Push Down Automata

(*) It is a way to implement CFG

(*) It is a regular automata with Stack

(*) It is more powerful & large memory than regular finite automata

(*) It is represented by 7 Tuples

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

Q → Finite set of state

Σ → Finite set of Symbol

Γ → Finite set of alphabet

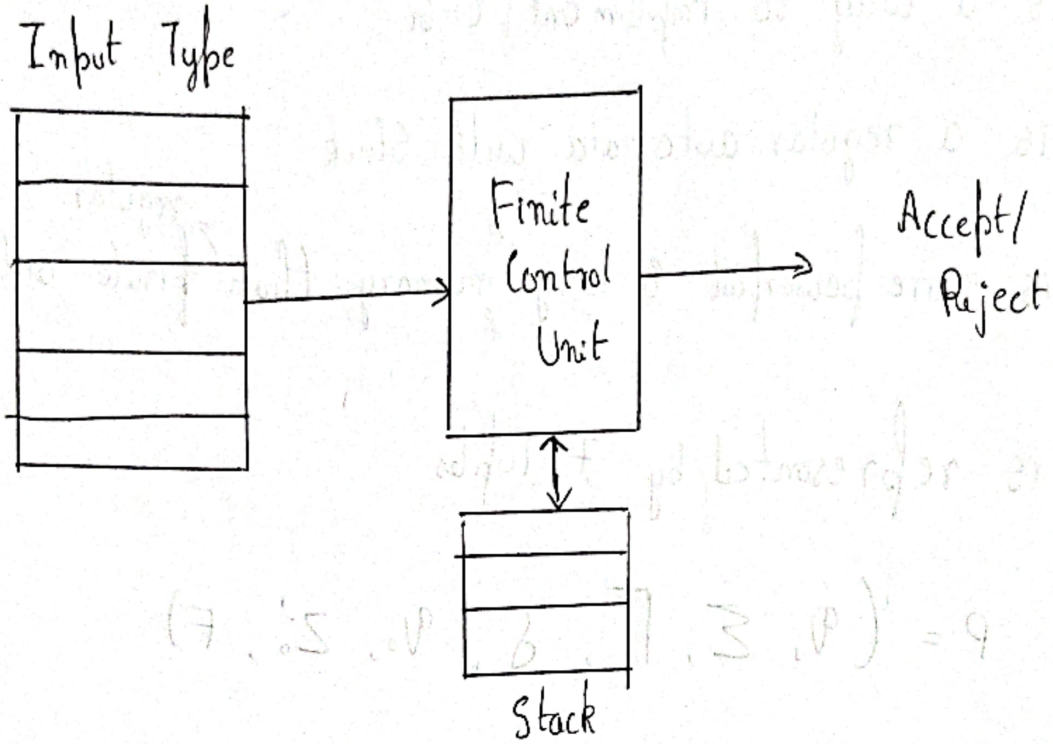
δ → Transition Function

q_0 → Intial State

Z_0 → Top of Stack

F → Final State

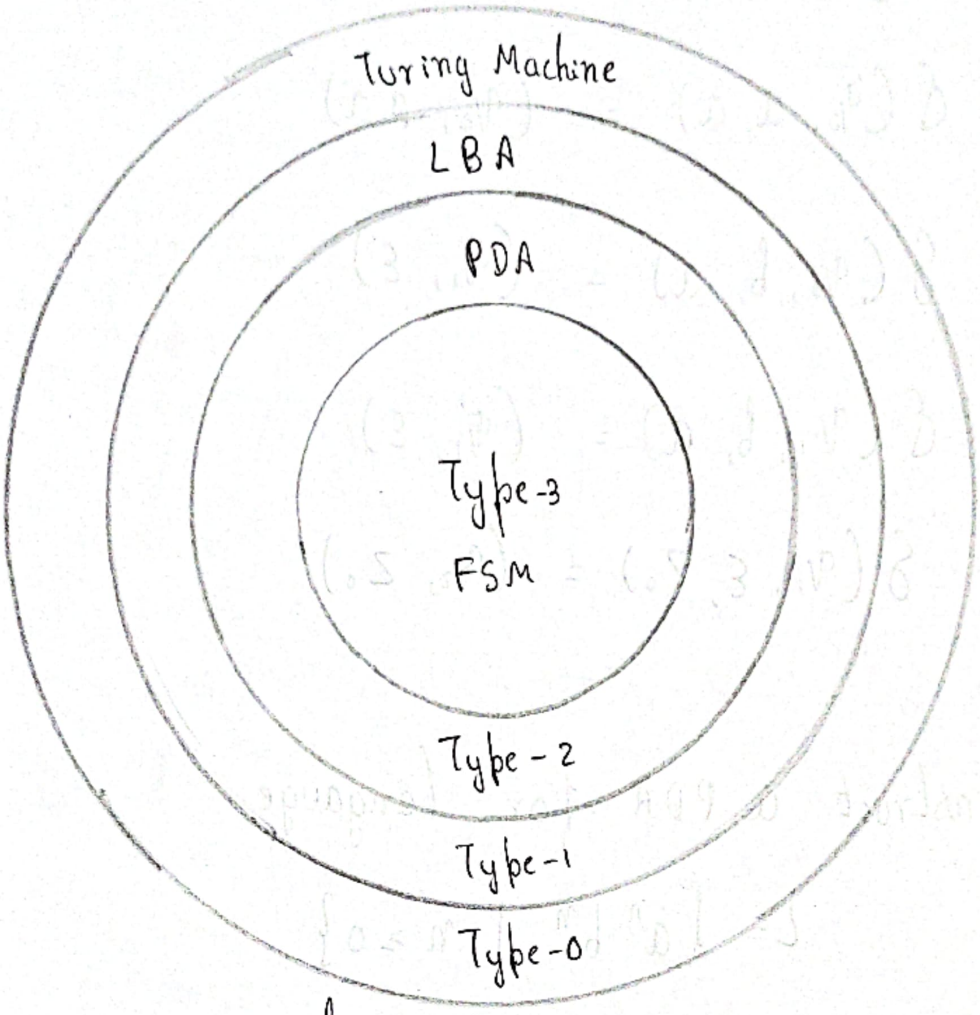
⇒ 3 Components of Push Down automata:-



⇒ Noam Chomsky defined 4 types of grammar:-

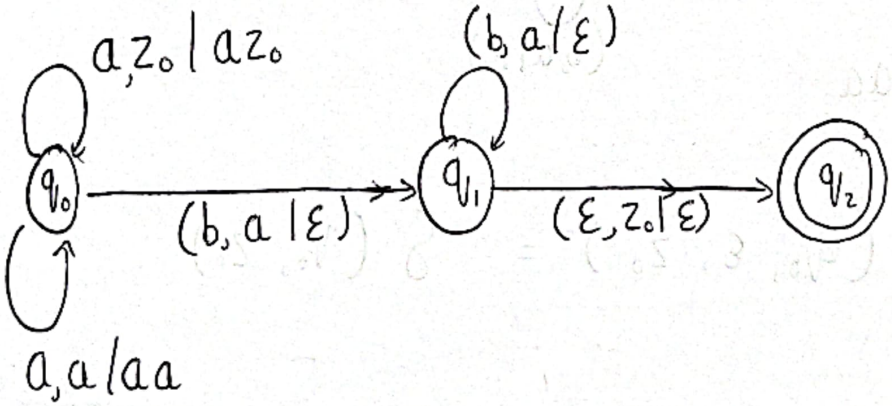
Type	Grammar Accepted	Language Accepted	System Accepted
Type-3	Regular	Regular	Finite State Machine
Type-2	Context-Free	Context-Free Language	Push Down Automata
Type-1	Context-Sensitive Grammar	Context-Sensitive Language	Linear Bounded Automata
Type-2	Unrestricted Grammar	Recursively Innumerable	Turing Machine

⇒ Chomsky Heirarchy :-



① Design a PDA for the language = $a^n b^n \mid n \geq 1$

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



$$\delta(q_0, a, z_0) = (q_0, a z_0)$$

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

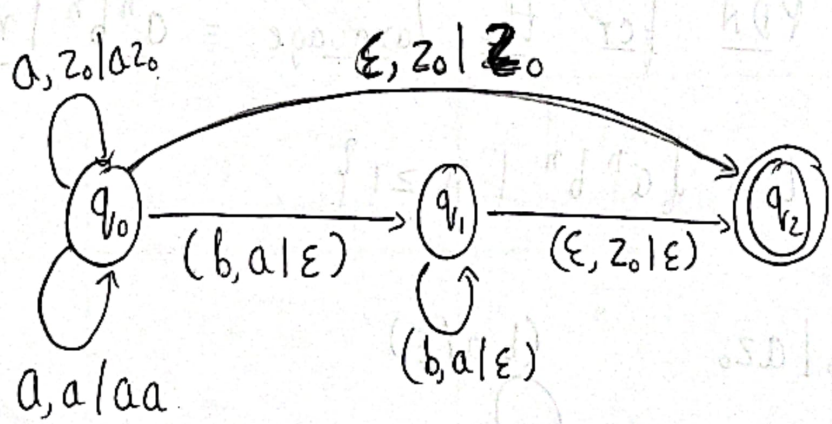
$$\delta(q_0, b, a) = (q_1, \varepsilon)$$

$$\delta(q_1, b, a) = (q_1, \varepsilon)$$

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

② Construct a PDA for langauge

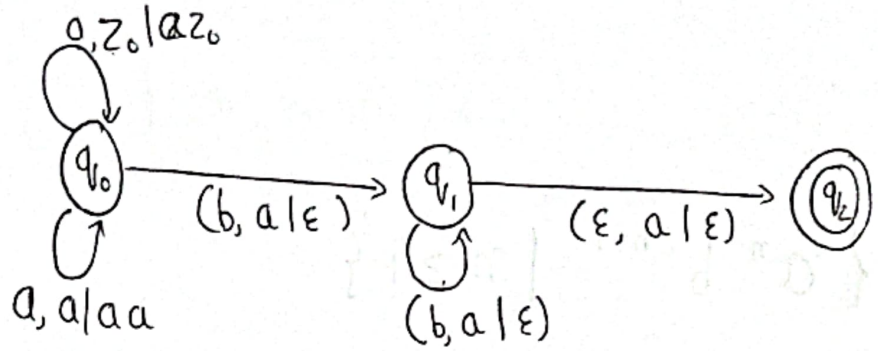
$$L = \{a^n b^n \mid n \geq 0\}$$



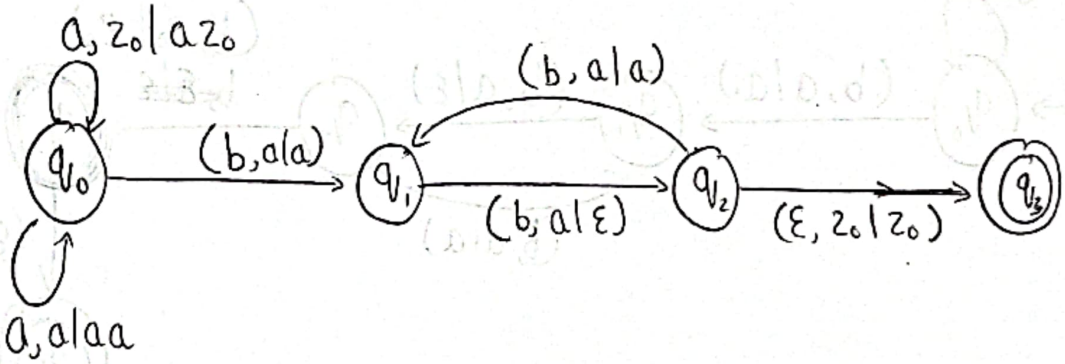
$$\delta(q_0, \varepsilon, z_0) = \delta(q_0, z_0)$$

③ $L = \{ a^n b^m \mid n \geq m,\ m \geq 1 \}$

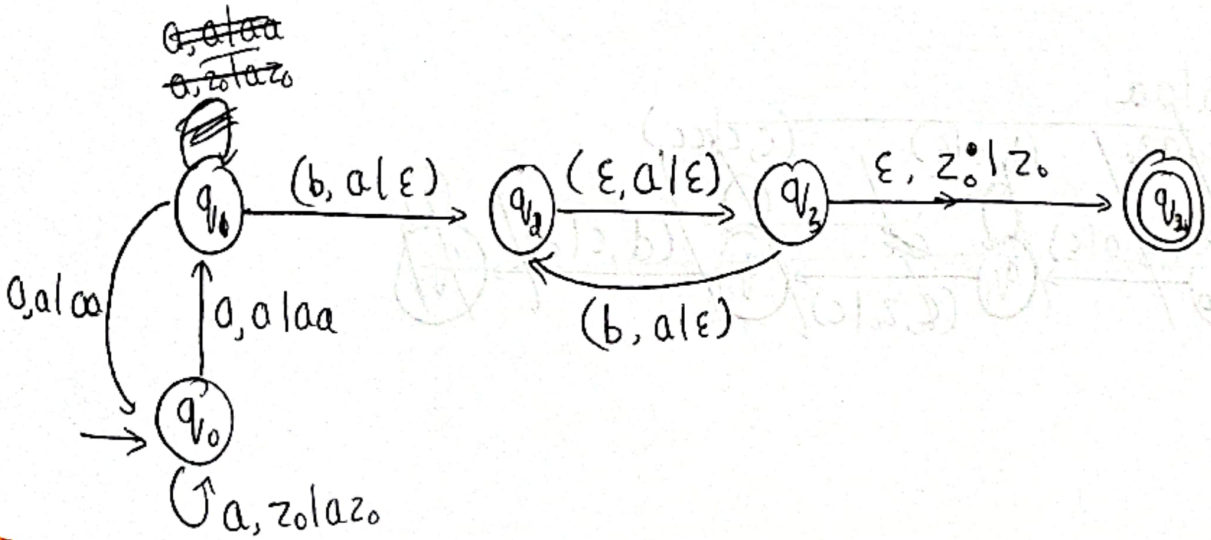
$L = \{ aab,\ aaabb,\ ab, \ldots \}$



④ $L = \{ a^n b^{2n} \mid n \geq 1 \}$

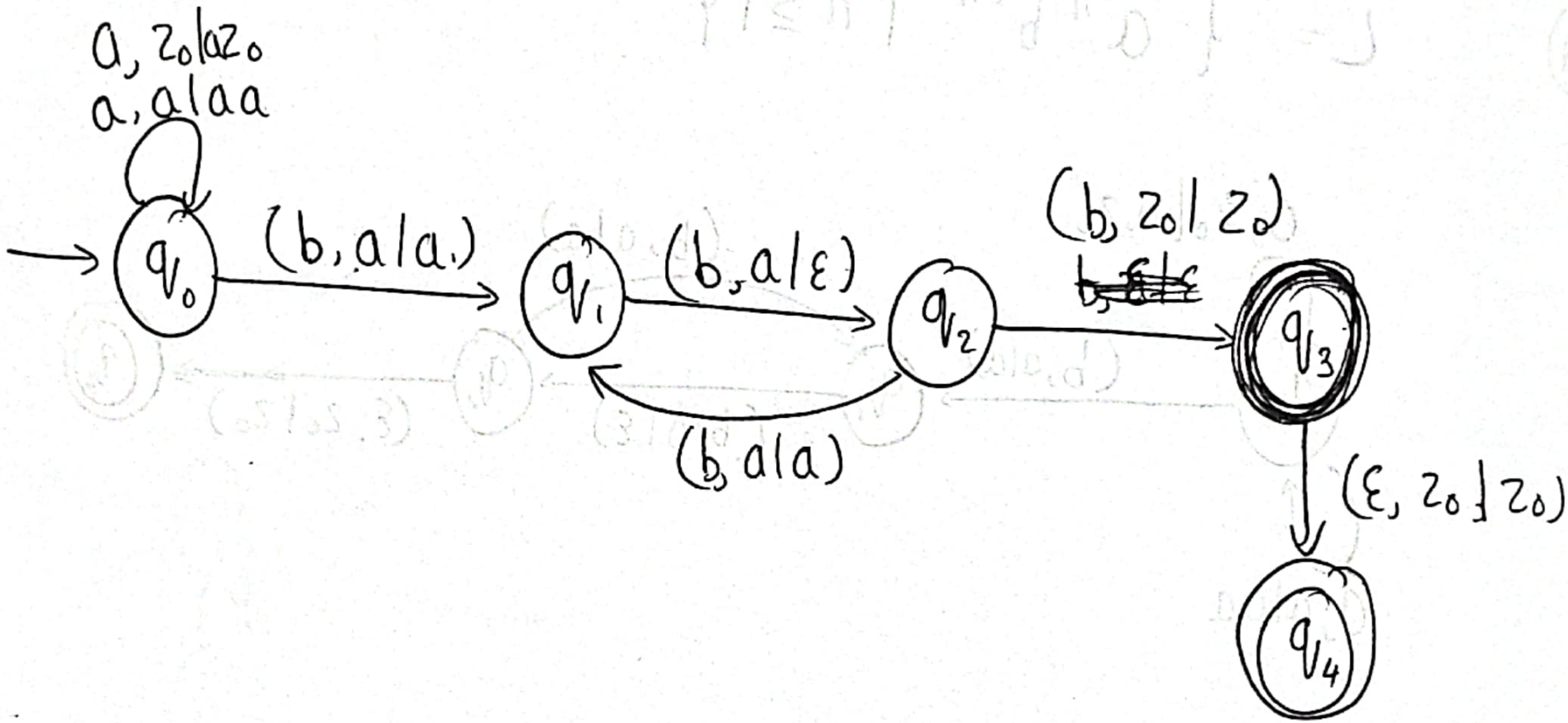


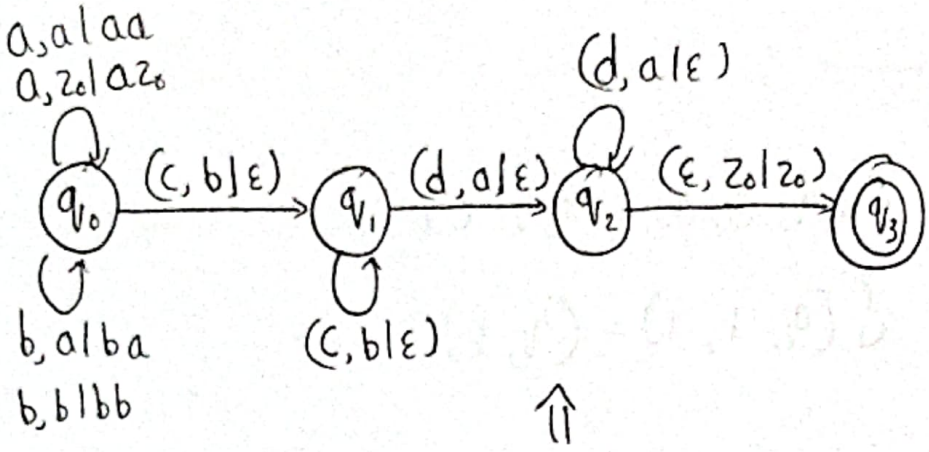
⑤ $L = \{ a^{2n} b^n \mid n \geq 1 \}$ i.e $L = \{ aab, aaaabb \ldots \}$



⑥ $L = \{ a^n b^{2n+1} \mid n \geq 1 \}$

$L = \{ abbb, aabbbbb, \ldots \}$





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

⇒ Conversion from CFG to PDA :-

① Convert the following CFG to PDA :-

$S \rightarrow 0S1 \mid 00 \mid 11$

Rules :-

① If a variable is encountered push into stack

$$\delta(q, \varepsilon, A) = (q, \beta)$$

② If a terminal is encountered, pop from stack

$$\delta(q, \varepsilon, a) = (q, \varepsilon)$$

Rule - 1 :-

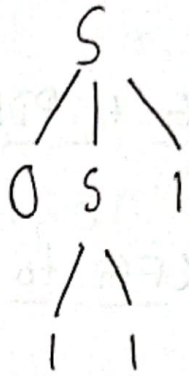
$$\delta(q, \varepsilon, S) = \{ (q, 0S1), (q, 00), (q, 11) \}$$

Rule ②:-

$$\delta(q, 0, 0) = (q, \epsilon)$$

$$\delta(q, 1, 1) = (q, \epsilon)$$

→ Parse Tree:-



=> Instanteneous Description (ID):-

Turnstile Notation ($\vdash$)

$(q, 0111, S) \vdash (q, 0111, 0S1)$

$\vdash (q, 111, S1)$

$\vdash (q, 111, 11)$

$\vdash (q, 11, 11)$

$\vdash (q, 1, 1)$

$\vdash (q, \epsilon, z_0)$

(2) Convert the following CFG to PDA and check if string "abbcbba" is accepted or not.

$S \rightarrow aSa$

$S \rightarrow bsb$

$S \rightarrow c$

The equivalent PDA is,

$\delta(q, \varepsilon, S) = \{(q, aSa), (q, bSb)$ $(q, c)\}$

$\delta(q, a, a) = (q, \varepsilon)$

$\delta(q, b, b) = (q, \varepsilon)$

$\delta(q, c, c) = (q, \varepsilon)$

ID :-

$(q, abbcbba, \$) \vdash (q, abbcbba, aSa)$

$\vdash (q, bbcbba, Sa)$

$\vdash (q, bbcbba, bsba)$

$\vdash (q, bcbba, bsbba)$

$\vdash (q, cbba, sbba)$

$\vdash (q, cbba, cbba)$

$\vdash (q, \varepsilon, Z_0)$ //

③ Convert the grammar to PDA

$$I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$$

$$E \rightarrow I \mid E \times E \mid E + E \mid (E)$$

Soln:-

The equivalent PDA is :-

$$\delta(q, \varepsilon, I) = \{(q, a), (q, b), (q, Ia), (q, Ib), (q, I0), (q, I1)\}$$

$$\delta(q, \varepsilon, E) = \{(q, I), (q, E \times E), (q, E+E), (q, (E))\}$$

$$\delta(q, a, a) = (q, \varepsilon) \qquad \delta(q, b, b) = (q, \varepsilon)$$

$$\delta(q, 0, 0) = (q, \varepsilon) \qquad \delta(q, 1, 1) = (q, \varepsilon)$$

$$\delta(q, *, *) = (q, \varepsilon) \qquad \delta(q, +, +) = (q, \varepsilon)$$

$$\delta(q,),)) = (q, \varepsilon) \qquad \delta(q, (, () = (q, \varepsilon)$$

⇒ Conversion of PDA to CFG:-

① Consider the PDA, Convert it into CFG:-

$$P = (\{q_0, q_1\}, \{a, b\}, \{a, z_0\}, \delta, q_0, z_0, \phi)$$

$$\delta(q_0, a, z_0) = (q_0, a z_0)$$

$$\delta(q_0, b, a) = (q_1, a)$$

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

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

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

$$\delta(q_1, \varepsilon, z_0) = (q_1, \varepsilon)$$

Soln :-

(i) $\delta(q_0, a, z_0) = (q_1, a z_0)$

$[q_0\ z_0\ q_0] \longrightarrow a\,[q_0\ a\ q_0][q_0\ z_0\ q_0]$

$[q_0\ z_0\ q_0] \longrightarrow a\,[q_0\ a\ q_1][q_1\ z_0\ q_0]$

$[q_0\ z_0\ q_1] \longrightarrow a\,[q_0\ a\ q_0][q_0\ z_0\ q_1]$

$[q_0\ z_0\ q_1] \longrightarrow a\,[q_0\ a\ q_1][q_1\ z_0\ q_1]$

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

$[q_0\ a\ q_0] \longrightarrow a\,[q_0\ a\ q_0]\,[q_0\ a\ q_0]$

$[q_0\ a\ q_0] \longrightarrow a\,[q_0\ a\ q_1]\,[q_1\ a\ q_0]$

$[q_0\ a\ q_1] \longrightarrow a\,[q_0\ a\ q_0]\,[q_0\ a\ q_1]$

$[q_0\ a\ q_1] \longrightarrow a\,[q_0\ a\ q_1]\,[q_1\ a\ q_1]$

(iii) $\delta(q_0, b, a) = (q_1, a)$

$[q_0\ b\ q_0] \longrightarrow b\,[q_0\ a\ q_0]$

$[q_0\ b\ q_1] \longrightarrow b\,[q_0\ a\ q_1]$

(iv) $\delta(q_1, b, a) = (q_1, a)$

$[q_1 a q_2] \longrightarrow b\,[q_1 a q_0]$

$[q_1 a q_1] \longrightarrow b\,[q_1 a q_1]$

(v) $\delta(q_1, b, a) = (q_1, \varepsilon)$

$[q_1 a q_1] \longrightarrow a$

(vi) $\delta(q_1, \varepsilon, z_0) = (q_1, \varepsilon)$

$[q_1 z_0 q_1] \rightarrow \varepsilon$

(2) Convert the following PDA to CFG:-

$P = (\{q_0, q_1\}\ \{a, b\}\ \{z_0, z\},\ \delta,\ q_0,\ z_0,\ \emptyset)$

and δ is given by

$\delta(q_0; b, z_0) = (q_0, z z_0)$ $\quad$ $\delta(q_0, \varepsilon, z_0) = (q_0, \varepsilon)$

$\delta(a, b, z) = (q, \varepsilon)$ $\quad$ $\delta(q_1, a, z_0) = (q_0, z_0)$

(i) $\delta(q_0, b, z_0) = (q_0, z z_0)$

$[q_0 z_0 q_0] \rightarrow b\,[q_0 z q_0][q_0 z_0 q_0]$

$[q_0 z_0 q_0] \longrightarrow b\,[q_0 z q_1][q_1 z_0 q_0]$

$[q_0 z_0 q_1] \longrightarrow b\,[q_1 z q_0][q_0 z_0 q_1]$

$[q_0 z_0 q_1] \longrightarrow b\,[q_1 z q_1][q_1 z_0 q_1]$

(ii) $\delta(q_1, a, z_0) = (q_0, z_0)$

$[q_1 z_0 q_0] \rightarrow a [q_0 z_0 q_0]$

$[q_1 z_0 q_1] \rightarrow a [q_0 z_0 q_1]$

(iii) $\delta(q_1, b, z) = (q_1, \varepsilon)$

$[q_1 z q_1] \rightarrow b$

(iv) $\delta(q_0, \varepsilon, z_0) = (q_0, \varepsilon)$

$[q_0 z_0 q_1] \rightarrow \varepsilon$

→ Languages Accepted by PDA:-

(i) Considering Final State

(ii) Considering Empty Stack.