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

Proof - step 1

Converting
Context-Free Grammars
to
NPDAs

We will convert any context-free grammar G

to an NPDA automaton M

Such that:

M Simulates leftmost derivations of G

Leftmost derivation

$G:$

$$S \Rightarrow \cdots\cdots \Rightarrow \underbrace{\sigma_1\sigma_2\cdots\sigma_k}_{\text{Input processed}}\underbrace{V_1V_2\cdots V_m}_{\text{Stack contents}} \Rightarrow \cdots$$

leftmost variable

$M:$

Simulation of derivation

Input

σ_1	σ_2	$\cdots$	σ_k	σ_{k+1}	$\cdots$

Stack

V_1
V_2
$\vdots$
V_m
$\$$

$G:$

Leftmost derivation

$$S \Rightarrow \cdots\cdots\cdots \Rightarrow \sigma_1\sigma_2\cdots\sigma_n$$

string of terminals

$M:$

Simulation of derivation

Stack

Input

σ_1	σ_2	$\cdots\cdots$	σ_n

$\$$

end of input is reached

An example grammar:

$$S \to aSTb$$

$$S \to b$$

$$T \to Ta$$

$$T \to \lambda$$

What is the equivalent NPDA?

Grammar:

$$S \to aSTb$$

$$S \to b$$

$$T \to Ta$$

$$T \to \lambda$$

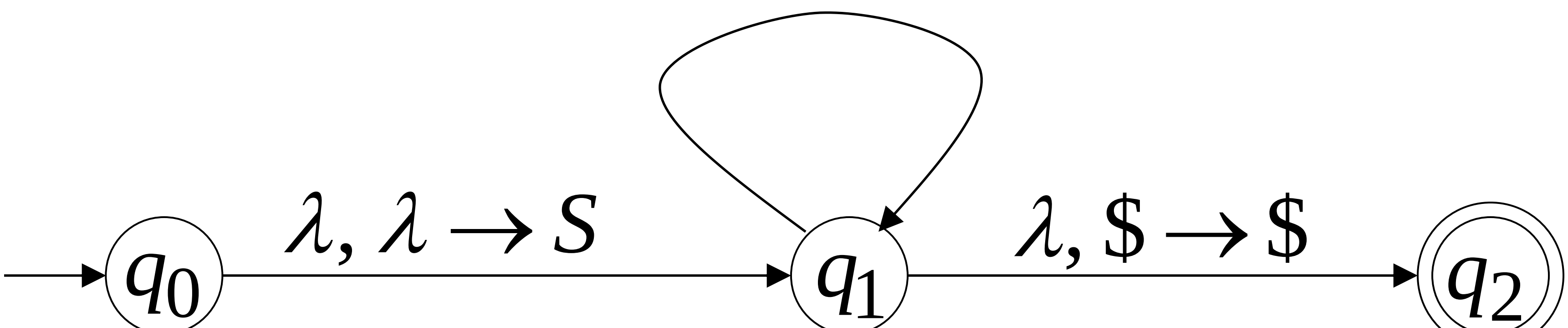
NPDA:

$$\lambda, S \to aSTb$$

$$\lambda, S \to b$$

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

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



Grammar:

$$S \to aSTb$$

$$S \to b$$

$$T \to Ta$$

$$T \to \lambda$$

A leftmost derivation:

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

Derivation:

Input

a	b	a	b

Time 0

$$\lambda, S \to aSTb$$

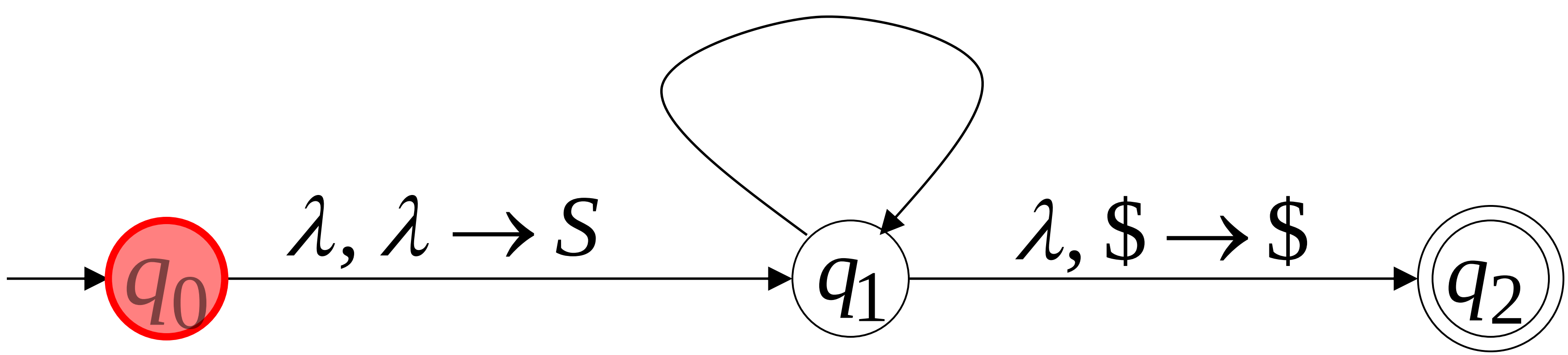
$$\lambda, S \to b$$

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

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

$\$$

Stack



Derivation: S

Input

a	b	a	b

Time 0

$$\lambda, S \rightarrow aSTb$$

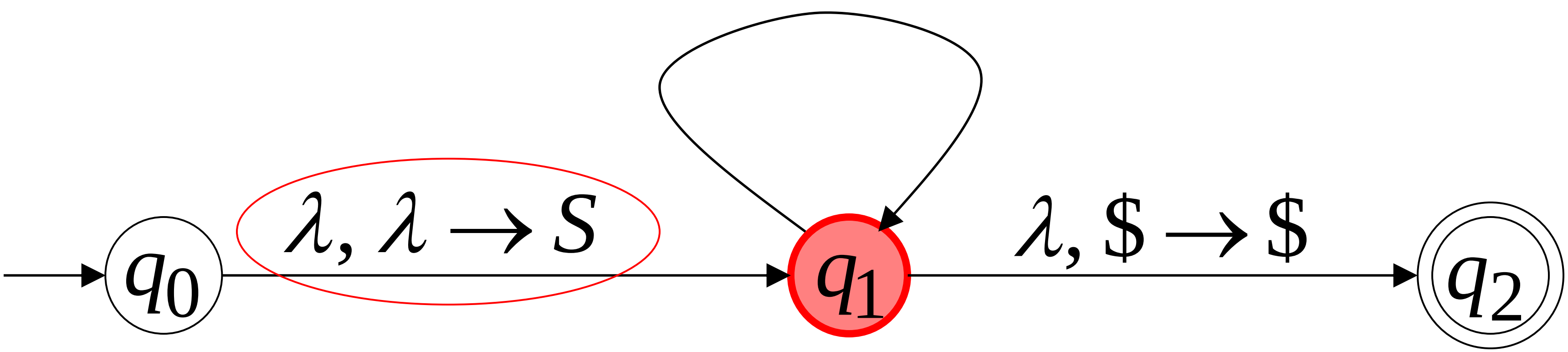
$$\lambda, S \rightarrow b$$

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

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

S
$\$$

Stack



Derivation: $S \Rightarrow aSTb$

Input

a	b	a	b

Time 1

$\lambda, S \rightarrow aSTb$

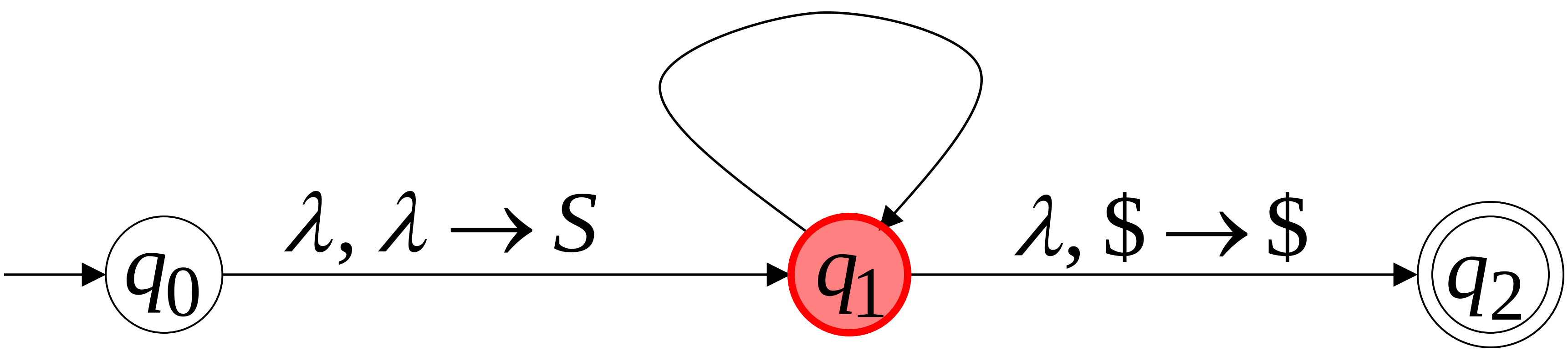
$\lambda, S \rightarrow b$

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

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

a
S
T
b
$\$$

Stack



Derivation: $S \Rightarrow aSTb$

Input

a	b	a	b

Time 2

$\lambda, S \rightarrow aSTb$

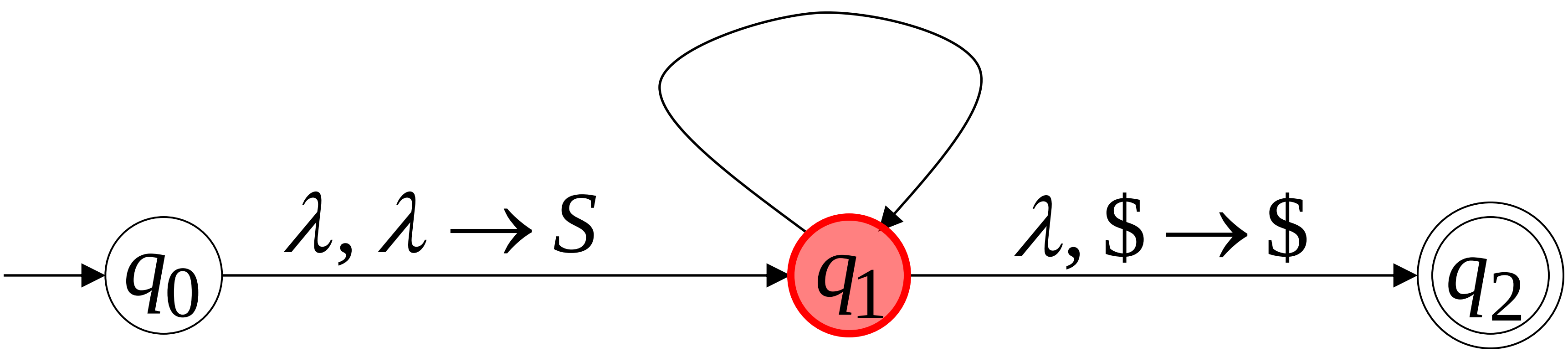
$\lambda, S \rightarrow b$

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

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

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

Stack



Derivation: $S \Rightarrow aSTb \Rightarrow abTb$

Input

a	b	a	b

Time 3

$\lambda, S \rightarrow aSTb$

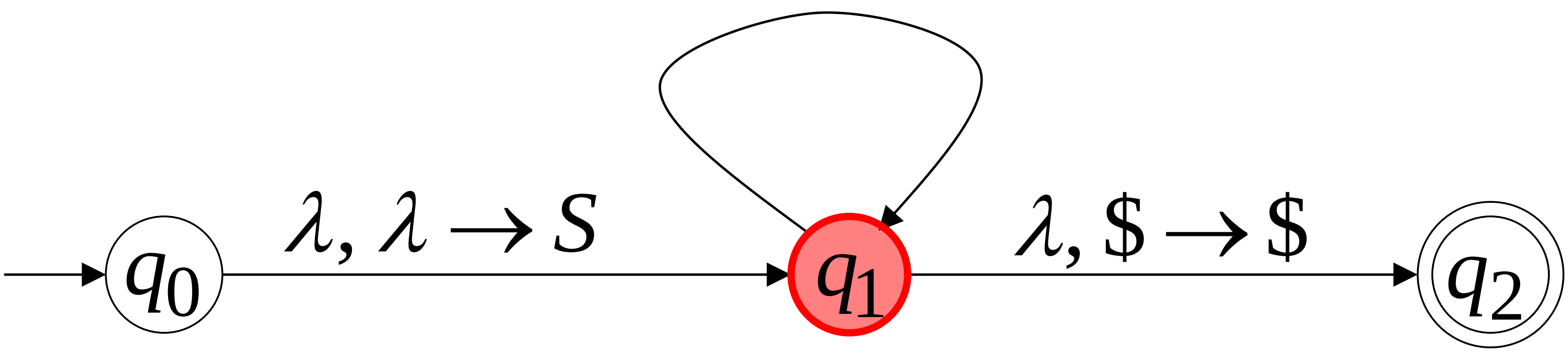
$\lambda, S \rightarrow b$

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

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

b
T
b
$\$$

Stack



Derivation: $S \Rightarrow aSTb \Rightarrow abTb$

Input

a	b	a	b

Time 4

$\lambda, S \rightarrow aSTb$

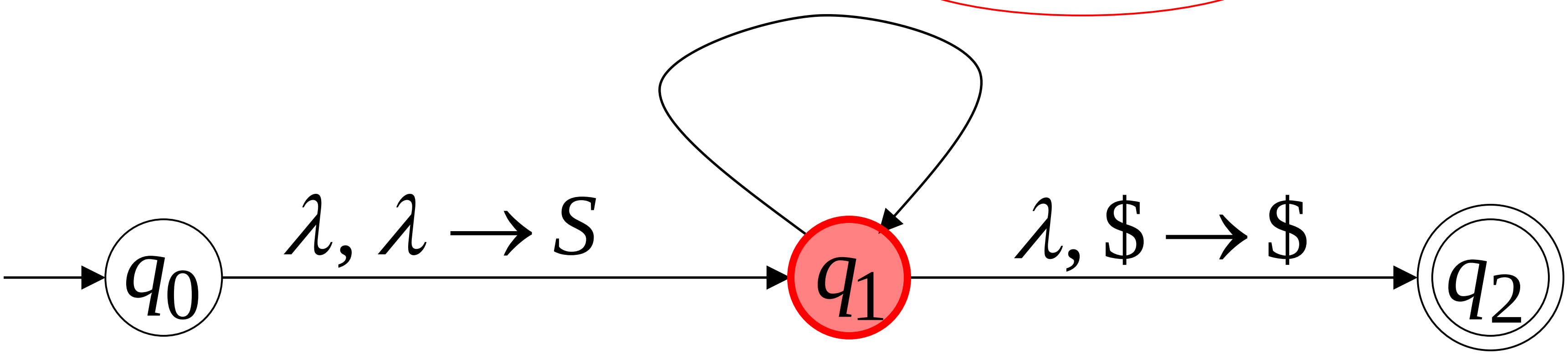
$\lambda, S \rightarrow b$

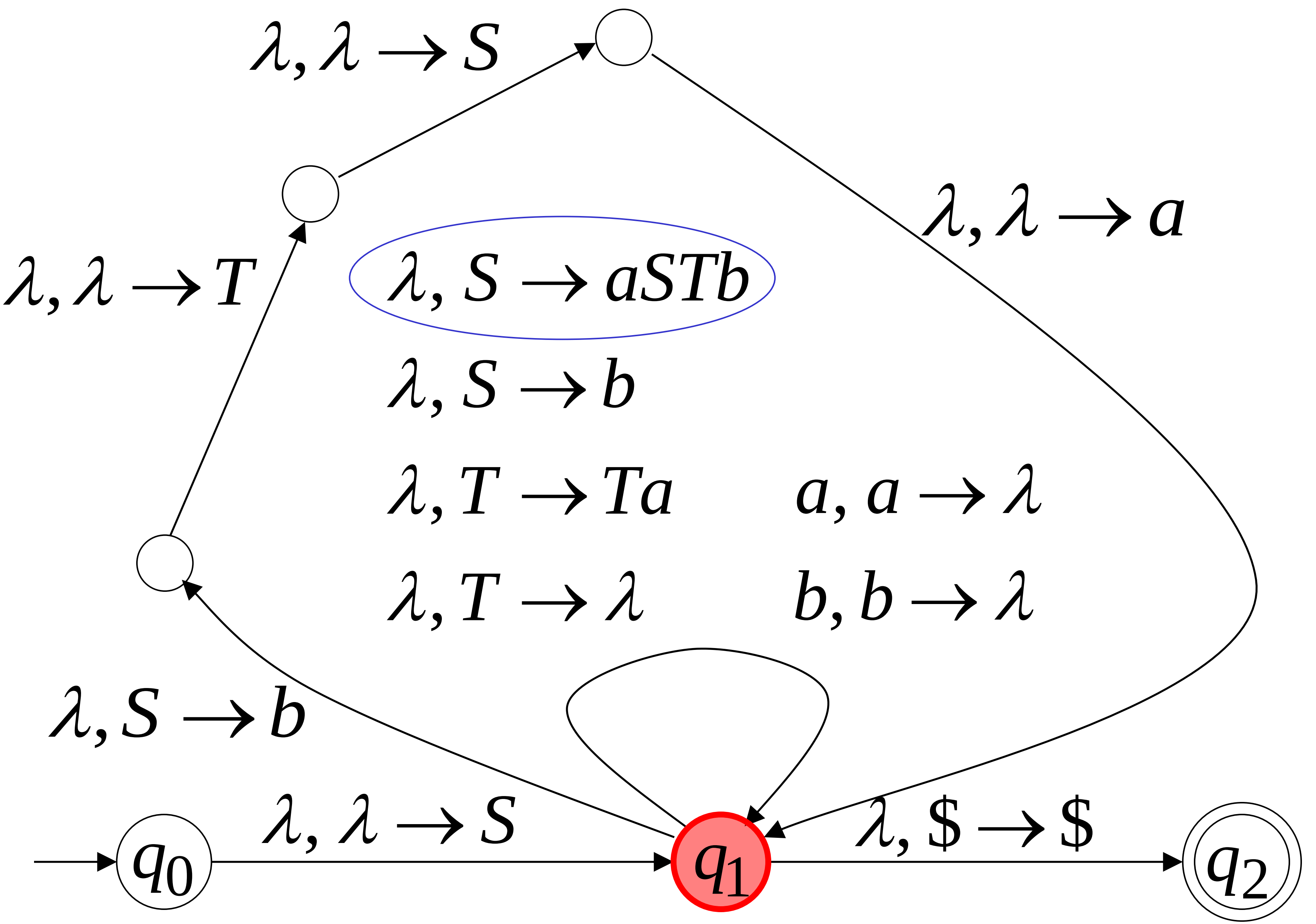
$\lambda, T \rightarrow Ta$ $\qquad a, a \rightarrow \lambda$

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

~~b~~
T
b
$\$$

Stack





$\lambda, \lambda \rightarrow S$
$\lambda, \lambda \rightarrow a$
$\lambda, \lambda \rightarrow T$
$\lambda, S \rightarrow aSTb$
$\lambda, S \rightarrow b$
$\lambda, T \rightarrow Ta$
$a, a \rightarrow \lambda$
$\lambda, T \rightarrow \lambda$
$b, b \rightarrow \lambda$
$\lambda, S \rightarrow b$
$\lambda, \lambda \rightarrow S$
$\lambda, \$ \rightarrow \$$
q_0
q_1
q_2

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

Input

a	b	a	b

Time 5

T
a
b
$\$$

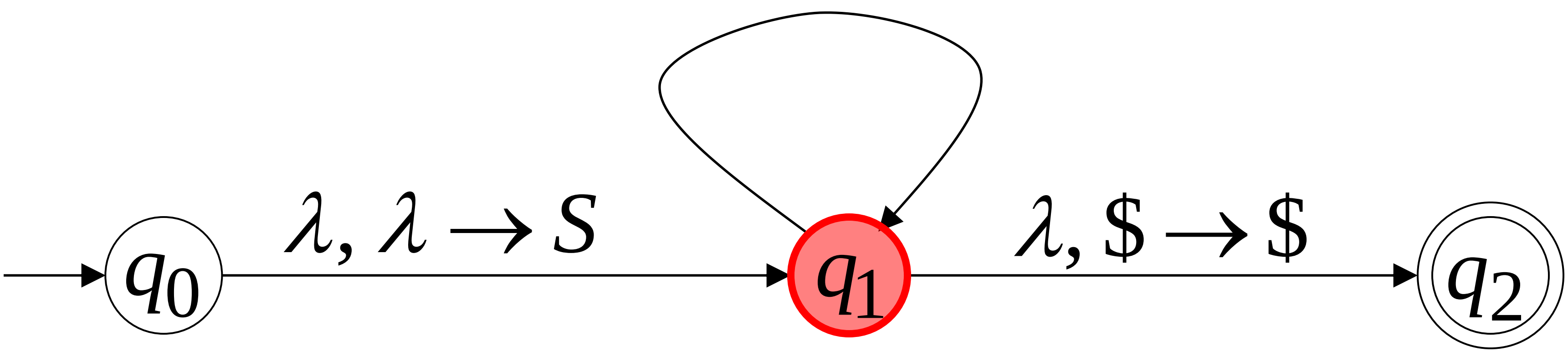
Stack

$\lambda, S \rightarrow aSTb$

$\lambda, S \rightarrow b$

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

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



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

Input

a	b	a	b

Stack

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

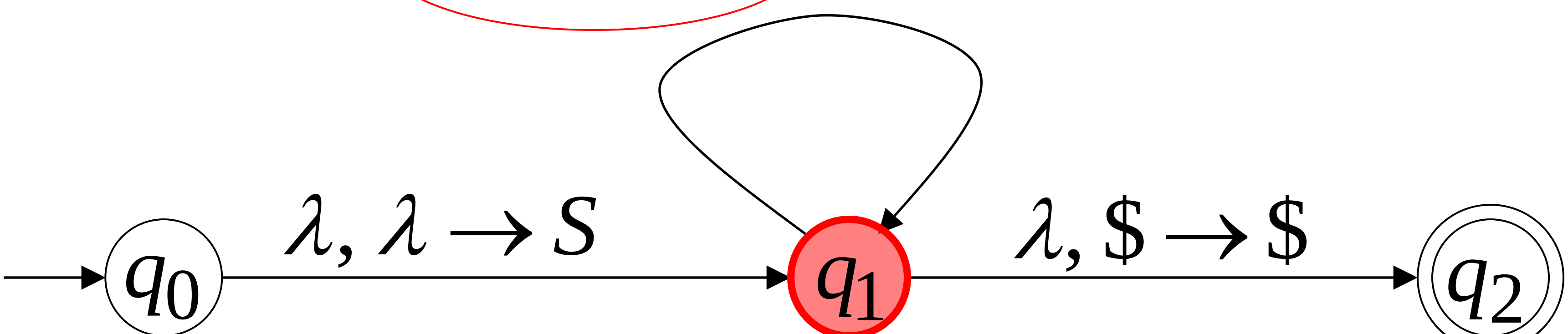
Time 6

$\lambda, S \rightarrow aSTb$

$\lambda, S \rightarrow b$

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

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



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

Input

a	b	a	b

Time 7

$\lambda, S \rightarrow aSTb$

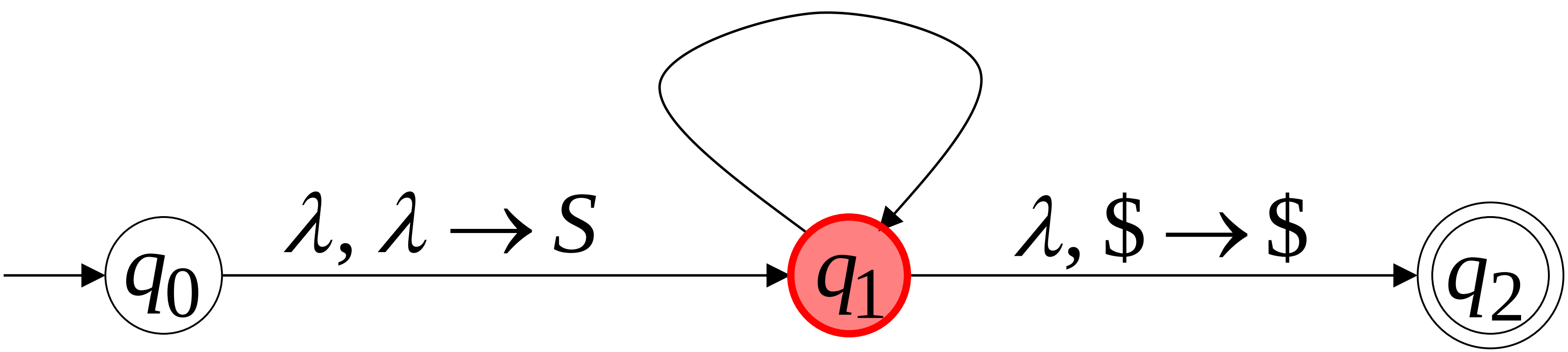
$\lambda, S \rightarrow b$

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

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

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

Stack



Derivation:

Input

a	b	a	b

Time 8

$\lambda, S \rightarrow aSTb$

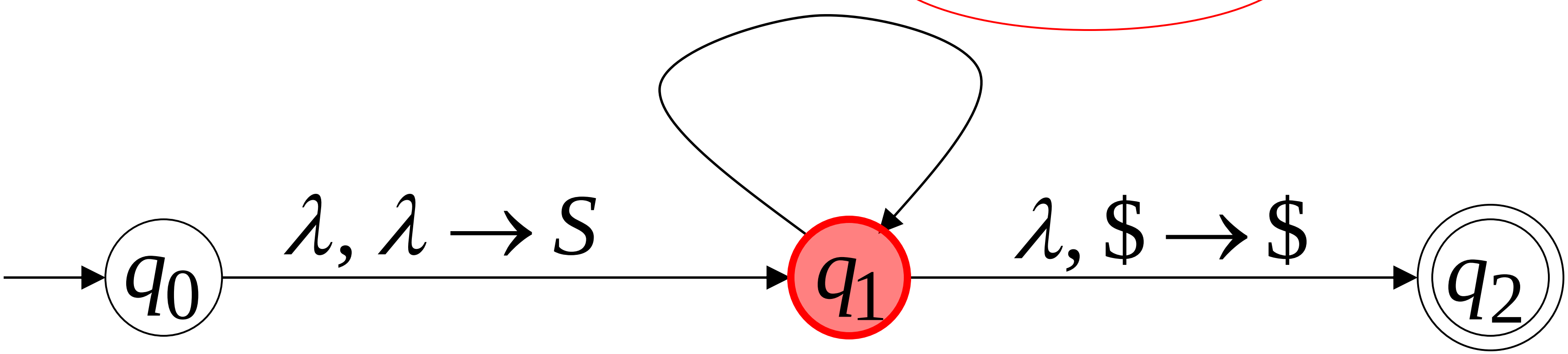
$\lambda, S \rightarrow b$

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

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

$\cancel{b}$
$\$$

Stack



Derivation:

Input

a	b	a	b

Time 9

$\lambda, S \rightarrow aSTb$

$\lambda, S \rightarrow b$

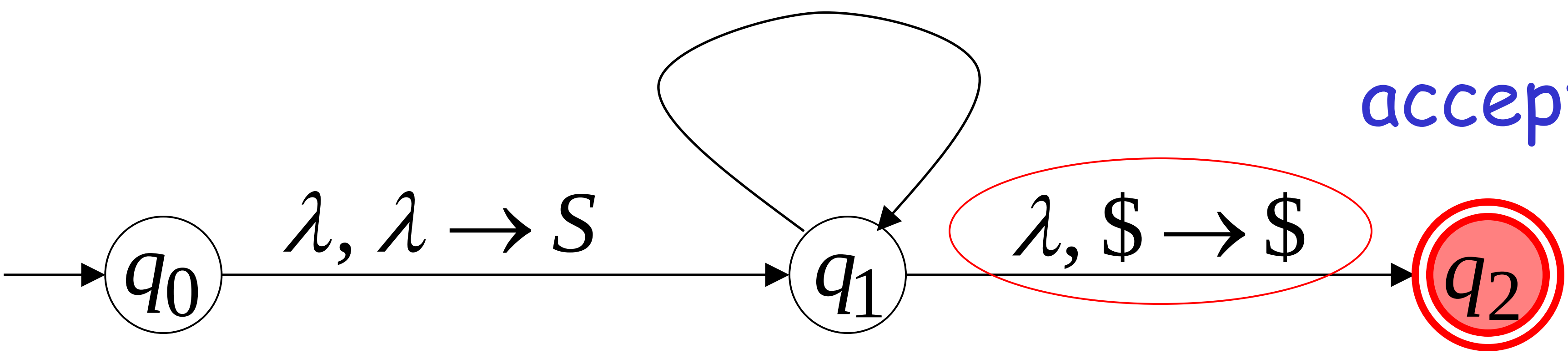
$\lambda, T \rightarrow Ta$ $\quad a, a \rightarrow \lambda$

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

$\$$

Stack

accept



In general:

Given any grammar G

We can construct a NPDA M

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

Constructing NPDA M from grammar G :

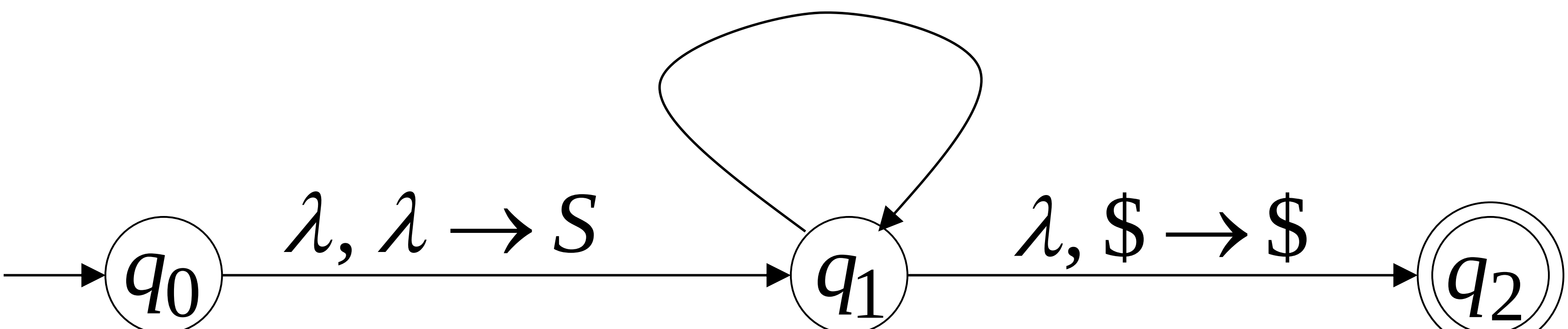
For any production

$$A \rightarrow w$$

For any terminal

$$a$$

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



Grammar G generates string w

if and only if

NPDA M accepts w

$\Downarrow$

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

Therefore:

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

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

Proof - step 2

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 :

terminals variables

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

Input processed Stack contents

Current configuration in NPDA M

Some Necessary Modifications

Modify (if necessary) the NPDA so that:

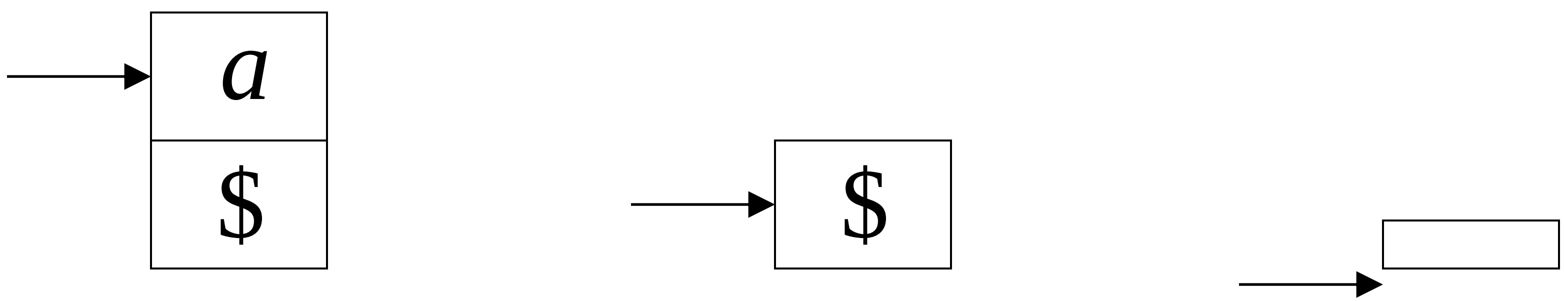
1) The stack is never empty

2) It has a single final state
   and empties the stack when it accepts a string

3) Has transitions in a special form

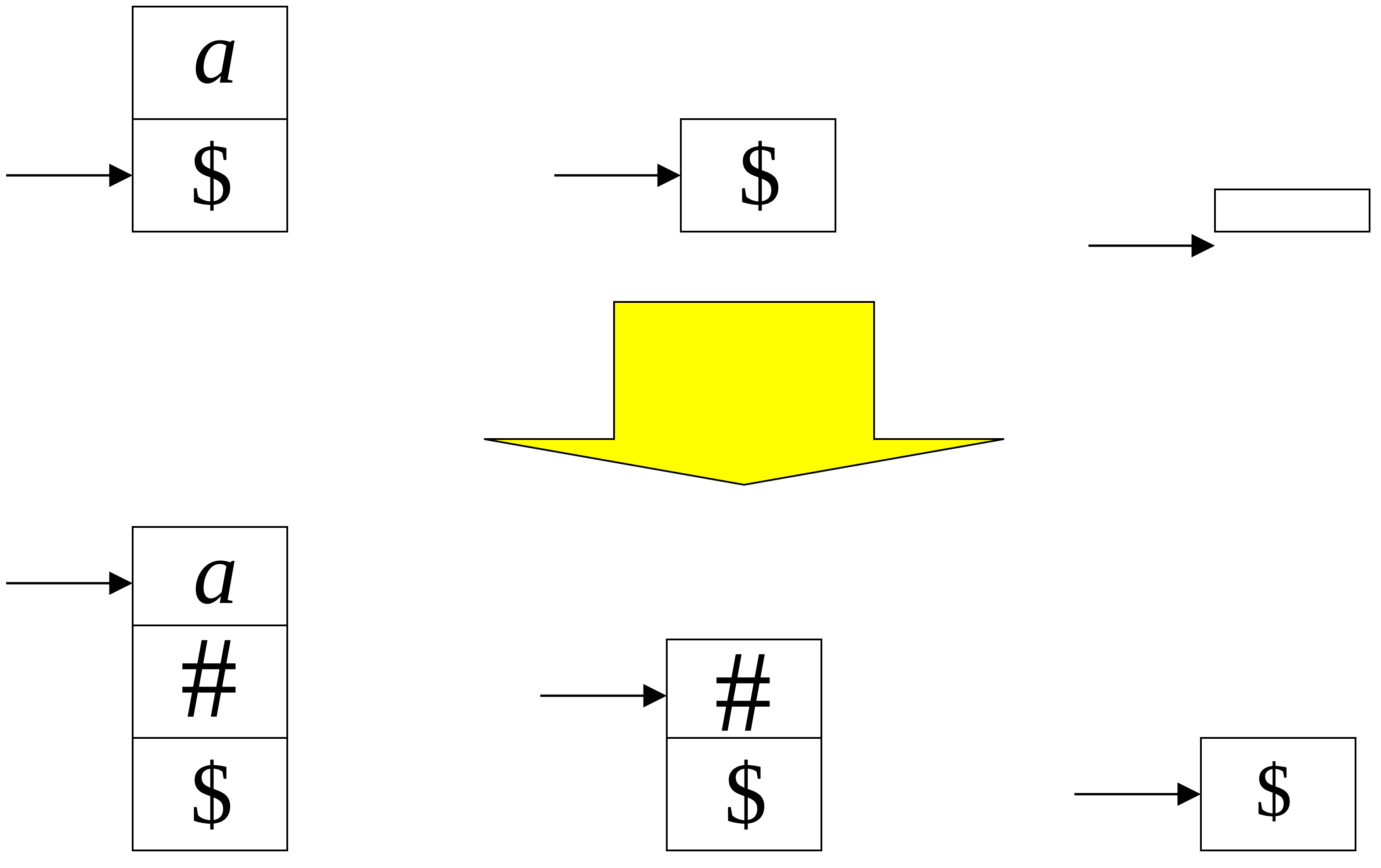
1) Modify the NPDA so that the stack is never empty

Stack

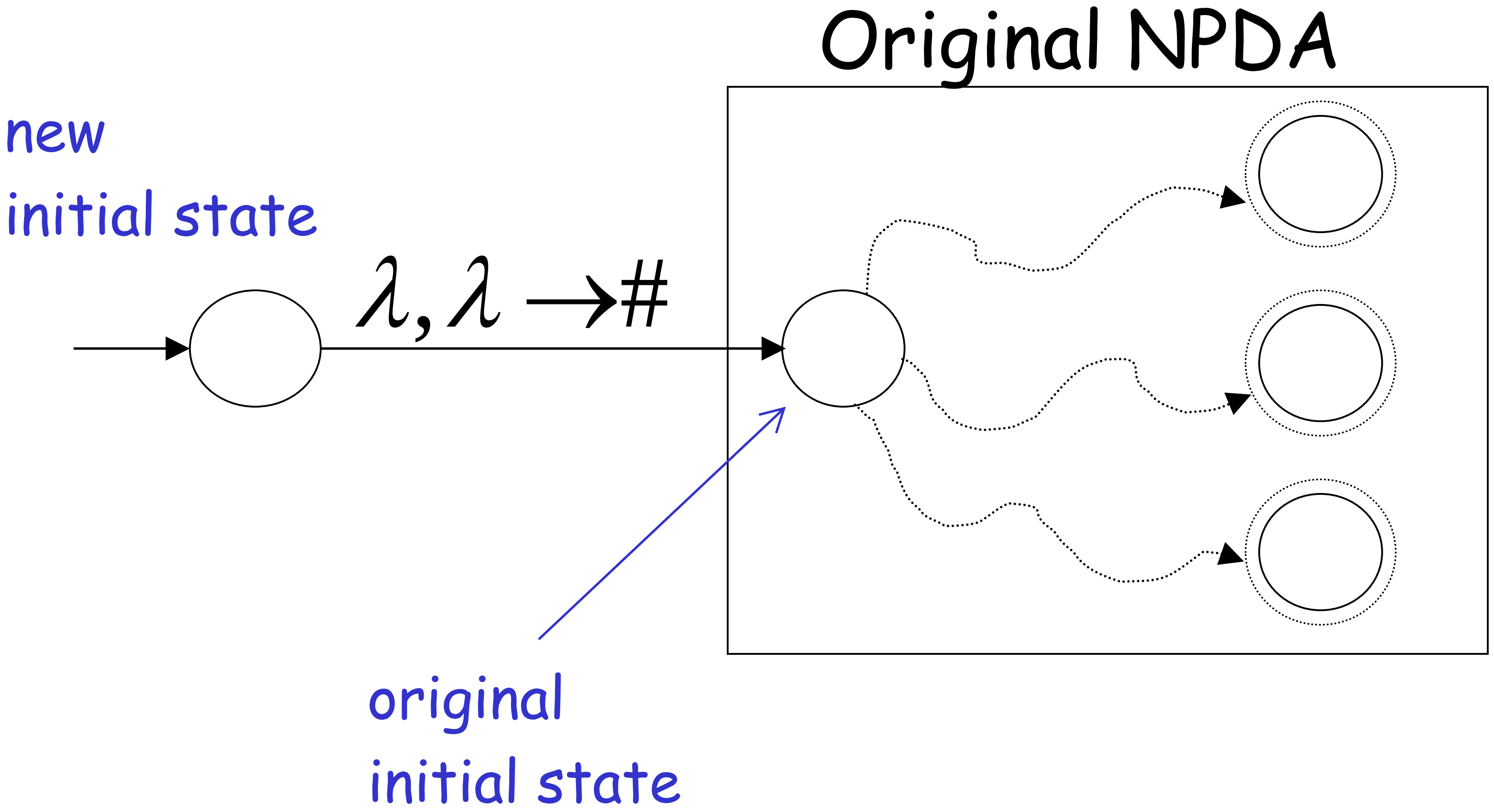


OK OK NOT OK

Introduce the new symbol # to denote the bottom of the stack



At the beginning push $\#$ into the stack



In transitions:

replace every instance of $\$$ with $\#$

Example:

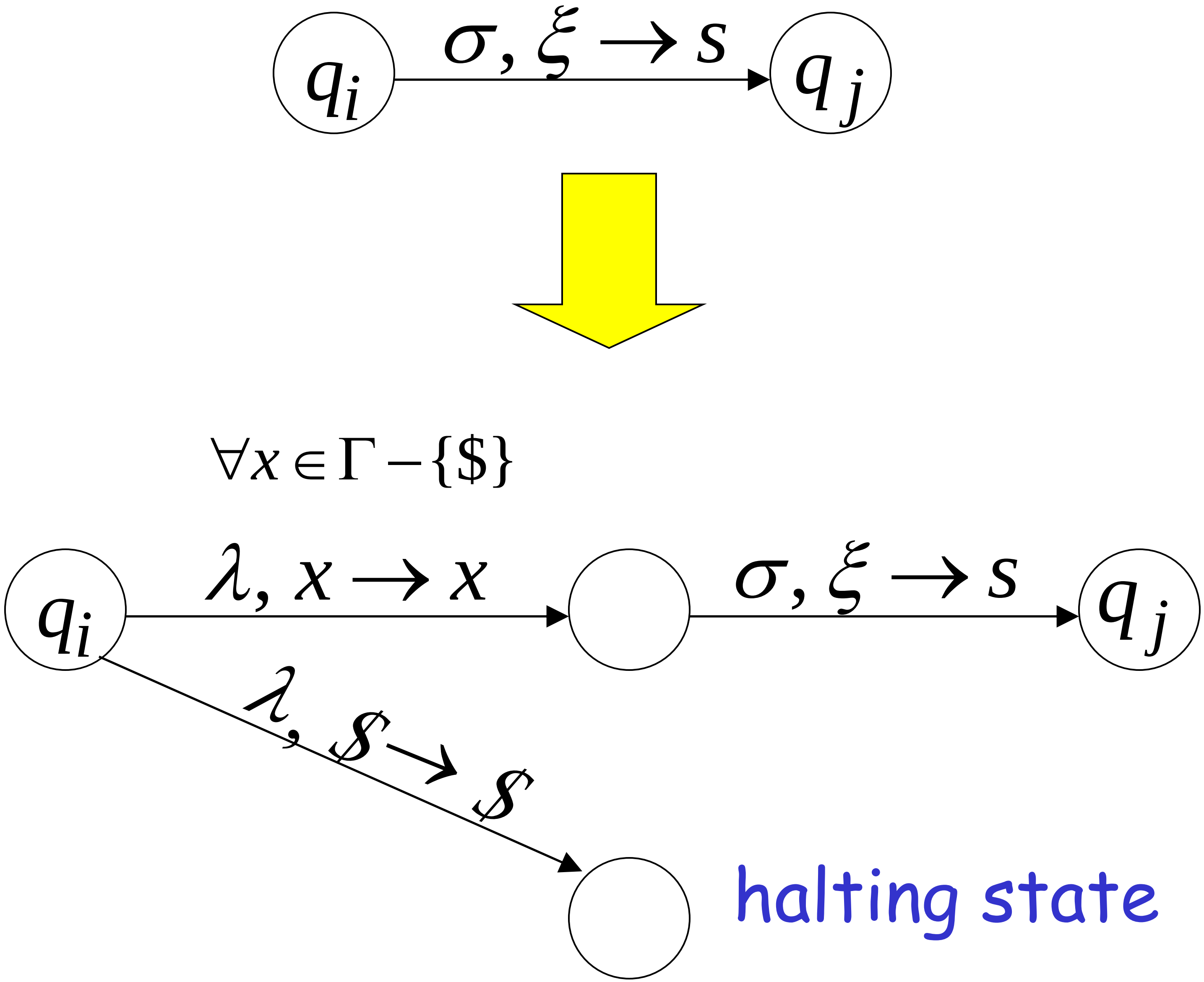
$$q_i \xrightarrow{a,\$\rightarrow b} q_j$$

$$q_i \xrightarrow{a,\#\rightarrow b} q_j$$

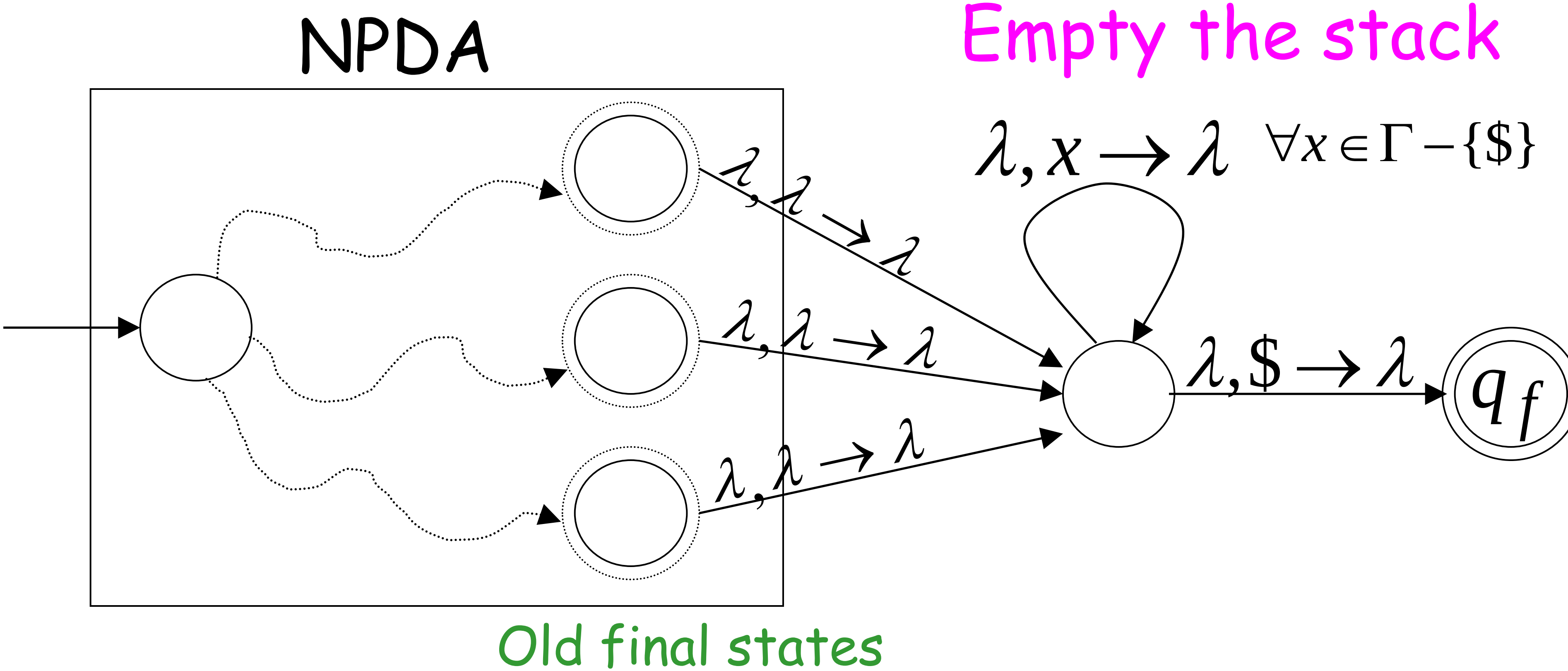
Convert all transitions so that:

if the automaton attempts to pop or replace $\$$ it will halt

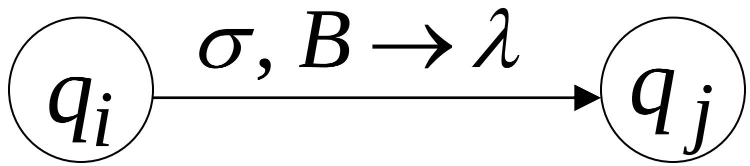
Convert transitions as follows:



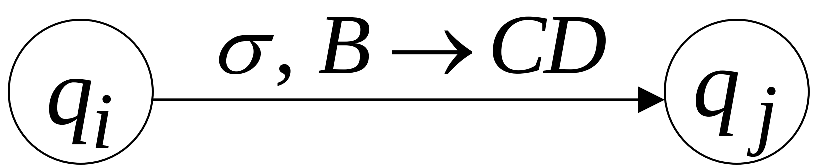
2) Modify the NPDA so that
it empties the stack
and has a unique final state



3) modify the NPDA so that transitions have the following forms:



OR



$B, C, D:$ stack symbols

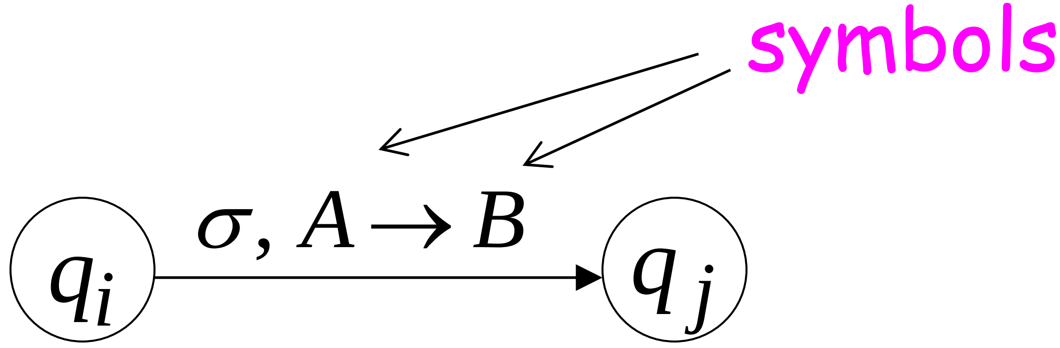
Convert:

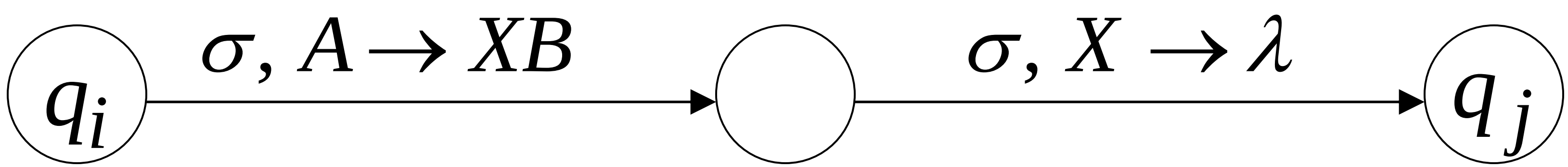
$$q_i \xrightarrow{\sigma,\, \lambda \rightarrow y} q_j$$

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

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

Convert:

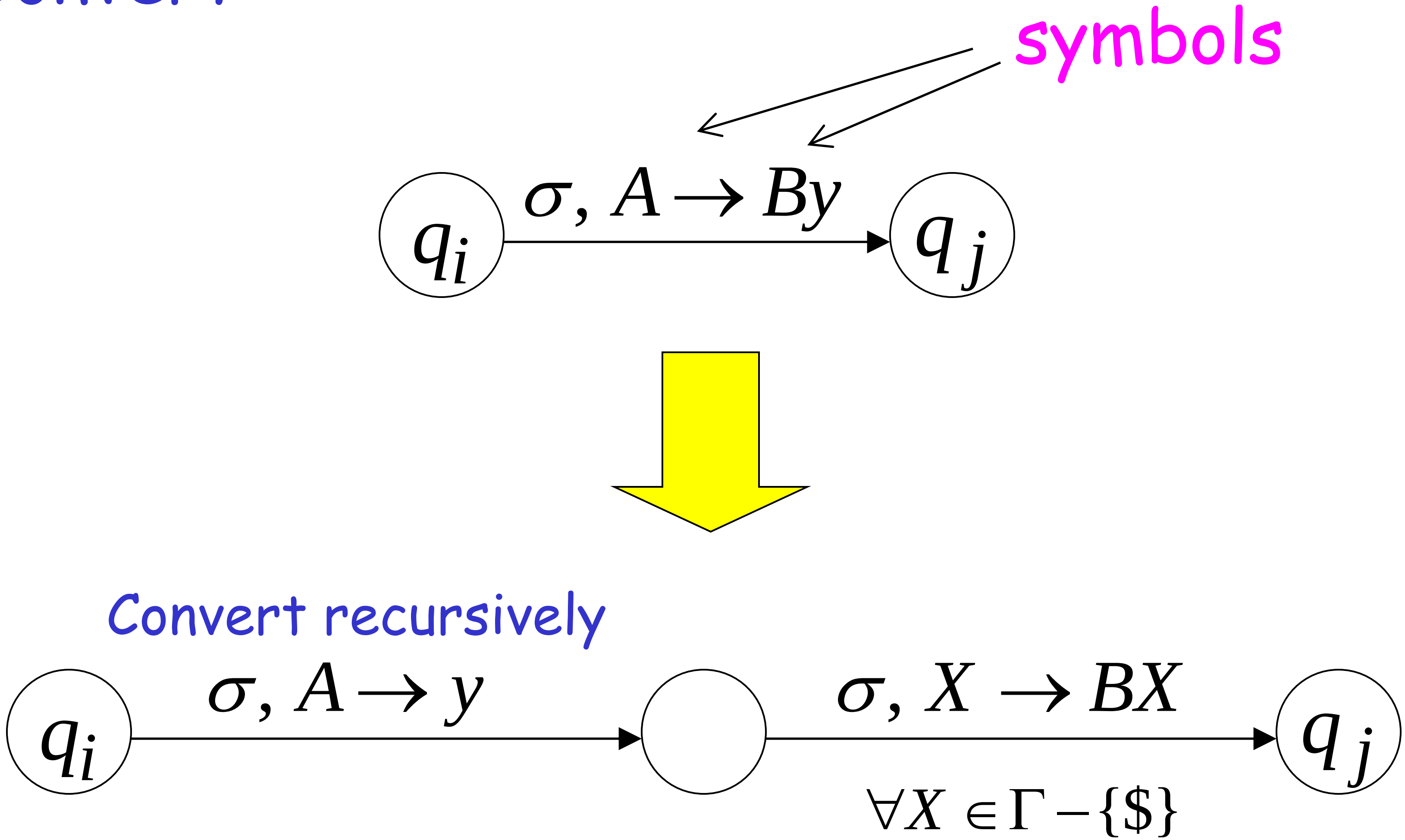




$$X \in \Gamma - \{\$\}$$

Convert:

$$|y| \geq 2$$

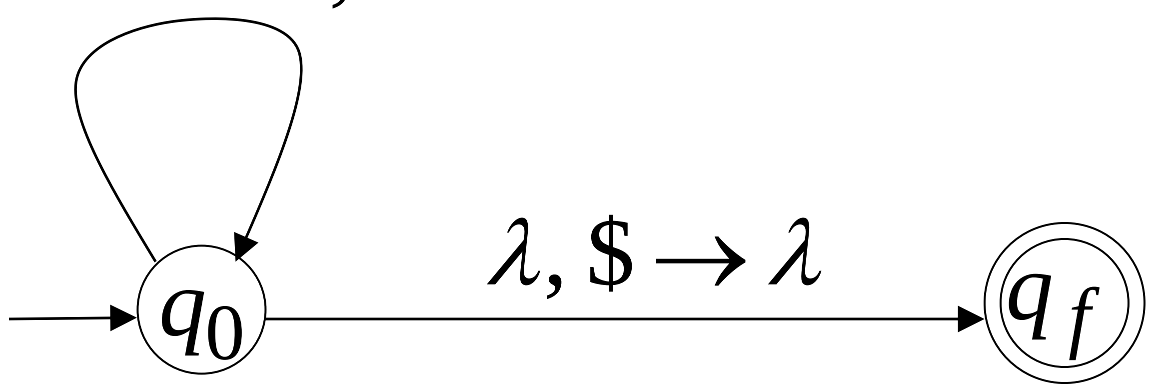


Example of a NPDA in correct form:

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

$\$$: initial stack symbol

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



The Grammar Construction

In grammar G:

Stack symbol

Variables: $(q_i B q_j)$

states

Terminals:

Input symbols of NPDA

For each transition $$q_i \xrightarrow{a,\ B \to \lambda} q_j$$

We add production $$(q_i B q_j) \to a$$

For each transition

$$q_i \xrightarrow{a,\ B \to CD} q_j$$

We add productions

$$(q_i B q_k) \to a(q_j C q_l)(q_l D q_k)$$

For all possible states q_k, q_l
in the automaton

Stack bottom symbol

Start Variable: $(q_o\$q_f)$

Start state

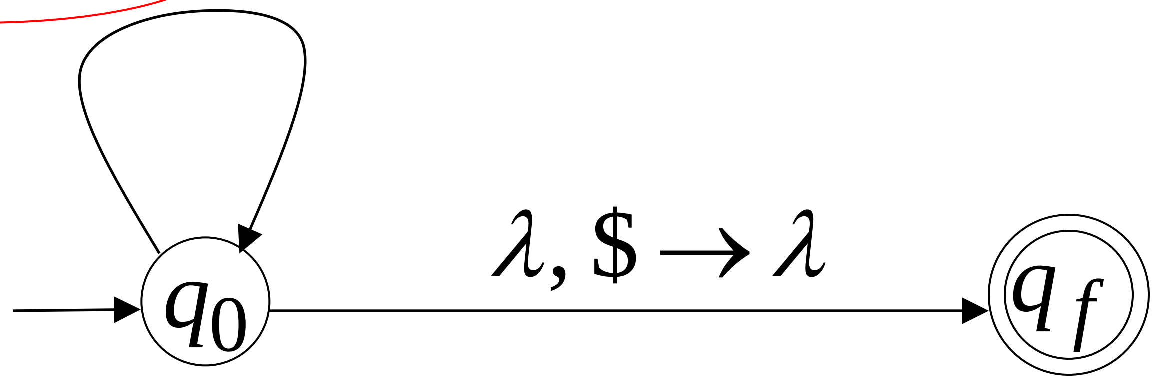
final state

Example:

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

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

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



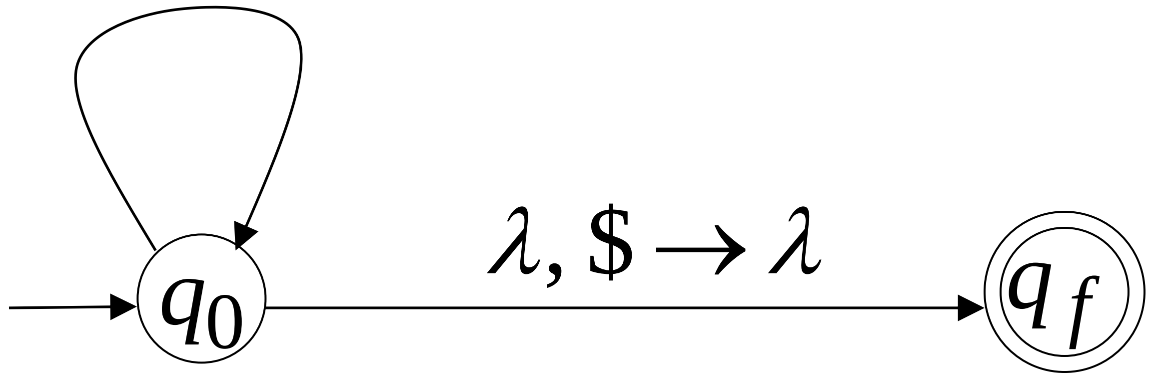
Grammar production: $(q_0 1 q_0) \to a$

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:

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

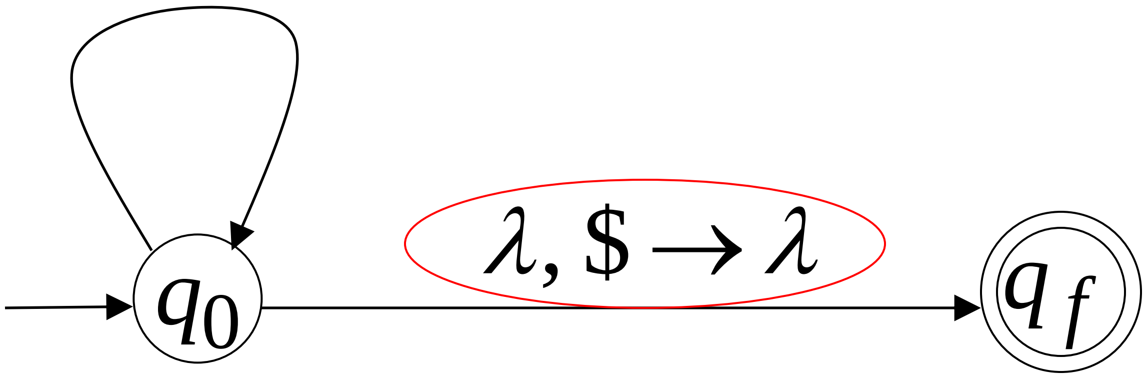
$$(q_0 \$ q_f) \rightarrow b(q_0 1 q_0)(q_0 \$ q_f) \mid b(q_0 1 q_f)(q_f \$ q_f)$$

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

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

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

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

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

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

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

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

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

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

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

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

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:

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

if and only if

the NPDA goes from q_i to q_j
by reading string w and
the stack doesn't change below A
and then A is removed from stack

Therefore:

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

if and only if

w is accepted by the NPDA

Therefore:

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

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

Deterministic PDA

DPDA

Deterministic PDA: DPDA

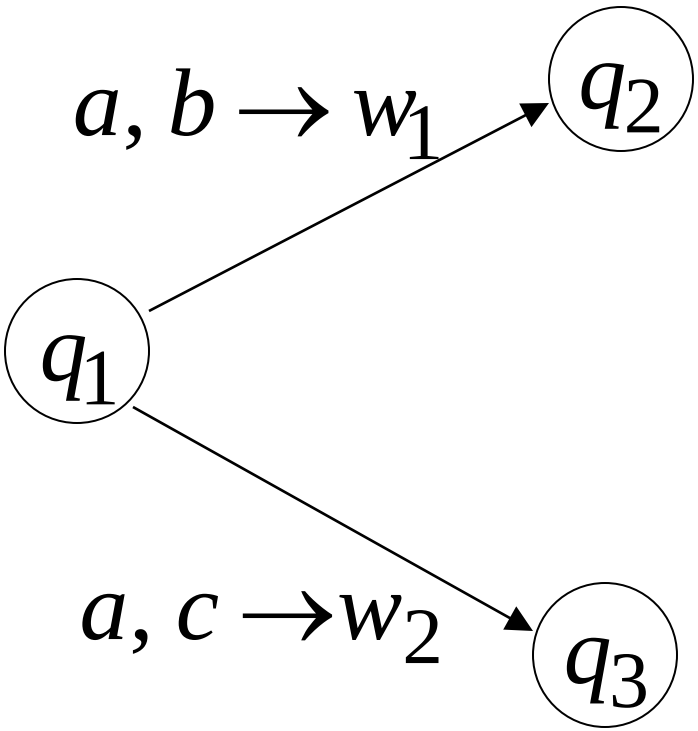
Allowed transitions:

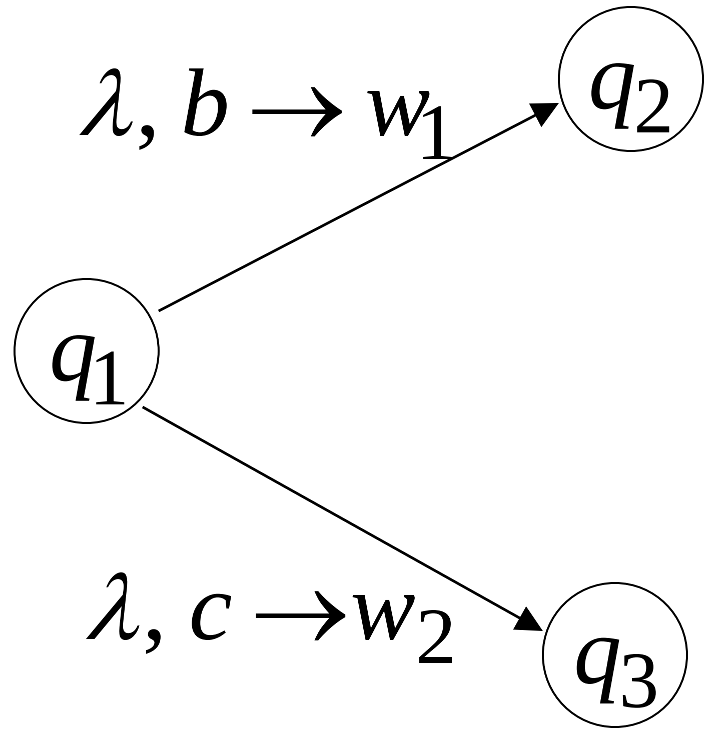
$$q_1 \xrightarrow{a,\, b \rightarrow w} q_2$$

$$q_1 \xrightarrow{\lambda,\, b \rightarrow w} q_2$$

(deterministic choices)

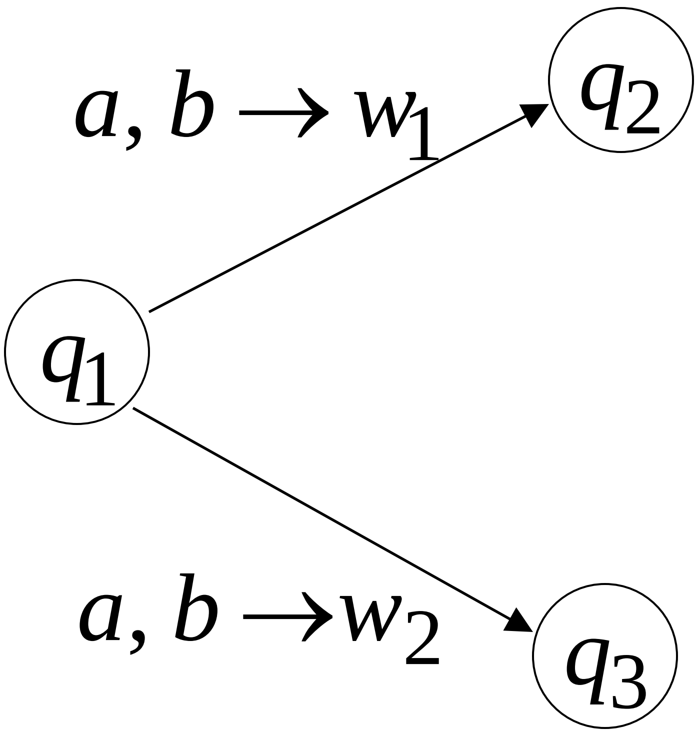
Allowed transitions:

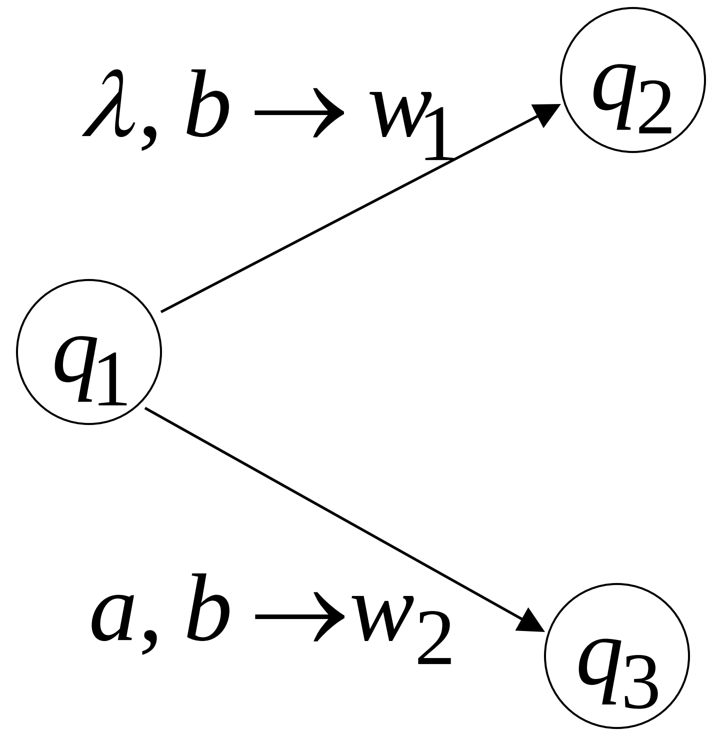




(deterministic choices)

Not allowed:

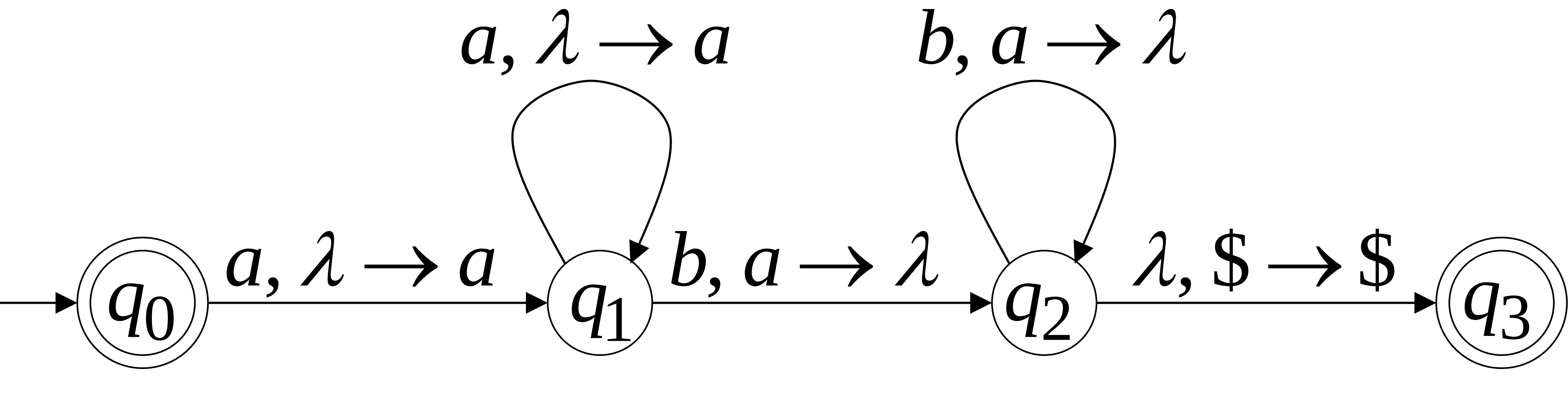




(non deterministic choices)

DPDA example

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



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

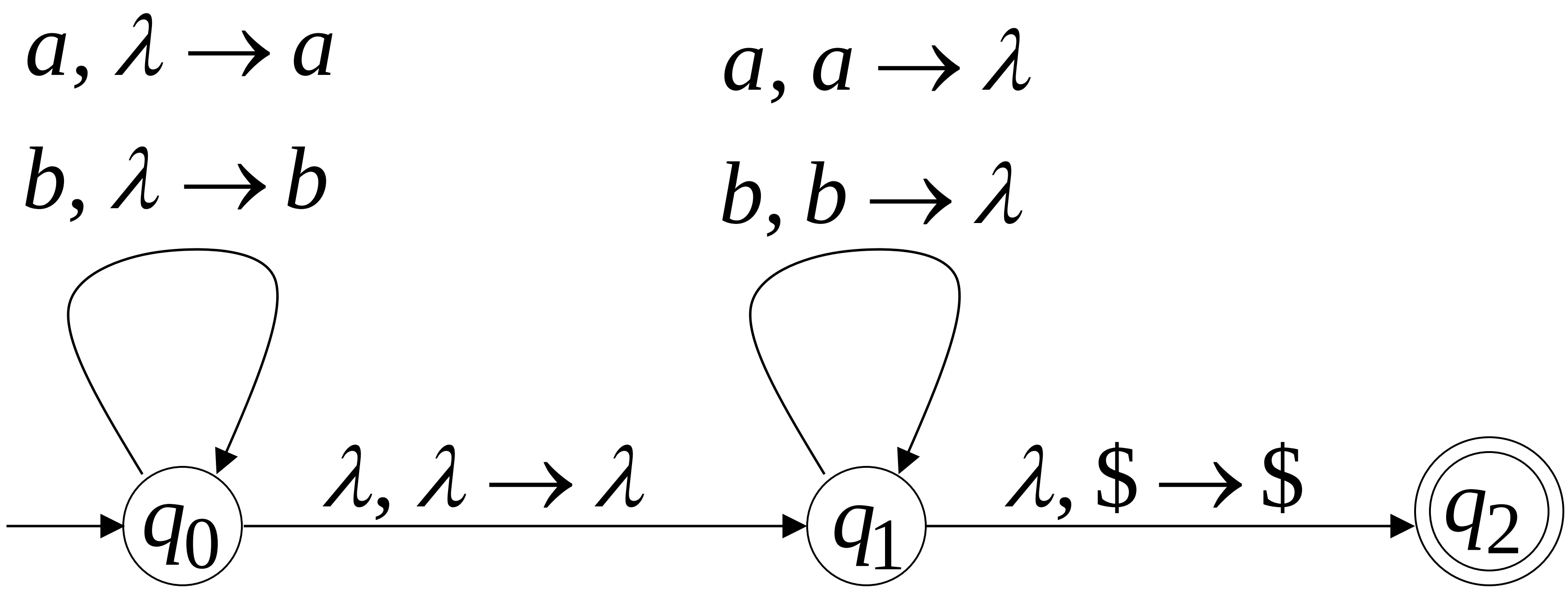
is **deterministic context-free**

Definition:

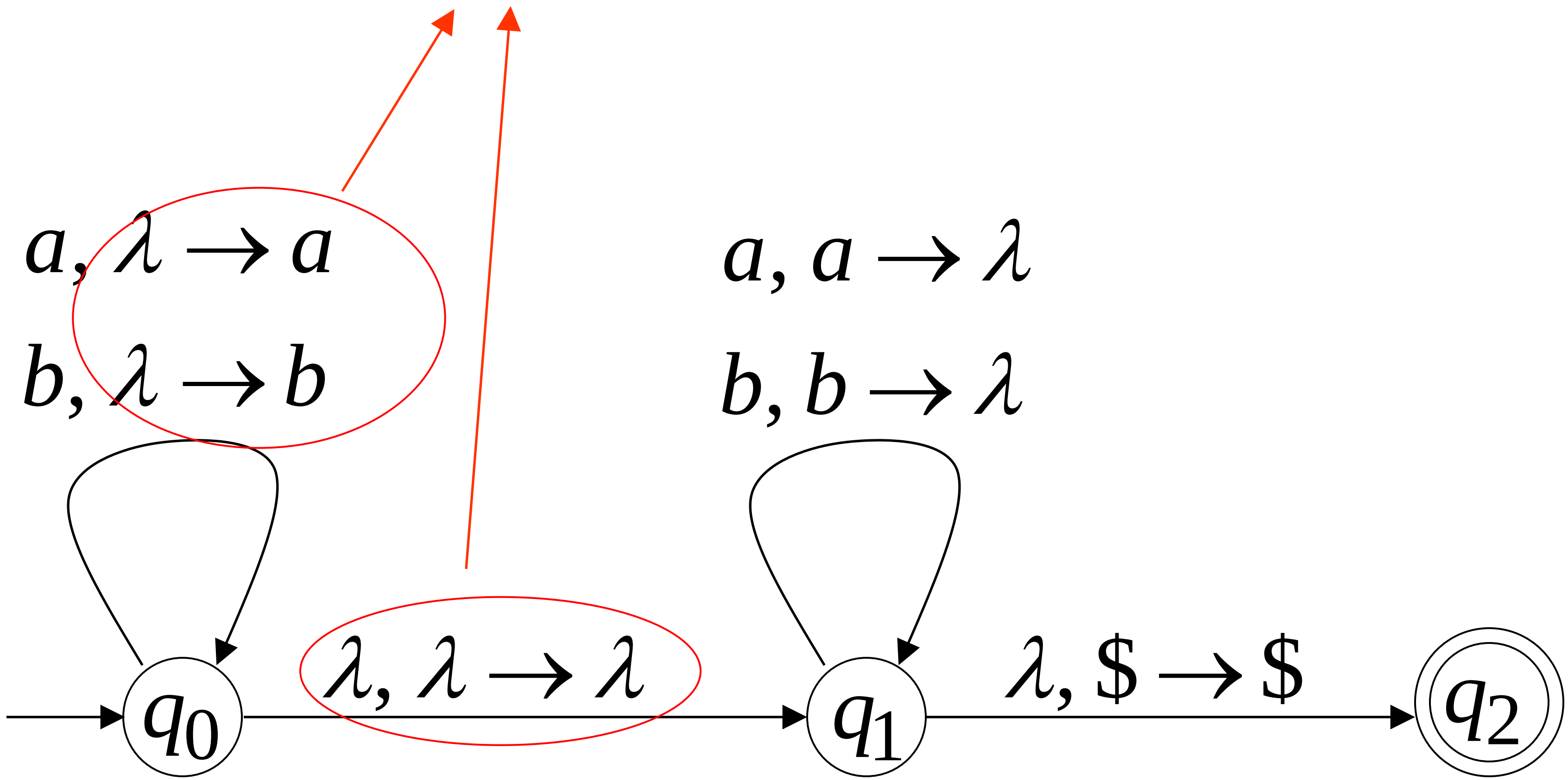
A language L is **deterministic context-free** if there exists some DPDA that accepts it

Example of Non-DPDA (NPDA)

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



Not allowed in DPDAs



NPDAs

Have More Power than

DPDAs

It holds that:

$$\left\{ \begin{array}{c} \text{Deterministic} \\ \text{Context-Free} \\ \text{Languages} \\ \text{(DPDA)} \end{array} \right\} \subseteq \left\{ \begin{array}{c} \text{Context-Free} \\ \text{Languages} \\ \text{NPDAs} \end{array} \right\}$$

Since every DPDA is also a NPDA

We will actually show:

$$L \notin \left\{ \begin{array}{c} \text{Deterministic} \\ \text{Context-Free} \\ \text{Languages} \\ \text{(DPDA)} \end{array} \right\} \subset \left\{ \begin{array}{c} \text{Context-Free} \\ \text{Languages} \\ \text{(NPDA)} \end{array} \right\} \ni L$$

We will show that there exists a context-free language L which is not accepted by any DPDA

The language is:

$$L = \{a^n b^n\} \cup \{a^n b^{2n}\} \qquad n \geq 0$$

We will show:

- L is context-free
- L is **not** deterministic context-free

$$L = \{a^n b^n\} \cup \{a^n b^{2n}\}$$

Language L is context-free

Context-free grammar for L :

$$S \rightarrow S_1 \mid S_2 \qquad \{a^n b^n\} \cup \{a^n b^{2n}\}$$

$$S_1 \rightarrow a S_1 b \mid \lambda \qquad \{a^n b^n\}$$

$$S_2 \rightarrow a S_2 bb \mid \lambda \qquad \{a^n b^{2n}\}$$

Theorem:

The language $L = \{a^n b^n\} \cup \{a^n b^{2n}\}$

is **not** deterministic context-free

(there is **no** DPDA that accepts L)

Proof: Assume for contradiction that

$$L = \{a^n b^n\} \cup \{a^n b^{2n}\}$$

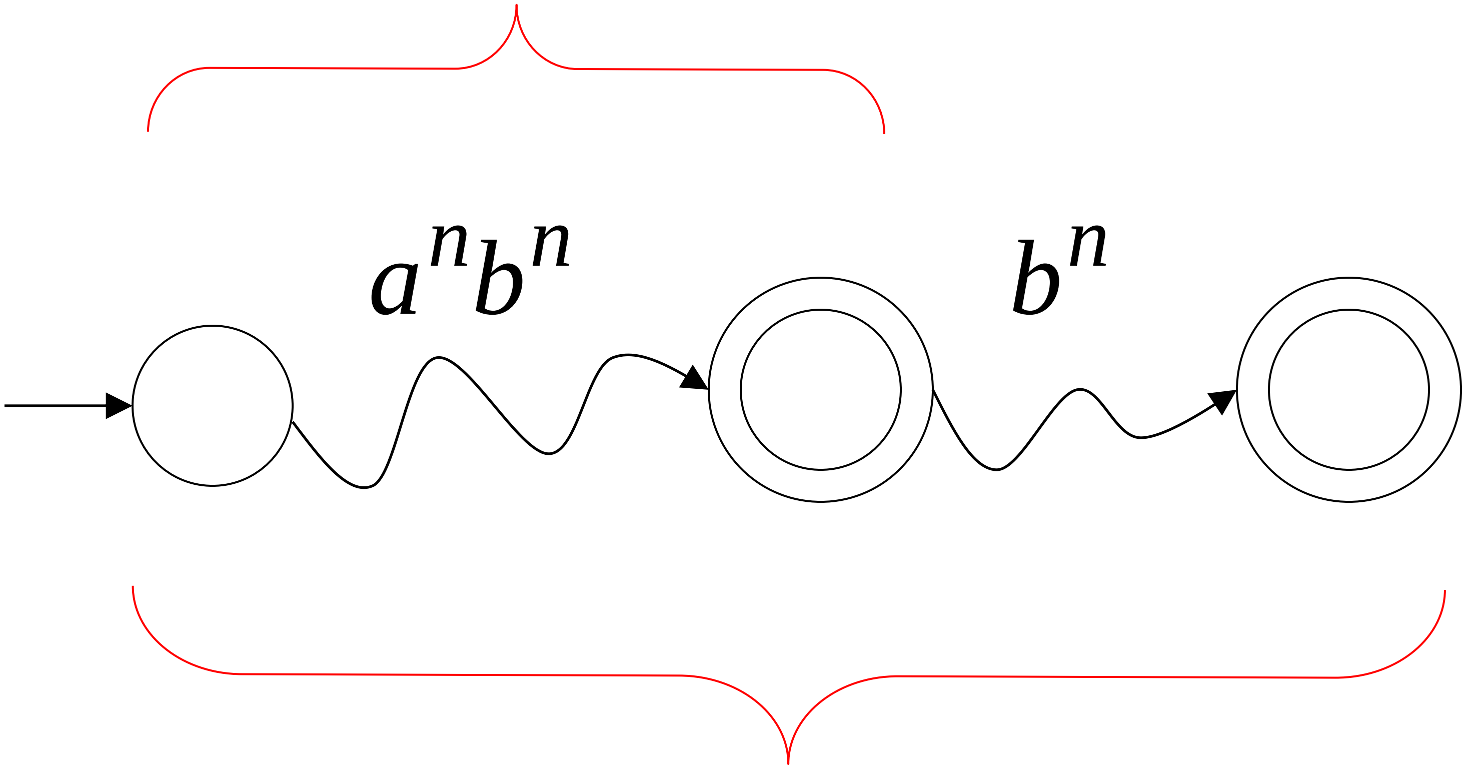
is deterministic context free

Therefore:

there is a DPDA M that accepts L

DPDA M with $L(M)=\{a^n b^n\} \cup \{a^n b^{2n}\}$

accepts $a^n b^n$

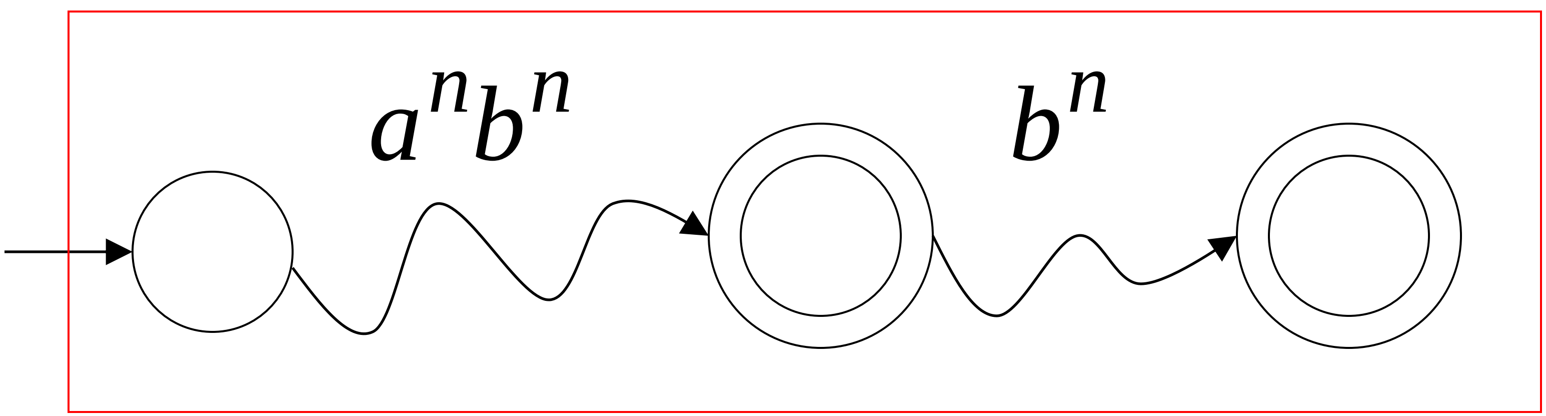


accepts $a^n b^{2n}$

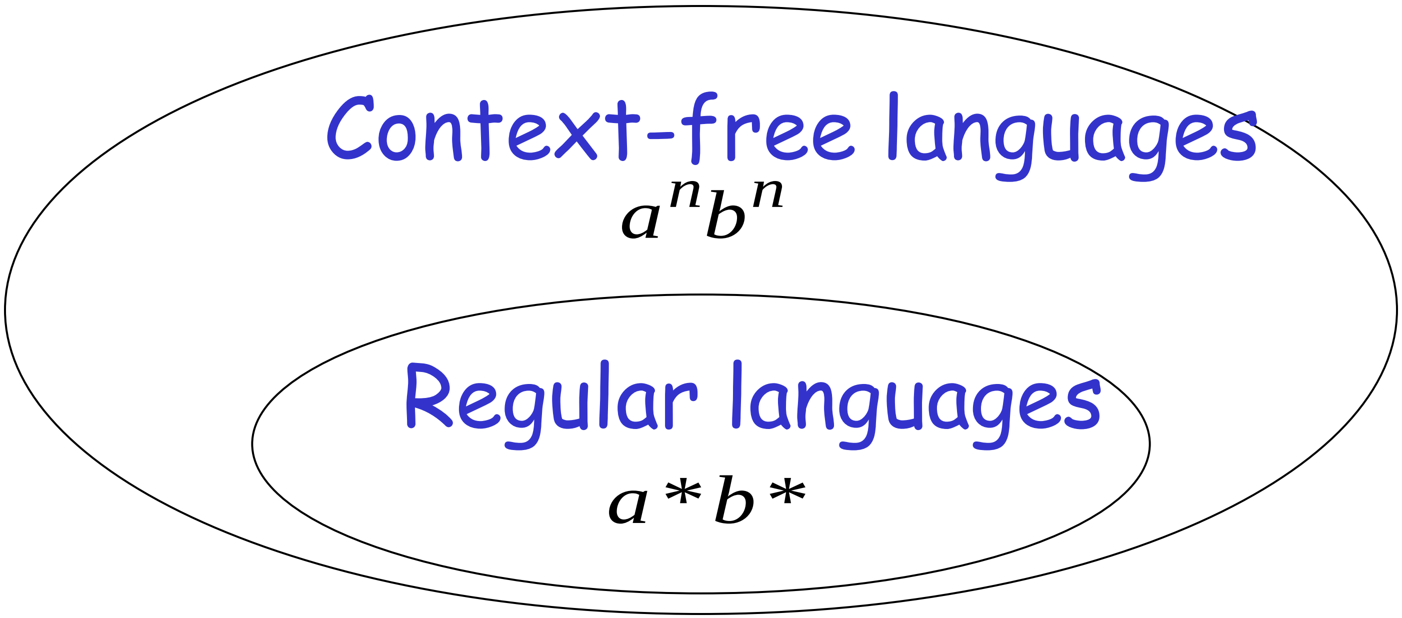
DPDA M with $L(M)=\{a^nb^n\}\cup\{a^nb^{2n}\}$

Such a path exists because of the determinism

M



Fact 1: The language $\{a^n b^n c^n\}$ is **not** context-free



(we will prove this later using pumping lemma for context-free languages)

Fact 2: The language $L \cup \{a^n b^n c^n\}$ is **not** context-free

$$(L = \{a^n b^n\} \cup \{a^n b^{2n}\})$$

(we can prove this using pumping lemma for context-free languages)

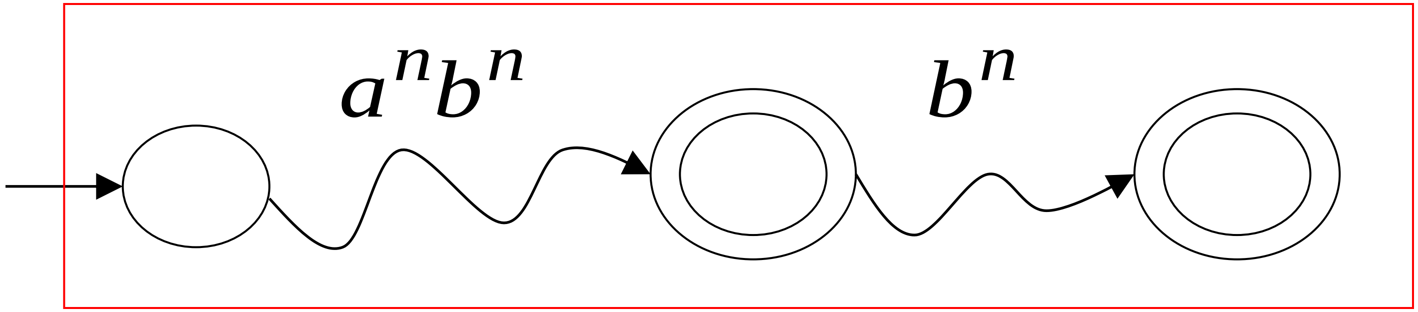
We will construct a NPDA that accepts:

$$L \cup \{a^n b^n c^n\}$$

$$(L = \{a^n b^n\} \cup \{a^n b^{2n}\})$$

which is a contradiction!

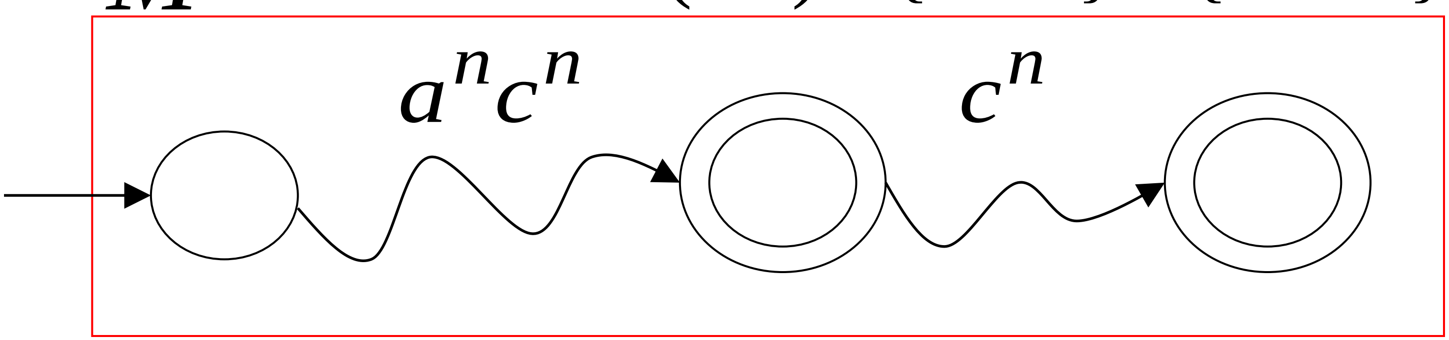
M $\quad L(M) = \{a^n b^n\} \cup \{a^n b^{2n}\}$



Modify M

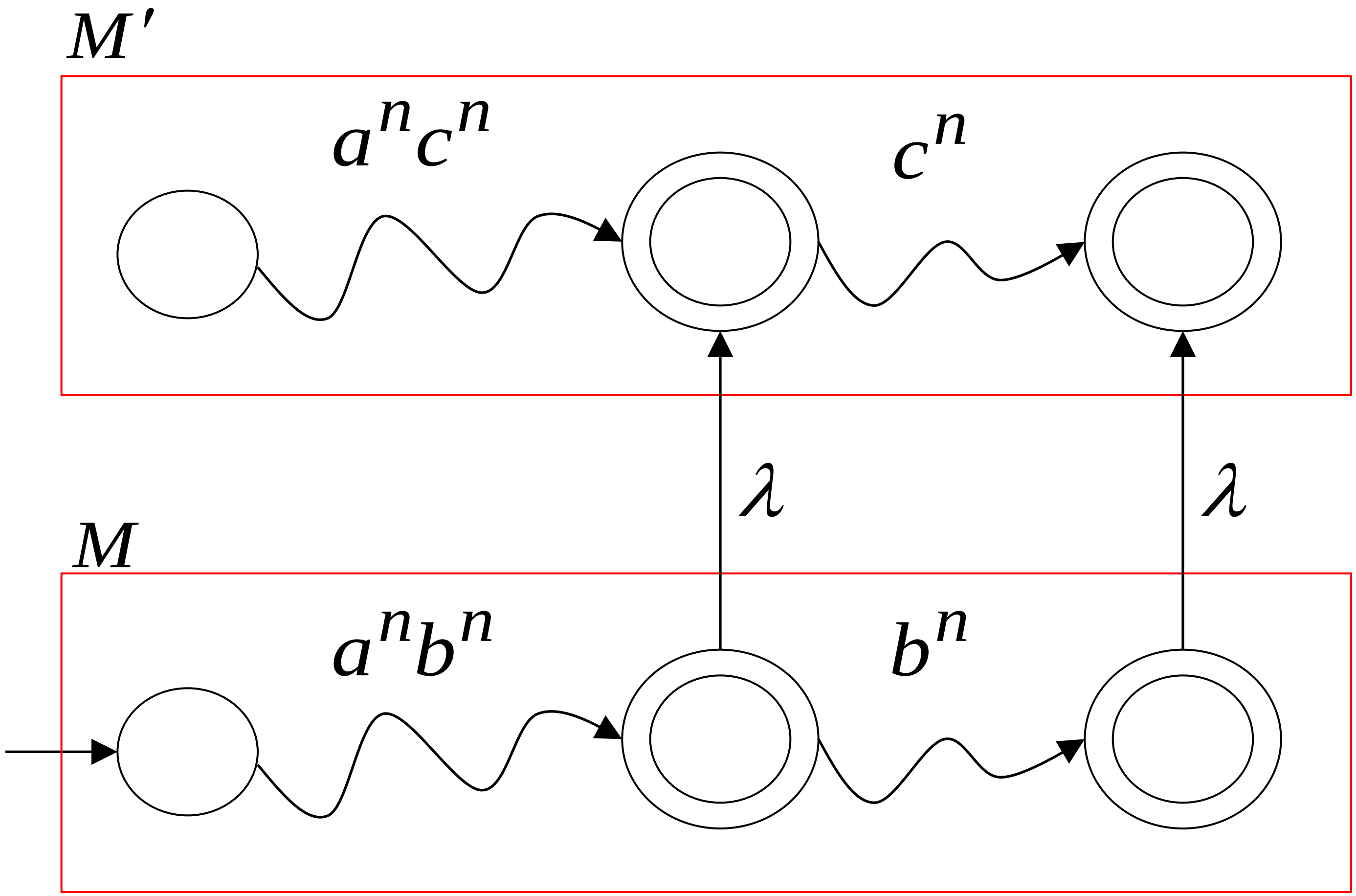
Replace b with c

M' $\quad L(M') = \{a^n c^n\} \cup \{a^n c^{2n}\}$



The NPDA that accepts $L \cup \{a^n b^n c^n\}$

Connect final states of M'
with final states of M



Since $L \cup \{a^n b^n c^n\}$ is accepted by a NPDA

it is context-free

Contradiction!

(since $L \cup \{a^n b^n c^n\}$ is not context-free)

Therefore:

Not deterministic context free

$$L = \{a^n b^n\} \cup \{a^n b^{2n}\}$$

There is **no** DPDA that accepts

End of Proof