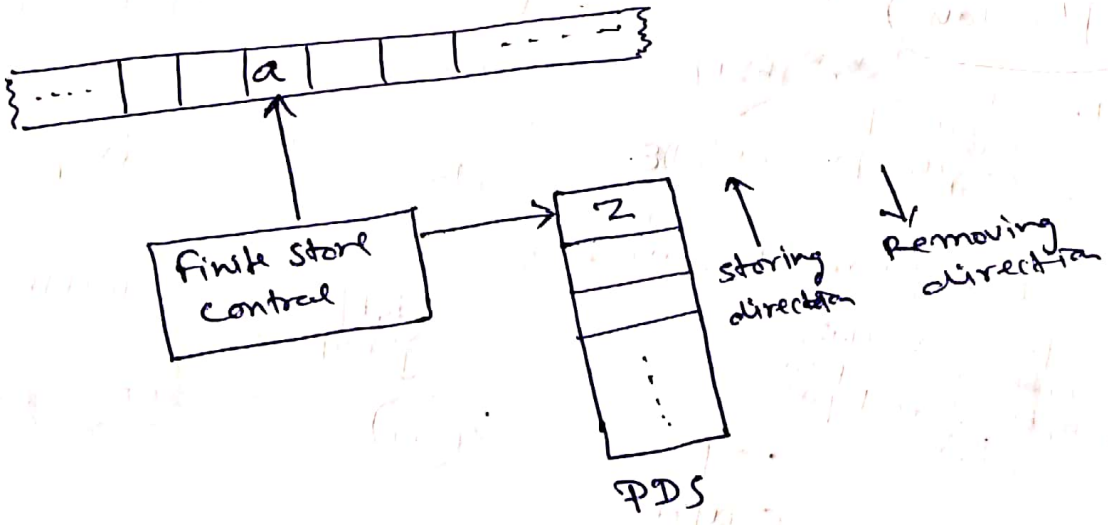
Vimal Singh
CSE Department

Push Down Automata (PDA): Description and Definition, Instantaneous Description, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic PDA. Equivalence of PDA and CFG, CFG to PDA and PDA to CFG, Two stack PDA.

Pushdown Automata Definition:- (PDA)

PDA has a read-only input tape, an input alphabet, a finite state control, a set of final states and an initial state as in case of FA. In addition to these, it has a stack called pushdown store (PDS). It is a read-write pushdown store as we add elements to PDS or remove elements from PDS.



Model of PDA

Transition Function of PDA:

for DPDA: $\delta : Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma^*$

for NDPDA: $\delta : Q \times \Sigma \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$

Difinition: A PDA consists of

(i) a finite non-empty set of state denoted by Q.
(ii) a finite non-empty set of input symbol denoted by Σ.
(iii) a finite non-empty set of pushdown symbol denoted by Γ.
(iv) initial state q_0.
(v) a special pushdown symbol called the initial symbol on the PDS denoted by Z_0.
(vi) a set of final states, a subset of Q denoted by F.
(vii) a transita function δ from $Q \times (\Sigma \cup \{\wedge\}) \times \Gamma$ to the set of finite subsets of $Q \times \Gamma^*$.

A PDA has 7 tupples

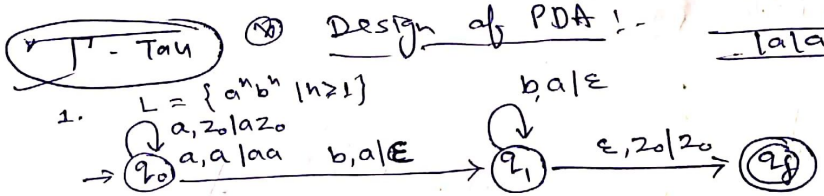
$(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

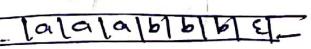
Note: when $\delta(q, a, Z) = \phi$ for $(q, a, Z) \in Q \times (\Sigma \cup \{\wedge\}) \times \Gamma$, we do not mention it.

Γ - Tau

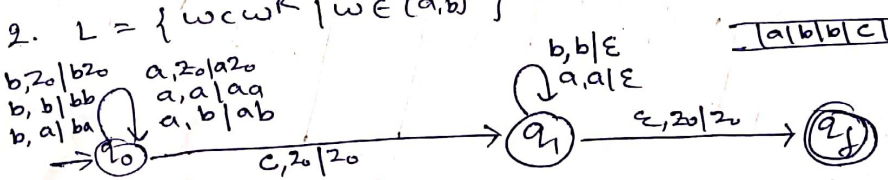
(N) Design of PDA :-

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



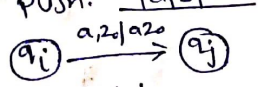


2. $L = \{wcw^R \mid w \in (a,b)^*\}$



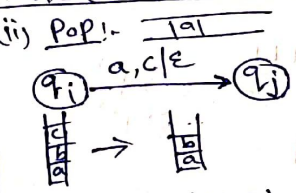
Basic operation of a stack (Instantaneous Description) :-

(i) PUSH:



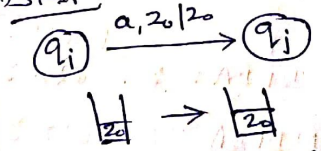
$\delta(q_i, a, Z_0) = (q_j, aZ_0)$

(ii) Pop :-



$\delta(q_i, a, c) = (q_j, \varepsilon)$

(iii) SKIP :-



$\delta(q_i, a, Z_0) = (q_j, Z_0)$

Push Down Automata (PDA) : Acceptance of String :-

Acceptance by Final State :-

Let $A = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a PDA

The set accepted by PDA by final state is defined by

$$T(A) = \{ w \in \Sigma^* \mid (q_0, w, z_0) \vdash^* (q_f, \Lambda, \alpha) \text{ for some } q_f \in F \text{ and } \alpha \in \Gamma^* \}$$

Acceptance by Empty Stack :-

Let $A = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a PDA

The set $N(A)$ accepted by empty stack is defined as

$$N(A) = \{ w \in \Sigma^* \mid (q_0, w, z_0) \vdash^* (q, \epsilon, \epsilon) \text{ for some } q \in Q \}.$$

Equivalence of PDA and CFG:-

Pushdown Automata and CFL:-

If L is a CFL, then we can construct a pda A accepted by empty store, i.e. $L = N(A)$.

Let $L = L(G)$, where $G = (V_N, \Sigma, P, S)$ is a CFG.

We construct a pda A as

$A = (\{q\}, \Sigma, V_N \cup \Sigma, \delta, q, S, \phi)$, q' is a new state not in q.

where δ is defined as follow

R_1: $\delta(q, \varepsilon, A) = \{(q, \alpha) \mid A \to \alpha \text{ is in } P\}$

R_2: $\delta(q, a, a) = \{(q, \varepsilon)\}$ for every a in Σ.

The pushdown symbol in A are variables and terminals.

(i) If the pda reads a variable A on the top of PDS, it makes a ε-move by placing RHS of any production.

(ii) If the pda read terminal a on PDS and if it matches with the current input symbol, then pda erases a.

(iii) In other cases pda halts.

Equivalence of PDA and CFG:-

$G = \{V, T, P, S\}$

Production $A \to \alpha$

$A \in V_N, \alpha \in (V \cup T)^*$

$M = \{\theta, \Sigma, \Gamma, \delta, q_0, Z_0, F\}$

The equivalent empty stack PDA will be

$M = \{q, T, V \cup T, \delta, q, S, \phi\}$

Assume Production $p: A \to \beta$

For Push: $(q, \varepsilon, A) \vdash (q, \beta)$

For Pop: $(q_0, a, a) \vdash (q, \varepsilon)$

Exp: $S \to \overset{①}{a}\, S\, b \mid \overset{②}{ab}$

Here $V_N = \{S\}$

$T = \{a, b\}$

① $\delta(q, \varepsilon, S) \to (q, asb)$

② $\delta(q, \varepsilon, S) \to (q, ab)$

$$\begin{cases} \delta(q, a, a) \to (q, \varepsilon) \\ \delta(q, b, b) \to (q, \varepsilon) \end{cases}$$

Ex: $S \Rightarrow asb \to aabb$

$(q, aabb, S) \vdash$ //

CFG to PDA :-

For converting given CFG to PDA, the necessary condition is that first symbol on RHS must be a terminal symbol.

Theorem: If L is a CFL, then we can construct a PDA A accepting L by empty store i.e. $L = N(A)$.

Let $L = L(G)$, $G = (V_N, \Sigma, P, S)$

the equivalent PDA will

$A = \{\{q\}, \Sigma, V_N \cup \Sigma, \delta, q, S, \phi\}$ (q' is new state not in q.)

where δ is defined by follow rules

R_1: $\delta(q, \varepsilon, A) = \{(q, \alpha) \mid A \to \alpha \text{ is in } P\}$

R_2: $\delta(q, a, a) = \{(q, \varepsilon)$, for every a in $\Sigma\}$

Q. Construct PDA A which is equivalent to CFG. follows:

$S \to OCC$, $C \to OS$, $C \to 1S$, $C \to 0$

Sol: The required PDA is as follow

$A = (\{q\}, \{0,1\}, \{S, C, 0, 1\}, \delta, q, S, \phi)$

δ is defined as follows rule

R_1: $\delta(q, \Lambda, S) = \{(q, OCC)\}$

$\delta(q, \Lambda, C) = \{(q, OS), (q, 1S), (q, 0)\}$

R_2: $\delta(q, 0, 0) = \{(q, \varepsilon)\}$

$\delta(q, 1, 1) = \{(q, \varepsilon)\}$

Q. Convert the grammer to equivalent PDA (empty)

$S \to 0S1 | A \quad , A \to 1A0 | S | \varepsilon$

Sol. The PDA can be

$A = \{\{q\}, \{0,1\}, \{S, A, 0, 1\}, \delta, q, S, \phi\}$

Where δ will be,

R_1: $\delta(q, \varepsilon, S) = \{(q, 0S1), (q, A)\}$

$\delta(\delta, \varepsilon, A) = \{(q, 1A0), (q, S), (q, \varepsilon)\}$

R_2: $\delta(q, 0, 0) = \{(q, \varepsilon)\}$

$\delta(q, 1, 1) = \{(q, \varepsilon)\}$

Q. Show that the language

$L = \{0^n 1^n | n \geq 1\} \cup \{0^n 1^{2n} | n \geq 1\}$

is CFL that not accepted by DPDA.

Sol: We can write a CFG for the language

$L = \{0^n 1^n | n \geq 1\} \cup \{0^n 1^{2n} | n \geq 1\}$

The production rules for the CFG would be

$S \to S_1 | S_2$

$S_1 \to aS_1b | ab$

$S_2 \to aS_2bb | abb$

The CNF of above CFG

$S \to S_1 | S_2$

$S_1 \to aS_1X | aX$

$X \to b$

$S_2 \to aS_2Y | aY$

$Y \to bX$

Now convert CFG to PDA

R_1: $(q, \varepsilon, S) = (q, S_1) | (q, S_2)$ — ①

$(q, \varepsilon, S_1) = (q, S_1X) | (q, X)$ — ②

$(q, \varepsilon, X) = (q, b)$ — ③

$(q, \varepsilon, S_2) = (q, aS_2Y) | (q, aY)$ ④

$(q, \varepsilon, Y) = (q, bX)$ — ⑤

equation ① and ④ are nondeterministic rules hence the above PDA is actually NPDA.

PDA to CFG :-

Let $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is a PDA then equivalent grammar as $G = (V, \Sigma, P, S)$. ($T = \Sigma$)

1. Construction of set of Non-terminal.

$$V = \{S\} \cup \{ [q, Z, q'] \mid q, q' \in Q, Z \in \Gamma \}$$

2- Construction of Production.

(i) S-Production: $S \to [q_0, Z_0, q]$, $q \in Q$

(ii) For Pop operation: if $\delta(q, a, Z) = (q', \varepsilon)$
then $(q, Z, q') \to a$

(iii) For Push: (SKIP) $\delta(q, a, Z) = (q_1, Z_1 Z_2 \cdots)$
then $(q, Z, q') \Rightarrow a[q_1, Z_1, q_2][q_2, Z_2, q_3] \cdots (q_m, Z_m, q')$
where $q', q_2, q_3 \cdots q_m \in Q$.

Exp:- Construct CFG for PDA $A = \{(q_0 q_1), (a, b), (Z_0 Z_1), \delta, q_0, Z_0, \phi\}$

where

$\delta(q_0, b, Z_0) = (q_0, Z Z_0)$	$\delta(q_0, \varepsilon, Z_0) = (q_0, \varepsilon)$
$\delta(q_0, b, Z) = (q_0, ZZ)$	$\delta(q_0, a, Z) = (q_1, Z)$
$\delta(q_1, b, Z) = (q_1, \varepsilon)$	$\delta(q_1, a, Z_0) = (q_0, Z_0)$

Sol: (1) $V = S, [q_0, Z_0, q_0]\ [q_0, Z_0, q_1]$
$[q_0, Z, q_0]\ [q_0, Z, q_1]$
$[q_1, Z_0, q_0]\ [q_1, Z_0, q_1]$
$[q_1, Z, q_0]\ [q_1, Z, q_1]$

2-(i) $S \to [q_0, Z_0, q_0]$
$S \to [q_0, Z_0, q_1]$

(ii) $\delta(q_0, b, Z_0) = (q_0, Z Z_0)$ (PUSH operation)

$[q_0, Z_0, q_0] \to b[q_0, Z, q_0][q_0, Z_0, q_0]$
$b[q_0, Z, q_1][q_1, Z_0, q_0]$
$[q_0, Z_0, q_1] \to b[q_0, Z, q_0][q_0, Z_0, q_1]$
$b[q_0, Z, q_1][q_1, Z_0, q_1]$

(iii) $\delta(q_0, \varepsilon, z_0) = (q_0, \varepsilon)$ (Pop operation)

$(q_0, z_0, q_0) \rightarrow \varepsilon$

(iv) $\delta(q_0, a, z) = (q_1, z)$

$[q_0, z, q_0] \rightarrow a[q_1, z, q_0]$

$[q_0, z, q_1] \rightarrow a[q_1, z, q_1]$

Two Stack PDA :-

Two stack PDA has six tuples $(Q, \Sigma, \Gamma, \delta, q_0, F)$

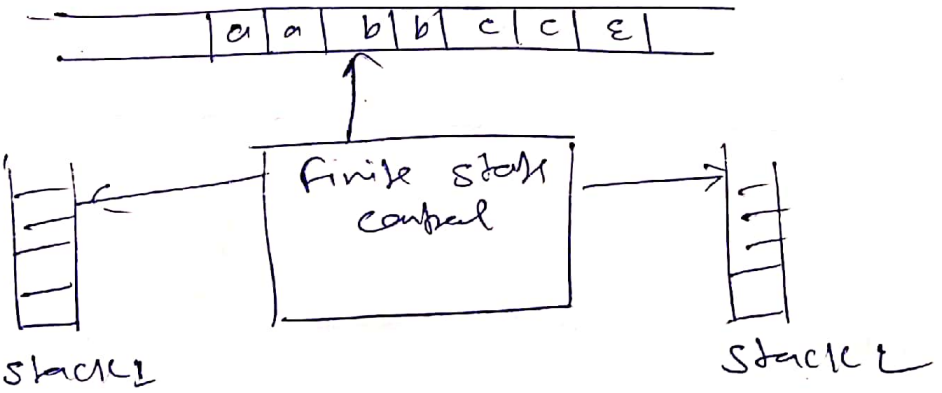
$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times \Gamma \to Q \times \Gamma^* \times \Gamma^*$$

A machine (PDA) with two stacks accepts the recursively enumerable languages.

Now the example for the language

$L_1 = \{ a^n b^n c^n \mid n \geq 1 \}$

The language L_1 easily recognized by two stack PDA. For L_1 store the numbers of a's on both stacks, it can be then used to check both the numbers of b's and c's.



Two stack PDA

Exp. $L = \{ a^n b^n c^n \mid n \geq 1 \}$