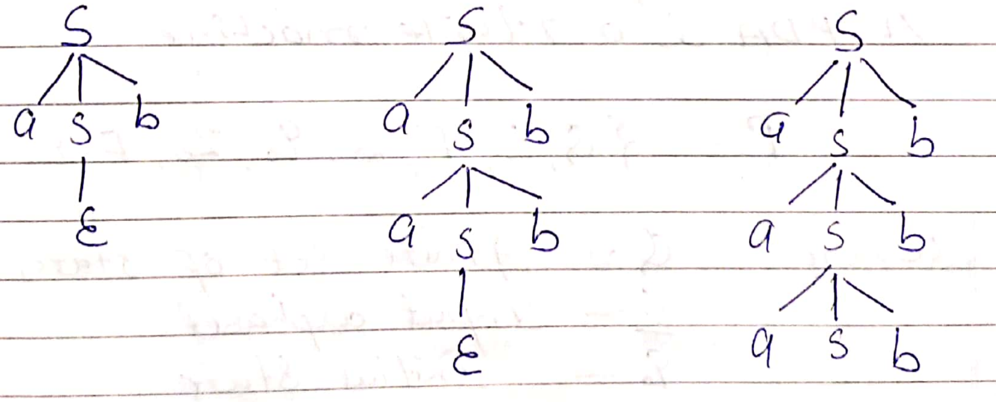
# Unit IV

## PushDown Automata (PDA)
### (finite Automata with Stack)

- Pushdown automata is used to accept Context free language.
- PDA is identical to finite automata except for the addition of Stack.
- Introduction of Stack allows us to recognize Some of the non-regular language, as finite automata Can recognize only regular language.
- Non-regular languages Can be generated by CFG.

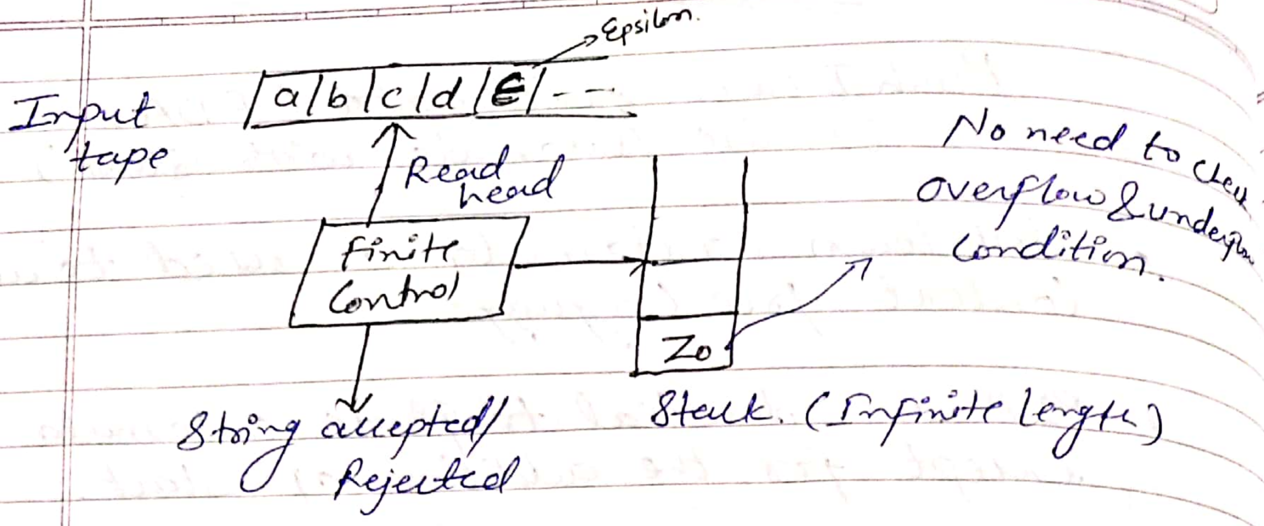
CFG

$S \to aSb \,/\, \epsilon \quad \Rightarrow L(G) = \{a^n b^n \,/\, n \geq 0\}$



String: ab      aabb      aaabbb.

(finite automata - Regular language ko accept Karte Yadi non-Regular language ko accept Karwana hai to PDA ka use Karte hai).




A PDA has - an input tape
↳ finite Control
↳ stack. (temporary memory).

- PDA has medium computing power.
- finite automata has no temporary memory and small computing power.
- PDA is more powerful than FSM.

Formal definition of PDA.
A PDA is a 7 tuple machine.

$$P = \{Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\}$$

where
$Q$ — finite set of states
$\Sigma$ — Input alphabet.
$q_0$ — Initial state
$F$ — set of final states
(Gamma) $\Gamma$ — stack alphabet.
$Z_0$ — Bottom/Initial Stack Symbol.
$\delta$ — transition function

## Deterministic PDA (DPDA):-

$$\delta : Q \times \Sigma \times \Gamma = \xrightarrow{\text{Replaced by}} Q \times \Gamma^*$$

↑ String of Stack Symbols.

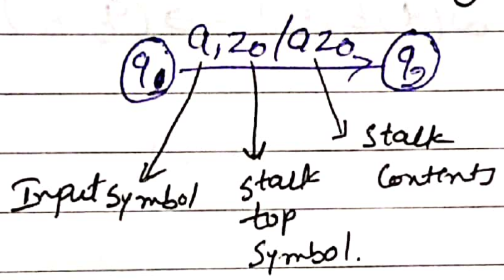
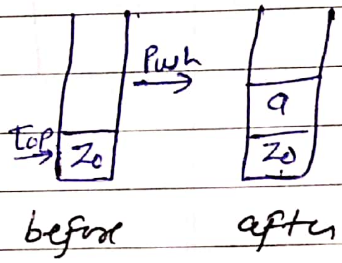
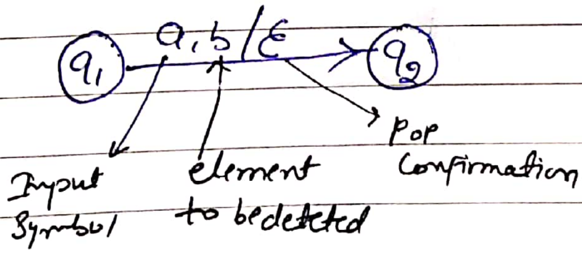
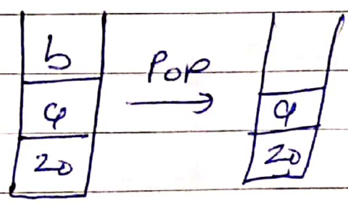
or

$$Q \times (\Sigma \cup \epsilon) \times \Gamma = Q \times \Gamma^*$$

## Non deterministic PDA (NPDA).

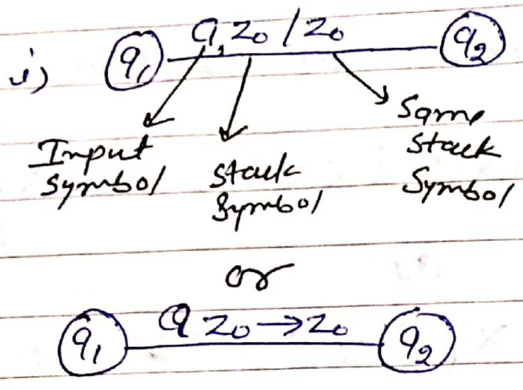
$$\delta : Q \times \Sigma \times \Gamma = 2^{(Q \times \Gamma^*)}$$

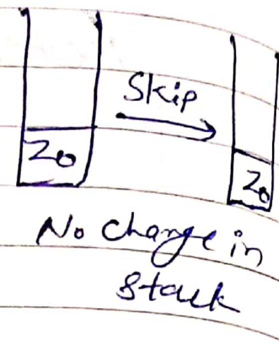
## * Operations performed on Stack :-

| Operation | Transition diagram | Stack. |
|---|---|---|
| 1) PUSH |   $\delta(q_1, a, z_0) \rightarrow (q_2, az_0)$ |   |
| 2) POP |   $\delta(q_1, a, b) \rightarrow (q_2, \epsilon)$ |   |

3) Skip

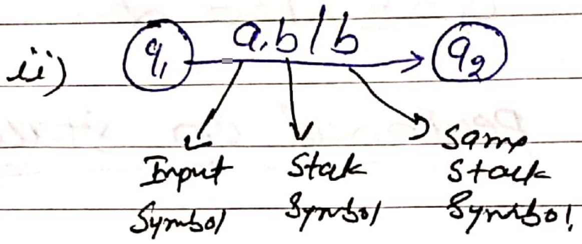
i)

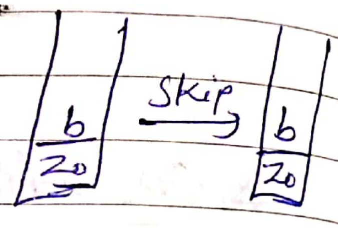



$\delta(q_1, a, Z_0) \rightarrow (q_2, Z_0)$

ii)




$\delta(a, b, b) \rightarrow (q_2, b)$

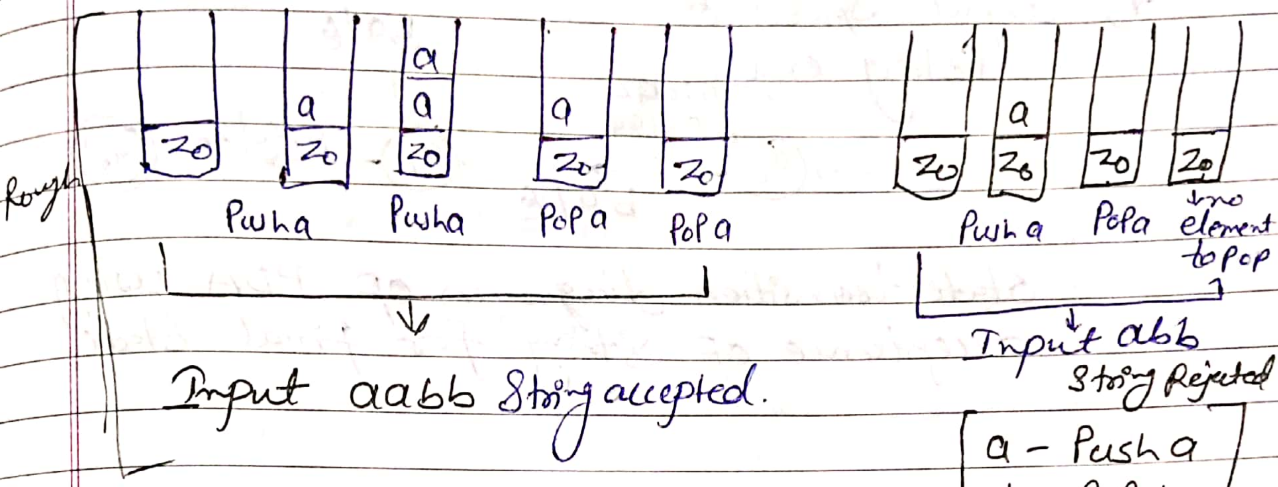
# Numericals on PDA.

Q. Construct a pushdown automata for language

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

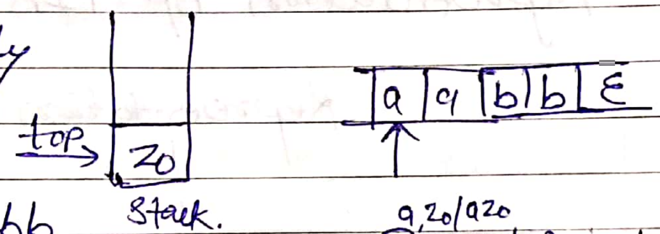
Sol:- $L = \{ab, aabb, aaabbb \ldots\}$

Rough



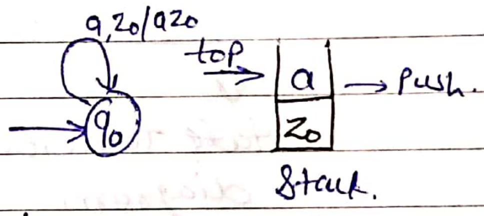

a - Push a
b - Pop b

Step 1:- Initially

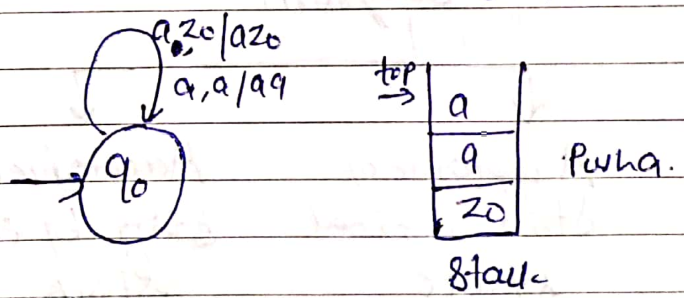



Let String aabb

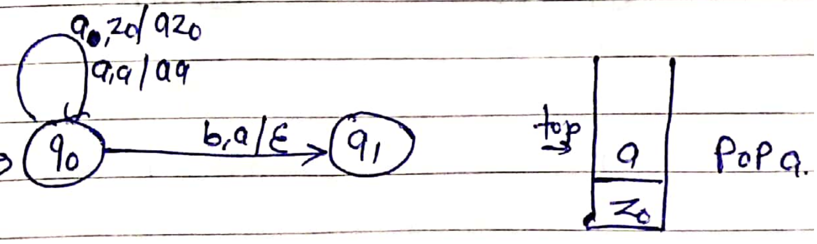
i) Input Symbol a



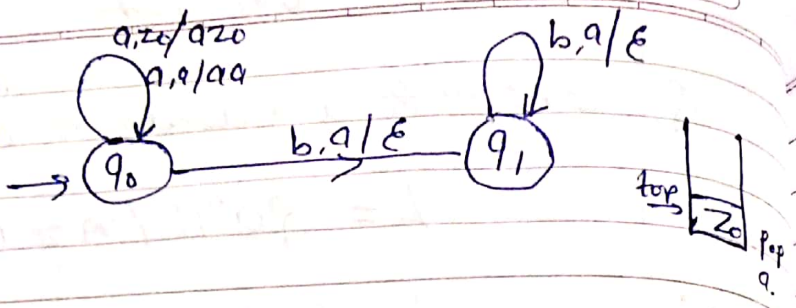

ii) Input Symbol a.
(any no. of a's)



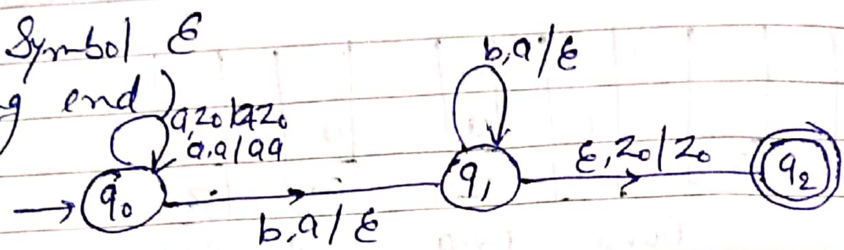

iii) Input Symbol b. →



iv] input Symbol b
(any no. of b's)




v) input Symbol ε
(String end)




State transition diagram of PDA using acceptance of String for final State.

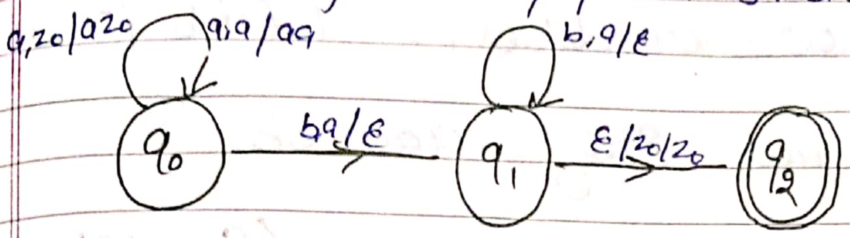
★ Representation of PDA :-

Representation of PDA.

- State transition diagram
  - Acceptance of String by final State
  - Acceptance of String by empty Stack.
- State transition function.
  - Acceptance of String by final State
  - Acceptance of String by empty stack.

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

Sol: PDA acceptance by final state :-




final State

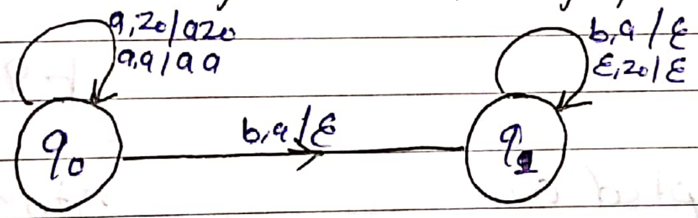
State transition diagram.

State transition function :-

$\delta(q_0, a, z_0) = (q_0, az_0)$ (Push)
$\delta(q_0, a, a) = (q_0, aa)$ (Push)
$\delta(q_0, b, a) = (q_1, \epsilon)$ (PoP)
$\delta(q_1, b, a) = (q_1, \epsilon)$ (PoP)
$\delta(q_1, \epsilon, z_0) = (q_2, z_0)$ (Skip).

or

PDA Acceptance by empty Stack.




(State transition diagram).

$\delta(q_0, a, z_0) = (q_0, az_0)$ (Push)
$\delta(q_0, a, a) = (q_0, aa)$ (Push)
$\delta(q_0, b, a) = (q_1, \epsilon)$ (PoP)

$\delta(q_1, b, a) = (q_1, \epsilon)$ (PoP)
$\delta(q_1, \epsilon, z_0) = (q_1, \epsilon)$ (PoP $z_0$)

* How to check acceptance or rejection String using PDA.

Sol:- $w =$ aaabbb $= L(a^n b^n / n \geq 1)$

Let Input String aaabbb

$(q_0, aaabbb, z_0)$ (Push) $(q_0, aaabbb, z_0)$

$\vdash q_0, aabbb, az_0$ (Push) $\longrightarrow$ (Same)

$\vdash q_0, abbb, aaz_0$ (Push)

$\vdash q_0, bbb, aaaz_0$ (POP)

$\vdash q_1, bb, aaz_0$ (POP)

$\vdash q_1, b, az_0$ (POP)

$\vdash q_1, \epsilon, z_0$ (Skip) $\vdash q_1, \epsilon, z_0$

$\vdash q_2, z_0$ $\vdash q_1, \epsilon$

String accepted as $q_2$ is final State

String acceptance by empty State

Input String abb

$(q_0, abb, z_0)$

$\vdash q_0, bb, az_0)$

$\vdash q_0, b, z_0$

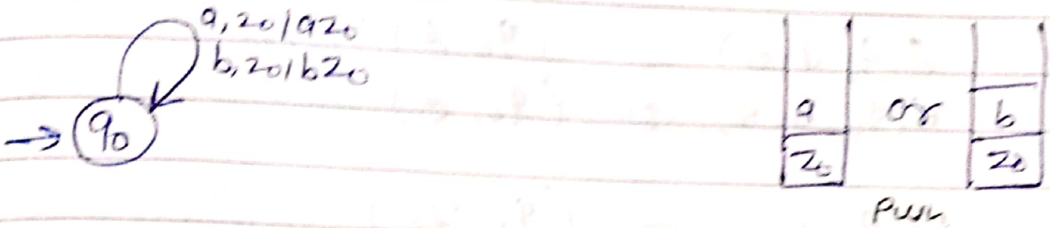
no Such transition available in PDA $\therefore$ String Rejected by PDA.

Q. Construct PDA for language

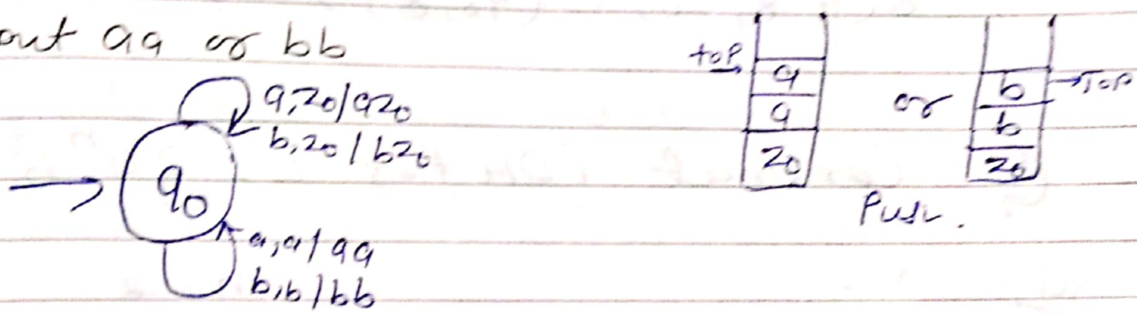
$$L = \{ w \mid n_a(w) = n_b(w) \}$$

Sol:- $L = \{ab, aabb, bbaa, baba, abab \ldots\}$

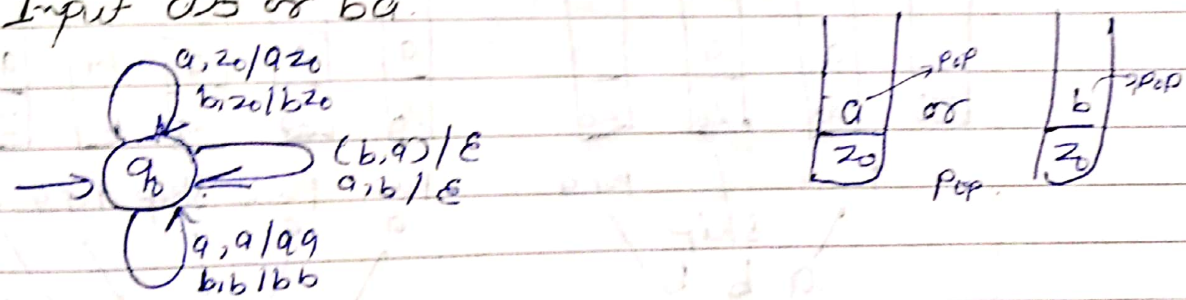
i) input a or b



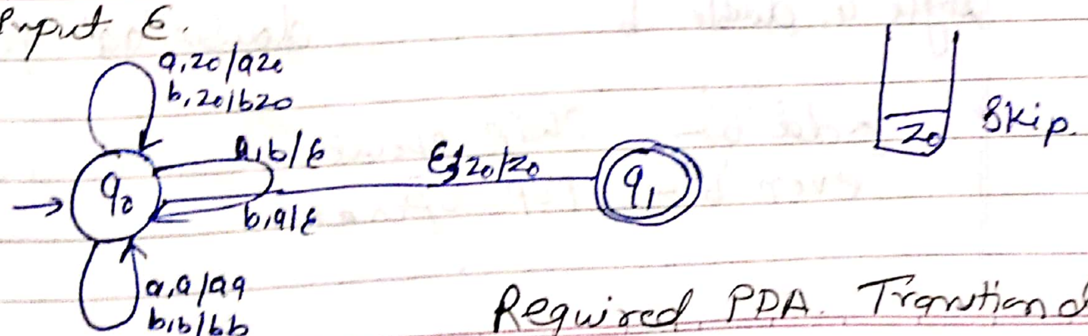

ii) input aa or bb




iii) Input ab or ba.




iv) Input ε.




Required PDA Transition diagram

Transition function:-

$$\delta(q_0, a, z_0) = (q_0, az_0)$$
$$\delta(q_0, b, z_0) = (q_0, bz_0)$$
$$\delta(q_0, a, a) = (q_0, aa)$$
→ Push
$$\delta(q_0, b, b) = (q_0, bb)$$

$$\delta(q_0, a, b) = (q_0, \epsilon)$$
$$\delta(q_0, b, a) = (q_0, \epsilon)$$
– Pop

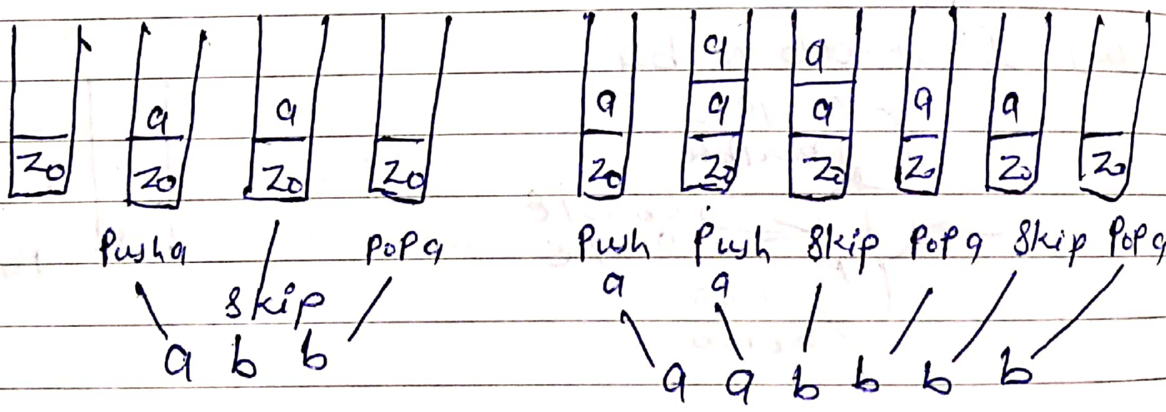
$$\delta(q_0, \epsilon, z_0) = (q_1, z_0)$$

or

$$\delta(q_0, \epsilon, z_0) = (q_0, \epsilon) \rightarrow \text{no final state.}$$

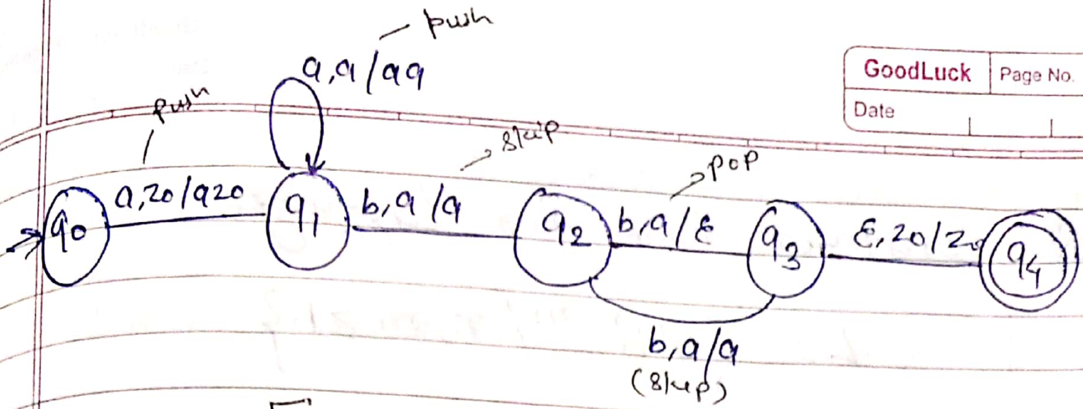
Q. Construct PDA for $L = \{a^n b^{2n} / n \geq 1\}$

Sol:- $n=1$ abb $\quad$ $n=2$ aabbbb




Single 'a', double 'b' $\quad$ double 'aa', four 'bbbb'

odd 'b' – Skip operation
even 'b' – Pop operation




[PDA with final state]

- Transition function:

$\delta(q_0, a, z_0) = (q_1, az_0)$ ] - Push 'a'

$\delta(q_1, a, a) = (q_1, aa)$

$\delta(q_1, b, a) = (q_2, a)$ - Skip

$\delta(q_2, b, a) = (q_3, \varepsilon) \rightarrow$ Pop a

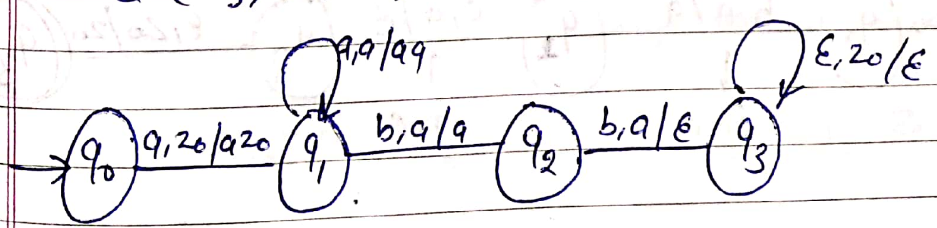
$\delta(q_3, b, a) = (q_2, a)$ - Skip.

$\delta(q_3, \varepsilon, z_0) = (q_4, z_0)$

Acceptance by final state

Empty Stack. transition function.

$\delta(q_3, \varepsilon, z_0) = (q_3, \varepsilon)$



Date

Q. Construct PDA for language:

$$L = \{a^n b^n c^m \mid n, m \geq 1\}$$

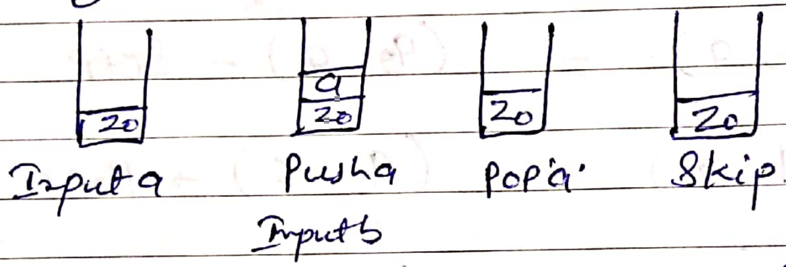
**Sol:-** $L = \{abc, abcc, aabbc, \ldots\}$

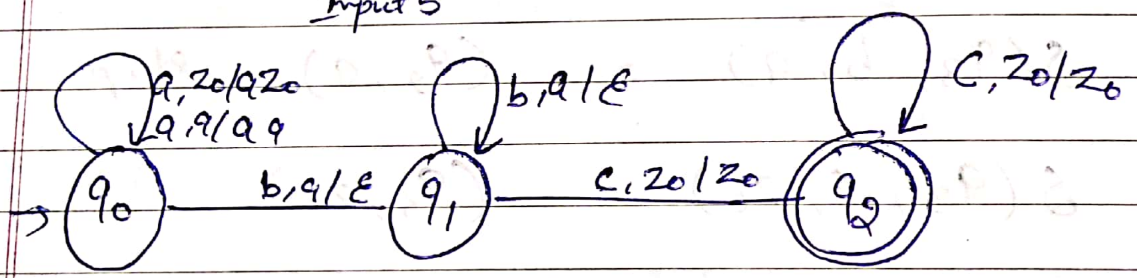
$a \rightarrow b \longrightarrow c^+$ — 1 or more

equal — accept.

a - Push 'a'
b - Pop 'a'

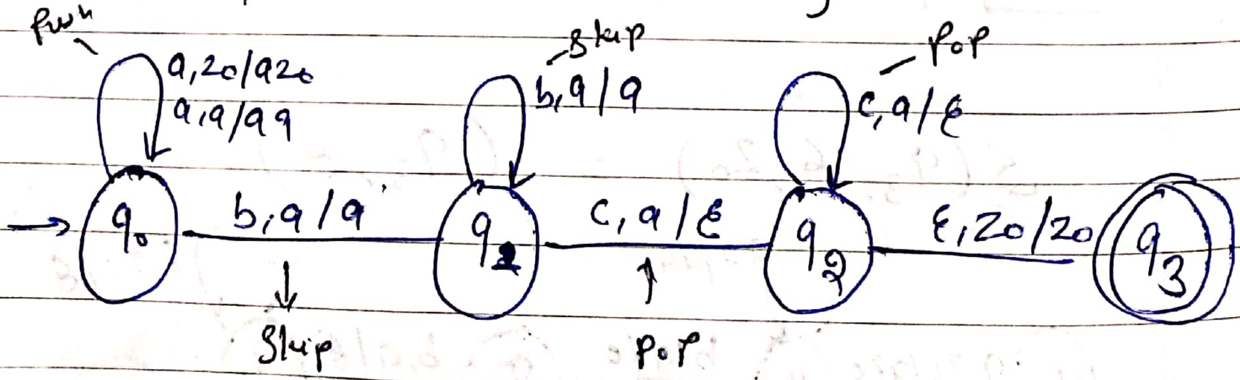
String $w = abc$




Q. $L = \{a^n b^m c^n \mid n, m \geq 1\}$

**Sol!-**



Q. Design a PDA to check for well formed parentheses.

Sol:- Well formed parentheses means

i) ( )   (ii) (( ))   (iii) ( )( )

not well formed parenthesis (i) (   (ii) (( )   (iii) ( )(

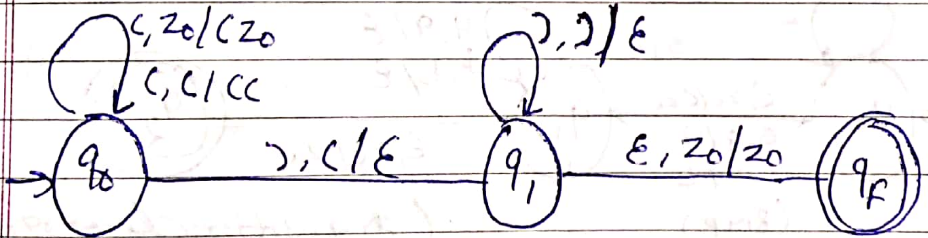
Transition function for PDA.

$\delta(q_0, (, Z_0) = (q_0, (Z_0)$   Push (

$\delta(q_0, (, () = (q_0, (()$   Push (

$\delta(q_0, ), () = (q_1, \epsilon)$   Pop (

$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$




Transition diagram. (accepted by final State)

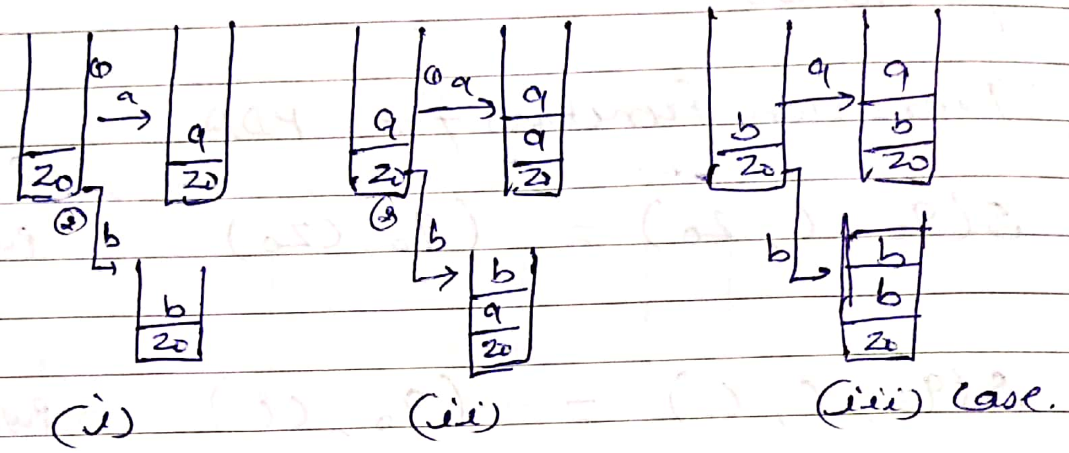
PDA $m = \{(q_0, q_1, q_f)\ ((, ))\ , ((, Z_0),\ Z_0, q_0, \delta, q_f\}$

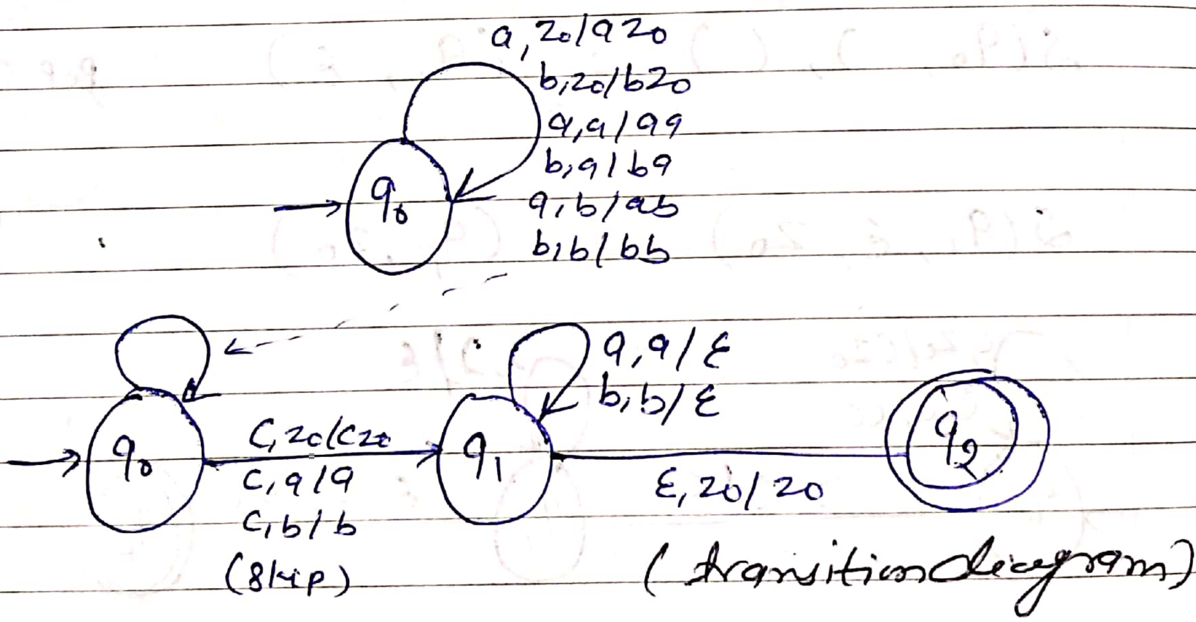
Seven tuples

Q Construct a PDA for language
$L = \{ wcw^R \mid w \in (a+b)^* \}$ [odd Pallindrom]

Sol:- $L = \{ c, aca, abcba, abbcbba, bacab, \ldots \}$

$w = abcba$




Transition function:-

$\delta(q_0, a, z_0) = (q_0, az_0)$
$\delta(q_0, b, z_0) = (q_0, bz_0)$
$\delta(q_0, a, a) = (q_0, aa)$

Push

a b c b a ε

$\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$ ] push
$\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, c, z_0) = (q_1, z_0)$
$\delta(q_0, c, a) = (q_1, a)$ ] skip
$\delta(q_0, c, b) = (q_1, b)$

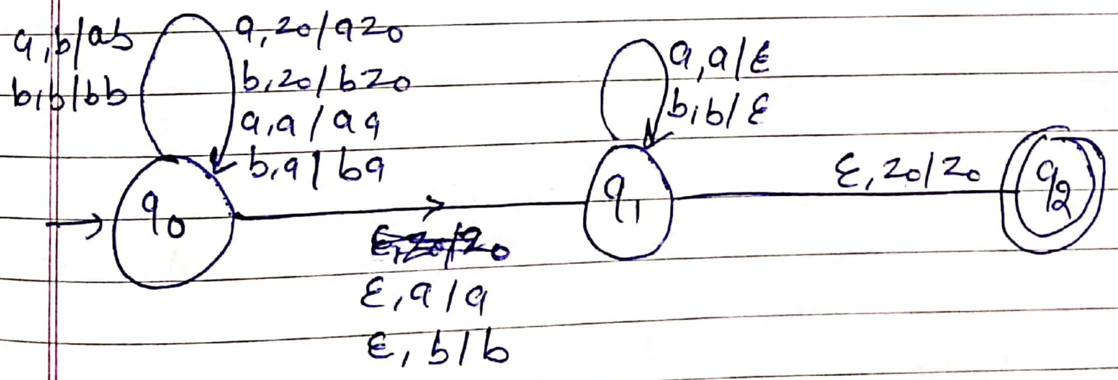
$\delta(q_1, a, a) = (q_1, \epsilon)$
$\delta(q_1, b, b) = (q_1, \epsilon)$ ] pop

$\delta(q_1, \epsilon, z_0) = (q_2, z_0)$

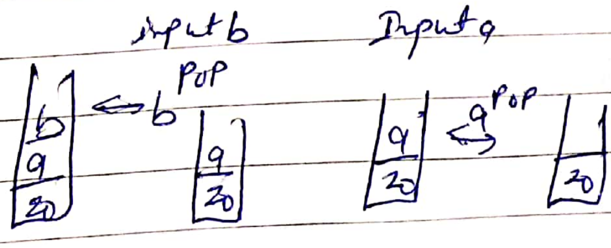
Q Construct PDA for language (non-deterministic)

$L = \{WW^R \mid W \in (a+b)^+\}$ [Even Pallindrom]

Sol $L = \{abba, aba, aba \; ....\}$




$w = abba$



Date

Transition function:-

$$\left.\begin{aligned}
\delta(q_0, a, z_0) &= (q_0, az_0) \\
\delta(q_0, b, z_0) &= (q_0, bz_0) \\
\delta(q_0, a, a) &= (q_0, aa) \\
\delta(q_0, b, a) &= (q_0, ba) \\
\delta(q_0, a, b) &= (q_0, ab) \\
\delta(q_0, b, b) &= (q_0, bb)
\end{aligned}\right] \text{Push}$$

$$\left.\begin{aligned}
\delta(q_0, \epsilon, a) &= (q_1, a) \\
\delta(q_0, \epsilon, b) &= (q_1, b)
\end{aligned}\right] \text{Skip}$$

$$\left.\begin{aligned}
\delta(q_1, a, a) &= (q_1, \epsilon) \\
\delta(q_1, b, b) &= (q_1, \epsilon)
\end{aligned}\right] \text{Pop}$$

$$\delta(q_1, \epsilon, z_0) = (q_2, z_0)$$

String accepted by final state

Conversion of PDA to CFG.

Q. Construct equivalent CFG for the following PDA.

$\delta(q_0, 1, z_0) = (q_0, Xz_0)$
$\delta(q_0, 1, X) = (q_0, XX)$
$\delta(q_0, 0, X) = (q_1, X)$
$\delta(q_1, 1, X) = (q_1, \epsilon)$
$\delta(q_1, 0, z_0) = (q_0, z_0)$
$\delta(q_0, \epsilon, z_0) = (q_0, \epsilon)$

Sol:- CFG is defined by 4 tuples.

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

where

$T = \{0, 1\}$ S - Starting Symbol

P :- Start production

① $S \rightarrow [q_0, z_0, q_0]$
② $S \rightarrow [q_0, z_0, q_1]$

$\delta(q_0, 1, z_0) = (q_0, Xz_0)$

③ $[q_0, z_0, q_0] \longrightarrow 1 [q_0, X, q_0] [q_0, z_0, q_0]$

④ $[q_0, z_0, q_0] \longrightarrow 1 [q_0, X, q_1] [q_1, z_0, q_0]$

⑤ $[q_0, z_0, q_1] \longrightarrow 1 [q_0, X, q_0] [q_0, z_0, q_1]$

⑥ $[q_0, z_0, q_1] \longrightarrow 1 [q_0, X, q_1] [q_1, z_0, q_1]$

$\delta[q_0, 1, X] = (q_0, XX)$ Push

(7) $[q_0, X, q_0] \to 1[q_0, X, q_0][q_0, X, q_0]$

(8) $[q_0, X, q_0] \to 1[q_0, X, q_1][q_1, X, q_0]$

(9) $[q_0, X, q_1] \to 1[q_0, X, q_0][q_0, X, q_1]$

(10) $[q_0, X, q_1] \to 1[q_0, X, q_1][q_1, X, q_1]$

$\delta(q_0, 1, X) = (q_1, \epsilon)$ Pop.

(11) $[q_1, X, q_1] \to 1$

$\delta(q_0, \epsilon, Z_0) = (q_0, \epsilon)$ Pop

(12) $[q_0, Z_0, q_0] \to \epsilon$

$\delta(q_0, 0, X) = (q_1, X)$ Skip.

(13) $[q_0, X, q_0] \to 0[q_1, X, q_0]$

(14) $[q_0, X, q_1] \to 0[q_1, X, q_1]$

$\delta(q_1, 0, Z_0) = (q_0, Z_0)$ Skip.

(15) $[q_1, Z_0, q_0] \to 0[q_0, Z_0, q_0]$

(16) $[q_1, Z_0, q_1] \to 0[q_0, Z_0, q_1]$

Let $[q_0, z_0, q_0] = A$ $\quad [q_0, z_0, q_1] = B$

$[q_1, z_0, q_0] = C$ $\quad [q_1, z_0, q_1] = D$

$[q_0, X, q_0] = E$ $\quad [q_0, X, q_1] = F$

$[q_0, X, q_0] = G$ $\quad [q_1, X, q_1] = H$

After substituting in above production we get.

$S \to A \mid B$

$A \to 1EA \mid 1FC \mid \epsilon$

$B \to 1EB \mid 1FD$ [non terminating]

$E \to 1EE \mid 1FG \mid 0G$ [useless]

$F \to 1EF \mid 1FH \mid 0H$

$H \to 1$

$C \to 0A$

$D \to 0B$ [useless]

In above grammer B, E, D, G are useless so remove it from production

P:

$S \to A$

$A \to 1FC \mid \epsilon$

$F \to 1FH \mid 0H$

$H \to 1$

$C \to 0A$

$V = \{S, A, F, H, C\}$

$T = \{0, 1\}$

S – Starting Symbol.

Required CFG.

## * Conversion of CFG to PDA (non-deterministic PDA) :-

Sol :-

PDA $P = \{Q, \Sigma, \delta, q_0, \Gamma, Z_0, F\}$

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

Here

$Q = \{q\}$

$\Sigma = T$

$q_0 = q$

$F = \phi$

$Z_0$ = bottom of stack.

$\Gamma = \{V\} \cup \{T\} \cup \{Z_0\}$

Step 1 :-

Transition function:

i) Initially.

$$\boxed{\delta(q, \epsilon, Z_0) = (q, SZ_0)}$$

Start symbol is pushed in the stack.

ii) $A \to \alpha$ or $A \to aBB$

$$\boxed{\delta(q, \epsilon, A) = (q, \alpha)} \quad \delta(q, \epsilon, A) = (q, aBB)$$

iii) $A \to a \quad \forall a \in \Sigma$

$$\boxed{\delta(q, a, a) = (q, \epsilon)}$$ Pop.

iv) $$\boxed{\delta(q, \epsilon, Z_0) = (q, \epsilon)}$$ empty stack.

Page No
Date

Q. Convert CFG to PDA.

$S \to aAA$
$A \to aS/bS/a$

Sol:- $G = \{V, T, P, S\}$

CFG = $G = \{(S, A), (a, b), P, S\}$

PDA = $P = \{Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\}$

$= \{\{q\}, \{a, b\}, \{S, A, a, b, Z_0\}$
$\delta, q_0, Z_0, \phi\}$

Transition function:

1) $\delta(q, \epsilon, Z_0) = (q, SZ_0)$
push start symbol in the stack.

For all NT. $A \to a$

2) $\delta(q, \epsilon, S) = \{q, aAA)$

3) $\delta(q, \epsilon, A) = \{q, aS\}, \{q, bS\} \{q, a\}$

for all T. $A \to a$

4) $\delta(q, a, a) = (q, \epsilon)$

5) $\delta(q, b, b) = (q, \epsilon)$

6) $\delta(q, \epsilon, Z_0) = (q, \epsilon)$ empty stack.