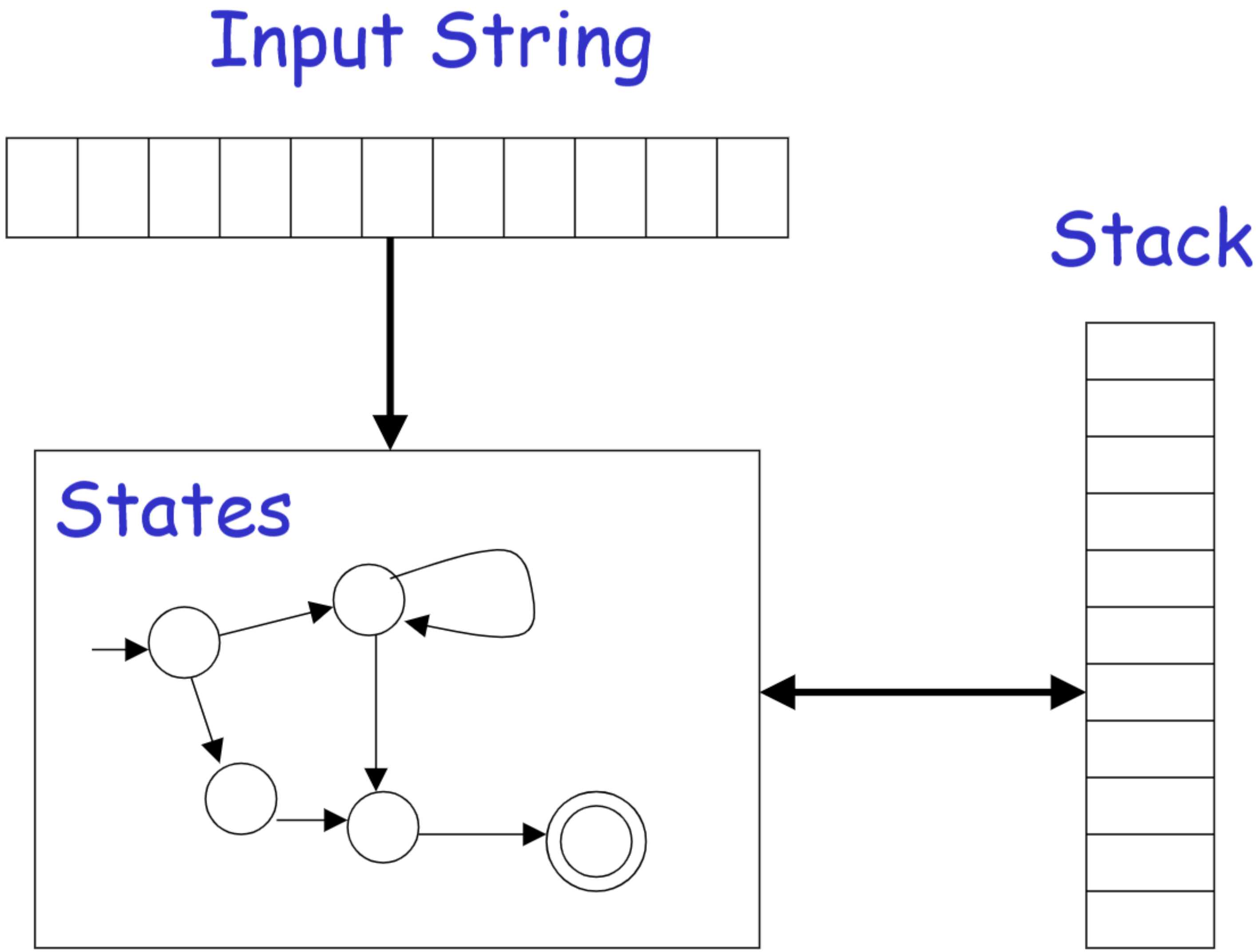
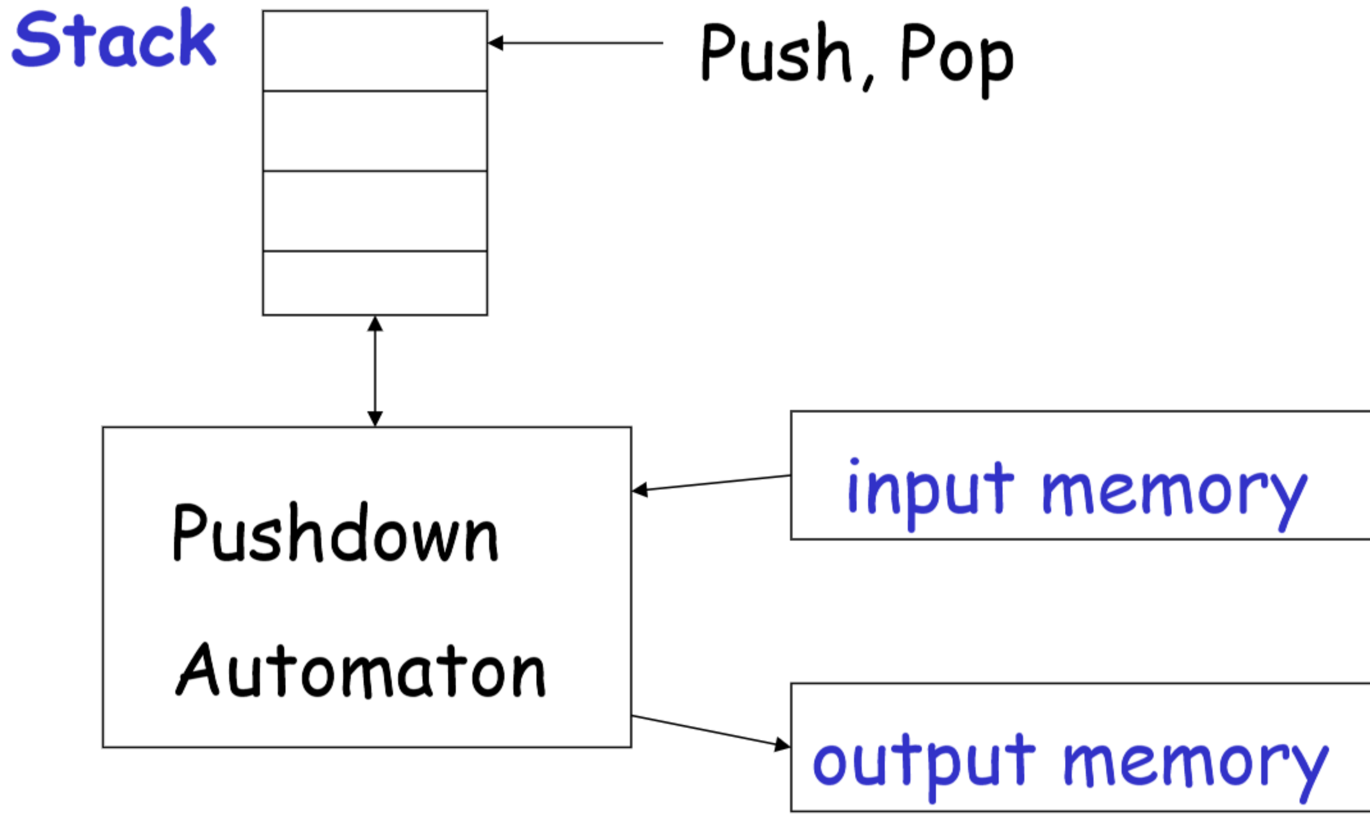
Pushdown Automaton -- PDA

PUSHDOWN AUTOMATA AND PROPERTIES OF CONTEXT-FREE LANGUAGES: Nondeterministic Pushdown Automata, Pushdown Automata and Context–Free Languages, Deterministic Pushdown Automata and Deterministic Context-Free Languages, A Pumping Lemma for Context Free Languages, Closure properties and Decision Algorithms for Context-Free Languages. **Text Book 1:** Chapter 7: 7.1 – 7.3, Chapter 8: 8.1,8.2	**06 Hours**

Pushdown Automaton -- PDA



Pushdown Automaton



Example: Compilers for Programming Languages

(medium computing power)

Definition of PDA: $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$

7 tuple or septuple

A nondeterministic pushdown accepter (npda) is defined by the septuple

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

where

Q is a finite set of internal states of the control unit,

Σ is the input alphabet,

Γ is a finite set of symbols called the **stack alphabet**,

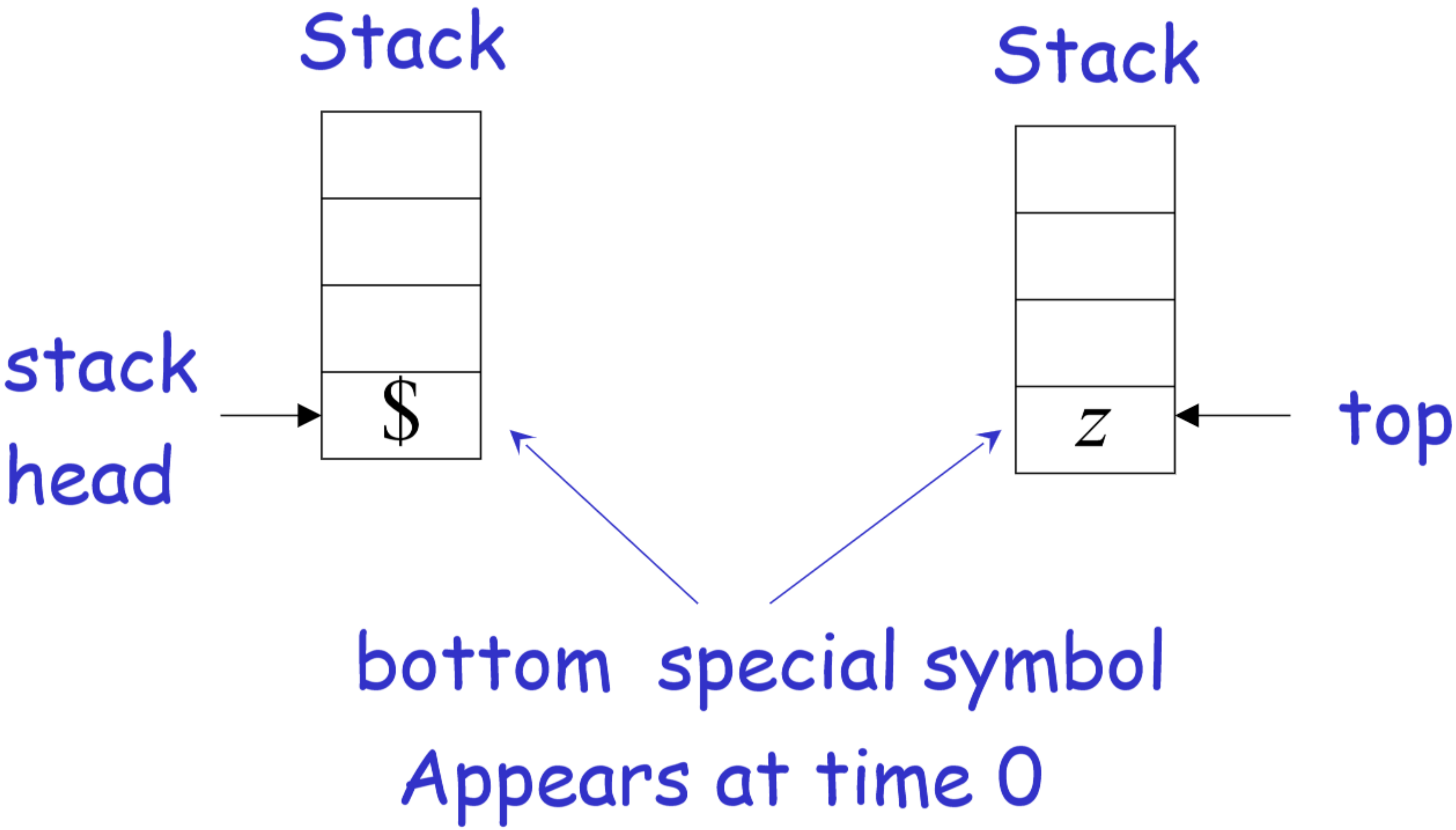
$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow$ set of finite subsets of $Q \times \Gamma^*$ is the transition function,

$q_0 \in Q$ is the initial state of the control unit,

$z \in \Gamma$ is the **stack start symbol**,

$F \subseteq Q$ is the set of final states.

Initial Stack Symbol



Another PDA example

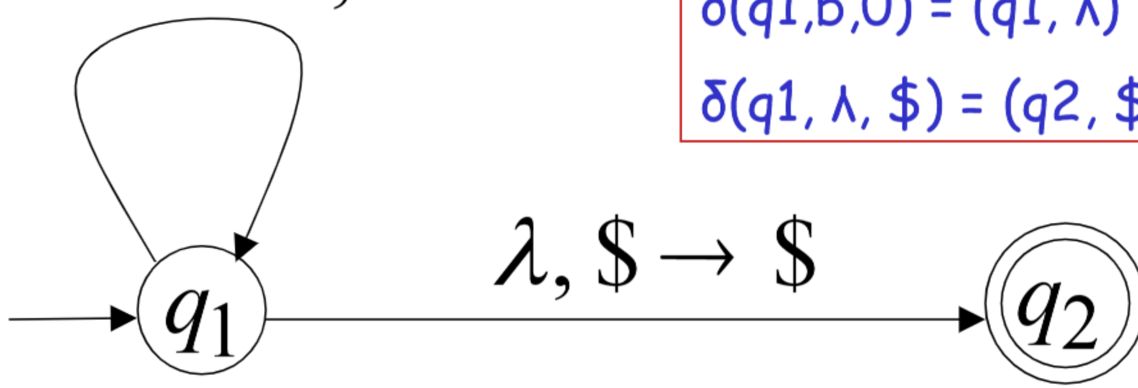
$$L(M) = \{w \in \{a,b\}^* : \; n_a(w) = n_b(w)\}$$

PDA M

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

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

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



δ(q1,a,$) = (q1, 0$)

δ(q1,a,0) = (q1, 00)

δ(q1,b,$) = (q1, 1$)

δ(q1,b,1) = (q1, 11)

δ(q1,a,1) = (q1,Λ)

δ(q1,b,0) = (q1, Λ)

δ(q1, Λ, $) = (q2, $) or (q2, Λ)

Execution Example: Time 0

Input

a	b	b	b	a	a

$\$$

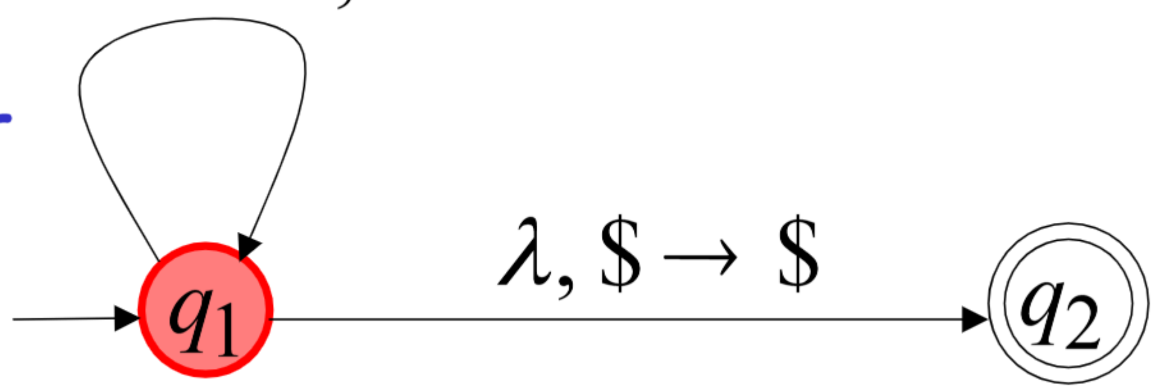
Stack

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

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

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

current state



Time 1

Input

a	b	b	b	a	a

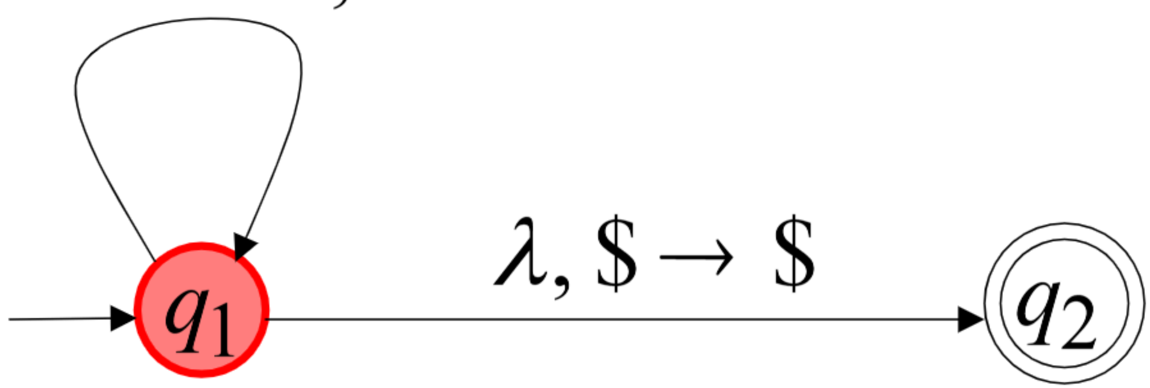
0
\$

Stack

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

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

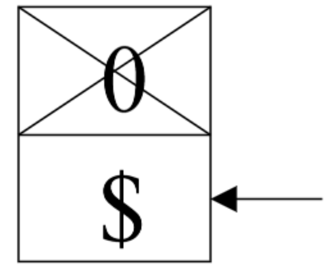
$a, 1 \to \lambda$ $\quad$ $b, 0 \to \lambda$



Time 3

Input

a	b	b	b	a	a

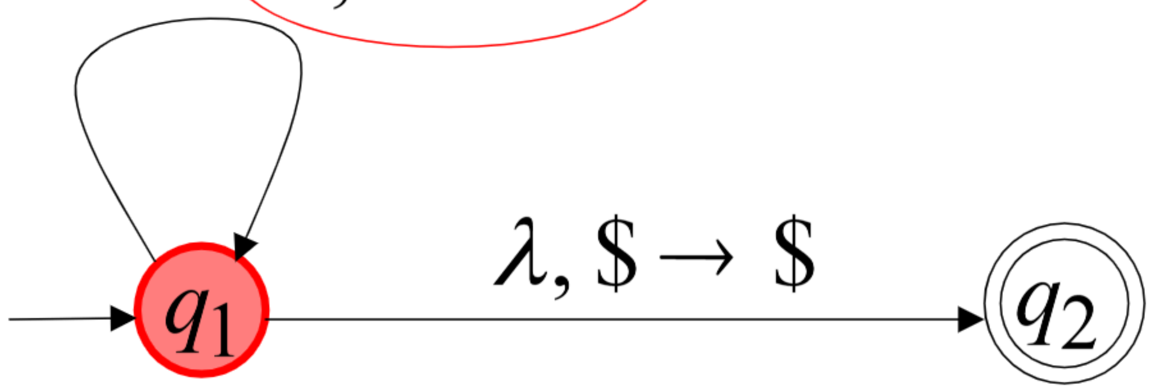


Stack

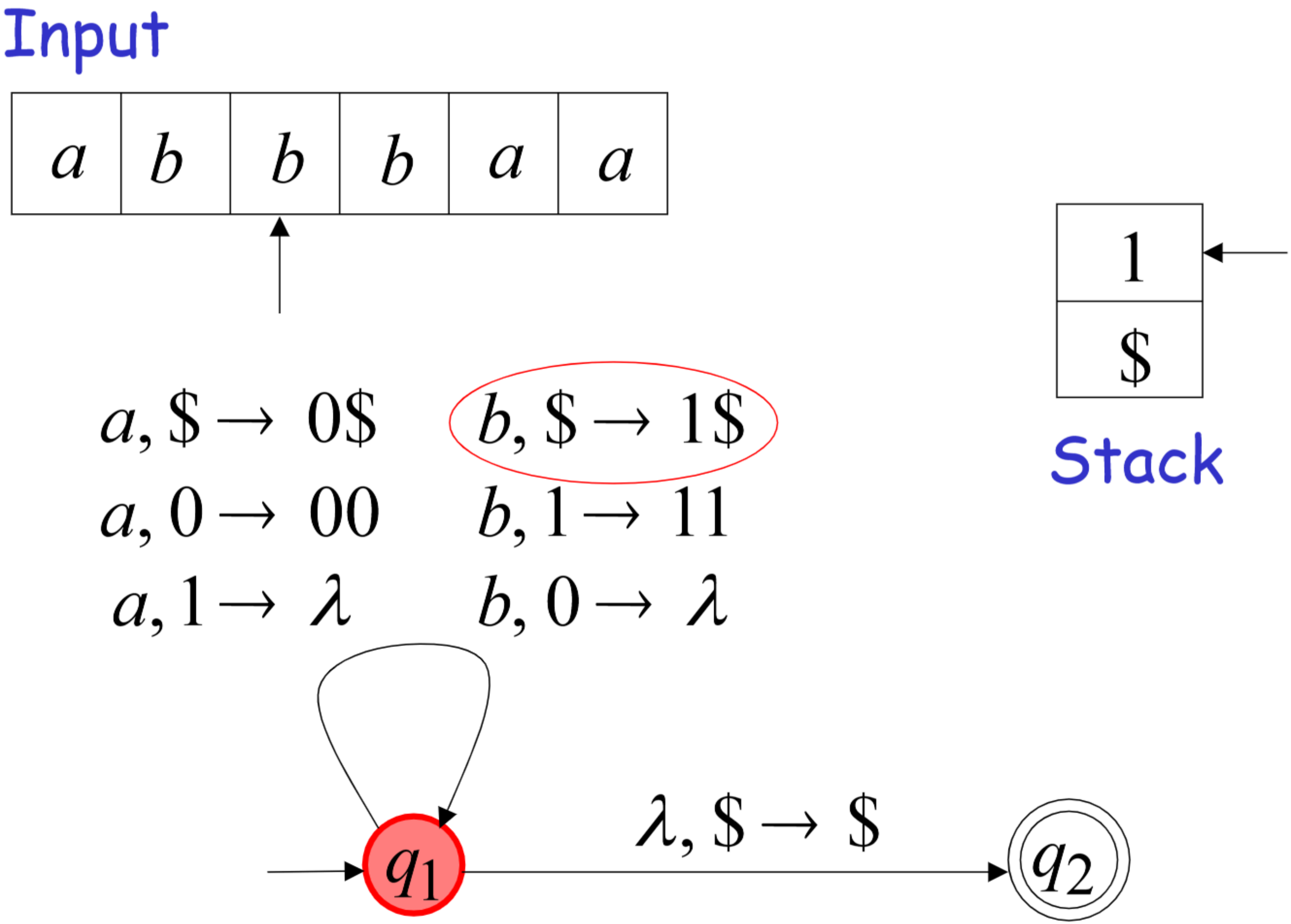
$a, \$ \rightarrow 0\$$ $\quad$ $b, \$ \rightarrow 1\$$

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

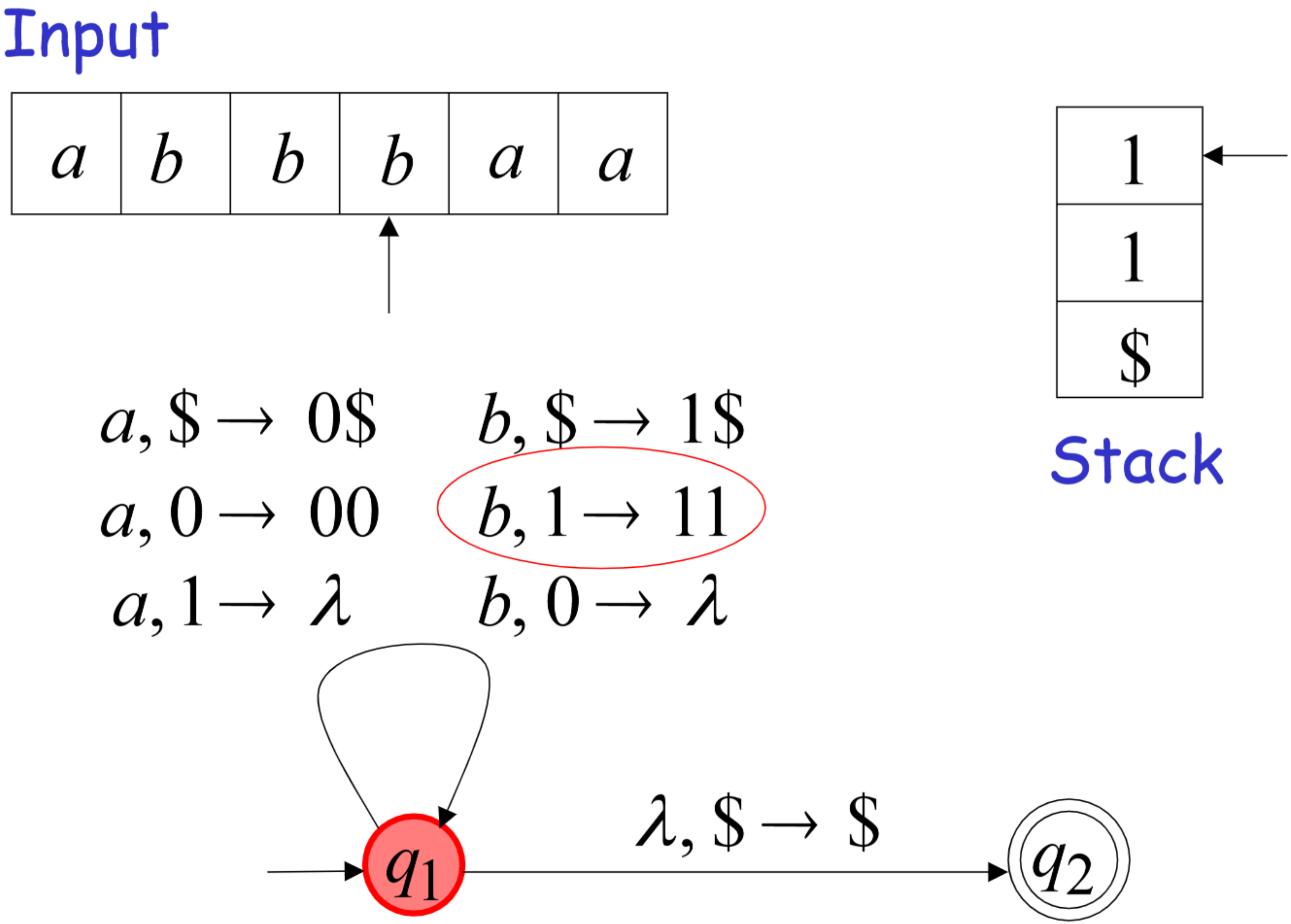
$a, 1 \rightarrow \lambda$ $\quad$ $b, 0 \rightarrow \lambda$



Time 4

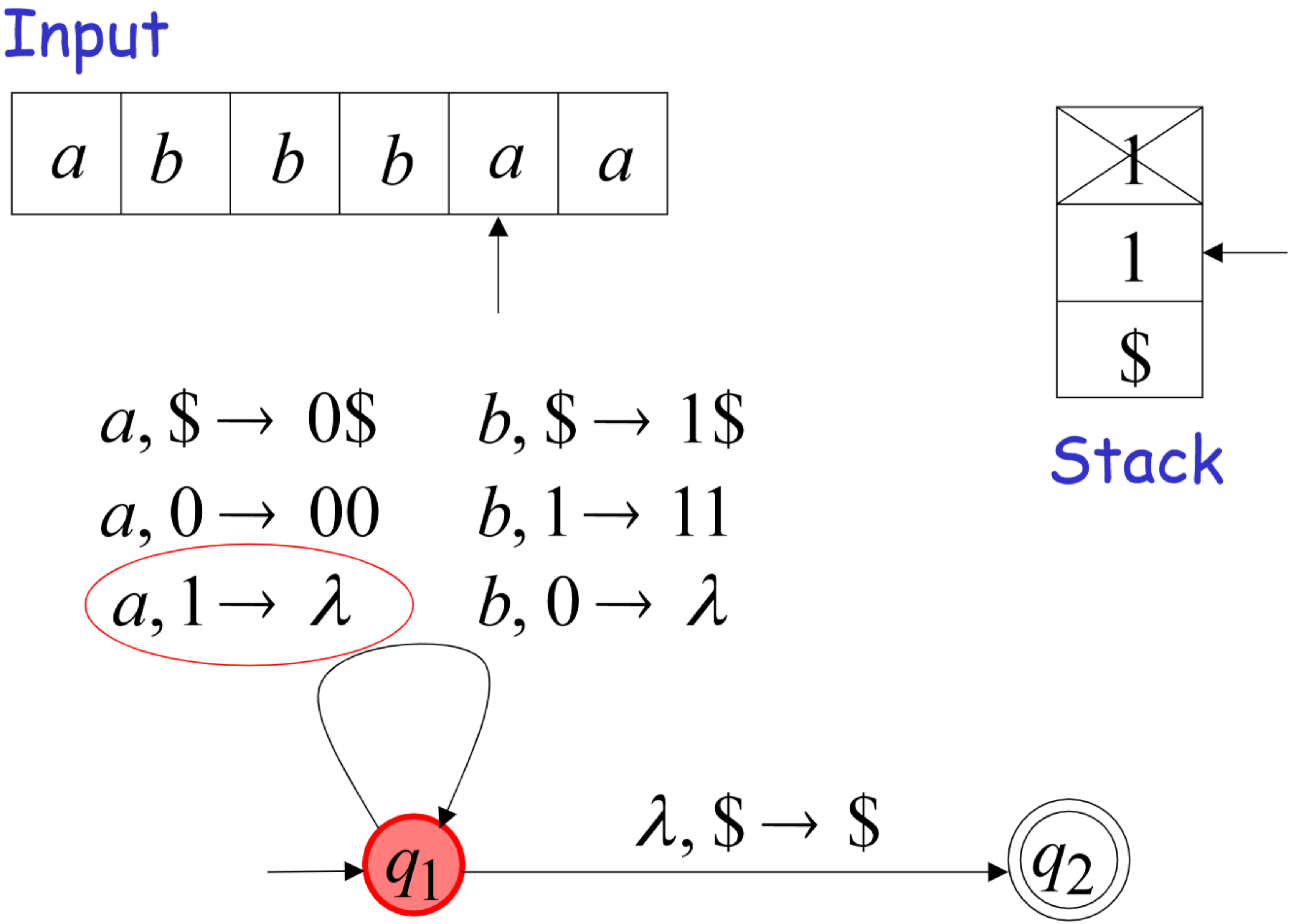

Input
a b b b a a
1
$
Stack
$a, \$ \to 0\$$
$a, 0 \to 00$
$a, 1 \to \lambda$
$b, \$ \to 1\$$
$b, 1 \to 11$
$b, 0 \to \lambda$
$\lambda, \$ \to \$$
q_1
q_2

Time 5

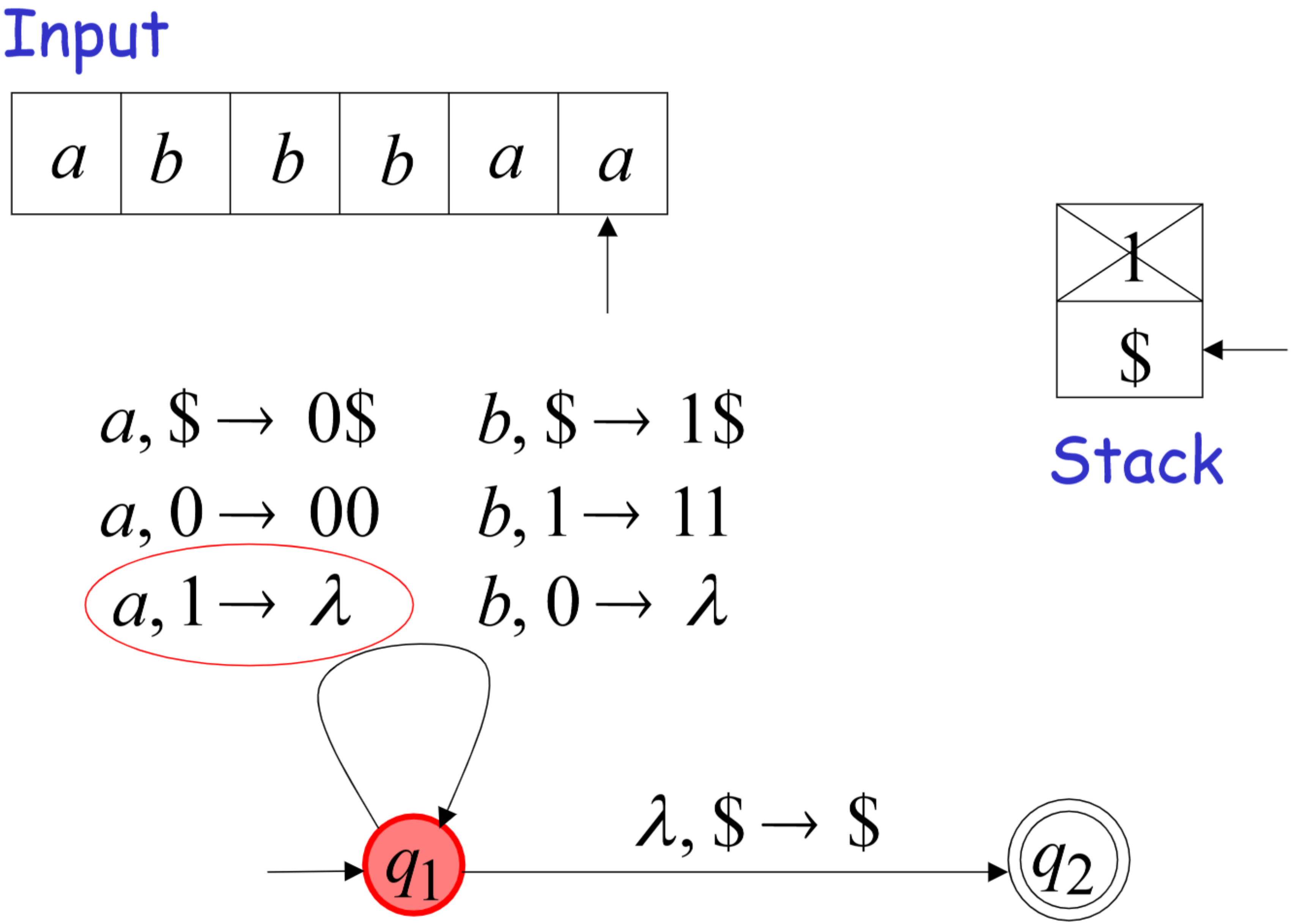


Input
a b b b a a
1
1
$
Stack
$a, \$ \rightarrow 0\$$
$a, 0 \rightarrow 00$
$a, 1 \rightarrow \lambda$
$b, \$ \rightarrow 1\$$
$b, 1 \rightarrow 11$
$b, 0 \rightarrow \lambda$
$\lambda, \$ \rightarrow \$$
q_1
q_2

Time 6



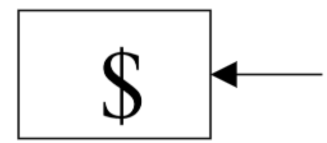
Time 7



Time 8

Input

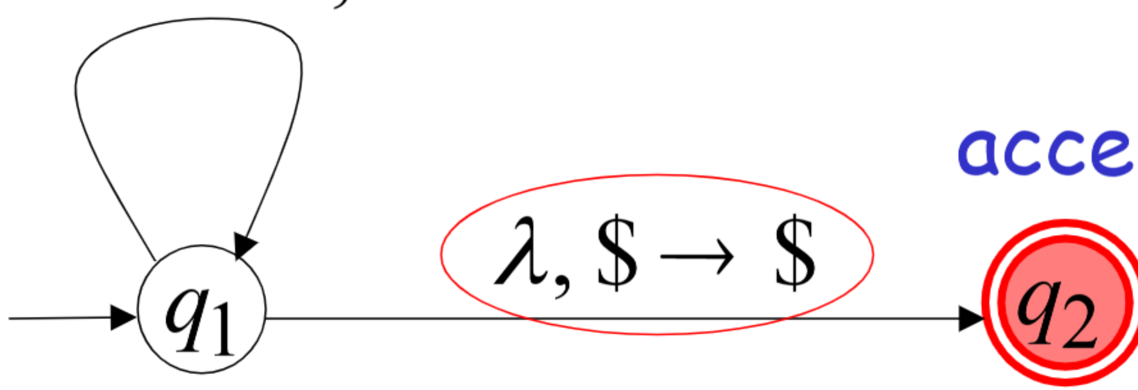
a	b	b	b	a	a



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

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

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

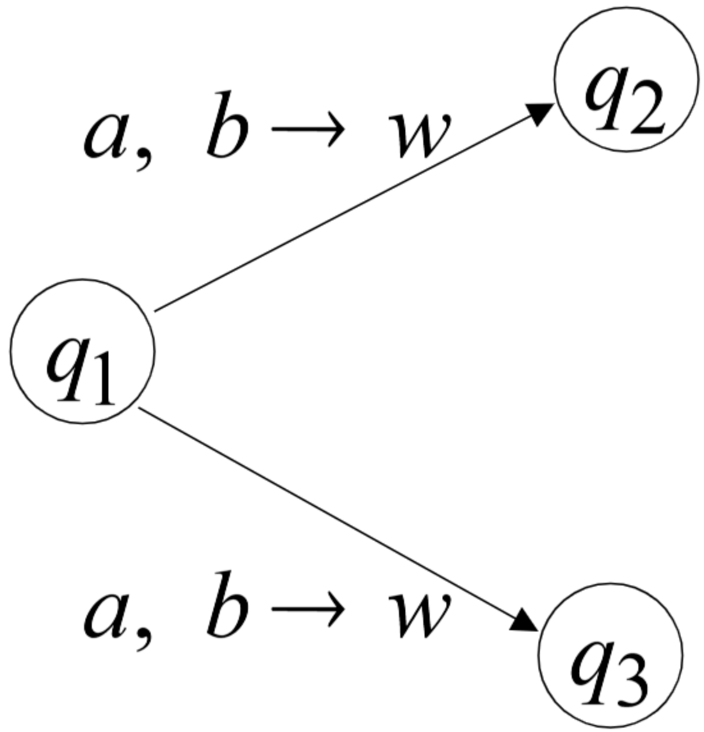


PDAs: Formal Definition

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

Transition function:

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



Transition function:

$$\delta(q_1, a, b) = \{(q_2, w), \ (q_3, w)\}$$

Formal Definition

Pushdown Automaton (PDA)

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

States

Input alphabet

Stack alphabet

Transition function

Initial state

Stack start symbol

Final states

Pushdown Automaton

Note: If $\delta(q_0, a, z) = (q_1, \beta)$ means if NPDA is in state q_0 with input symbol a and z the top symbol on the stack can enter state q_1, replace symbol z by string β.

Language accepted by a PDA

Two types of acceptance: Acceptance by final state or Acceptance by empty stack

Definition: Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ be a NPDA. We define L(M), the language accepted by final state is the set

$$L(M) = \{ w \in \Sigma^* : (q_0, w, z) \mid\!-_M (p, \lambda, u), p \in F, u \in \Gamma^* \}$$

In other words, the language accepted by M is the set of all strings that can put M into a final state.

Definition:

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ be a NPDA. We define $L(M)$, the language accepted by empty stack is the set

$$L(M) = \{ w \in \Sigma^* : (q_0, w, z) \vdash_M (p, \lambda, \lambda), p \in Q\}$$

When acceptance is by empty stack, the set of final states is to be empty set.

Example 1:

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

Q={q0,q1,q2,qf}

Σ={a,b}

Γ={0,1,z}

F={qf}

δ(q0, a, z) = (qf, ʌ) or (q1, 0z)

δ(q0, ʌ , z) = (qf, ʌ)

δ(q1, a, 0) = (q1, 00)

δ(q1, b, 0) = (q2, ʌ)

δ(q2, b, 0) = (q2, ʌ)

δ(q2, ʌ ,z) = (qf, z) Or (qf, ʌ)

Instantaneous Description: check with the string "aabb"

(q0, aabb, z) ⊢ (q1, abb, 0z) ⊢ (q1, bb, 00z) ⊢(q1, bb, 00z) ⊢(q2, b, 0z) ⊢

(q2, ʌ, z) ⊢ (qf, ʌ, z) or (qf, ʌ, ʌ)

Example 1:

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

Consider an npda with

$$\begin{aligned} Q &= \{q_0, q_1, q_2, q_3\}, \\ \Sigma &= \{a, b\}, \\ \Gamma &= \{0, 1\}, \\ z &= 0, \\ F &= \{q_3\}, \end{aligned}$$

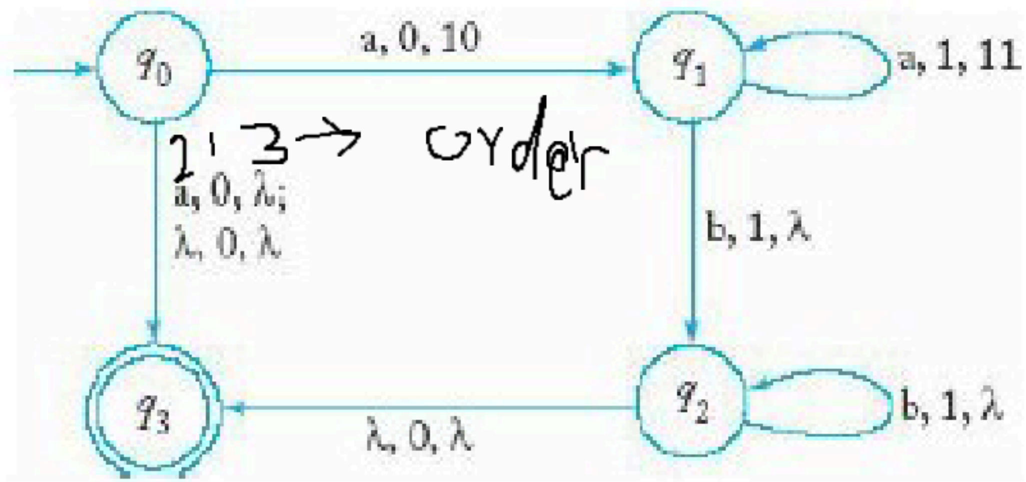
with initial state q_0 and

$$\begin{aligned} \delta(q_0, a, 0) &= \{(q_1, 10), (q_3, \lambda)\}, \\ \delta(q_0, \lambda, 0) &= \{(q_3, \lambda)\}, \\ \delta(q_1, a, 1) &= \{(q_1, 11)\}, \\ \delta(q_1, b, 1) &= \{(q_2, \lambda)\}, \\ \delta(q_2, b, 1) &= \{(q_2, \lambda)\}, \\ \delta(q_2, \lambda, 0) &= \{(q_3, \lambda)\}. \end{aligned}$$

What can we say about the action of this automaton?

The npda in Example 7.2 is represented by the transition graph in Figure 7.2.

Figure 7.2



Instantaneous Description:

The triplet (q, w, u) where *q* is the state of the control unit, *w* is the unread part of the input string, and *u* is the stack contents (with the leftmost symbol indicating the top of the stack), is called an **instantaneous description** of a pushdown automaton.

- A move from one instantaneous description to another will be denoted by the symbol |- thus

$$(q_1, aw, bx) \vdash (q_2, w, yx)$$

is possible if and only if

$$(q_2, y) \in \delta(q_1, a, b).$$

- Example: Instantaneous description for language $L = \{a^n b^n : n \geq 1\}$, $Q = \{q_0, q_1, q_2, q_3\}$

Take string w = aabb

$(q_0, aabb, Z) \vdash (q_1, abb, aZ) \vdash (q_1, bb, aaZ) \vdash$
$(q_2, b, aZ) \vdash (q_2, \Lambda, Z) \vdash (q_3, \Lambda, Z)$

Construct an npda for the language

Use $Q = \{ q_0, q_f)$

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

$\delta(q_0, a, z) = (q_0, az)$

$\delta(q_0, b, z) = (q_0, bz)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, a, b) = (q_0, \lambda)$

$\delta(q_0, b, a) = (q_0, \lambda)$

$\delta(q_0, \lambda, z) = (q_f, \lambda)$

Construct an npda for the language

Use $Q = \{ q_o, q_f \}$

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

$\delta(q_o, a, z) = (q_0, 0z)$

$\delta(q_o, b, z) = (q_0, 1z)$

$\delta(q_o, a, 0) = (q_0, 00)$

$\delta(q_o, b, 1) = (q_0, 11)$

$\delta(q_o, a, 1) = (q_0, \lambda)$

$\delta(q_o, b, 0) = (q_0, \lambda)$

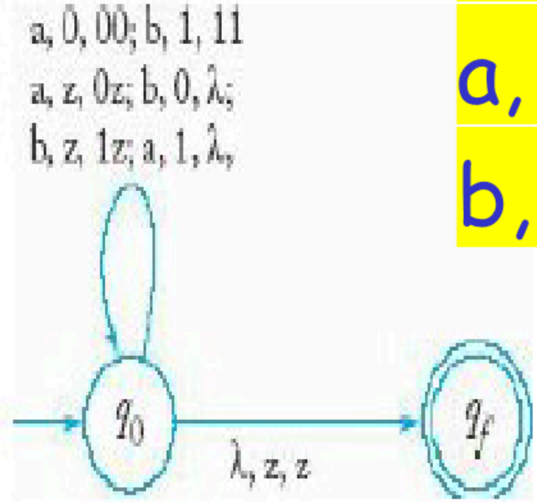
$\delta(q_o, \lambda, z) = (q_f, \lambda)$ (shows empty stack)

Or $\delta(q_o, \lambda, z) = (q_f, z)$ (not an empty stack)

- When a is read insert 0 in the stack
- When b is read insert 1 in the stack
- Stack top symbol is z

Construct an npda for the language

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



a, z | 0z & b, z | 1z

a, 0 | 00 & b,1 | 11

b, 0 | λ & a, 1 | λ

In processing the string *baab*, the npda makes the moves

$$(q_0, baab, z) \vdash (q_0, aab, 1z) \vdash (q_0, ab, z)$$
$$\vdash (q_0, b, 0z) \vdash (q_0, \lambda, z) \vdash (q_f, \lambda, z)$$

- To construct an npda for accepting the language $L = \{ ww^R : w \in \{a, b\}^+ \}$

problem is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$, where

$$Q = \{q_0, q_1, q_2\},$$

$$\Sigma = \{a, b\},$$

$$\Gamma = \{a, b, z\},$$

$$F = \{q_2\}.$$

$$\delta(q_0, a, a) = \{(q_0, aa)\},$$
$$\delta(q_0, b, a) = \{(q_0, ba)\},$$
$$\delta(q_0, a, b) = \{(q_0, ab)\},$$
$$\delta(q_0, b, b) = \{(q_0, bb)\},$$
$$\delta(q_0, a, z) = \{(q_0, az)\},$$
$$\delta(q_0, b, z) = \{(q_0, bz)\},$$

a set to guess the middle of the string, where the npda switches from state q_0 to q_1

$$\delta(q_0, \lambda, a) = \{(q_1, a)\},$$
$$\delta(q_0, \lambda, b) = \{(q_1, b)\},$$

a set to match w^R against the contents of the stack,

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

and finally

$$\delta(q_1, \lambda, z) = \{(q_2, z)\},$$

The sequence of moves in accepting $abba$ is

$$(q_0, abba, z) \vdash (q_0, bba, az) \vdash (q_0, ba, baz) \vdash (q_1, ba, baz) \vdash (q_1, a, az) \vdash (q_1, \lambda, z) \vdash (q_2, z).$$

Try for the language $L = \{w\ c\ w^R \mid w \in (a,b)^*\}$

$Q = \{q0, q1, q2\}$, stack top symbol be z, $\tau = \{a, b, z\}$

- Tyr these examples....

1. $L = \{ w \mid w \in (a, b)^* \text{ and } n_a(w) > n_b(w)\}$, $Q = q0, q1$, Γ - $\{a,b, z\}$

2. $L = \{ w \mid w \in (a, b)^* \text{ and } n_a(w) < n_b(w)\}$ $Q = q0, q1$, Γ - $\{a,b, z\}$

3. $L = \{ a^n b^{2n} \mid n >= 1\}$ $Q = q0, q1, q2$, Γ - $\{a, z\}$

4. Check for balanced parenthesis Where $\Sigma = \{ (,),[,] \}$

   check for [()()([ ])] $Q = q0, q1$, Γ - $\{(,[, z\}$

Pushdown Automata for Context-Free Languages

Construct a pda that accepts the language generated by a grammar with productions

$$S \to aSbb|a.$$

We first transform the grammar into Greibach normal form, changing the productions to

$$\begin{aligned} S &\to aSA|a, \\ A &\to bB, \\ B &\to b. \end{aligned}$$

The corresponding automaton will have three states $\{q_0, q_1, q_2\}$, with initial state q_0 and final state q_2.
First, the start symbol S is put on the stack by

$$\delta(q_0, \lambda, z) = \{(q_1, Sz)\}.$$

$$\delta(q_1, a, S) = \{(q_1, SA), (q_1, \lambda)\}.$$

In an analogous manner, the other productions give

$$\delta(q_1, b, A) = \{(q_1, B)\},$$
$$\delta(q_1, b, B) = \{(q_1, \lambda)\}.$$

The appearance of the stack start symbol on top of the stack signals the completion of the derivation and the pda is put into its final state by

$$\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}.$$

Consider the grammar

$$
\begin{aligned}
S &\to aA, \\
A &\to aABC \mid bB \mid a, \\
B &\to b, \\
C &\to c.
\end{aligned}
$$

Since the grammar is already in Greibach normal form, we can use the construction in the previous theorem immediately. In addition to rules

$$\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$$

and

$$\delta(q_1, \lambda, z) = \{(q_f, z)\},$$

the pda will also have transition rules

$$\begin{aligned}
\delta(q_1, a, S) &= \{(q_1, A)\}, \\
\delta(q_1, a, A) &= \{(q_1, ABC), (q_1, \lambda)\}, \\
\delta(q_1, b, A) &= \{(q_1, B)\}, \\
\delta(q_1, b, B) &= \{(q_1, \lambda)\}, \\
\delta(q_1, c, C) &= \{(q_1, \lambda)\}.
\end{aligned}$$

The sequence of moves made by M in processing $aaabc$ is

$$\begin{aligned}
(q_0, aaabc, z) &\vdash (q_1, aaabc, Sz) \\
&\vdash (q_1, aabc, Az) \\
&\vdash (q_1, abc, ABCz) \\
&\vdash (q_1, bc, BCz) \\
&\vdash (q_1, c, Cz) \\
&\vdash (q_1, \lambda, z) \\
&\vdash (q_f, \lambda, z).
\end{aligned}$$

$$S \Rightarrow aA \Rightarrow aaABC \Rightarrow aaaBC \Rightarrow aaabC \Rightarrow aaabc.$$

CFG to PDA

Example 2:

- Obtain a PDA to accept the language

$L = \{a^n b^n \mid n>=1\}$

$S \rightarrow aSb$

$S \rightarrow ab$

Step 1: Convert into GNF

$S \rightarrow aSB \mid aB$

$B \rightarrow b$

Step 2:

Find PDA…

Deterministic Pushdown Automata and Deterministic Context-Free Languages

→ A **deterministic pushdown accepter (dpda)** is a pushdown automaton that never has a choice in its move

• A pushdown automaton $M = (Q, \Sigma, \Gamma, \delta, q0, z, F)$ is said to be deterministic if it is an automaton as defined that, for every $q \in Q$, $a \Sigma \cup \{\lambda\}$ and $b \in \Gamma$,

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

e language

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

a deterministic context-free language. The pda $M = (\{q_0, q_1, q_2\}, \{a, b\}, \{0, 1\}, \delta, q_0, 0, \{q_0\})$ with

$$\begin{aligned}
\delta(q_0, a, 0) &= \{(q_1, 10)\}, \\
\delta(q_1, a, 1) &= \{(q_1, 11)\}, \\
\delta(q_1, b, 1) &= \{(q_2, \lambda)\}, \\
\delta(q_2, b, 1) &= \{(q_2, \lambda)\}, \\
\delta(q_2, \lambda, 0) &= \{(q_0, \lambda)\}
\end{aligned}$$

cepts the given language. It satisfies the conditions of Definition 7.3 and is therefore deterministic.

- # Another Example

The language $L = \{a^n b^n | n \geq 1\}$ is a deterministic CFL.

Proof: The PDA $M = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \{0, 1\}, \delta, q_0, Z, \{q_3\})$ with

$$\delta(q_0, a, Z) = \{(q_1, 1Z)\},$$

$$\delta(q_1, a, 1) = \{(q_1, 11)\},$$

$$\delta(q_1, b, 1) = \{(q_2, \lambda)\},$$

$$\delta(q_2, b, 1) = \{(q_2, \lambda)\},$$

$$\delta(q_2, \lambda, Z) = \{(q_3, \lambda)\}$$

accepts the given language.

It satisfies the conditions for being deterministic.

Our previous PDA for $\{ww^R | w \in \Sigma^+\}, \Sigma = \{a, b\}$ is nondeterministic.

It contains these transitions:

$\delta(q_0, a, a) = \{(q_0, aa)\}$

$\delta(q_0, \lambda, a) = \{(q_1, a)\}$

$$\delta(q_0, a, a) = \{(q_0, aa)\},$$
$$\delta(q_0, b, a) = \{(q_0, ba)\},$$
$$\delta(q_0, a, b) = \{(q_0, ab)\},$$
$$\delta(q_0, b, b) = \{(q_0, bb)\},$$
$$\delta(q_0, a, z) = \{(q_0, az)\},$$
$$\delta(q_0, b, z) = \{(q_0, bz)\},$$

To construct an npda for accepting the language
$L = \{ ww^R : w \in \{a, b\}^+ \}$

a set to guess the middle of the string, where the npda switches from state q_0 to q_1

$$\delta(q_0, \lambda, a) = \{(q_1, a)\},$$
$$\delta(q_0, \lambda, b) = \{(q_1, b)\},$$

a set to match w^R against the contents of the stack,

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

and finally

$$\delta(q_1, \lambda, z) = \{(q_2, z)\},$$

Try…

Check DPDA or NPDA?

$L = \{x \in \{0, 1\}^* \mid n_0(x) > n_1(x)\}$