2021

Theory of Computation

Kun-Ta Chuang
Department of Computer Science and Information Engineering
National Cheng Kung University

Outline

1. Nondeterministic Pushdown Automata
2. Pushdown Automata and Context-Free Languages
3. Deterministic Pushdown Automata and Deterministic CFLs

Formal Definition

Non-Deterministic Pushdown Automaton (NPDA)

$$M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$$

Final states

States

Input alphabet

Stack alphabet

Transition function

Initial state

Stack start symbol

Transition Function

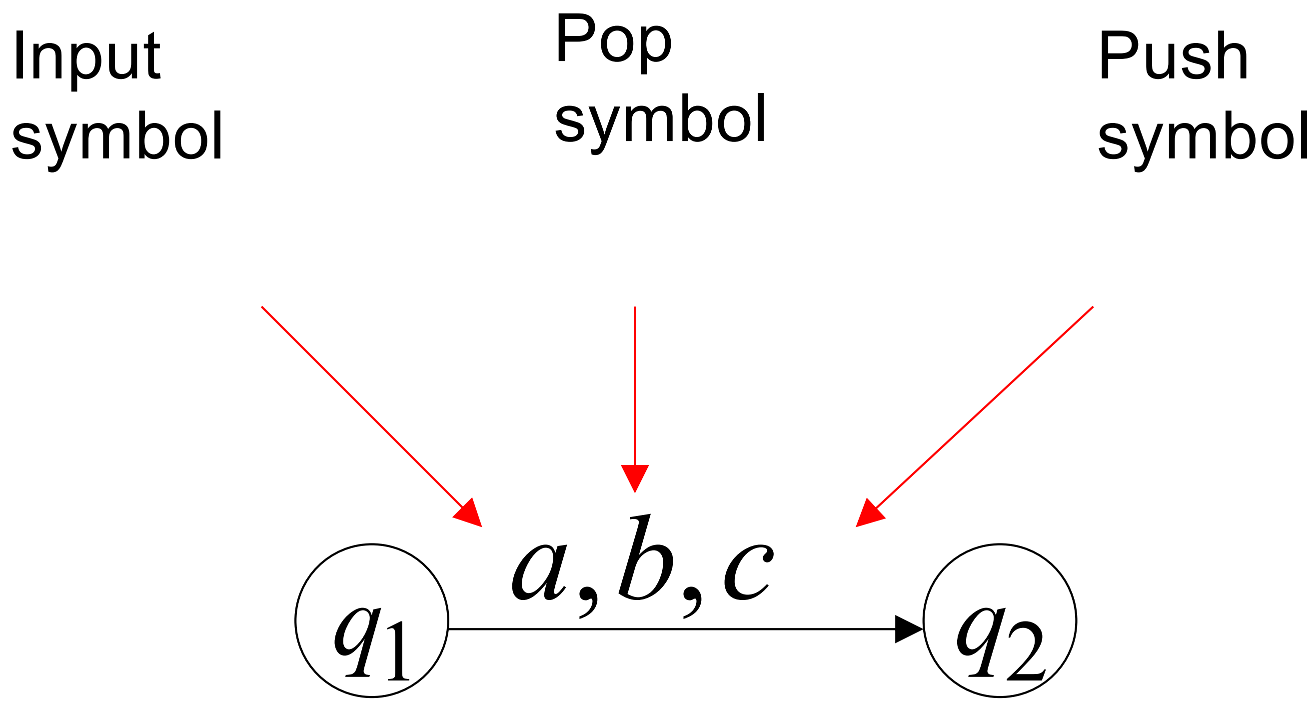
$$\delta(q,\ u,\ s)$$

Current state (Q)

Current input symbol (Σ U {λ})

Current symbol on top of the stack (Γ)

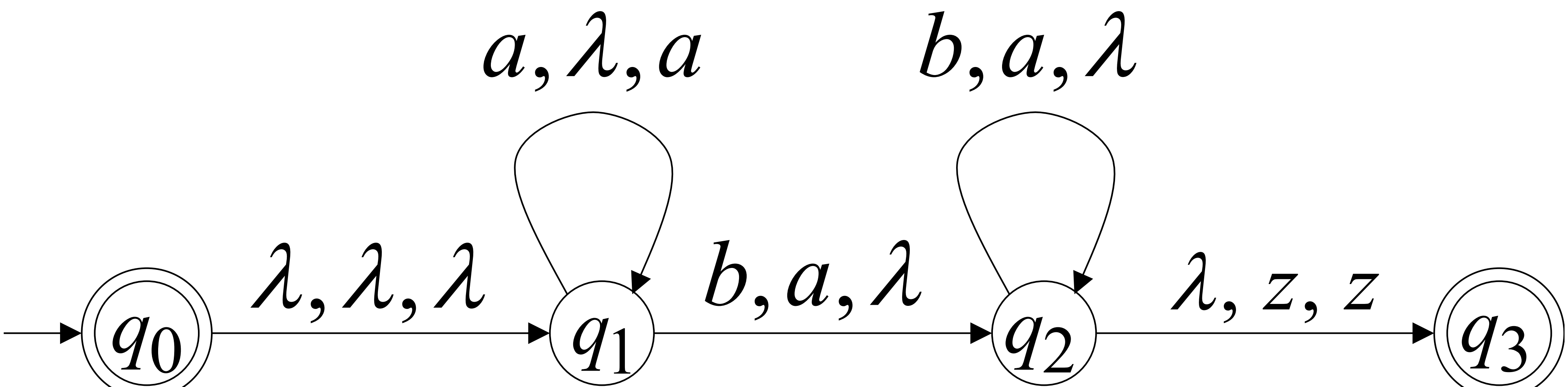
The States



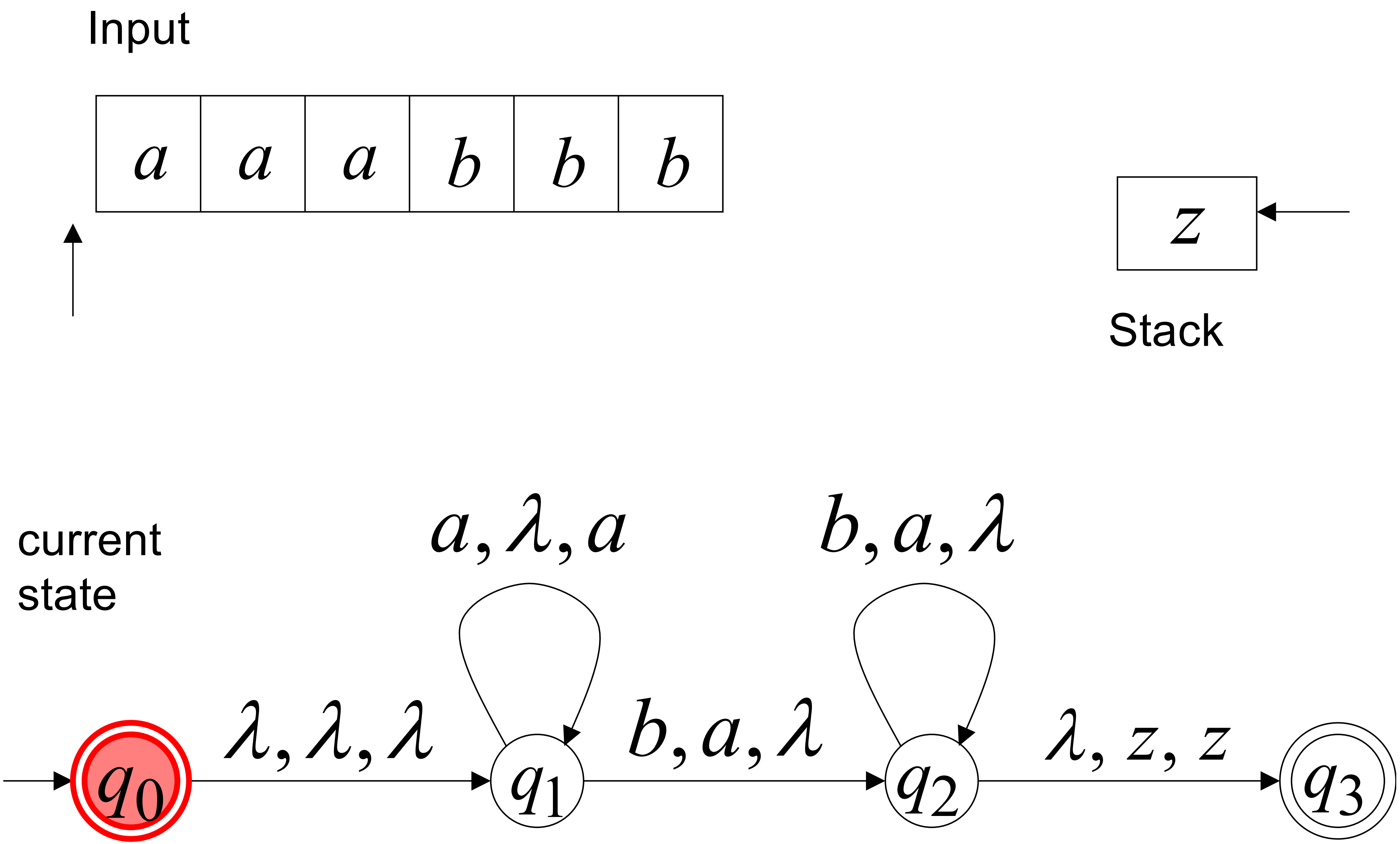
$$\delta(q_1,\ a,\ b) = \{(q_2,\ c)\}$$

NPDA: Non-Deterministic PDA

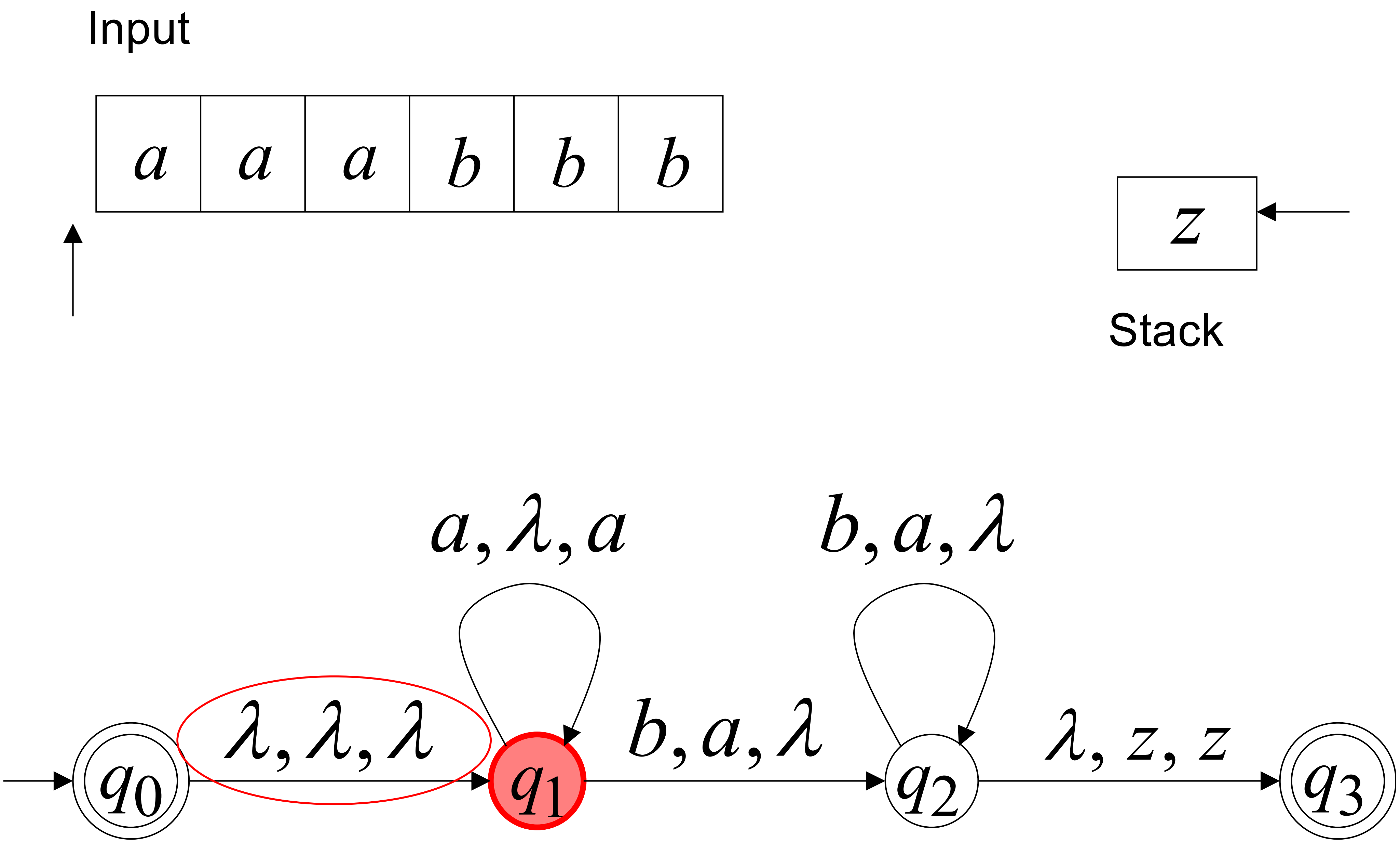
Example: $L = \{a^n b^n : n \geq 0\}$



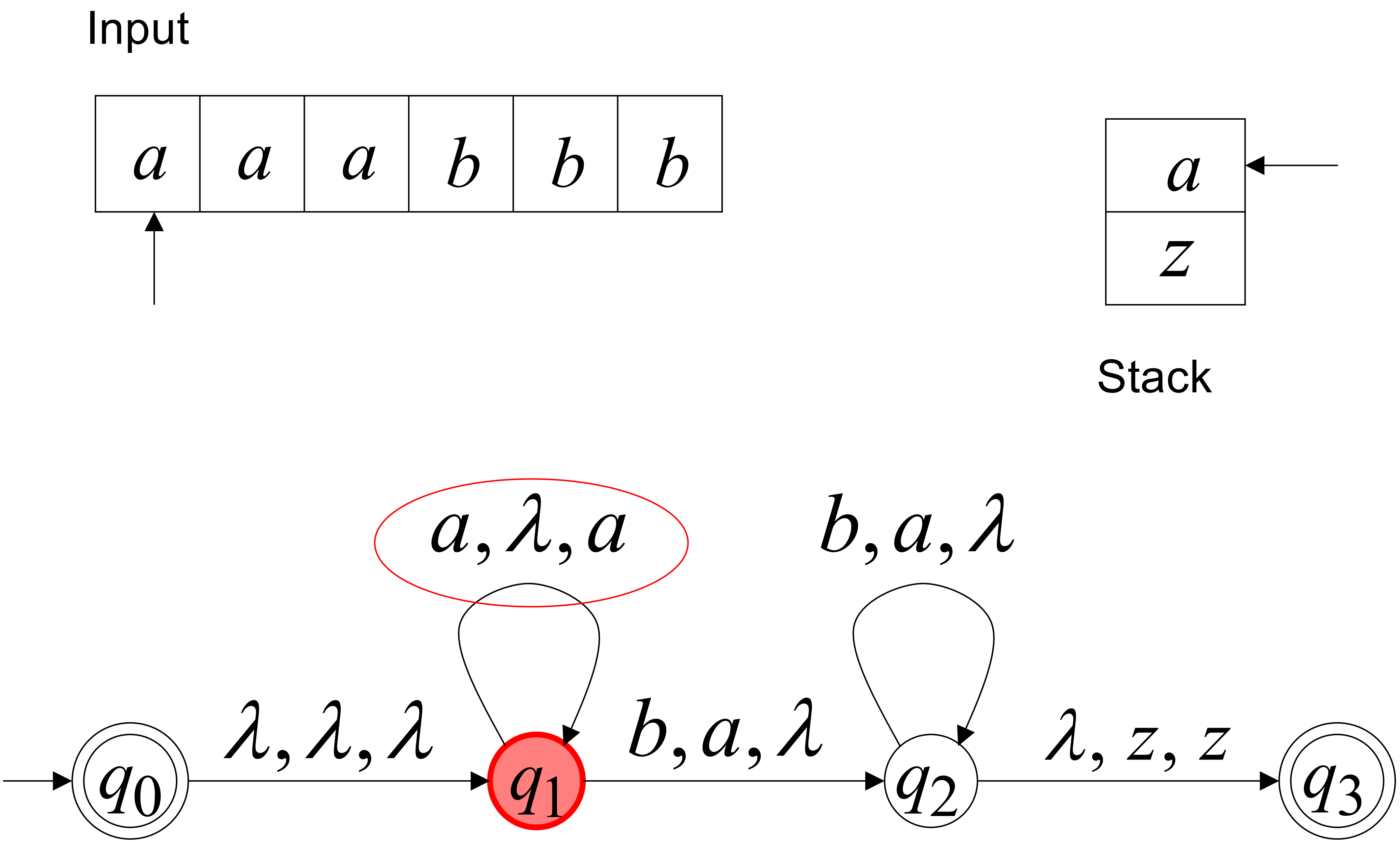
Execution Example: Time 0



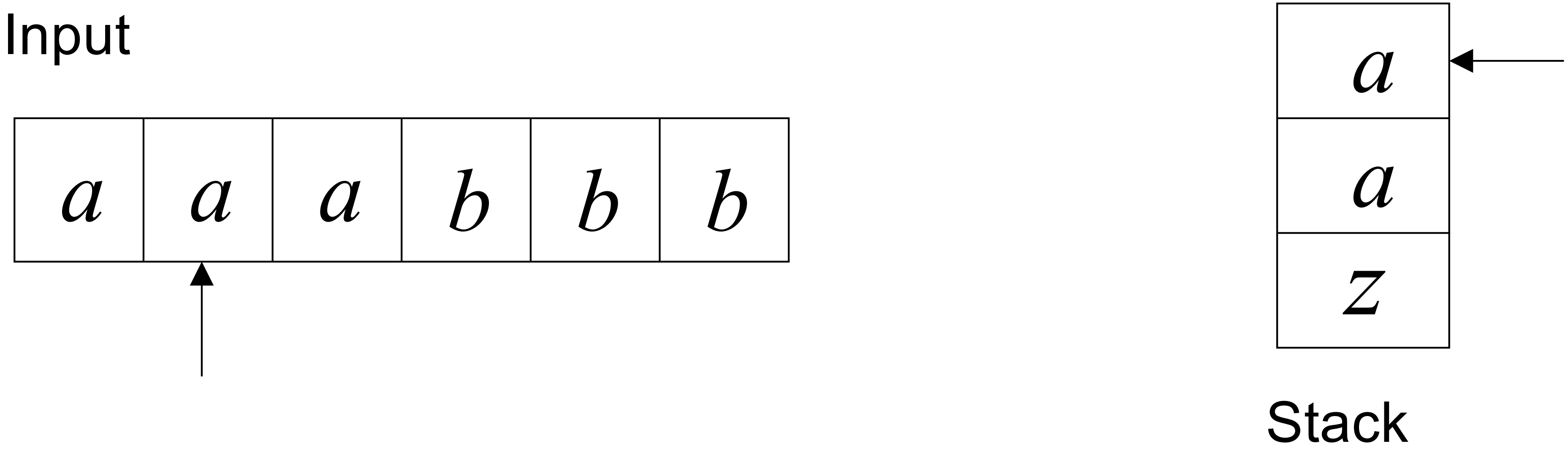
Time 1

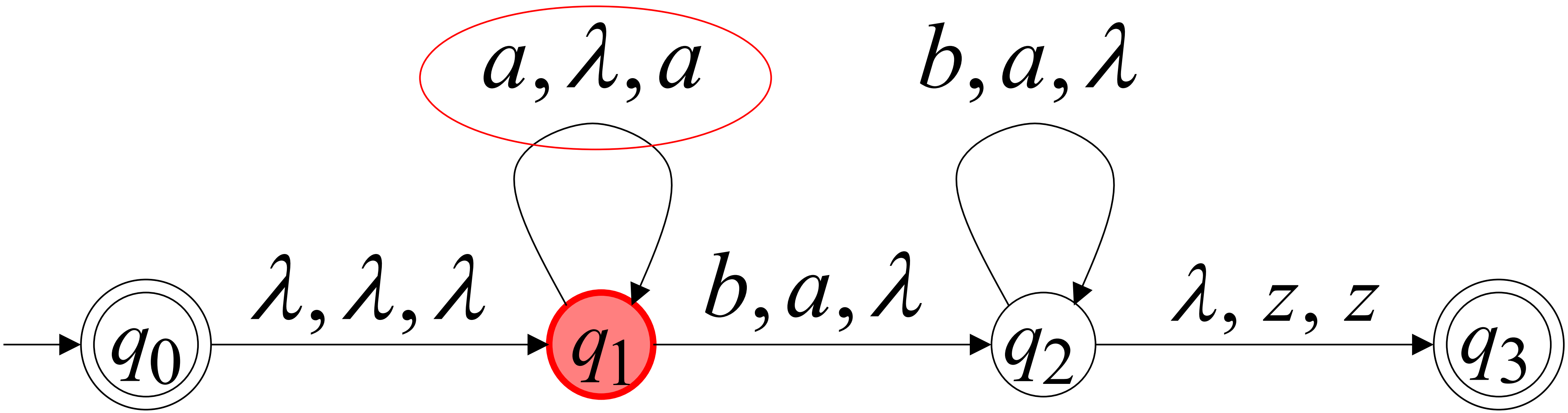

Input
a
a
a
b
b
b
z
Stack
a, λ, a
b, a, λ
$\lambda, \lambda, \lambda$
b, a, λ
λ, z, z
q_0
q_1
q_2
q_3

Time 2

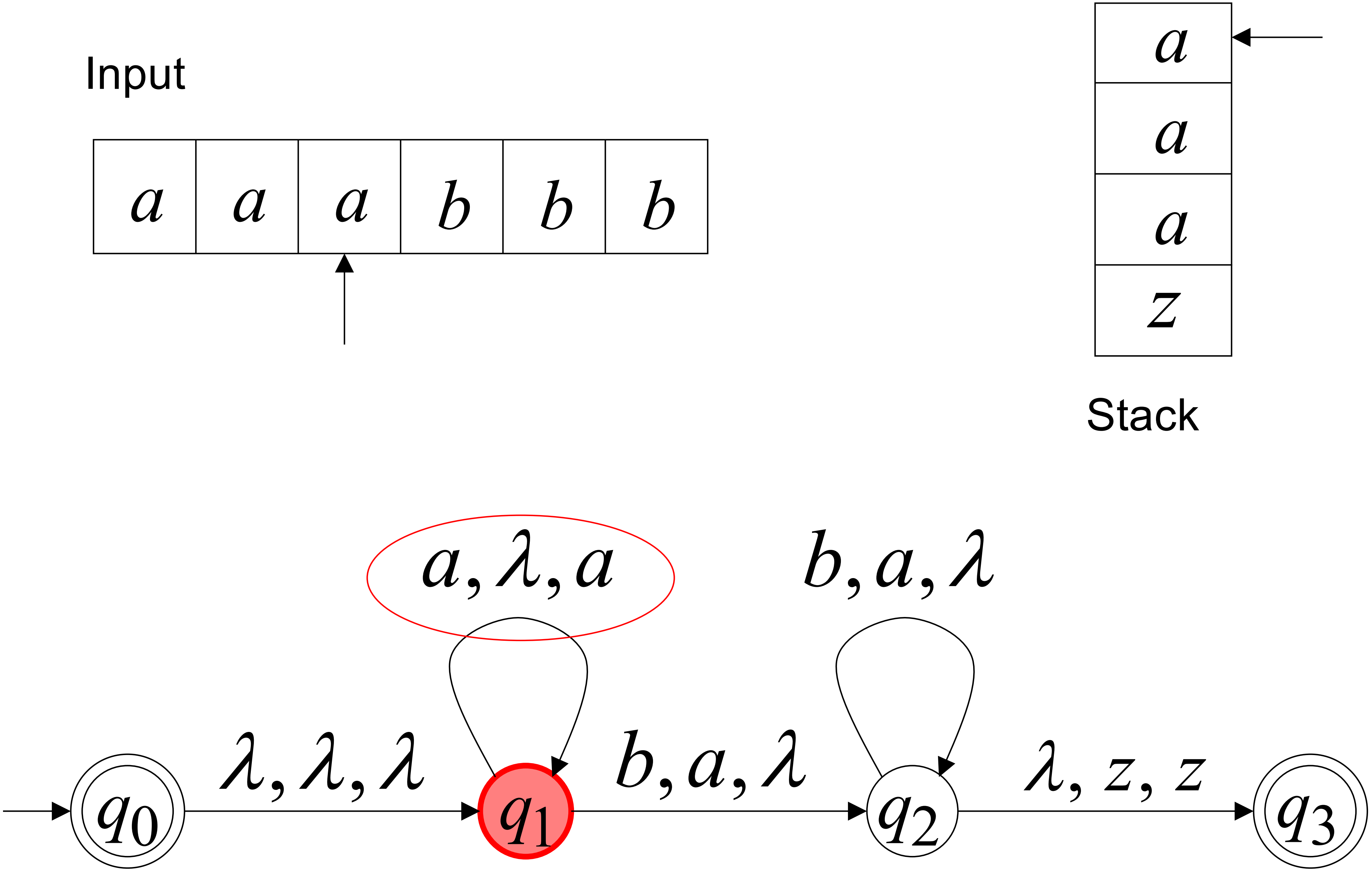


Time 3

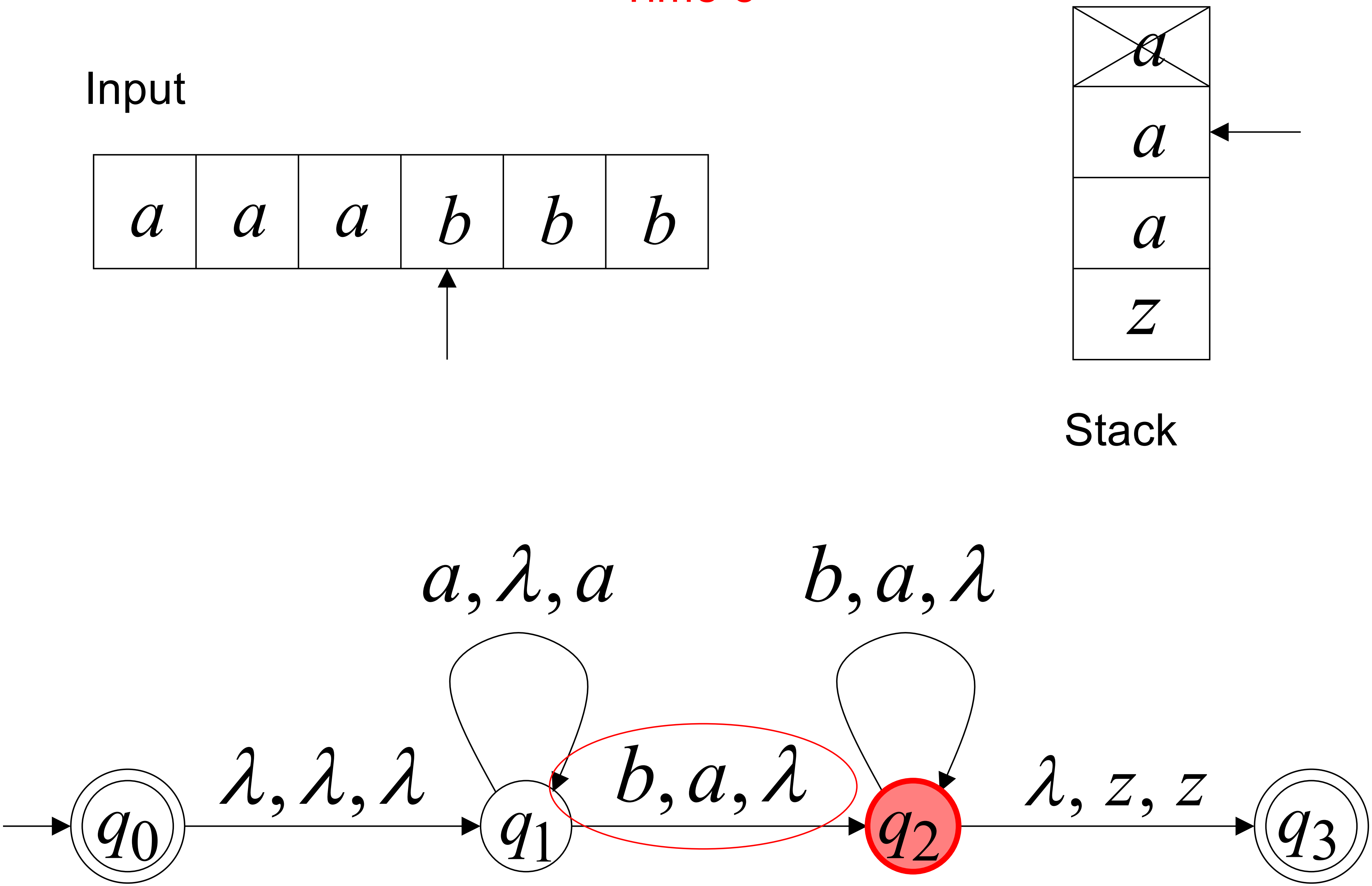

Input
a
a
a
b
b
b
a
a
z
Stack


a, λ, a
b, a, λ
$\lambda, \lambda, \lambda$
b, a, λ
λ, z, z
q_0
q_1
q_2
q_3

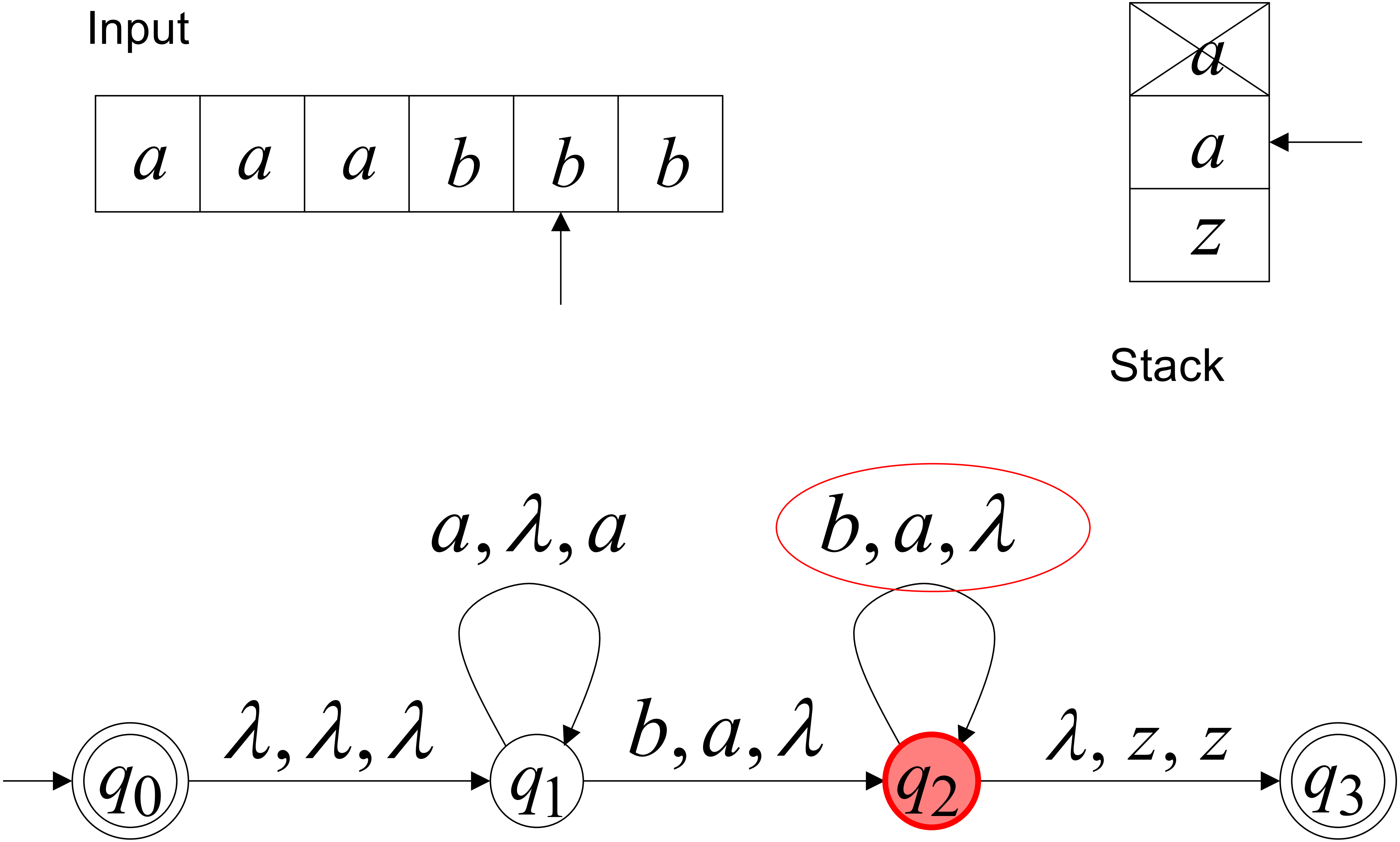
Time 4



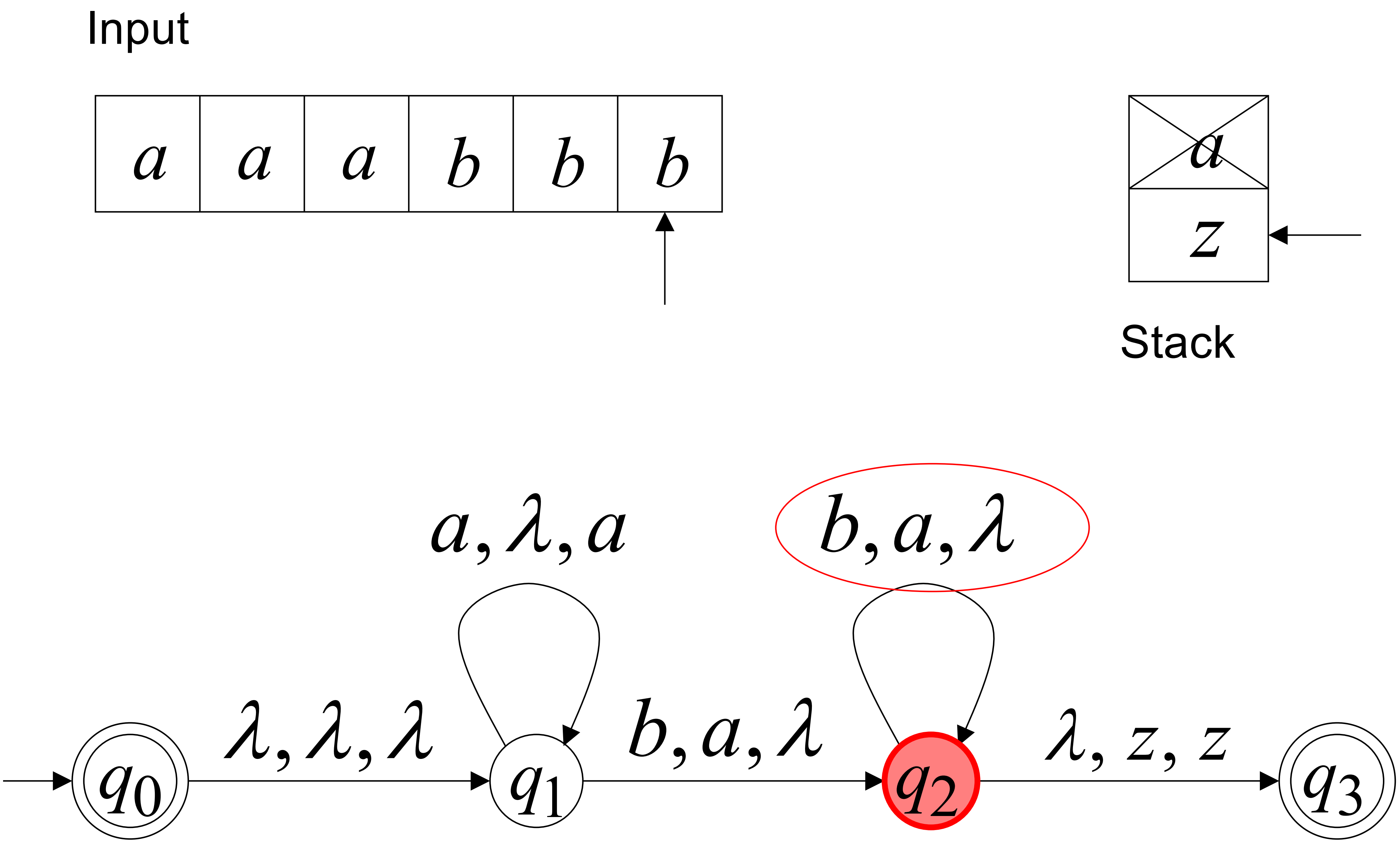
Time 5



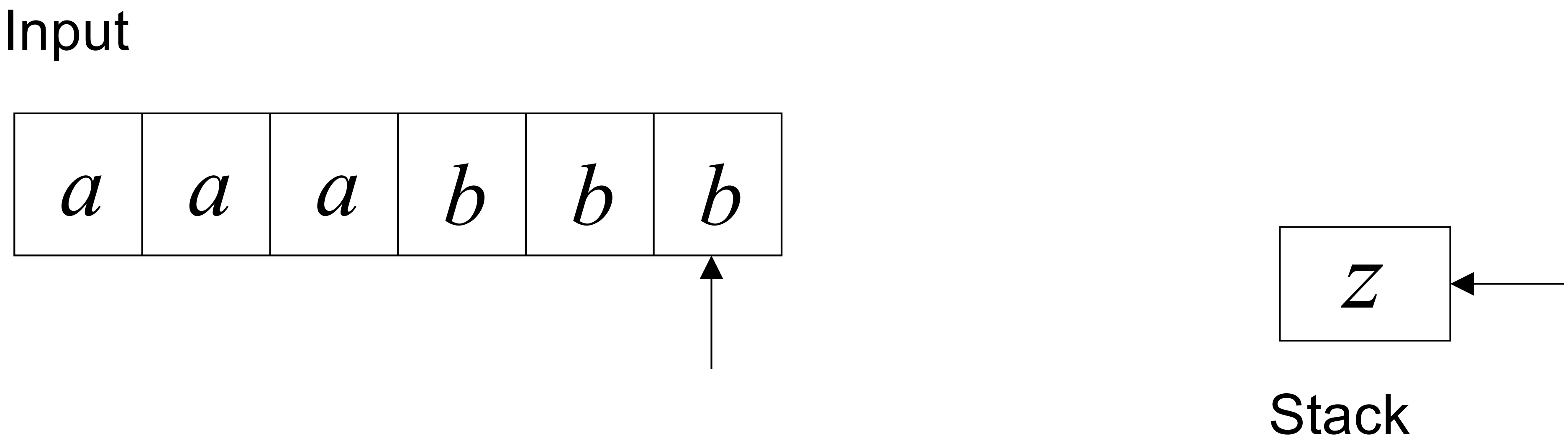
Time 6

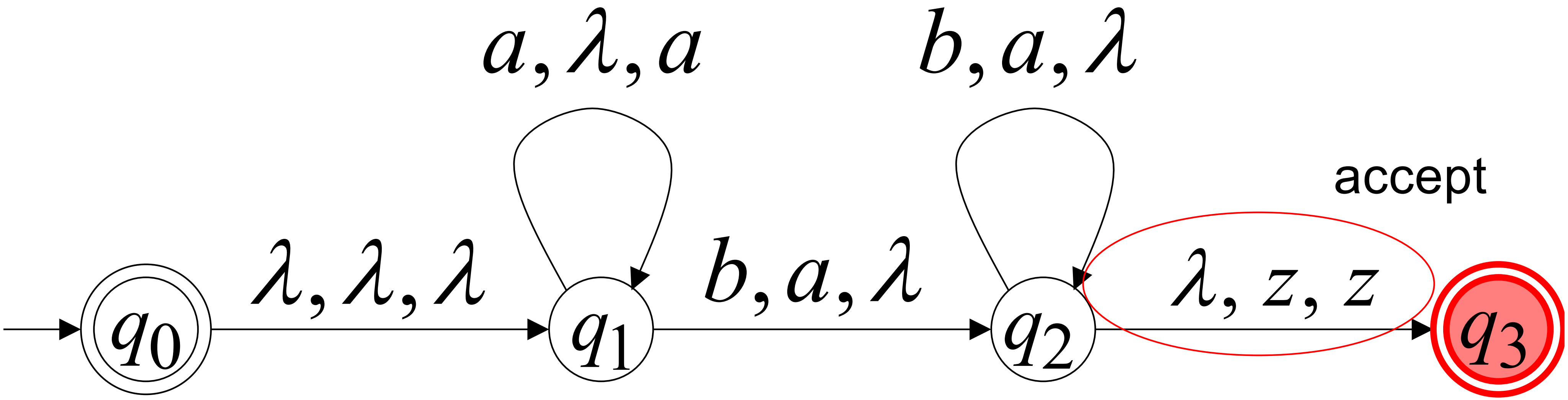


Time 7



Time 8


Input
a
a
a
b
b
b
z
Stack


a, λ, a
b, a, λ
accept
$\lambda, \lambda, \lambda$
b, a, λ
λ, z, z
q_0
q_1
q_2
q_3

Instantaneous Description (ID)

$$(q,\ u,\ s)$$

Current state

Remaining input

Current stack contents

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:

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

Theorem 7.1

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

Proof - Step 2:

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

Theorem 7.2

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

Assume G in Greibach normal form

$$A \to a\, \underbrace{V_1 V_2 \cdots V_k}_{\text{New Variable in stack}}$$

Top of stack

Input

Leftmost derivation

$$G: \quad S \Rightarrow \cdots\cdots \Rightarrow \underbrace{\sigma_1\sigma_2\cdots\sigma_k}_{\text{Input processed}} V_1 \underbrace{V_2\cdots V_m}_{\text{Stack contents}} \Rightarrow \cdots$$

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
Z

$G:$

Leftmost derivation

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

string of terminals

$M:$

Simulation of derivation

Input

σ_1	σ_2	$\cdots\cdots$	σ_n

end of input is reached

Stack

z

Example 7.6

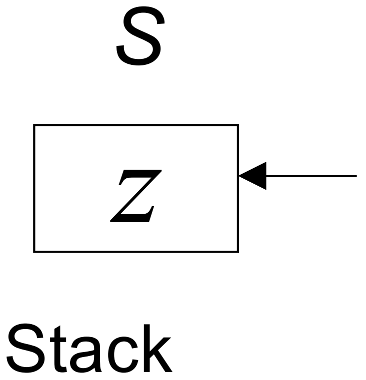
$$S \to aSbb \mid a$$

Greibach Normal Form

$$S \to aSA \mid a$$

$$A \to bB$$

$$B \to b$$

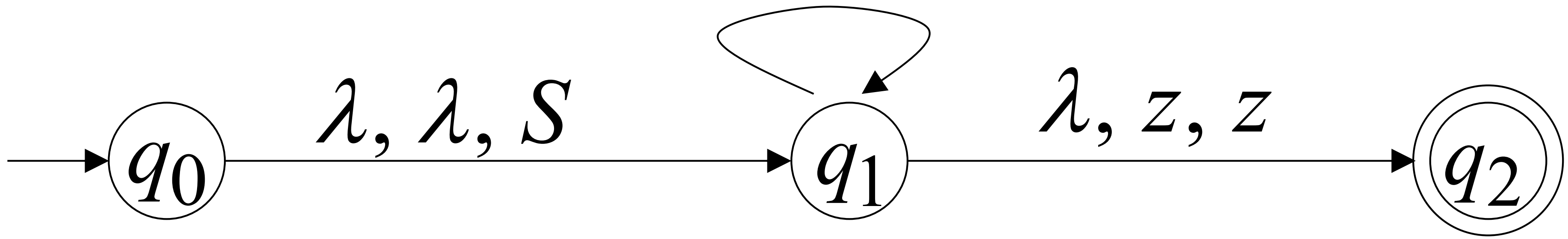


a, S, SA

a, S, λ

b, A, B

b, B, λ



Example 7.7

$$S \to aA$$

$$A \to aABC \mid bB \mid a$$

$$B \to b$$

$$C \to c$$

a, S, A

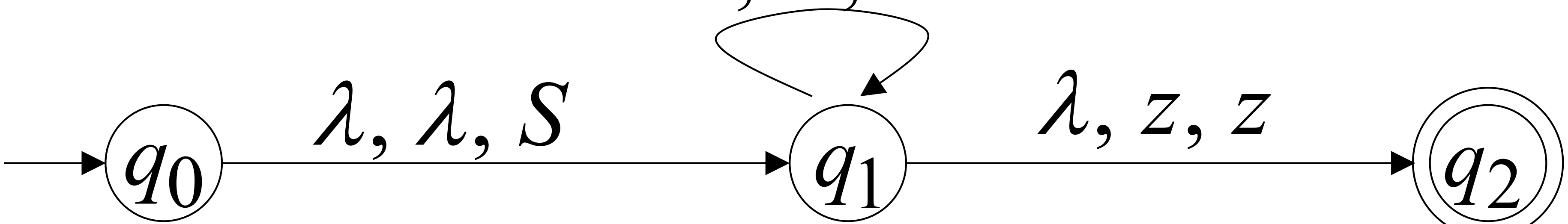
a, A, ABC

a, A, λ

b, A, B

b, B, λ

c, C, λ



Yet another approach…

An example grammar:

$$S \to aSTb$$

$$S \to b$$

$$T \to Ta$$

$$T \to \lambda$$

What is the equivalent NPDA?

Grammar:

$$S \to aSTb$$

$$S \to b$$

$$T \to Ta$$

$$T \to \lambda$$

A leftmost derivation:

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

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

Grammar:

$$S \rightarrow aSTb$$

$$S \rightarrow b$$

$$T \rightarrow Ta$$

$$T \rightarrow \lambda$$

NPDA:

$$\lambda, S, aSTb$$

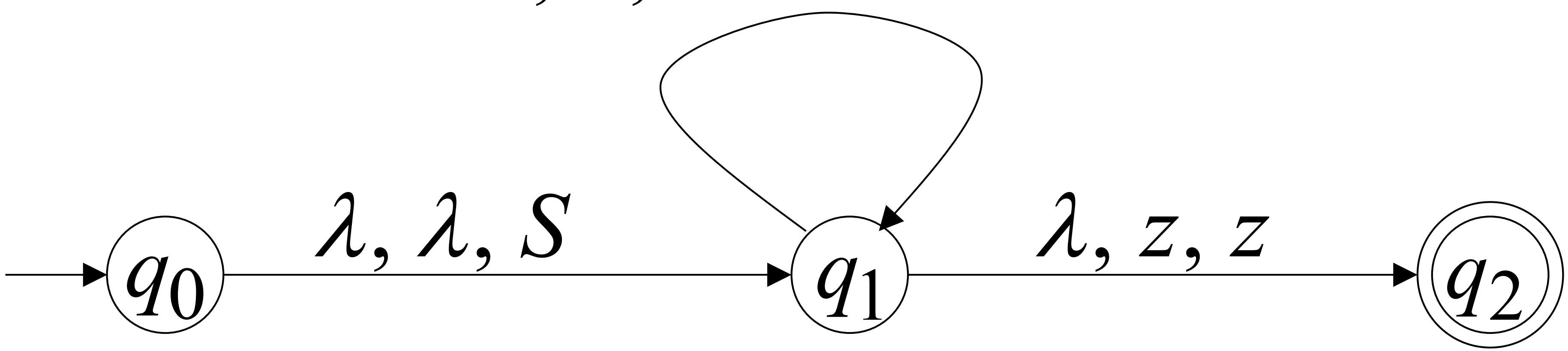
$$\lambda, S, b$$

$$\lambda, T, Ta$$

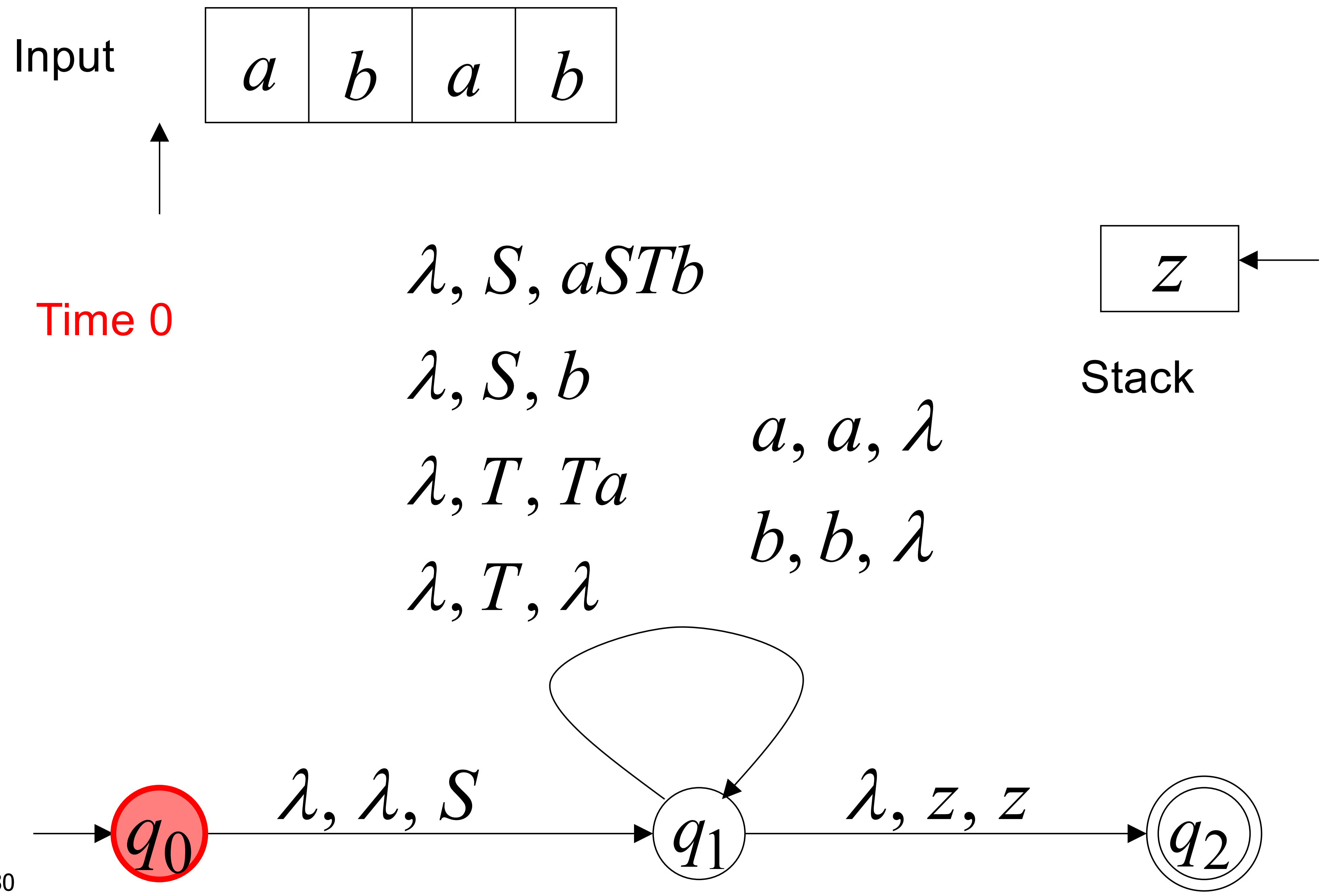
$$\lambda, T, \lambda$$

$$a, a, \lambda$$

$$b, b, \lambda$$

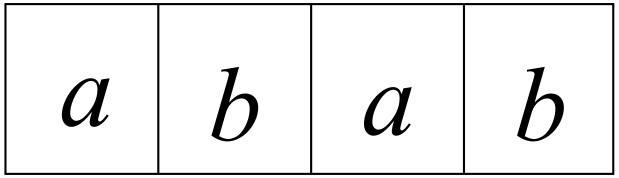


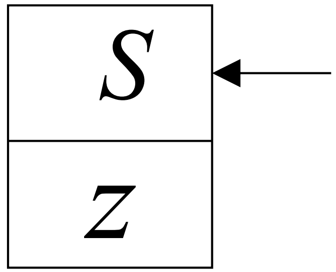
Derivation:


Input
a b a b
Time 0
$\lambda, S, aSTb$
λ, S, b
λ, T, Ta
λ, T, λ
a, a, λ
b, b, λ
z
Stack
q_0
λ, λ, S
q_1
λ, z, z
q_2

Derivation: S

Input





Stack

Time 0

$\lambda, S, aSTb$

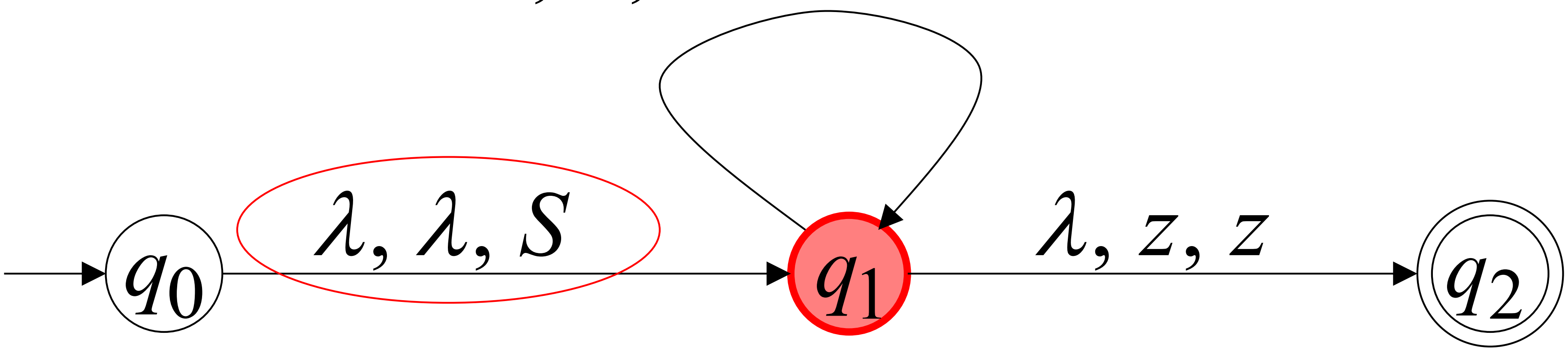
λ, S, b

λ, T, Ta

λ, T, λ

a, a, λ

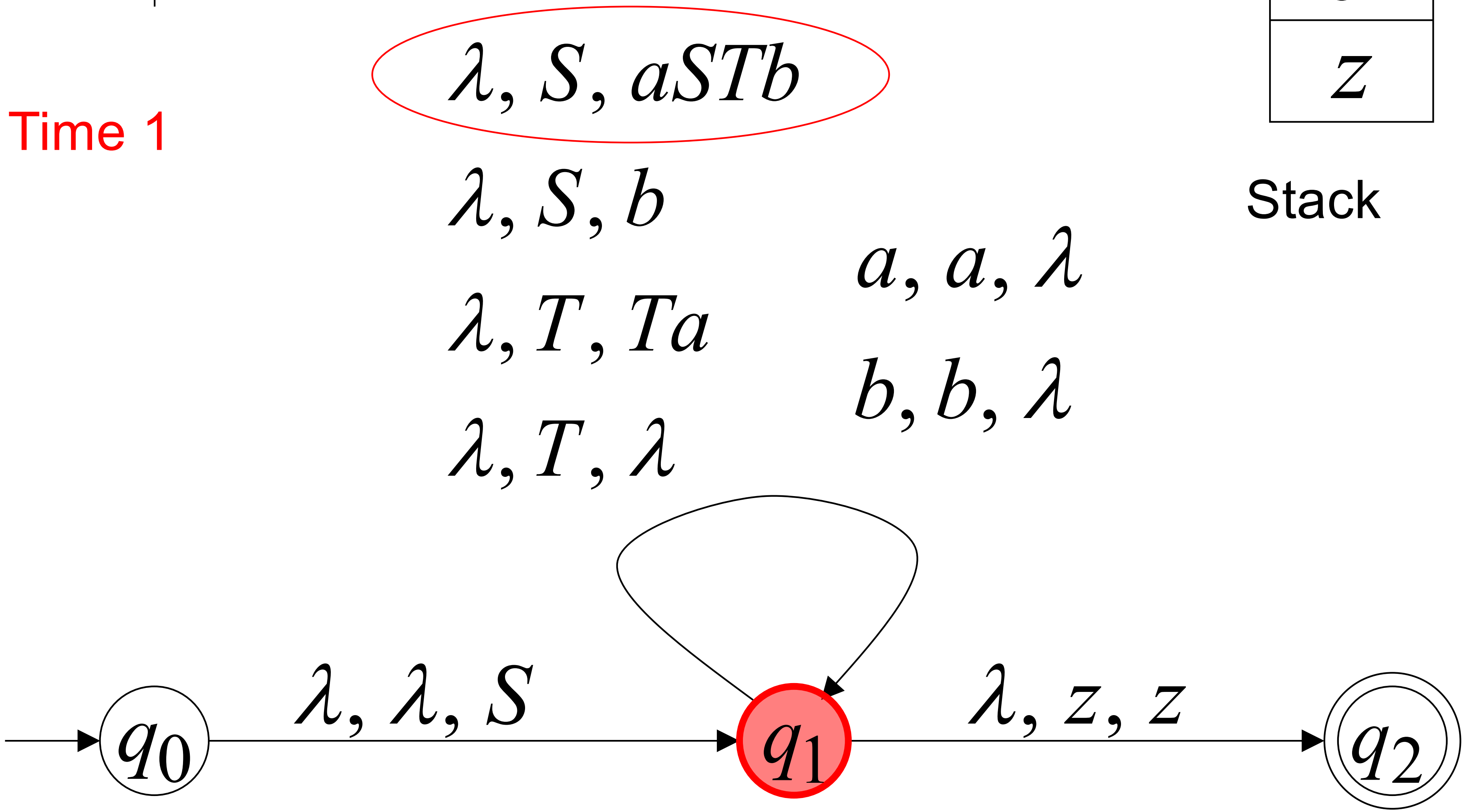
b, b, λ



Derivation: $S \Rightarrow aSTb$

Input

a	b	a	b



a
S
T
b
z

Stack

Derivation: $S \Rightarrow aSTb$

Input

a	b	a	b

Stack

$\cancel{a}$
S
T
b
z

Time 2

$\lambda, S, aSTb$

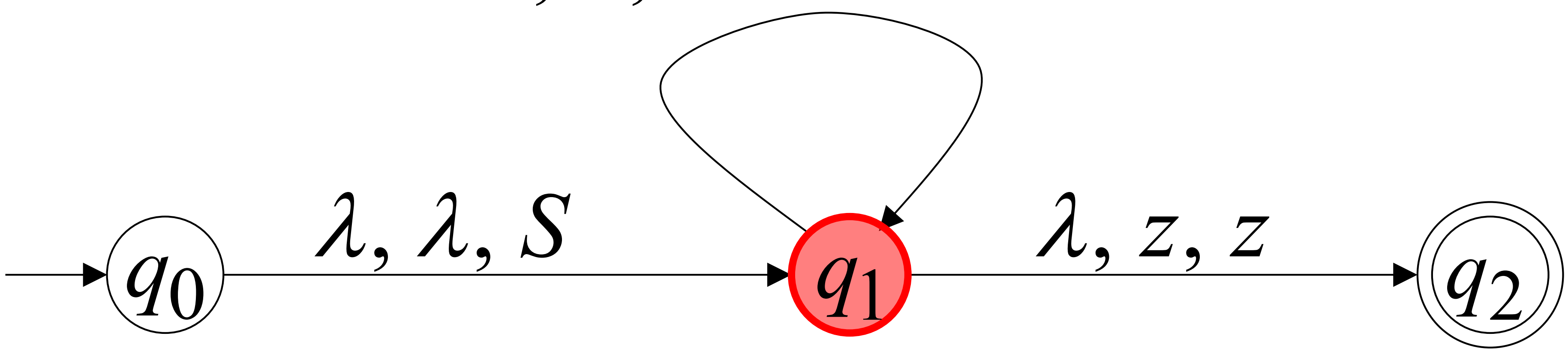
λ, S, b

λ, T, Ta

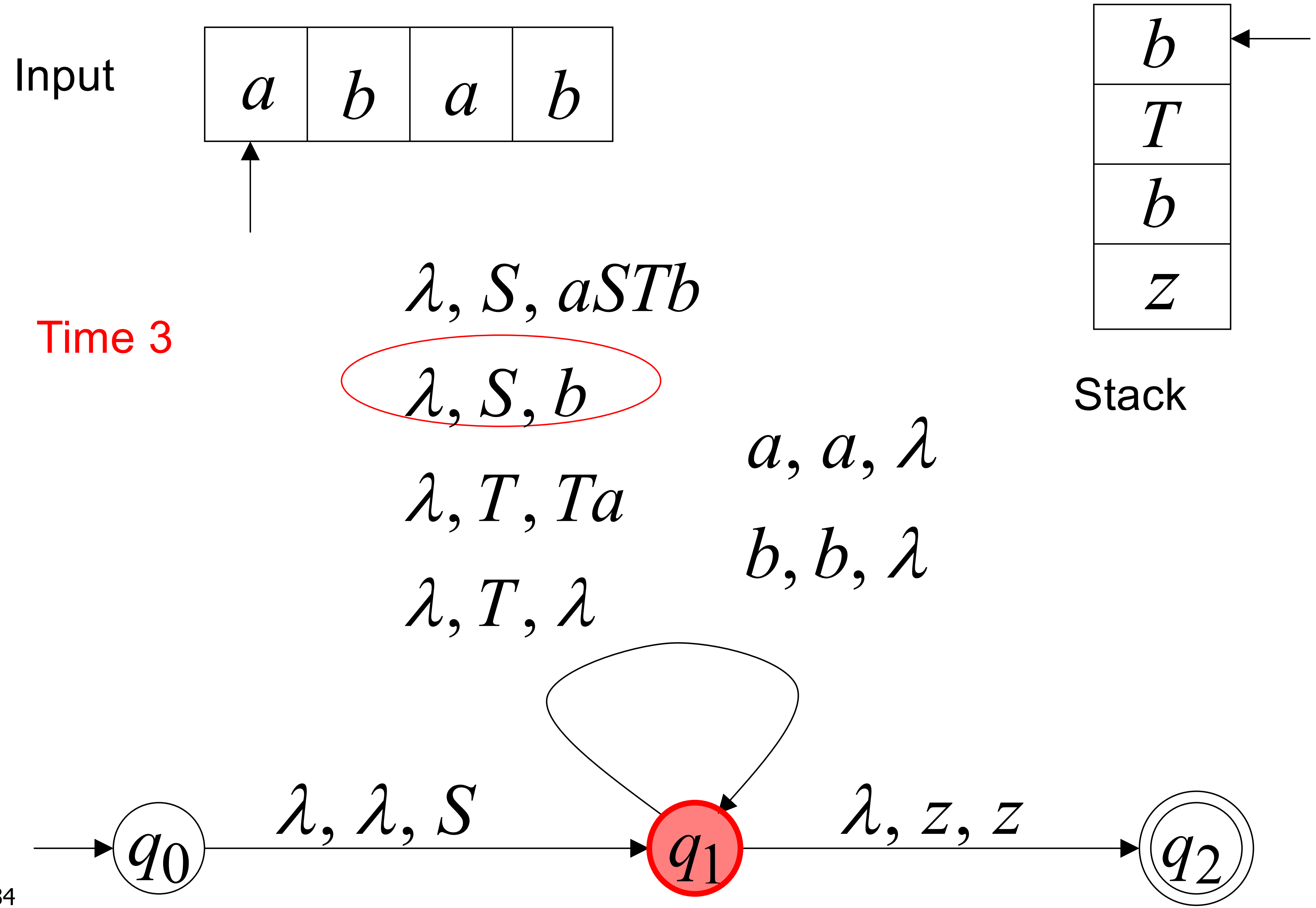
λ, T, λ

a, a, λ

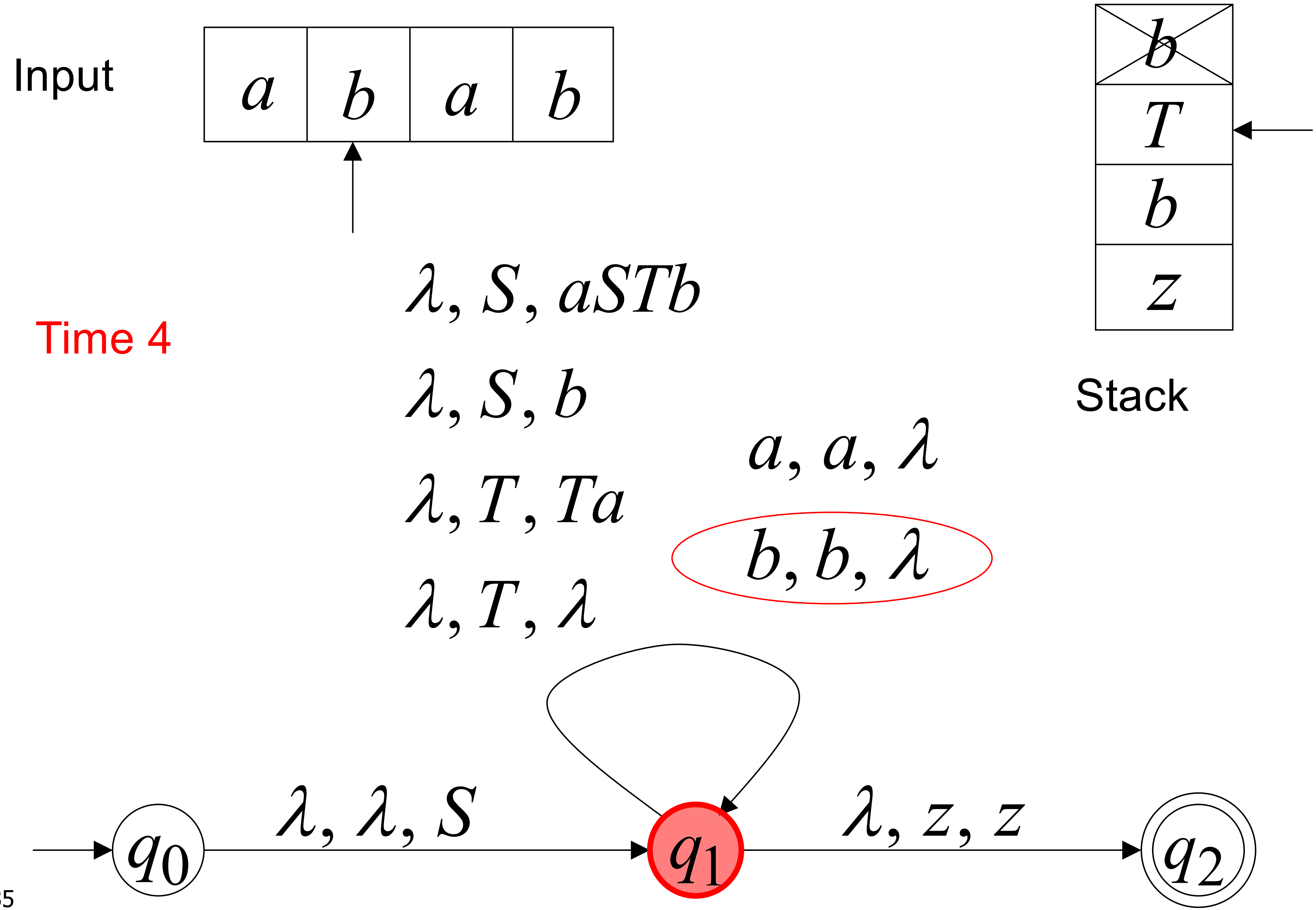
b, b, λ



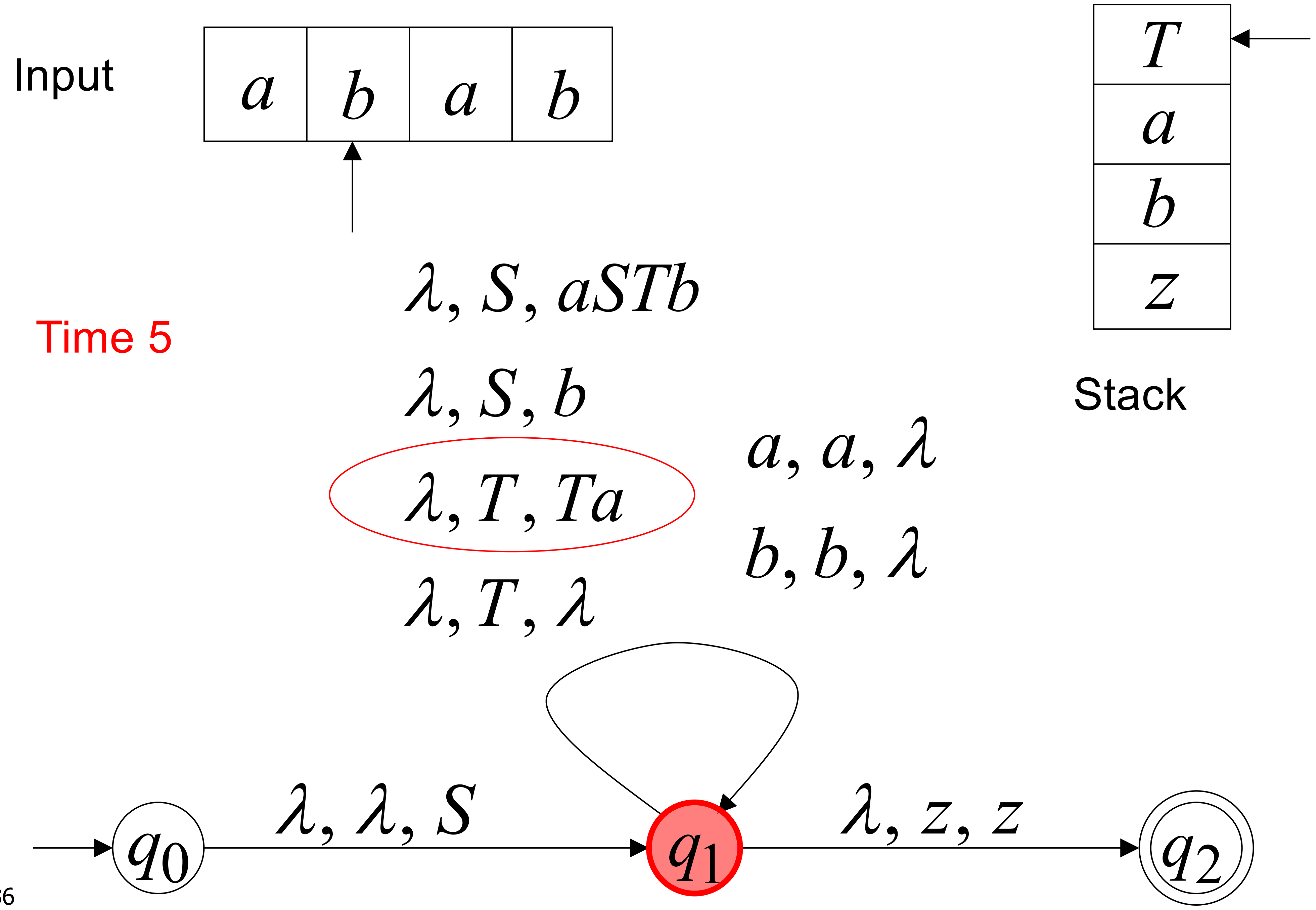
Derivation: $S \Rightarrow aSTb \Rightarrow abTb$



Derivation: $S \Rightarrow aSTb \Rightarrow abTb$

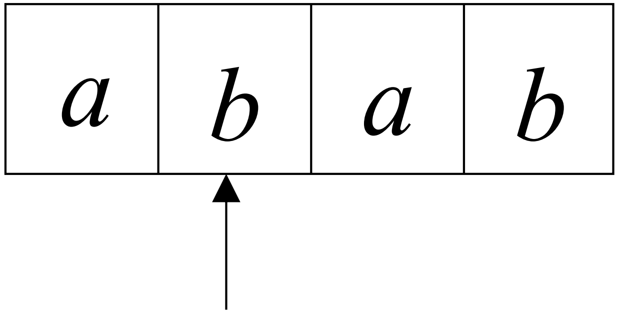


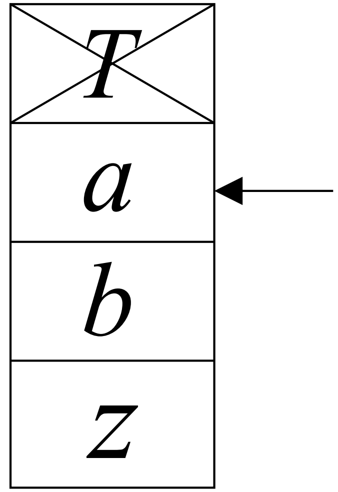
Derivation: $S \Rightarrow aSTb \Rightarrow abTb \Rightarrow abTab$



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

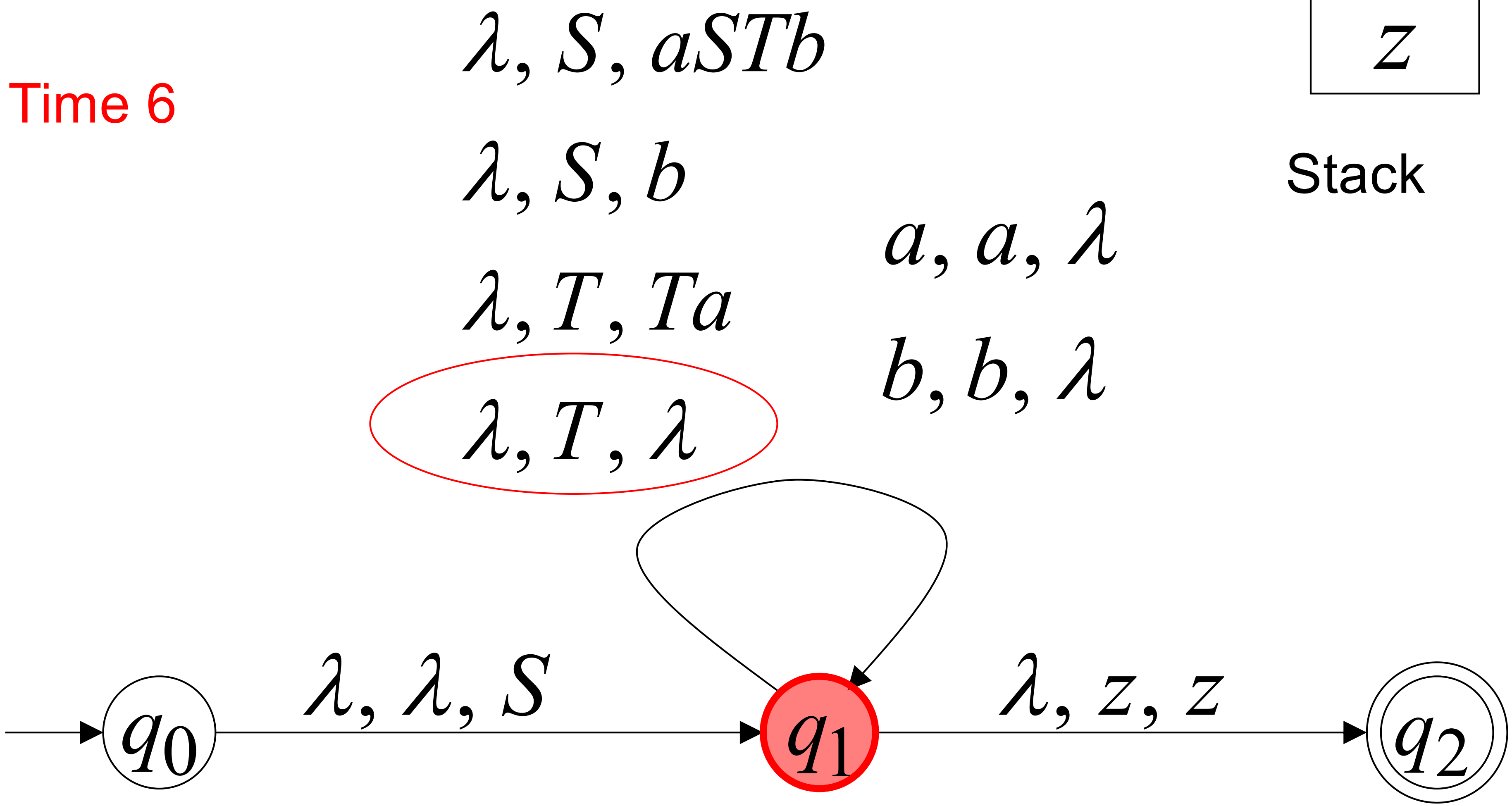
Input



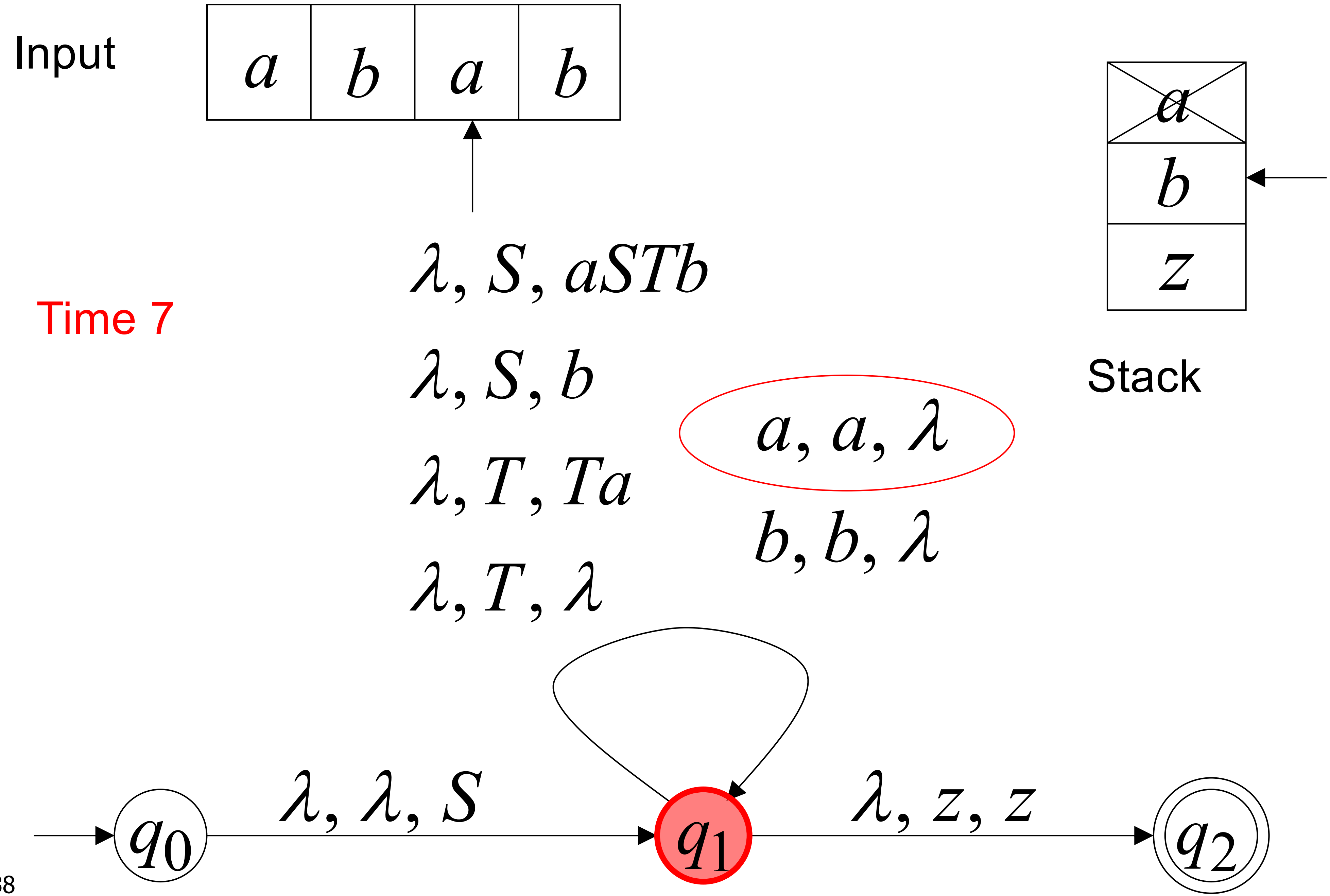


Stack

Time 6

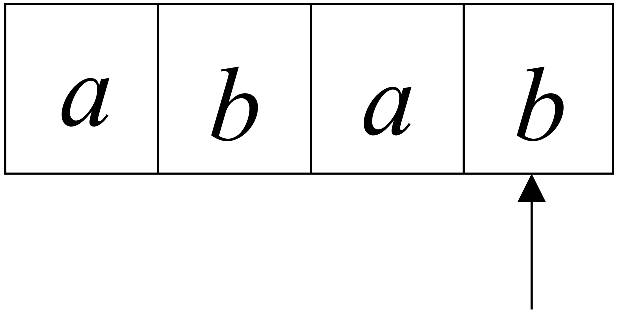


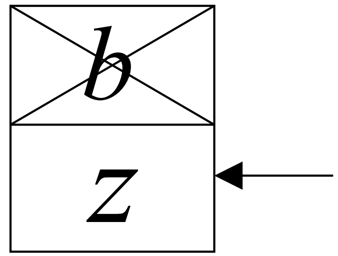
Derivation: $S \Rightarrow aSTb \Rightarrow abTb \Rightarrow abTab \Rightarrow abab$



Derivation:

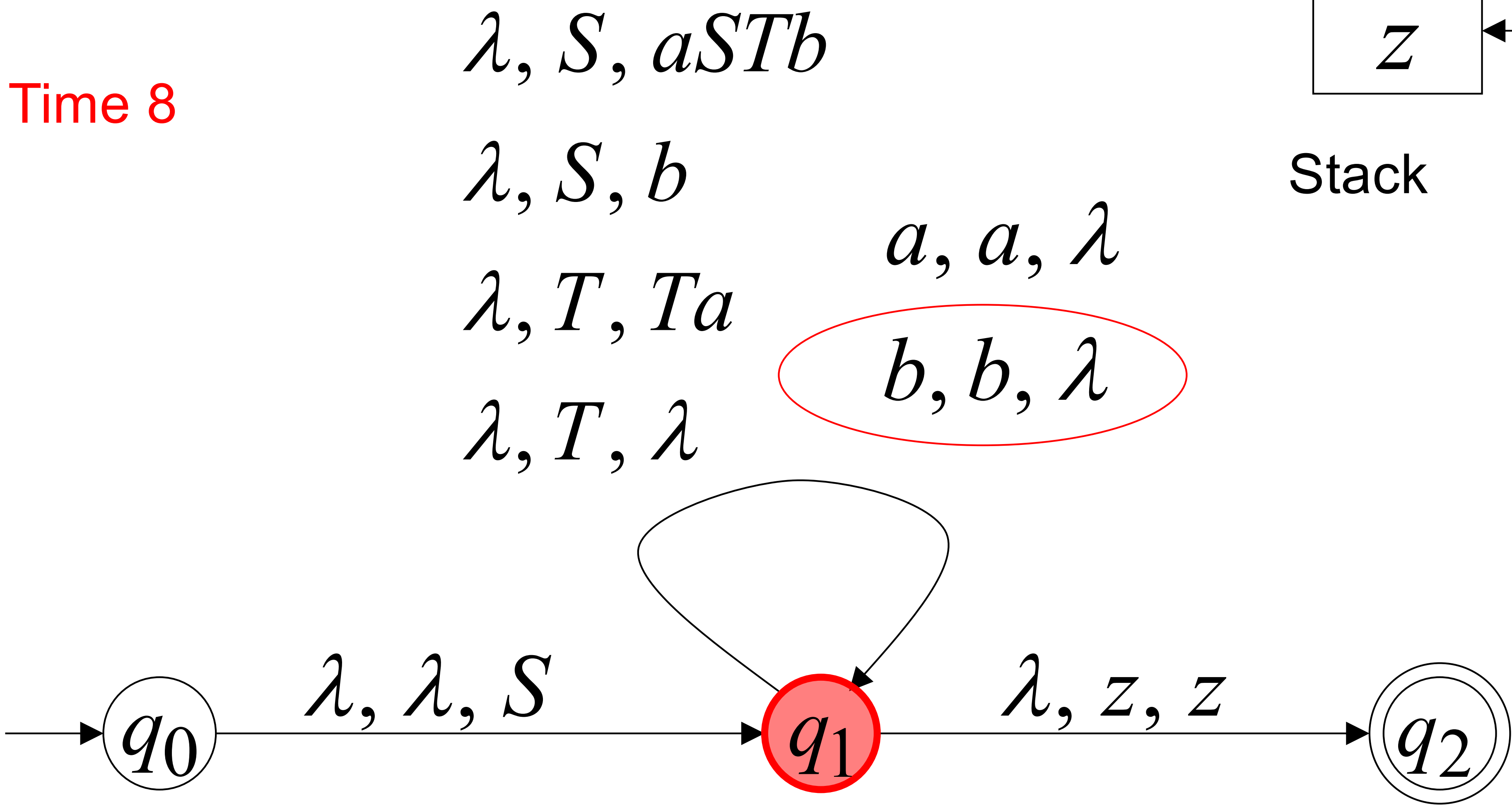
Input



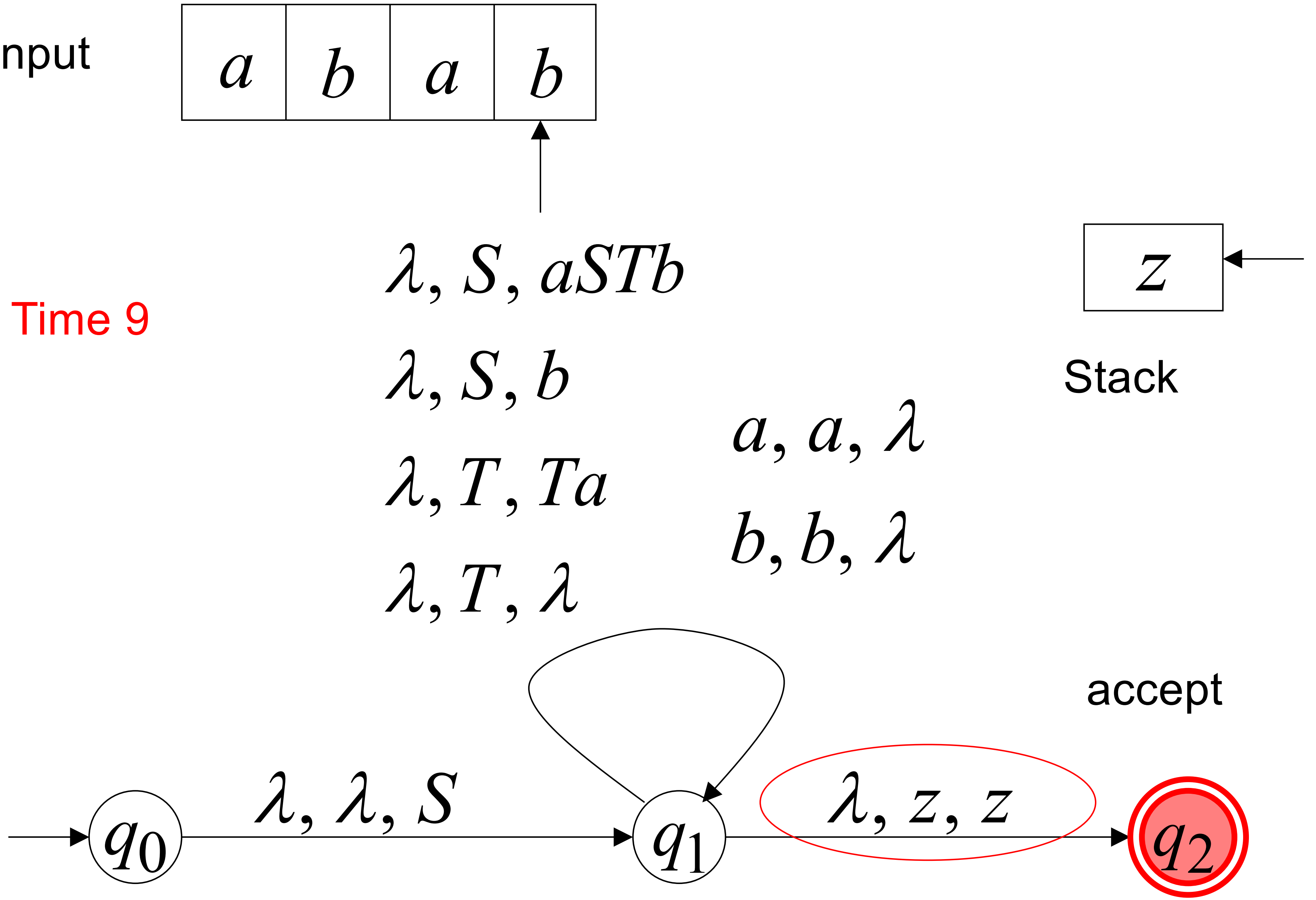


Stack

Time 8



Derivation:


Input
a b a b
Time 9
λ, S, aSTb
λ, S, b
λ, T, Ta
λ, T, λ
a, a, λ
b, b, λ
z
Stack
accept
q0
λ, λ, S
q1
λ, z, z
q2

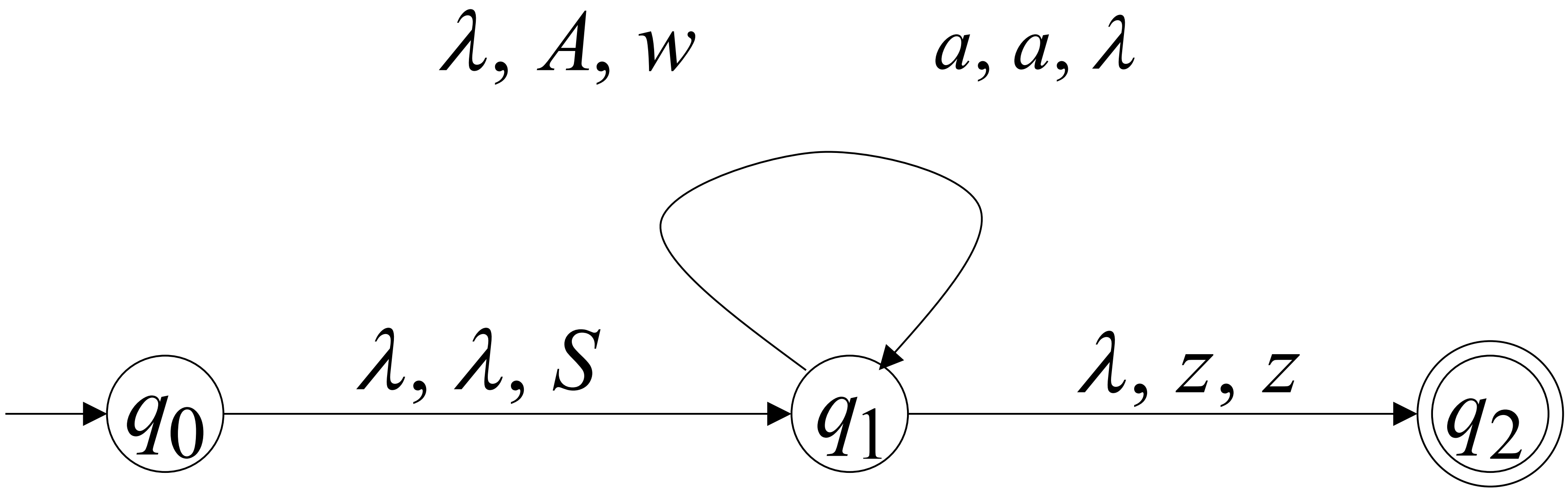
Constructing NPDA M from grammar G :

For any production

$$A \to w$$

For any terminal

$$a$$



In general:

Given any CFG G

We can construct an NPDA M

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

CFG 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 an 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 moves of machine

A derivation in Grammar G :

terminals variables

$$S \Rightarrow \cdots \Rightarrow \overset{\text{terminals}}{abc}\ldots\overset{\text{variables}}{ABC}\ldots \Rightarrow \ldots \Rightarrow abc\ldots$$

Processed input

Stack contents

Current configuration in NPDA M

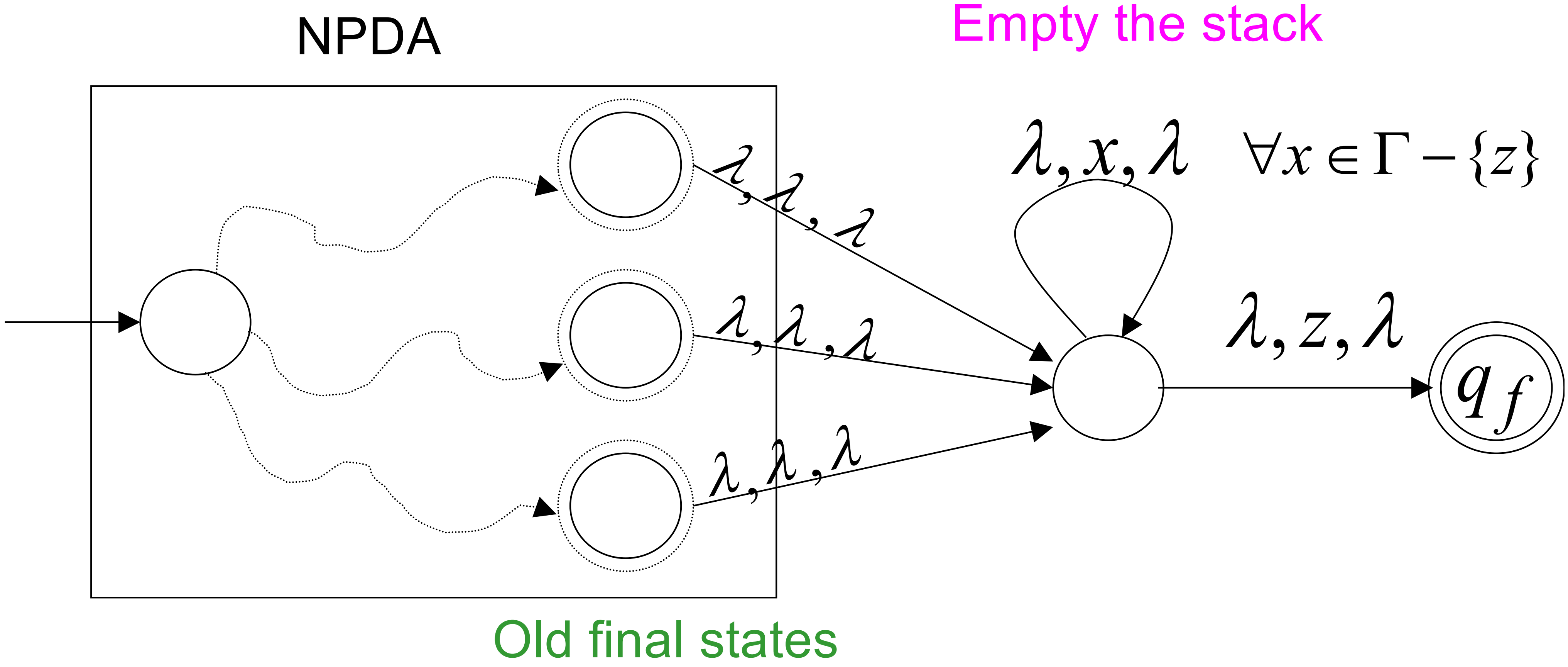
Some Necessary Modifications

Modify (if necessary) the NPDA so that:

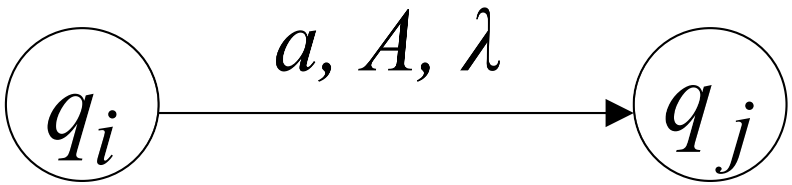
1) It has a single final state
   and empties the stack when it accepts a string

2) Has transitions in a special form

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



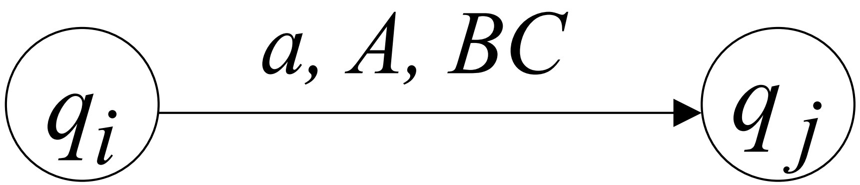
2) modify the NPDA so that
transitions have the following forms:



$a \in \Sigma \cup \{\lambda\}$

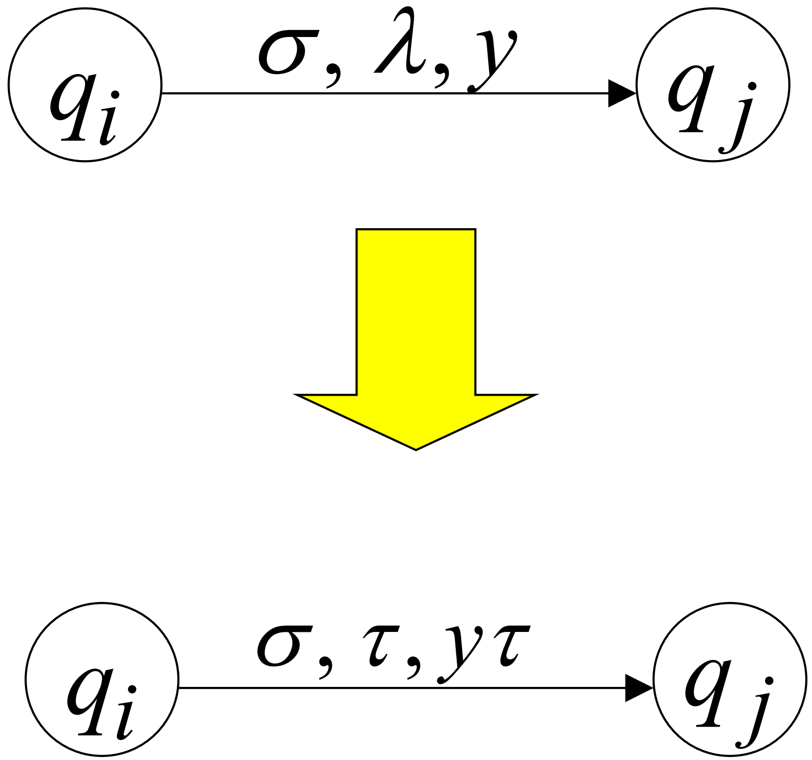
OR

A, B, C: stack symbols



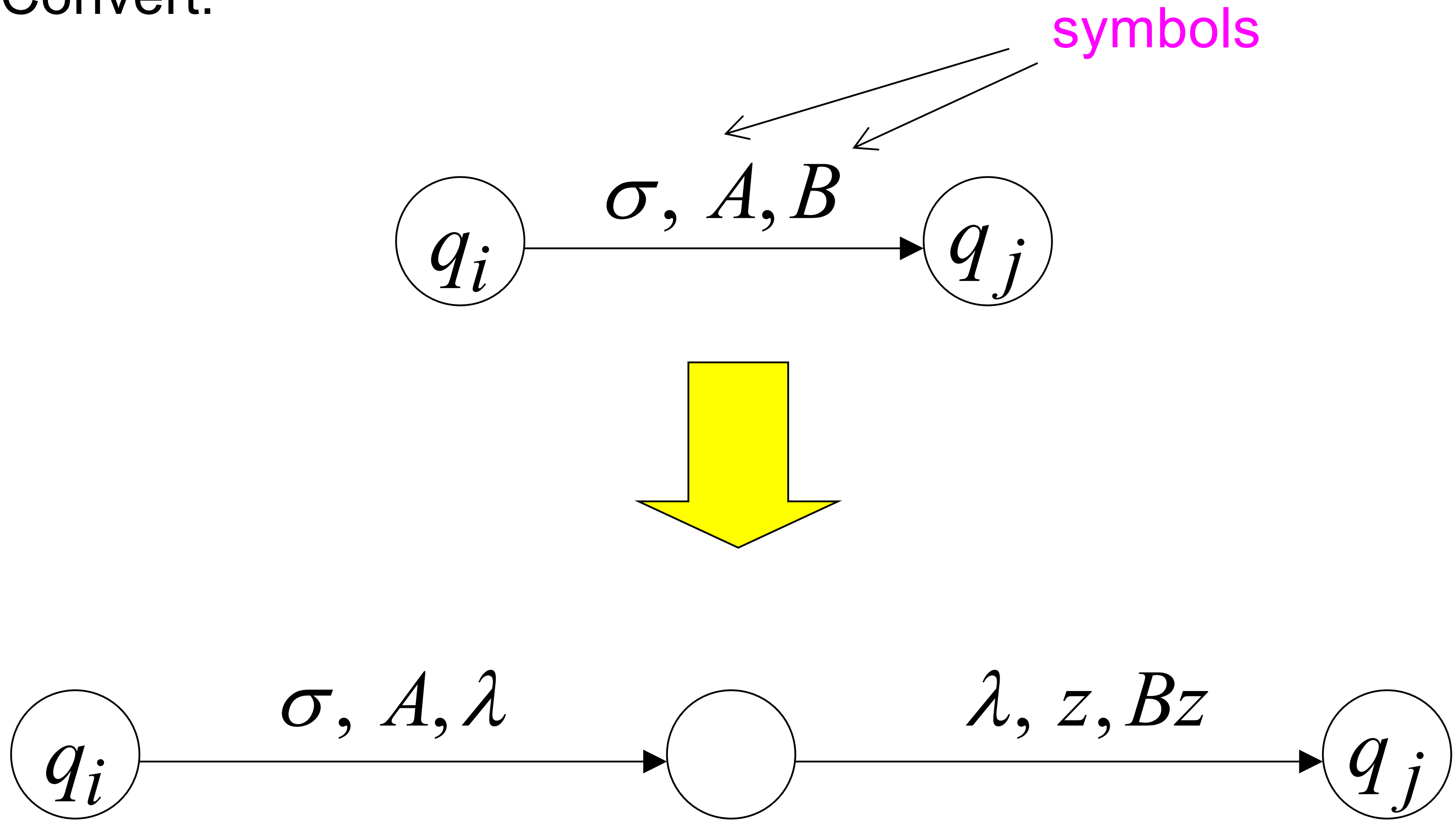
Each move either increases or decreases
the stack content by a single symbol

Convert:

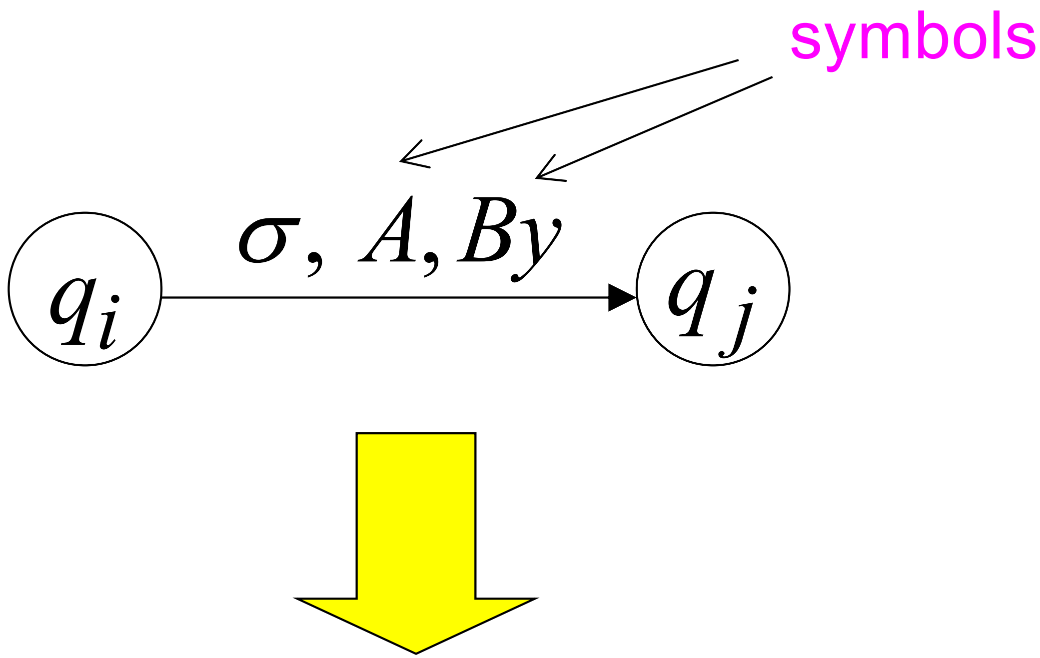


$$\forall \tau \in \Gamma - \{z\}$$

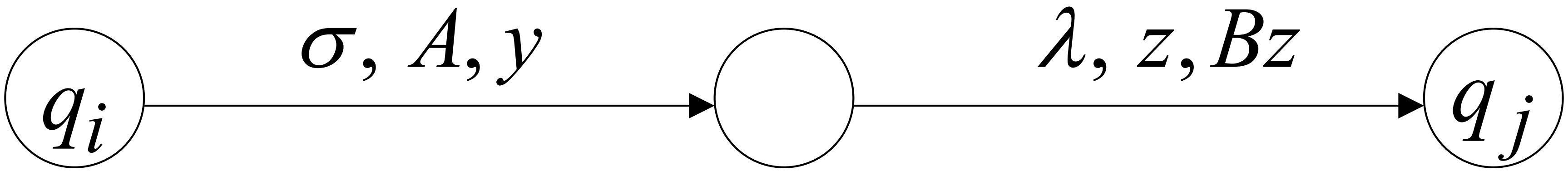
Convert:



Convert: $|y| \geq 2$



Convert recursively



Example of an NPDA in correct form:

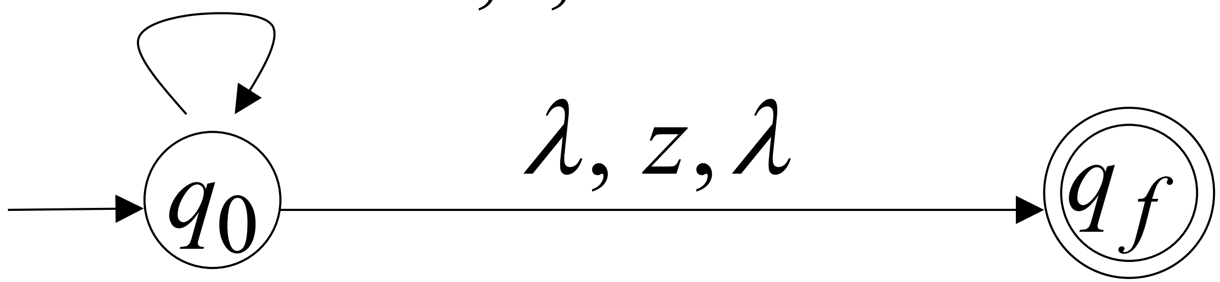
$$L(M) = \{w : \quad n_a = n_b\}$$

$$z : \text{initial stack symbol}$$

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

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

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



each variable is of the form $(q_i A q_j)$

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

corresponding move in npda:

- erasing A from the stack
- reading v
- going from state q_i to state q_j

The Grammar Construction

In grammar G :

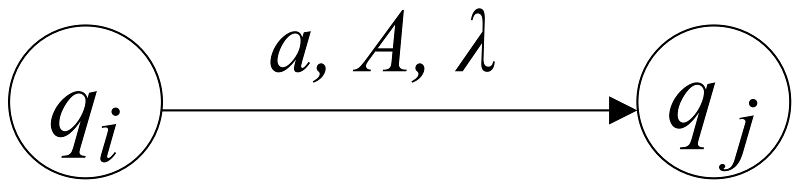
Stack symbol

Variables: $(q_i A q_j)$

states

Terminals:
Input symbols of NPDA

For each transition



We add production

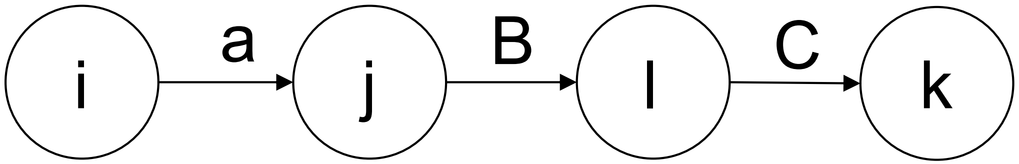
$$(q_i A q_j) \rightarrow a$$

For each transition

We add productions

$$(q_i A q_k) \rightarrow a(q_j B q_l)(q_l C q_k)$$

for all possible states q_k, q_l in the automaton



Stack bottom symbol

Start Variable: $(q_o z q_f)$

Start state

final state

Example 7.8

- Consider the NPDA with transitions

$$\delta(q_0, a, z) = \{(q_0, Az)\}$$
$$\boxed{\delta(q_0, a, A) = \{(q_0, A)\}}$$
$$\delta(q_0, b, A) = \{(q_1, \lambda)\}$$
$$\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$$

$$\Rightarrow$$

$$\delta(q_0, a, z) = \{(q_0, Az)\}$$
$$\boxed{\begin{gathered}\delta(q_3, \lambda, z) = \{(q_0, Az)\} \\ \delta(q_0, a, A) = \{(q_3, \lambda)\}\end{gathered}}$$
$$\delta(q_0, b, A) = \{(q_1, \lambda)\}$$
$$\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$$

$$q_0 \xrightarrow{a,\ A, \lambda} q_3 \xrightarrow{\lambda,\ z,\ Az} q_0$$

Example 7.8

$$\delta(q_0, a, z) = \{(q_0, Az)\}$$

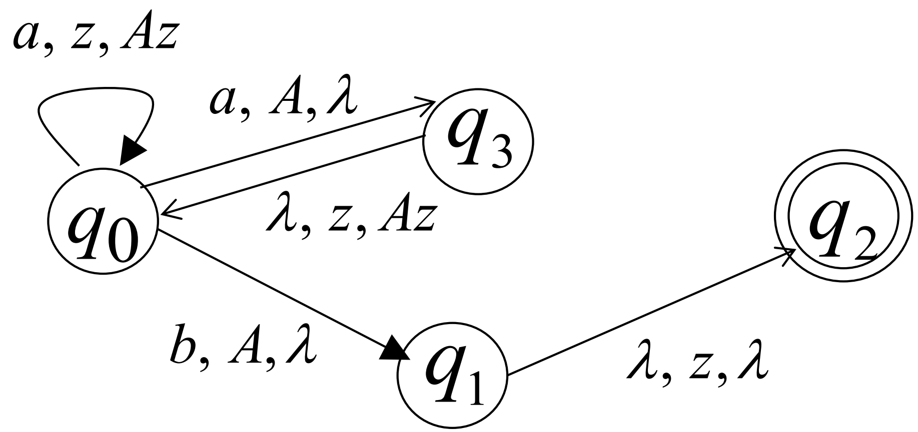
$$\delta(q_3, \lambda, z) = \{(q_0, Az)\}$$

$$\boxed{\begin{aligned} &\delta(q_0, a, A) = \{(q_3, \lambda)\} \\ &\delta(q_0, b, A) = \{(q_1, \lambda)\} \\ &\delta(q_1, \lambda, z) = \{(q_2, \lambda)\} \end{aligned}}$$

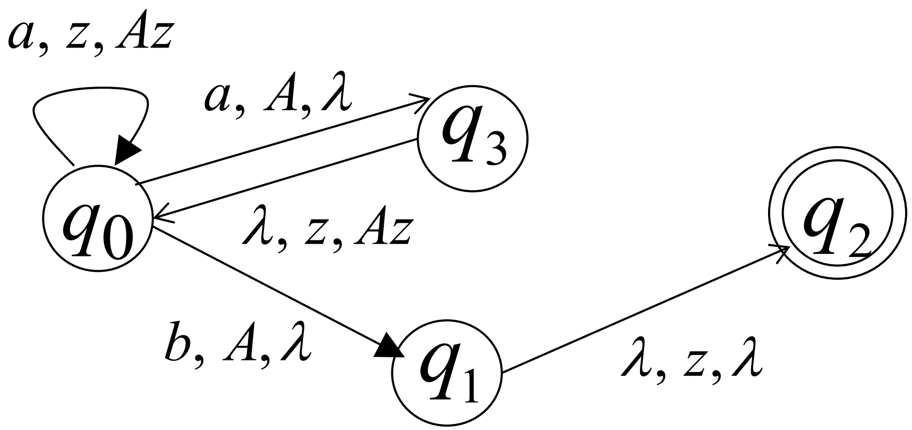
$$(q_0 A q_3) \rightarrow a$$

$$(q_0 A q_1) \rightarrow b$$

$$(q_1 z q_2) \rightarrow \lambda$$



$$\delta(q_0, a, z) = \{(q_0, Az)\}$$



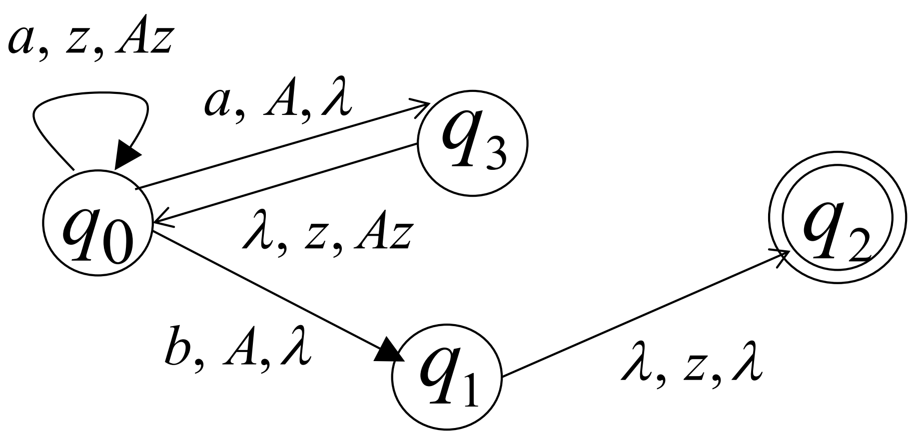
$$(q_i A q_k) \rightarrow a(q_j B q_l)(q_l C q_k)$$

$$\begin{aligned}
(q_0 z q_0) &\rightarrow \cancel{a(q_0 A q_0)(q_0 z q_0)} \mid \cancel{a(q_0 A q_1)(q_1 z q_0)} \\
&\quad\ \cancel{a(q_0 A q_2)(q_2 z q_0)} \mid a(q_0 A q_3)(q_3 z q_0) \\
(q_0 z q_1) &\rightarrow \cancel{a(q_0 A q_0)(q_0 z q_1)} \mid \cancel{a(q_0 A q_1)(q_1 z q_1)} \\
&\quad\ \cancel{a(q_0 A q_2)(q_2 z q_1)} \mid a(q_0 A q_3)(q_3 z q_1) \\
(q_0 z q_2) &\rightarrow \cancel{a(q_0 A q_0)(q_0 z q_2)} \mid a(q_0 A q_1)(q_1 z q_2) \\
&\quad\ \cancel{a(q_0 A q_2)(q_2 z q_2)} \mid a(q_0 A q_3)(q_3 z q_2) \\
(q_0 z q_3) &\rightarrow \cancel{a(q_0 A q_0)(q_0 z q_3)} \mid \cancel{a(q_0 A q_1)(q_1 z q_3)} \\
&\quad\ \cancel{a(q_0 A q_2)(q_2 z q_3)} \mid a(q_0 A q_3)(q_3 z q_3)
\end{aligned}$$

$(q_0 A q_0)$ *and* $(q_0 A q_2)$ do not occur on the left side of any production must be useless

no path from q_1 to q_0, from q_1 to q_1, from q_1 to q_3, and from q_2 to q_2,

$$\delta(q_3, \lambda, z) = \{(q_0, Az)\}$$



$$(q_i A q_k) \rightarrow a(q_j B q_l)(q_l C q_k)$$

$$\begin{aligned}
(q_3 z q_0) &\rightarrow \cancel{(q_0 A q_0)(q_0 z q_0)} \mid \cancel{(q_0 A q_1)(q_1 z q_0)} \\
&\quad \cancel{(q_0 A q_2)(q_2 z q_0)} \mid (q_0 A q_3)(q_3 z q_0) \\
(q_3 z q_1) &\rightarrow \cancel{(q_0 A q_0)(q_0 z q_1)} \mid \cancel{(q_0 A q_1)(q_1 z q_1)} \\
&\quad \cancel{(q_0 A q_2)(q_2 z q_1)} \mid (q_0 A q_3)(q_3 z q_1) \\
(q_3 z q_2) &\rightarrow \cancel{(q_0 A q_0)(q_0 z q_2)} \mid (q_0 A q_1)(q_1 z q_2) \\
&\quad \cancel{(q_0 A q_2)(q_2 z q_2)} \mid (q_0 A q_3)(q_3 z q_2) \\
(q_3 z q_3) &\rightarrow \cancel{(q_0 A q_0)(q_0 z q_3)} \mid \cancel{(q_0 A q_1)(q_1 z q_3)} \\
&\quad \cancel{(q_0 A q_2)(q_2 z q_3)} \mid (q_0 A q_3)(q_3 z q_3)
\end{aligned}$$

$(q_0 A q_0)$ *and* $(q_0 A q_2)$ do not occur on the left side of any production must be useless

no path from q_1 to q_0, from q_1 to q_1, from q_1 to q_3, and from q_2 to q_2, The final result

with start variable $(q_0 z q_2)$

$$(q_0 A q_3) \to a$$
$$(q_0 A q_1) \to b$$
$$(q_1 z q_2) \to \lambda$$
$$(q_0 z q_0) \to a(q_0 A q_3)(q_3 z q_0)$$
$$(q_0 z q_1) \to a(q_0 A q_3)(q_3 z q_1)$$
$$(q_0 z q_2) \to a(q_0 A q_1)(q_1 z q_2) \mid a(q_0 A q_3)(q_3 z q_2)$$
$$(q_0 z q_3) \to a(q_0 A q_3)(q_3 z q_3)$$
$$(q_3 z q_0) \to (q_0 A q_3)(q_3 z q_0)$$
$$(q_3 z q_1) \to (q_0 A q_3)(q_3 z q_1)$$
$$(q_3 z q_2) \to (q_0 A q_1)(q_1 z q_2) \mid (q_0 A q_3)(q_3 z q_2)$$
$$(q_3 z q_3) \to (q_0 A q_3)(q_3 z q_3)$$

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:

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

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

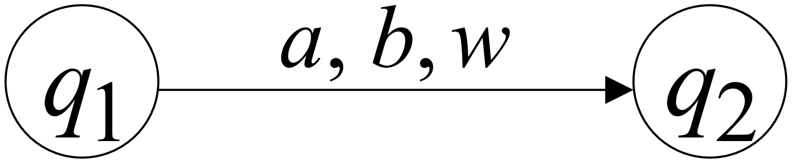
$$\left\{ \begin{array}{c} \text{Context-Free} \\ \text{Languages} \\ \text{(Grammars)} \end{array} \right\} \supseteq \left\{ \begin{array}{c} \text{Languages} \\ \text{Accepted by} \\ \text{NPDAs} \end{array} \right\}$$

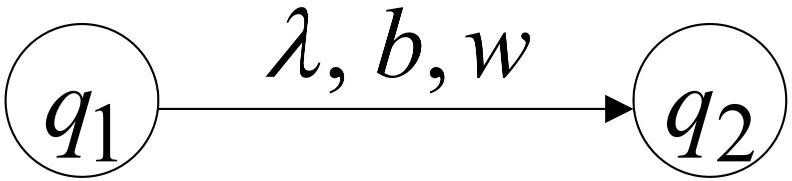
Outline

1. Nondeterministic Pushdown Automata
2. Pushdown Automata and Context-Free Languages
3. Deterministic Pushdown Automata and Deterministic CFLs

Deterministic PDA: DPDA

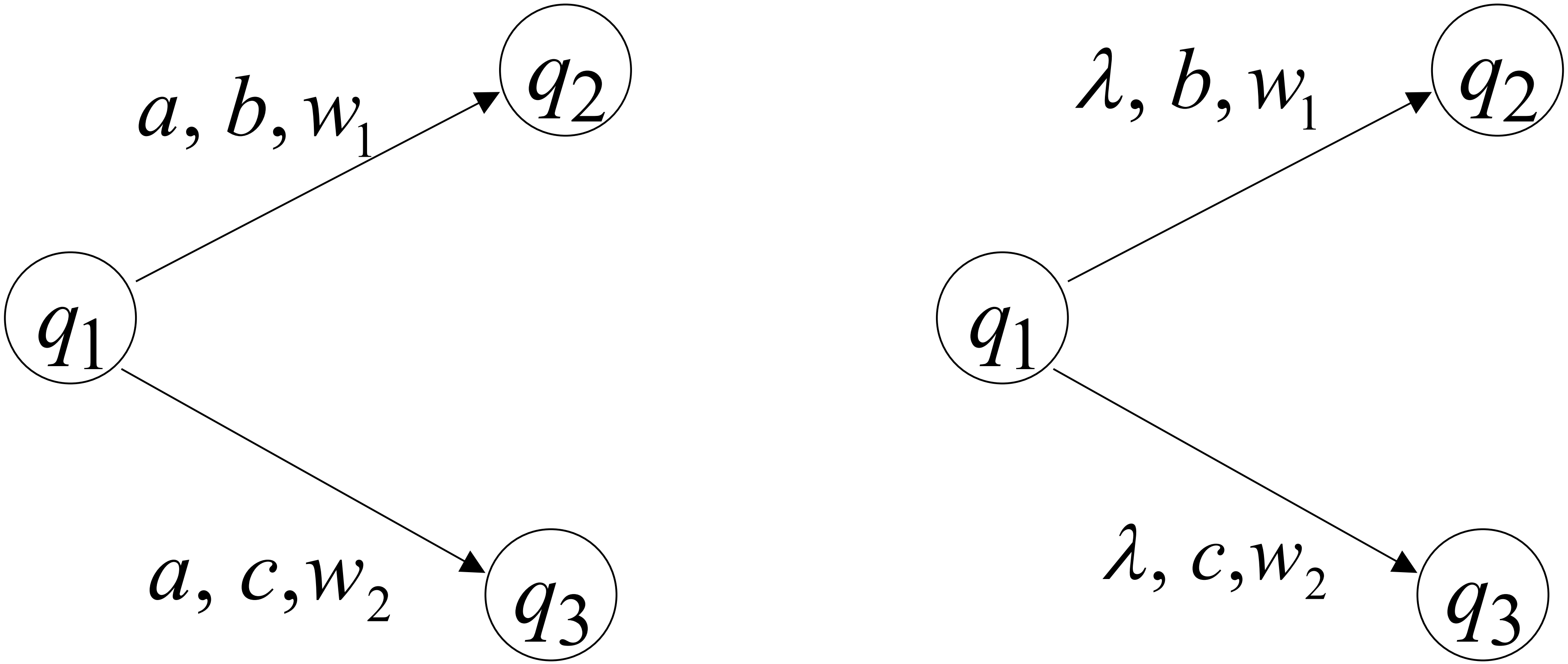
Allowed transitions:





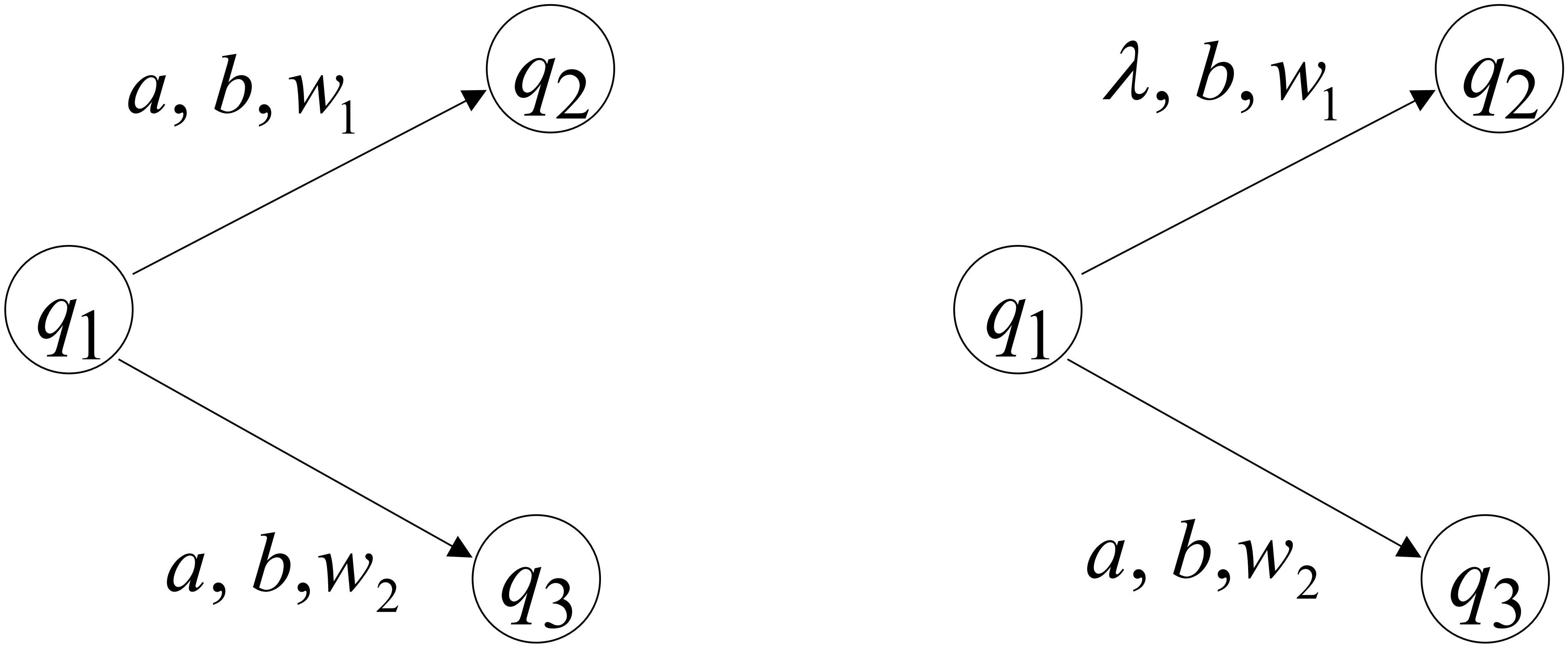
(deterministic choices)

Allowed transitions:



(deterministic choices)

Not allowed:



(non deterministic choices)

Deterministic PDA

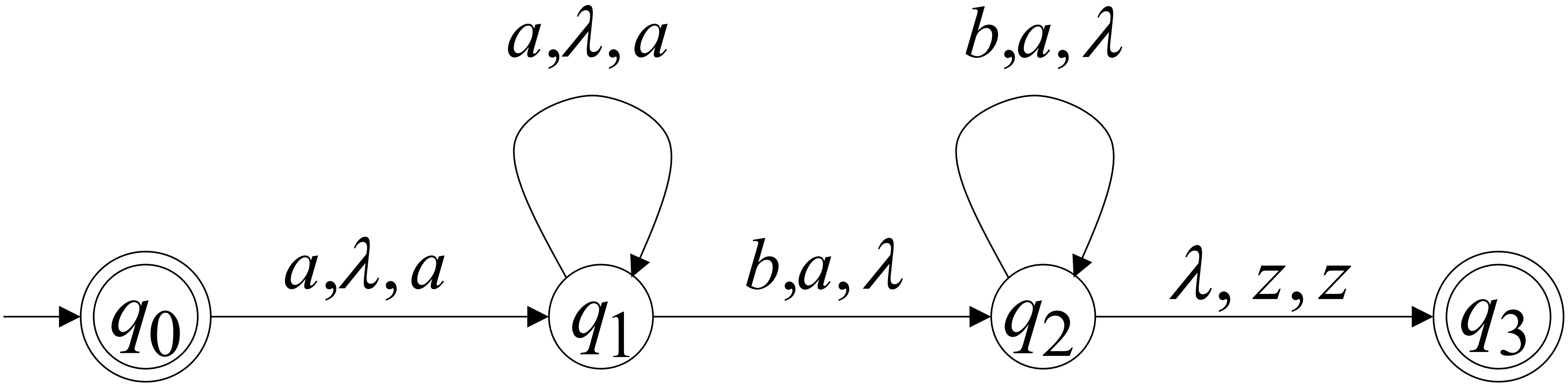
- $\delta(q, a, b)$ contains at most one element
- If $\delta(q, \lambda, b)$ is not empty, then $\delta(q, c, b)$ must be empty for every $c \in \Sigma$

- λ transition is possible
- Some transitions may be to the empty set

- At all times at most one possible move

DPDA example

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



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

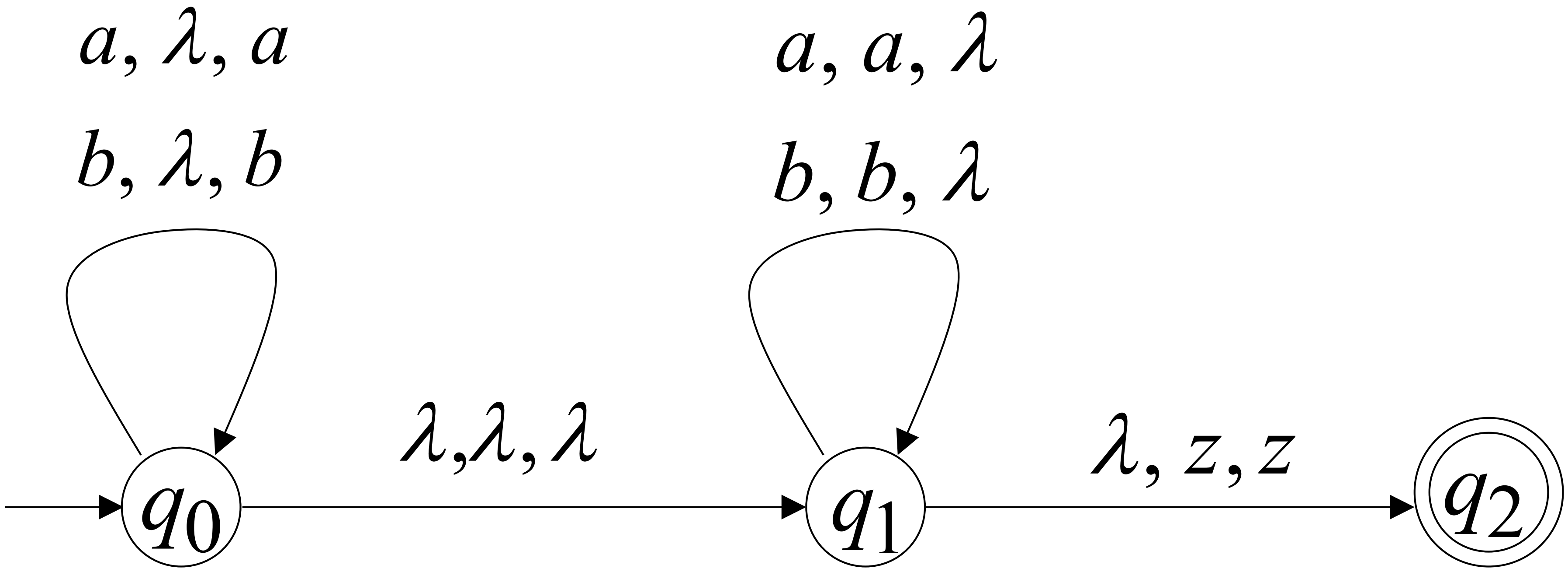
is **deterministic context-free**

Definition:

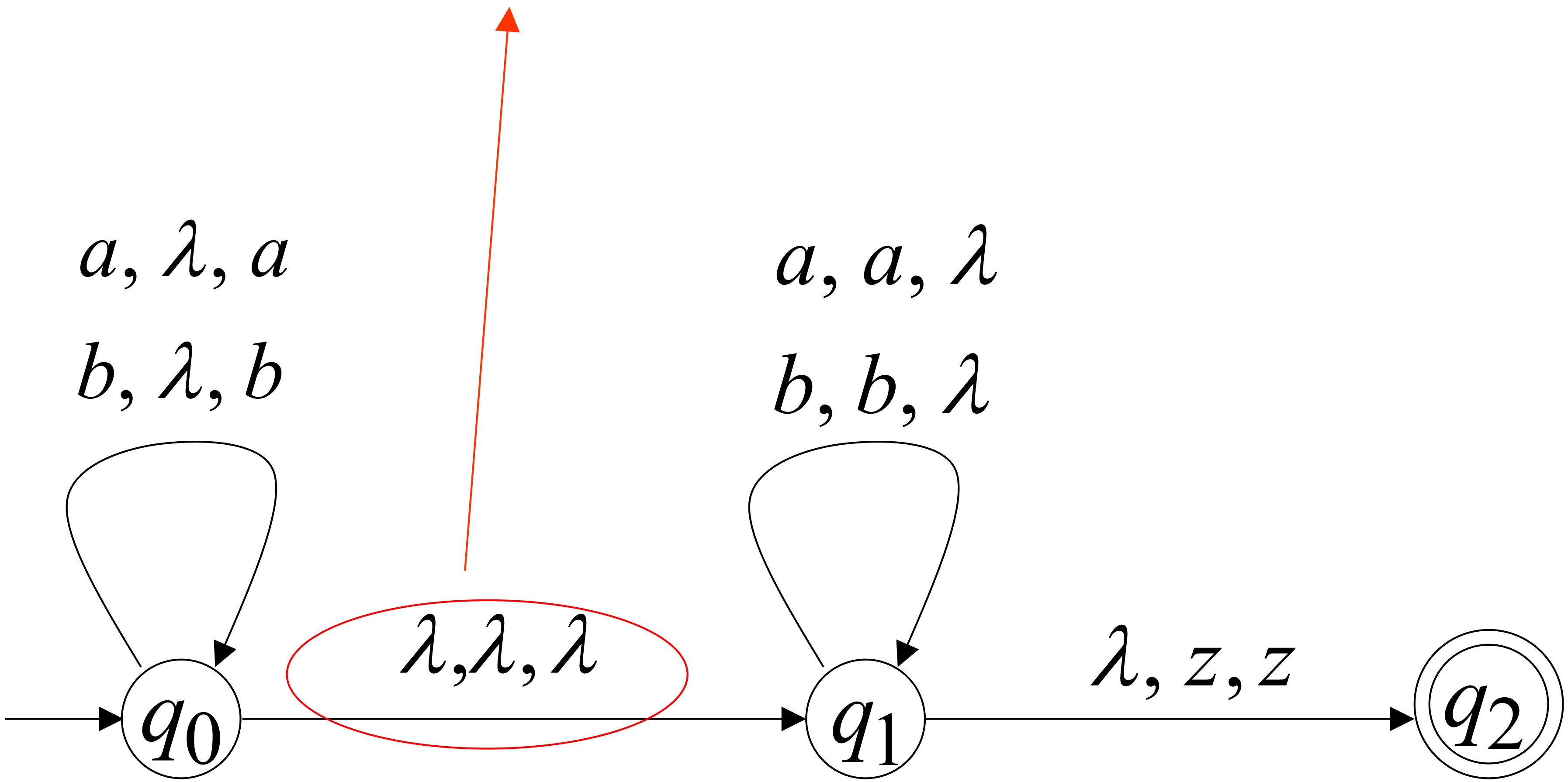
A language L is **deterministic context-free** if there exists some DPDA that accepts it

Example of Non-DPDA (NPDA)

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



Not allowed in DPDAs



NPDAs

Have More Power than

DPDAs

There are context-free languages that are not deterministic

It holds that:

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

Since every DPDA is also an NPDA

We will actually show:

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

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

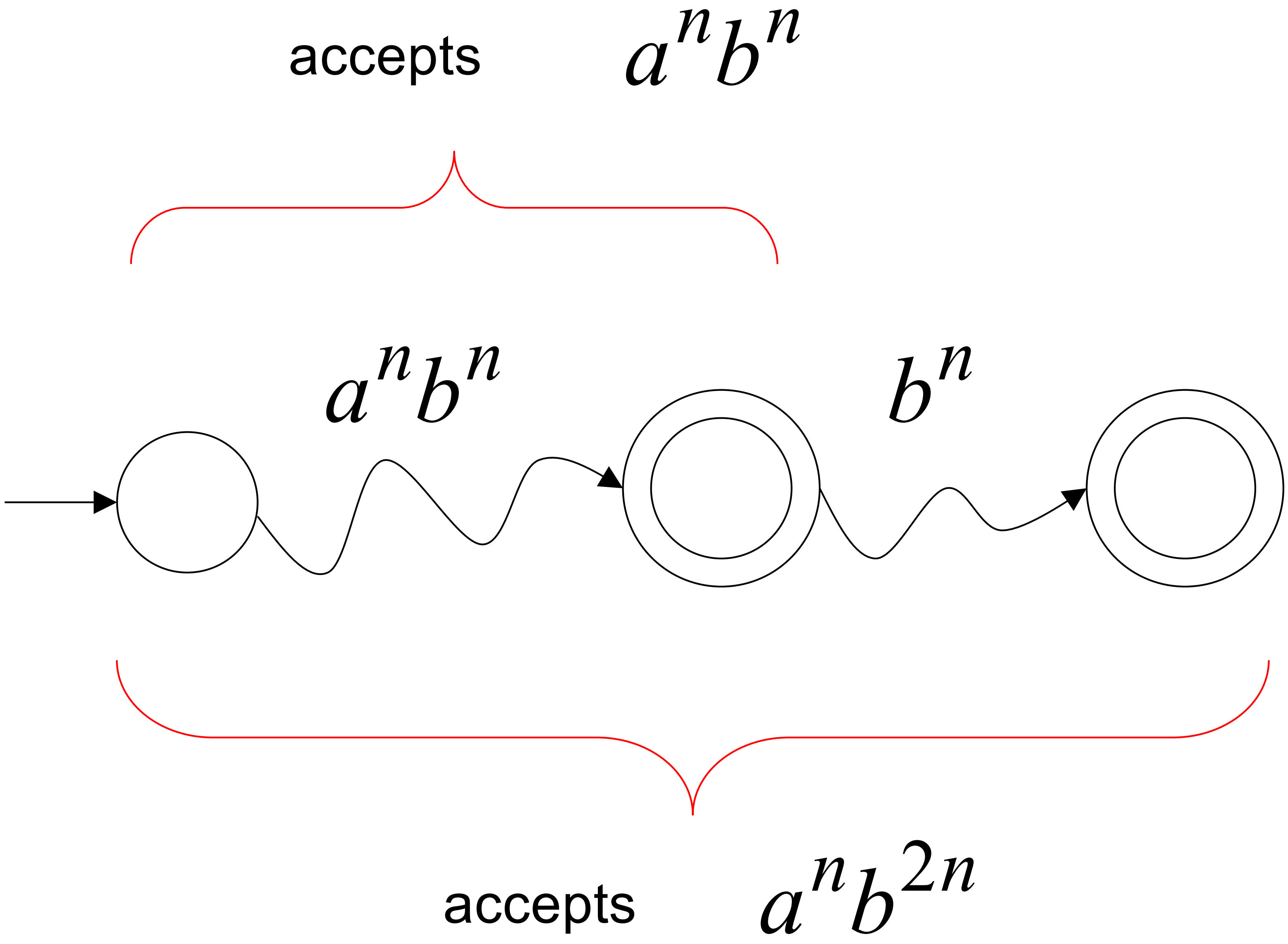
$$L = \{a^n b^n\} \cup \{a^n b^{2n}\}$$

is deterministic context free

Therefore:

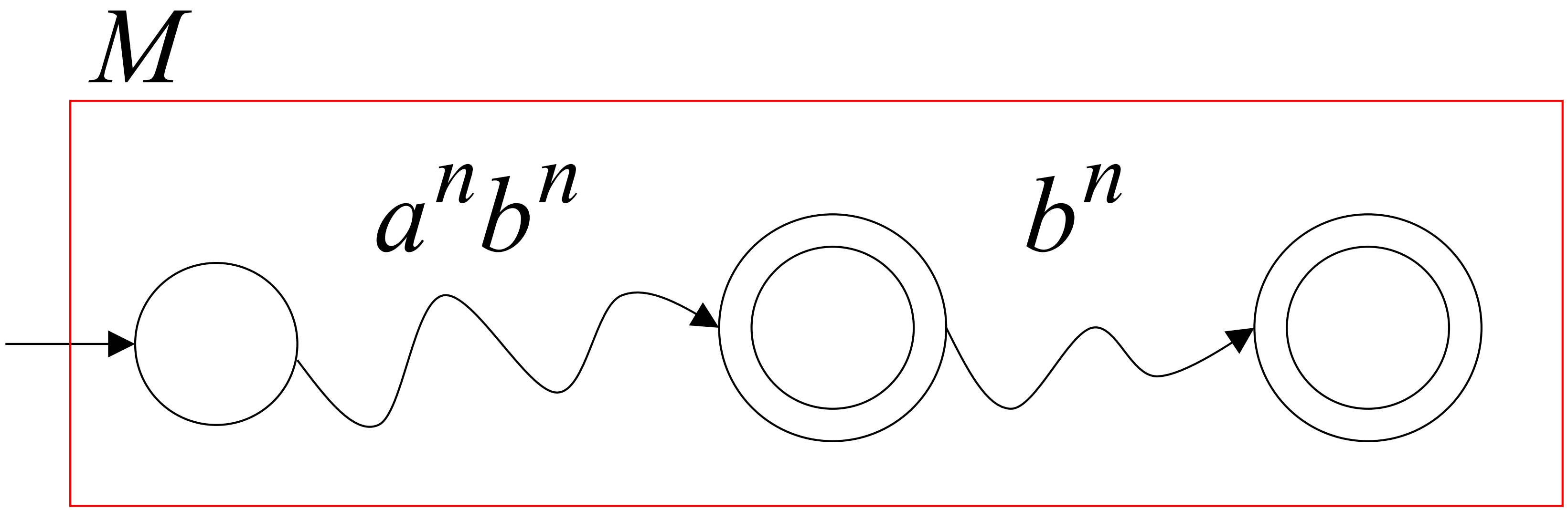
there is a DPDA M that accepts L

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

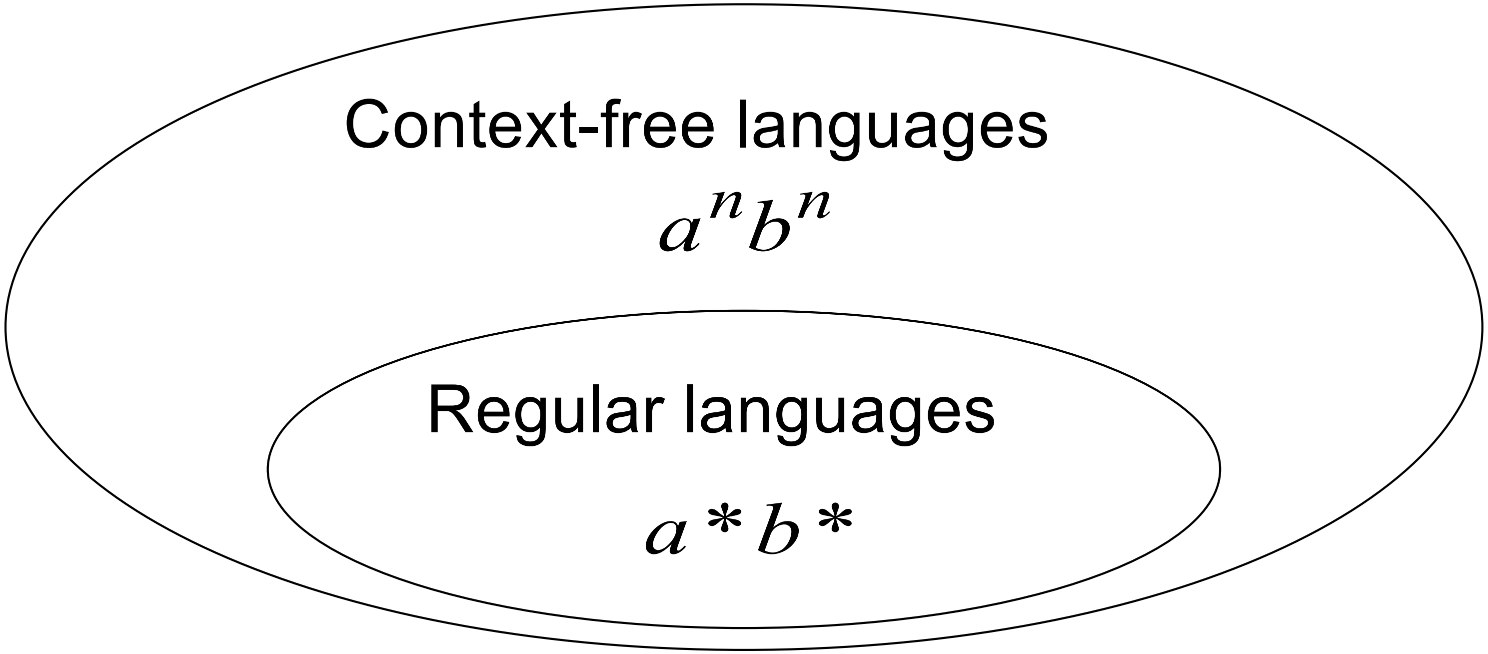


DPDA M with $L(M) = \{a^n b^n\} \cup \{a^n b^{2n}\}$

Such a path exists because of the determinism



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



(we will prove this 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)

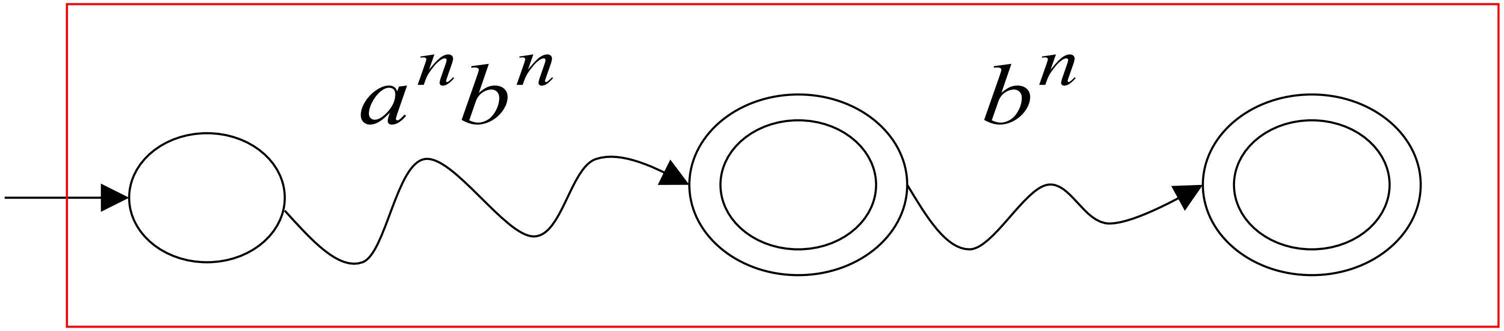
We will construct a NPDA that accepts:

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

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

which is a contradiction!

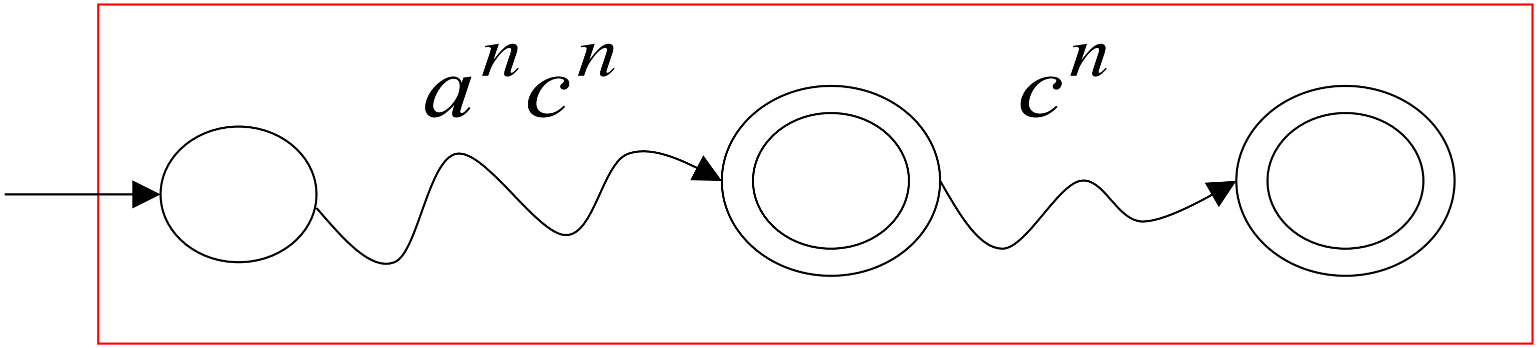
M $L(M)=\{a^nb^n\}\cup\{a^nb^{2n}\}$



Modify M

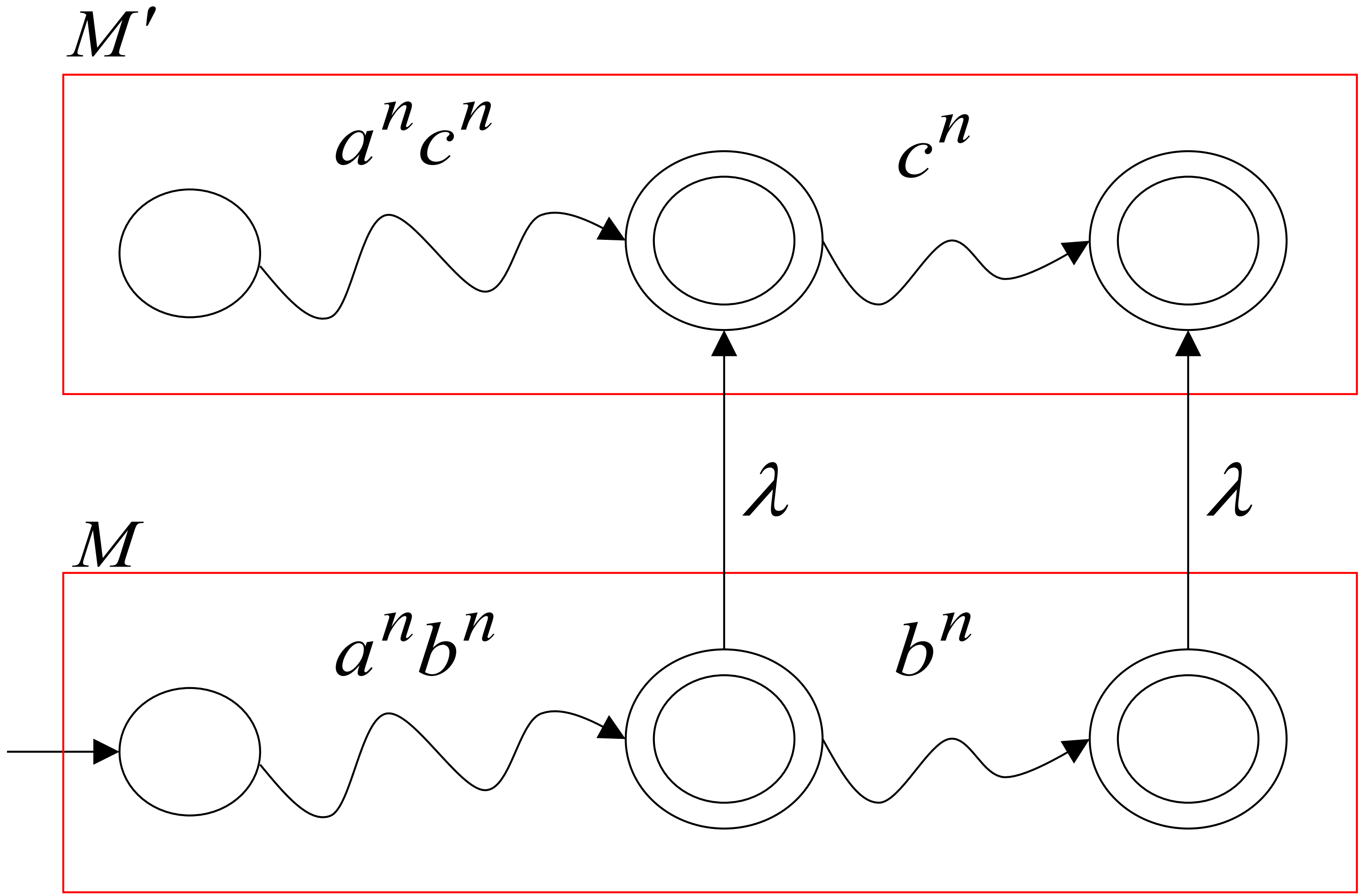
Replace b with c

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



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 it

End of Proof