$$(q_1 z_2 q_2) \to b(q_1 z_2 q_1)(q_1 z_2 q_2)$$
$$(q_1 z_2 q_2) \to b(q_1 z_2 q_2)(q_2 z_2 q_2).$$

Example 7.28 Convert the following PDA into an equivalent CFG.

$$\delta(q_0, a, z_0) \to (q_1, z_1 z_0)$$
$$\delta(q_0, b, z_0) \to (q_1, z_2 z_0)$$
$$\delta(q_1, a, z_1) \to (q_1, z_1 z_1)$$
$$\delta(q_1, b, z_1) \to (q_1, \lambda)$$
$$\delta(q_1, b, z_2) \to (q_1, z_2 z_2)$$
$$\delta(q_1, a, z_2) \to (q_1, \lambda)$$
$$\delta(q_1, \lambda, z_0) \to (q_1, \lambda) // acceptedbytheemptystack$$

Solution: The PDA contains two states: q_0 and q_1. The following productions are added to the CFG [according to rule (1)].

$$S \to [q_0 z_0 q_0]/[q_0 z_0 q_1]$$

Transitional functions (iv), (vi), and (vii) are in the form $\delta(q, a, Y) \to (r, \in)$. Thus, the following three productions are added to the CFG [according to rule (2)]

$$(q_1 z_1 q_1) \to b \text{ // From production (iii)}$$
$$(q_1 z_2 q_2) \to a \text{ // From production (iv)}$$
$$(q_1 z_0 q_1) \to \varepsilon \text{ // From production (vii)}$$

For the remaining transitional functions, the productions are as follows:

$$\delta(q_0, a, z_0) \to (q_1, z_1 z_0)$$
$$(q_0 z_0 q_0) \to a(q_0 z_1 q_0)(q_0 z_0 q_0)$$

$$(q_0 z_0 q_0) \to a(q_0 z_1 q_1)(q_1 z_0 q_0)$$
$$(q_0 z_0 q_1) \to a(q_0 z_1 q_0)(q_0 z_0 q_1)$$
$$(q_0 z_0 q_1) \to a(q_0 z_1 q_1)(q_1 z_0 q_1)$$
$$\delta(q0, b, z0) \to (q1, z2z0)$$
$$(q_0 z_0 q_0) \to b(q_0 z_2 q_0)(q_0 z_0 q_0)$$
$$(q_0 z_0 q_0) \to b(q_0 z_2 q_1)(q_1 z_0 q_0)$$
$$(q_0 z_0 q_1) \to b(q_0 z_2 q_0)(q_0 z_0 q_1)$$
$$(q_0 z_0 q_1) \to b(q_0 z_2 q_1)(q_1 z_0 q_1)$$
$$\delta(q_1, a, z_1) \to (q_1, z_1 z_1)$$
$$(q_1 z_1 q_0) \to a(q_1 z_1 q_0)(q_0 z_1 q_0)$$
$$(q_1 z_1 q_0) \to a(q_1 z_1 q_1)(q_1 z_1 q_0)$$
$$(q_1 z_1 q_1) \to a(q_1 z_1 q_0)(q_0 z_1 q_1)$$
$$(q_1 z_1 q_1) \to a(q_1 z_1 q_1)(q_1 z_1 q_1)$$

$$\begin{aligned}
&\delta(q_1, b, z_2) \to (q_1, z_2z_2)\\
&\quad (q_1z_2q_0) \to b(q_1z_2q_0)(q_0z_2q_0)\\
&\quad (q_1z_2q_0) \to b(q_1z_2q_1)(q_1z_2q_0)\\
&\quad (q_1z_2q_1) \to b(q_1z_2q_0)(q_0z_2q_1)\\
&\quad (q_1z_2q_1) \to b(q_1z_2q_1)(q_1z_2q_1)
\end{aligned}$$

7.6 Graphical Notation for PDA

The mathematical notation for a PDA is $(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$. In a PDA, the transitional function δ consists of three touples: fi rst is a present state, second is the present input, and the third is the stack top symbol, which generates one next state and the stack symbol(s) if a symbol is pushed into the stack or $\in$, if the top most symbol is popped from the stack. In the graphical notation of the PDA, there are states. Among them a circle with an arrow indicates a beginning state and a state

with double circle indicates a final state.The state transitions are denoted byarrows. The labels of the state transitions consists of input symbol, previous stack top symbol (at $t_i - 1$) and the current stack top symbol (at t_i) which is added after the transitions or null symbol (if a symbol is popped).

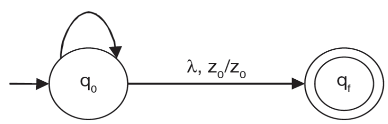
Example 7.29 Construct a PDA with a graphical notation to accept $L = (a, b)*$ with equal number of 'a' and 'b', i.e., $n_a(L) = n_b(L)$ by the fi nal state.

Solution: At the beginning of transition, the PDA is in state q_0 with stack z_0. The string may start with 'a' or 'b'. If the string starts with 'a', one z_1 is pushed into the stack. If the string starts with 'b', one z_2 is pushed into the stack. If 'b' is traversed after 'a', and the stack top is z_1, that stack top is popped. If 'a' is traversed after 'b', and the stack top is z_2, that stack top is popped. If 'a' is traversed after 'a', and the stack top is z_1, one $'z_1'$ is pushed into the stack. If 'b' is traversed after 'b', and the stack top is z_2, one $'z_2'$ is pushed into the stack.

The PDA in graphical notation is as follows:

$$a, z_0/z_1z_0$$
$$b, z_0/z_2z_0$$
$$a, z_1/z_1z_1$$
$$b, z_2/z_2z_2$$
$$b, z_1/\lambda$$
$$a, z_2/\lambda$$

1 picture

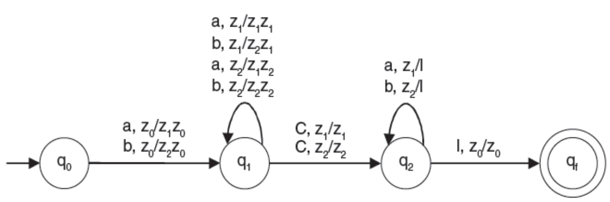


Example 7.30 `Construct a PDA with a graphical notation to accept the language L = WCWR, where W` $\in (a, b)+$ `and WR is the reverse of W by the fi nal state.`

z_1 or z_2, another z_1 or z_2 is pushed into the stack, respectively. In state q_1, if it gets the input 'b' with the stack top z_2 or z_1, another z_2 or z_1 is pushed into the stack, respectively. C is the symbol which differentiates W with WR. Before C, W is traversed. So, at the time of traversing C, the stack top symbol may be z_1 or z_2. In state q_1, if the PDA gets C as input and the stack top z_1 or z_2, no operation is done on the stack, but only the state is changed from q_1 to q_2. After C, the string WR is traversed. If the machine gets 'a' as input, the stack top must be z_1. And that z_1 is popped from the stack top. If the machine gets 'b' as input, the stack top must be z_2. And that z_2 is popped from the stack top. The state is not changed. By this process, the whole string WCWR is traversed. In state q_2, if the machine gets no input but stack top z_0, the machine goes to its fi nal state q_f.

The graphical notation for the PDA is as follows:

2 picture



7.7 Two-stack PDA

Finite automata recognize regular languages such as $\{an|n \geq 0\}$. Adding one stack to a fi nite automata, it becomes PDA which can recognize context-free language $\{a^n b^n, n \geq 0\}$. In the case of context-sensitive language such as $\{a^n b^n c^n, n \geq 0\}$, the PDA is helpless because of only one auxiliary storage. Now the question arises-if more than one stack is added in the form of auxiliary storage with PDA, does its power increase or not.

From this question, the concept of a two-stack PDA has come. Not only two stacks, but more than two stacks can be added to a PDA.

A PDA can be deterministic or non-deterministic, but two-stack PDA is deterministic and it accepts all context-free languages, which may be deterministic or non-deterministic, with context-sensitive language such as $\{a^n b^n c^n, n \geq 0\}$. In the Turing machine chapter, we shall learn that two-stack PDA is equivalent to the Turing machine. There, we shall also learn a theorem called the Minsky's theorem.

Defi nition: A two-stack PDA consists of a 9-tuple

$$M = (Q, \Sigma, \Gamma, \Gamma', \delta, q_0, z_1, z_2, F)$$

where

Q denotes a fi nite set of states.

Σ denotes a fi nite set of input symbols.

Γ denotes a fi nite set of fi rst stack symbols.

Γ' denotes a fi nite set of second stack symbols

δ denotes the transitional functions.

q_0 is the initial state of PDA $[q_0 \in Q]$.

z_1 is the stack bottom symbol of stack 1.

z_2 is the stack bottom symbol of stack 2.

F is the fi nal state of PDA.

In PDA, the transitional function δ is in the form

$$Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \times \Gamma' \to (Q, \Gamma, \Gamma')$$

Example 7.31 Construct a two-stack PDA for the language $L = \{anbncn, n \geq 0\}$.

Solution: While scanning 'a', push X into stack 1. While scanning 'b', push 'Y' into stack 2. While scanning 'c' with stack top X in 1 and stack top Y in 2, pop X and Y from stack 1 and 2, respectively.
The transitional functions are as follows:

$$\begin{aligned}
&\delta(q_0, \lambda, z_1, z_2) \to (q_f, z_1, z_2)\\
&\delta(q_0, a, z_1, z_2) \to (q_0, Xz_1, z_2)\\
&\delta(q_0, a, X, z_2) \to (q_0, XX, z_2)\\
&\delta(q_0, b, X, z_2) \to (q_0, X, Yz_2)\\
&\delta(q_0, b, X, Y) \to (q_0, X, YY)\\
&\delta(q_0, c, X, Y) \to (q_1, \lambda, \lambda)\\
&\delta(q_1, c, X, Y) \to (q_1, \lambda, \lambda)\\
&\delta(q_1, \lambda, z_1, z_2) \to (q_f, , z_1, z_2) \text{ // accepted by the fi nal state.}
\end{aligned}$$

Theorem 7.2: Intersection of RE and CFL is CFL.

Proof: Let L is a CFL accepted by a PDA $M_1 = \{Q_1, \Sigma, \Gamma_1, \delta_1, q_01, z_0, F_1\}$ and R be a regular expression accepted by a FA $M_2 = \{Q_2, \Sigma, \delta_2, q_02, F_2\}$. A new PDA $M_3 = \{Q, \Sigma, \Gamma, \delta, q_0, z_0, F\}$ is designed which performs computation of M_1 and M_2 in parallel and accepts a string accepted by both M_1 and M_2. M_3 is designed as follows.

$$\begin{aligned}
&Q = Q_1 \times Q_2 \text{ [Cartesian product]}\\
&\Sigma = \Sigma\\
&\Gamma = \Gamma_1\\
&F = F_1 \times F_2
\end{aligned}$$

$\delta : [(S_1, S_2), i/p, Z) \to (q_1, q_2, Z')$ if $\delta_1(S_1, i/p, Z) \to (q_1, Z')$ and $\delta_2(S_2, i/p) \to q_2$

[Z and $Z' \in \Gamma$]

$$q_0 = q_0 1 \cup q_0 2$$

The transitional function of M_3 keeps track of transaction from S_1 to q_1 in PDA M_1 and S_2 to q_2 in FA M_2 for same input alphabet. Thus a string W accepted by M_3 if and only if, it is accepted by both M_1 and M_2. Therefore $W \in L(M_1) \cap L(M_2)$. As W is accepted by a PDA, W is a CFL.

3 picture

What We Have Learned So Far

1. Pushdown automata (in short PDA) is the machine format of context-free language.

2. The mechanical diagram of a pushdown automata contains the input tape, reading head, fi nite control, and a stack.

3. A pushdown automata consists of a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ where Q, Σ, q_0, and F have their original meaning, Γ is fi nite set of stack symbols, and z_0 is the stack bottom symbol.

4. In a PDA, the transitional function δ is in the form $Q \times (\Sigma \cup \lambda) \times \Gamma \rightarrow (Q, \Gamma)$.