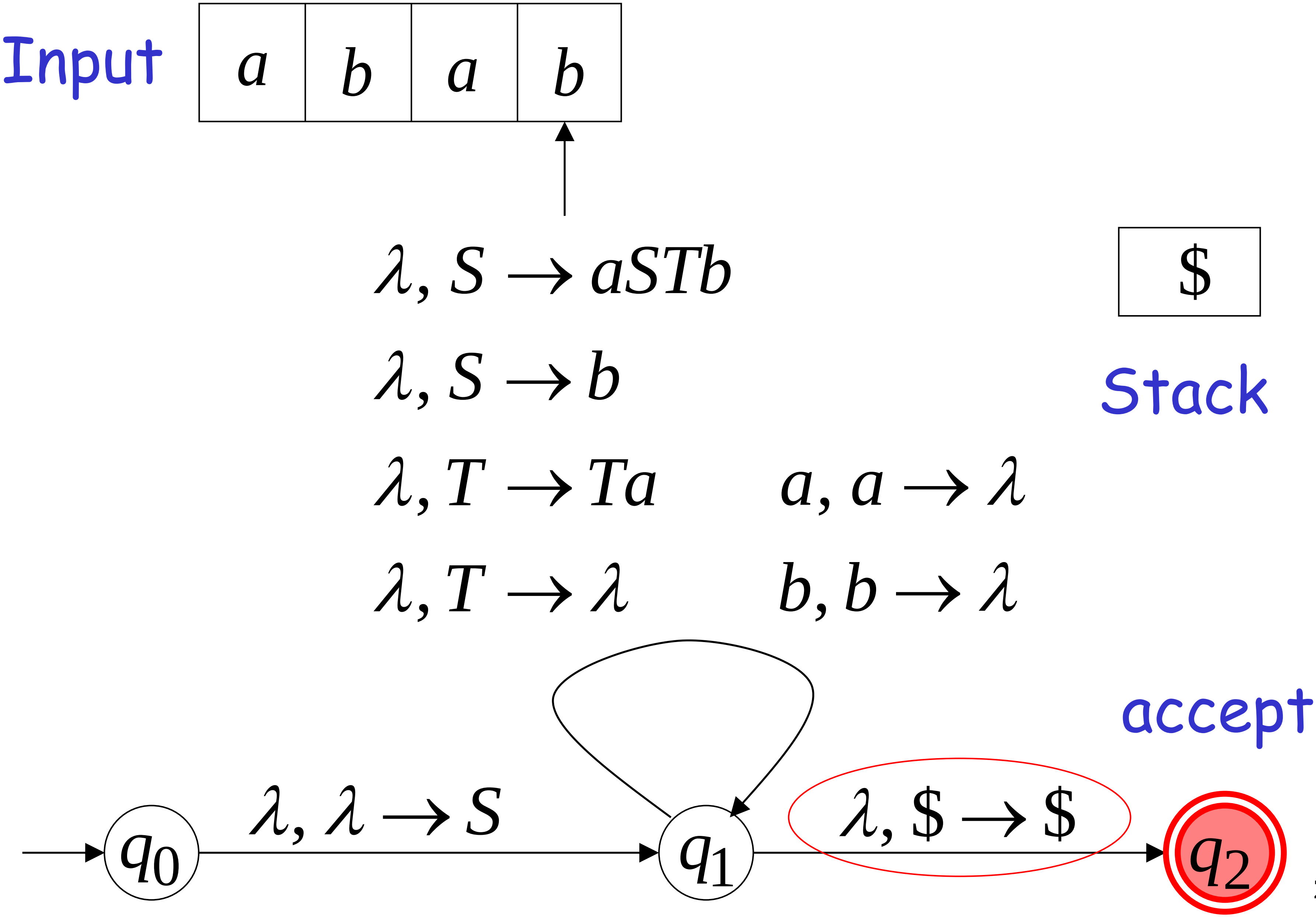
Time 8



Time 9



Time 10



Input
a
b
a
b
$\lambda, S \to aSTb$
$\lambda, S \to b$
$\lambda, T \to Ta$
$\lambda, T \to \lambda$
$a, a \to \lambda$
$b, b \to \lambda$
$
Stack
accept
$\lambda, \lambda \to S$
$\lambda, \$ \to \$$
q_0
q_1
q_2

In general:

Given any grammar G

We can construct a NPDA M

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

Constructing NPDA M from grammar G :

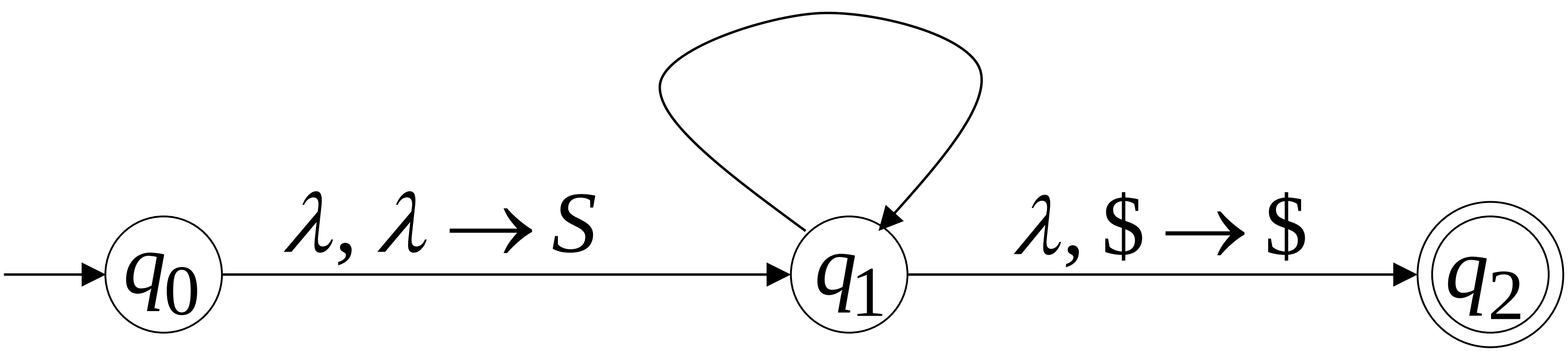
For any production

$$A \to w$$

For any terminal

$$a$$

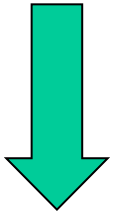
$$\lambda, A \to w \qquad a, a \to \lambda$$



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

Converting NPDAs to Context-Free Grammars

For any NPDA M

we will construct

a context-free grammar G with

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

Intuition: The grammar simulates the machine

A derivation in Grammar G :

$$S \Rightarrow \cdots \Rightarrow \underbrace{abc \ldots ABC \ldots}_{} \Rightarrow \ldots \Rightarrow abc \ldots$$

Current configuration in NPDA M

A derivation in Grammar G :

terminals　　variables

$$S \Rightarrow \cdots \Rightarrow abc\ldots ABC\ldots \Rightarrow \ldots \Rightarrow abc\ldots$$

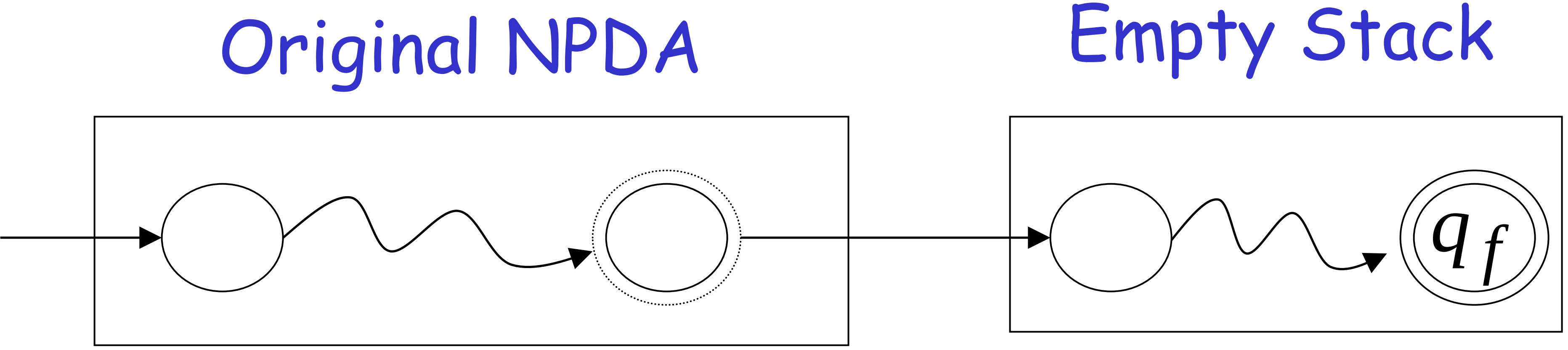
Input processed　　Stack contents

in NPDA M

Some Necessary Modifications

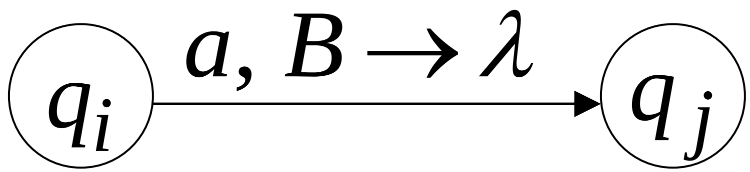
First, we modify the NPDA:

- It has a single final state q_f
- It empties the stack when it accepts the input

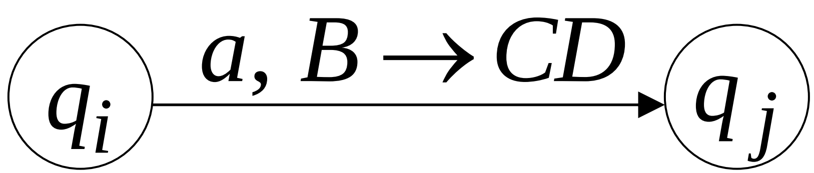


Second, we modify the NPDA transitions:

all transitions will have form



or



B, C, D: stack symbols

Example of a NPDA in correct form:

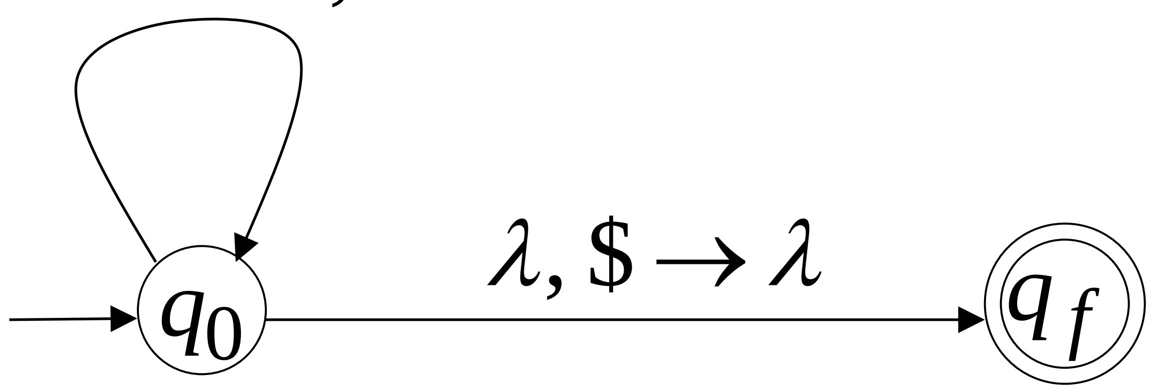
$$L(M) = \{w: \quad n_a = n_b\}$$

$: initial stack symbol

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

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

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



The Grammar Construction

In grammar G:

Stack symbol

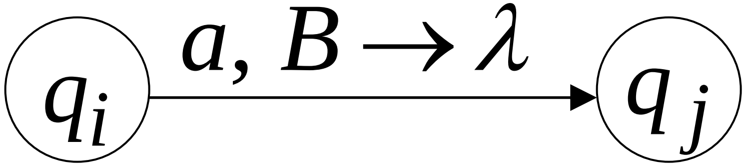
Variables: $(q_i B q_j)$

states

Terminals:

Input symbols of NPDA

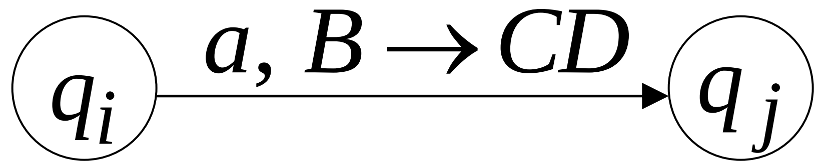
For each transition



We add production

$$(q_i B q_j) \rightarrow a$$

For each transition



We add production $(q_i B q_k) \rightarrow a(q_j C q_l)(q_l D q_k)$

For all states q_k, q_l

Stack bottom symbol

Start Variable: $(q_o\$q_f)$

Start state

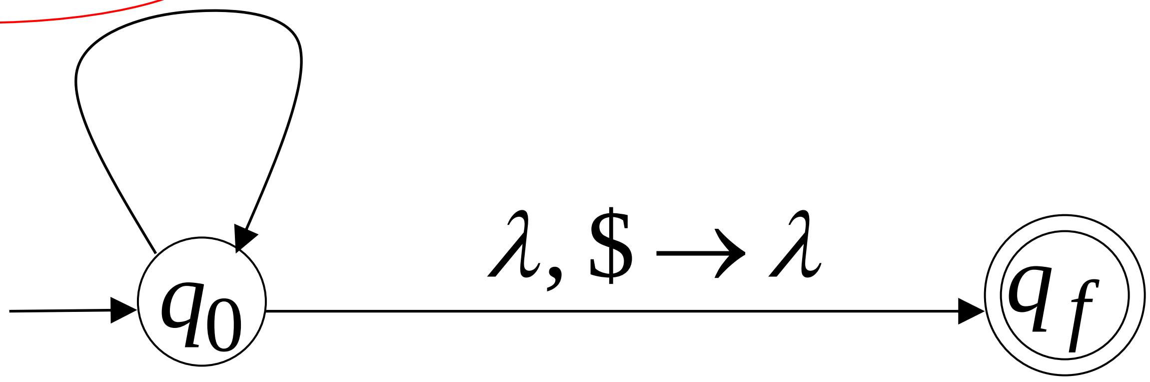
final state

Example:

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

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

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



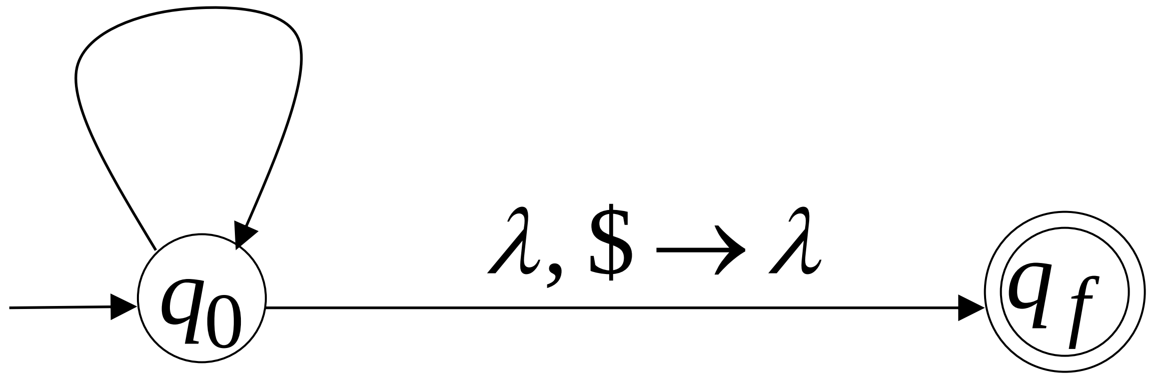
Grammar production: $(q_0 1 q_0) \rightarrow a$

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:

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

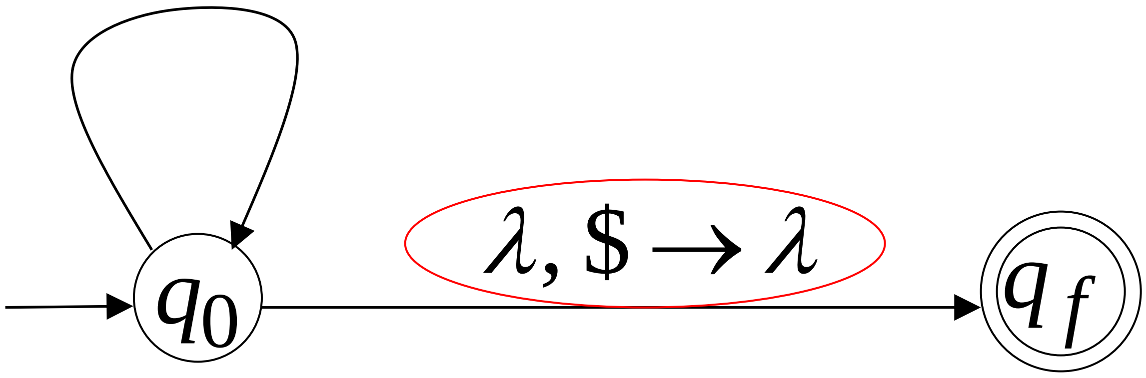
$$(q_0 \$ q_f) \rightarrow b(q_0 1 q_0)(q_0 \$ q_f) \mid b(q_0 1 q_f)(q_f \$ q_f)$$

Example:

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

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

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



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

Resulting Grammar:

$(q_0 \$ q_f)$: start variable

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

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

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

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

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

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

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

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

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

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

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

Derivation of string $abba$

$$(q_0\$q_f) \Rightarrow \; a(q_0 0 q_0)(q_0\$q_f) \Rightarrow$$

$$ab(q_0\$q_f) \Rightarrow$$

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

$$abba(q_0\$q_f) \Rightarrow \; abba$$

In general, in Grammar:

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

if and only if

w is accepted by the NPDA

Explanation:

By construction of Grammar:

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

if and only if

in the NPDA going from q_i to q_j
the stack doesn't change below A
and A is removed from stack