unit - 4

• Push Down Automata:

→ A PDA consists of 3 components:

i) Input tape
ii) Finite control
iii) Stack Structure.

→ The input tape consists of a linear configuration of cells each of which contains a character from the input Alphabet, the tape can be moved one cell at a time to the left.

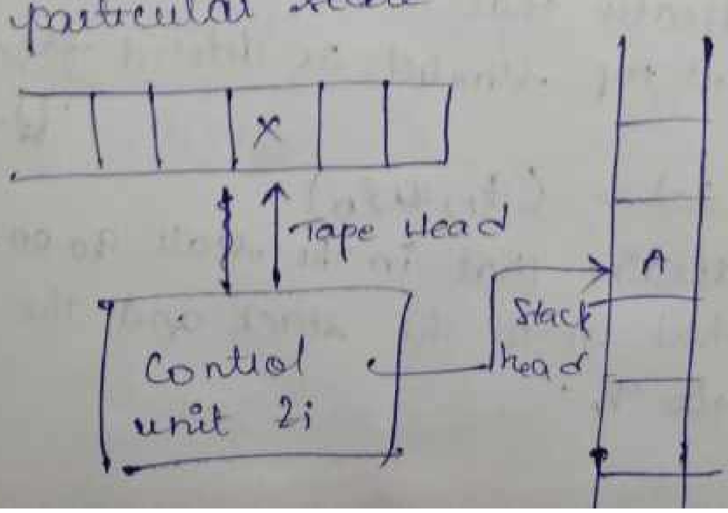
→ A stack is also a sequence structure that has a first element and grows in either directions from the other end.

→ Control unit has some pointer (head) which points the current symbol that is to be read.

→ The head position over the current stack element can read & write special stack characters from that position.

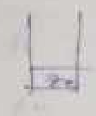
→ The current stack element is always the top element of the stack hence the name is stack.

→ The control unit contains both tape head and stack head and finds itself at any movement in a particular state.



→ A finite state PDA is a 7 tuple machine M

where $M = \{ Q, \Sigma, \delta, \Gamma, q_0, z_0, F \}$



Q = finite set of states

Σ = finite set of Input Alphabets

Γ = A finite set of stag Alphabets

q_0 = start state / Initial state

F = set of final states.

$\delta = Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$

z_0 = Initial stack symbol. $(z_0 \in \Gamma)$

A move on PDA indicates:

→ A move on PDA indicates

1) A element maybe added to the stack.
2) A element maybe deleted from the stack.
3) There may or maynot be change of stack (do nothing operation).

→ Operations:

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

→ This indicates that in the state q_0 on seeing a, a is pushed on to the stack and there is no change in the stack.

 [push]

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

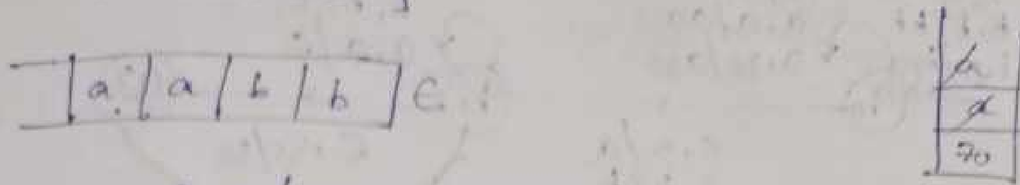
→ This indicates that in the state q_0 on seeing a, the current top symbol is deleted from the stack

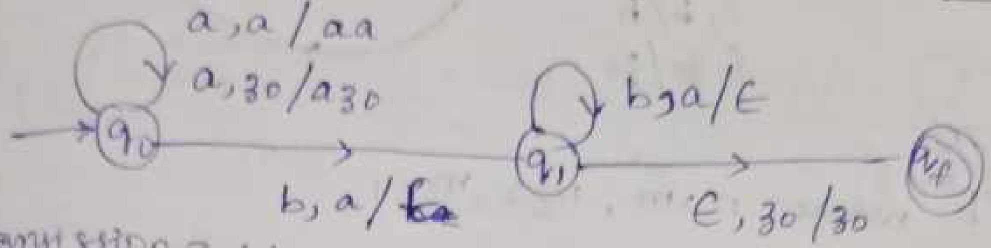
 (pop)

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

→ This indicates that in the state q_0 on seeing a, a is pushed on to the stack and the state is changed to q_1.

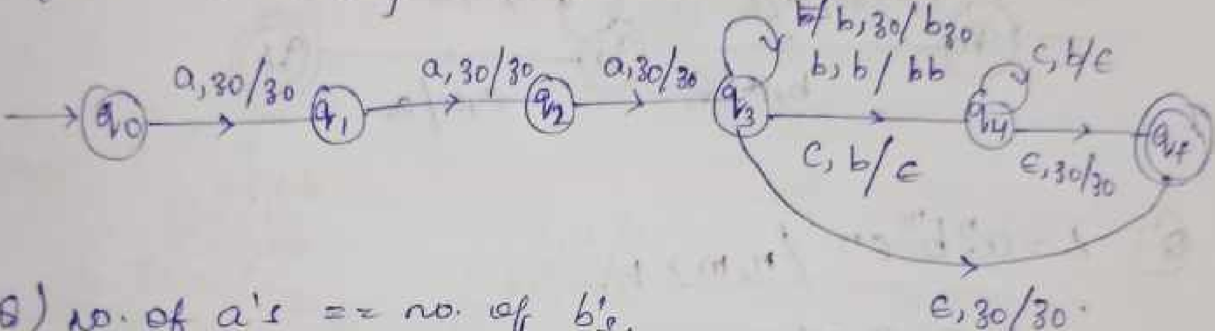
*Q) Design a PDA which accepts the language $L = \{a^n b^n, n \geq 1\}$





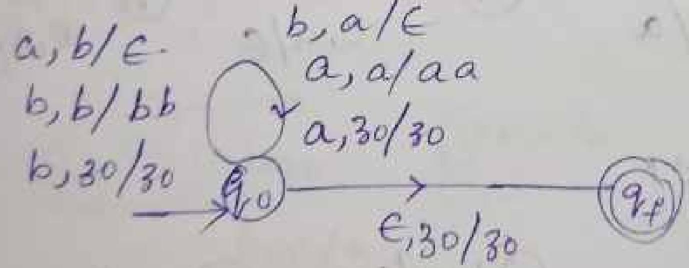
Transition Table.

Q) $\{a^3 b^n c^n / n \geq 0\}$

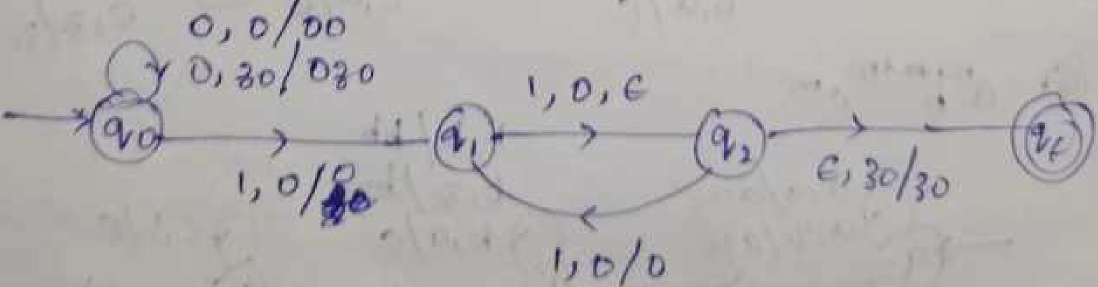


Q) no. of a's == no. of b's.

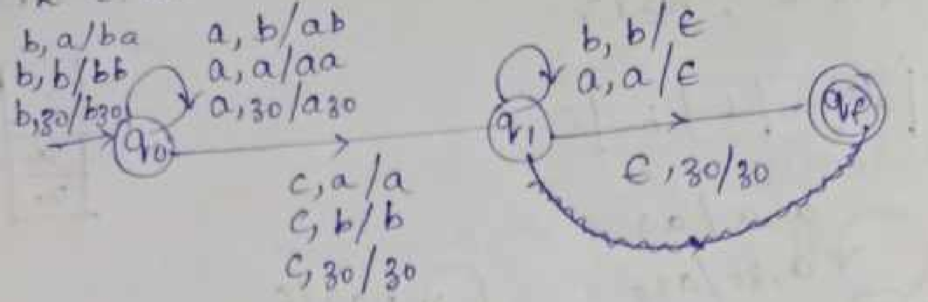
{ϵ, ab, aabb, abab, baba, ...}



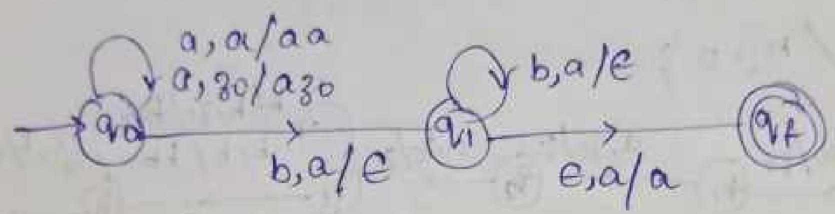
Q) Design a PDA that accepts $L = \{0^n 1^{2n}, n \geq 1\}$



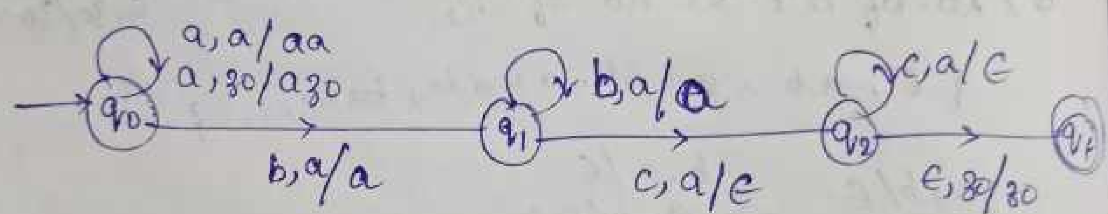
Q) $L = wcw^R$



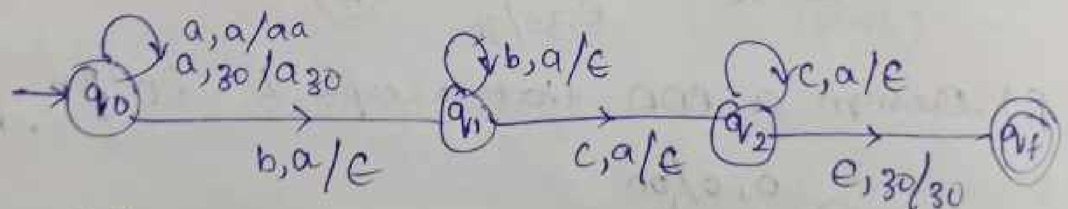
Q) $L = a^n b^m$ where $n > m$, $n \geq 2$, $m \geq 1$.



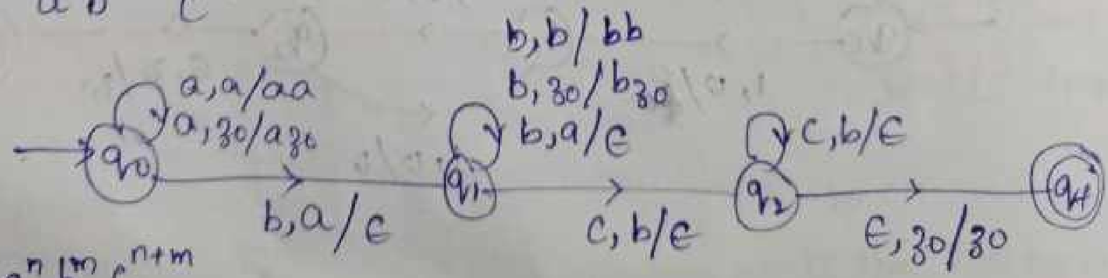
Q) $L = a^n b^m c^n$ / $n, m \geq 1$.



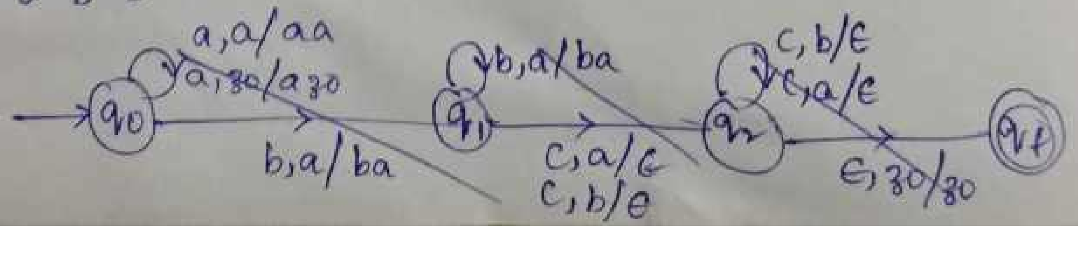
Q) $a^{m+n} b^m c^n$ / $n, m \geq 1$

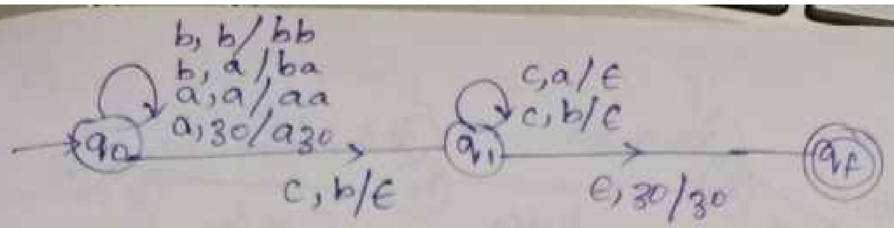


Q) $a^n b^{m+n} c^m$

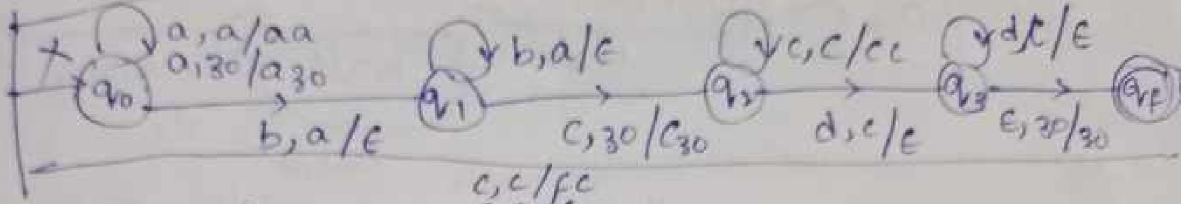


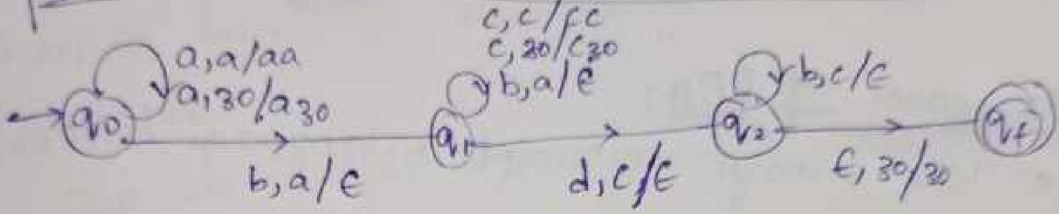
Q) $a^n b^m c^{n+m}$



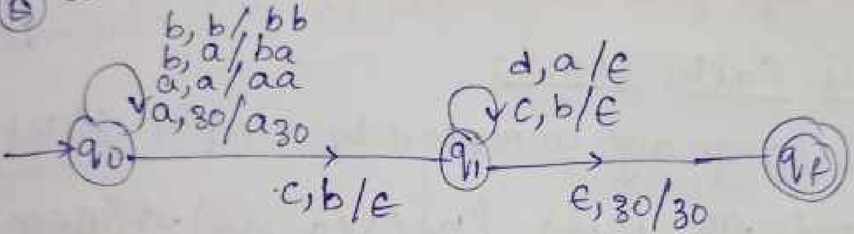


(Q) $a^m b^m c^n d^n$

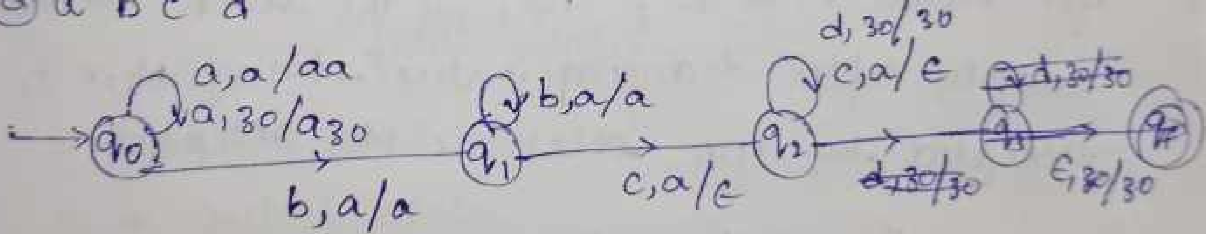




(Q) $a^m b^n c^n d^m$

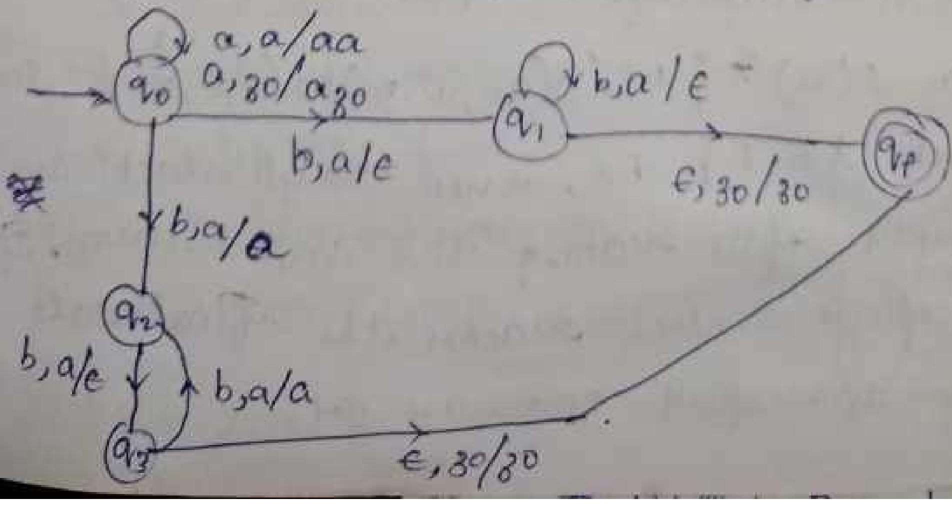


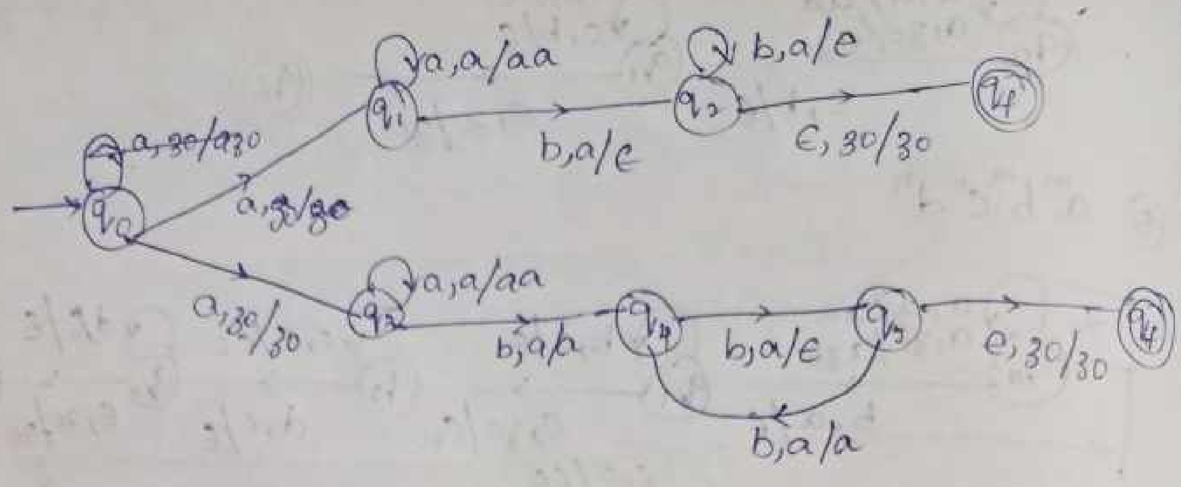
(Q) $a^m b^i c^m d^k$



(Q) $a^m b^n c^m d^n$, $a^n b^n c^n$ can't be designed by PDA.

Q) $L = \{a^n b^n / n \geq 1\} \cup \{a^n b^{2n} / n \geq 1\}$





Acceptance of PDA:

Accept a language by PDA:

→ There are 2 ways to accept

i) ~~Acceptance~~ Accepted by Empty stack.

ii) Accepted by final state.

i) Accepted by Empty stack:

→ The given language accepted by empty stack to be defined as $L(M) = \{ w / \delta(q_0, w, z_0) \Rightarrow (P, \epsilon, \epsilon)$ for some P in Q $\}$ i.e., if the stack becomes empty after scanning entire string, then it is accepted by PDA otherwise not accepted.

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

ii) Accepted by final state:

→ The given language accepted by final state is defined as $L(M) = \{ w / \delta(q_0, w, z_0) \Rightarrow (P, \epsilon, \lambda)$ for some $P \in F$ and $\lambda \in \Gamma \}$ i.e, even though stack is not empty after scanning the input string, if the ~~final~~ finite control reaches the final state then it is accepted otherwise not.

eg: $\{ a^n b^m / n > m, n \geq 2, m \geq 1 \}$

• Types of PDA:

1) DPDA (Deterministic Push Down Automata)
2) NPDA (Non- Deterministic Push Down Automata)

-> The PDA that has atmost one chance of move in any state is called a DPDA.

-> NPDA provides Non- Deterministic in the moves defined.

-> DPDA are useful in programming languages.

eg: parsars are used in yet another compiler compiler (YACC) are determined in PDA.

: DPDA:

-> A DPDA is a 7 tuple machine.

$M = \{ Q, \Sigma, q_0, \gamma, z_0, \delta, F \}$

Q = set of finite states.

Σ = set of input alphabets.

q_0 = Initial states

γ = finite set of stack alphabets

z_0 = Initial stack symbol

δ = Transition function or mapping function used for mapping current state to next state

F = set of final states

-> If a transition denotes a unique transition for each input then PDA is said to be a DPDA.

eg: $L = \{ a^n b^n / n \geq 1 \}$, $L = \{ wcw^R / w = (a+b)^* \}$

NPDA:

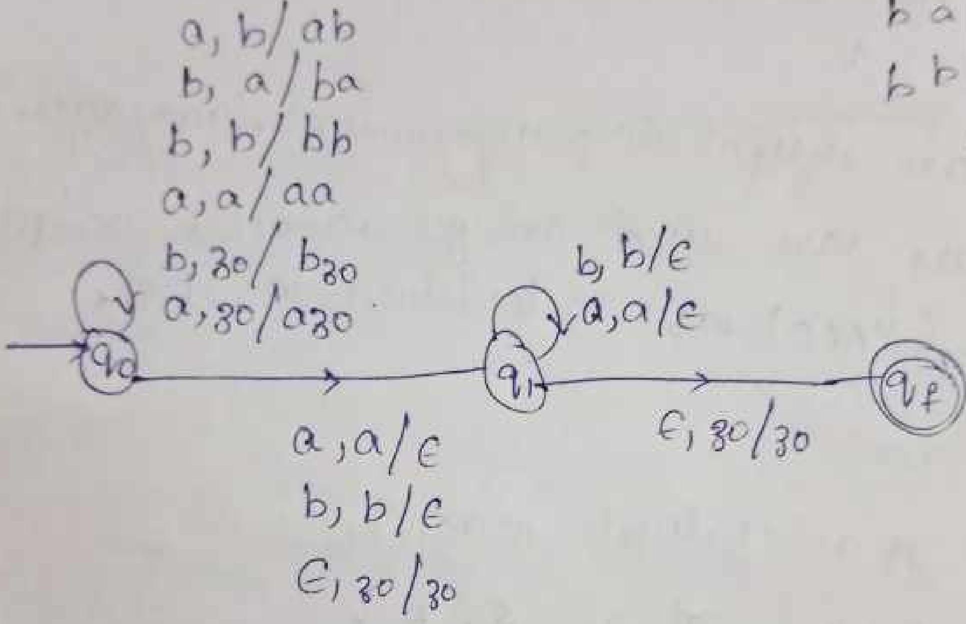
→ A NPDA is 7 tuple machine,

$M = \{Q, \Sigma, q_0, \Gamma, z_0, \delta, F\}$

→ If a transition denotes more than one transition for a particular input symbol then the PDA is said to be a NPDA.

* $L = (ww^R / w = (a+b)^*)$

a a	a a
a b	b a
b a	a b
b b	b b



- Equivalance of PDA's & CFG's:

→ Conversion of CFG to PDA:

1) For constructing a PDA from a given CFG, it is necessary to convert the CFG to some normal forms like GNF.

2) For converting a given CFG to PDA, by this method the necessary condition is that the first symbol on Right side of the production rule must be a terminal symbol. This rule that can be used to obtain PDA from CFG.

Rule-1:

→ For non-terminal symbols, $\delta(q, \epsilon, A) = (q, \alpha)$ where the production Rule is $A \longrightarrow \alpha$

(α is non-terminal)

Rule-2:

→ For each terminal symbol, $\delta(q, a, a) = (q, \epsilon)$ for every terminal symbol a in given CFG

Q) Construct a PDA equivalent to the following grammar

$$S \longrightarrow 0BB$$
$$B \longrightarrow 0S$$
$$B \longrightarrow 1S$$
$$B \longrightarrow 0$$

A) It is in minimised format.

from Rule-1: $A \rightarrow \alpha$, $\delta(q, \epsilon, A) = (q, \alpha)$.

$$\delta(q, \epsilon, S) = (q, 0BB)$$
$$\delta(q, \epsilon, B) = (q, 0S)$$
$$\delta(q, \epsilon, B) = (q, 1S)$$
$$\delta(q, \epsilon, B) = (q, 0)$$

from Rule-2:

$$\delta(q, 0, 0) = (q, \epsilon)$$
$$\delta(q, 1, 1) = (q, \epsilon)$$

(or)

$$\delta(q, 0, S) = (q, BB)$$
$$\delta(q, 0, B) = (q, S)$$
$$\delta(q, 1, B) = (q, S)$$
$$\delta(q, 0, B) = (q, \epsilon)$$

Constructing CFG for the given PDA:

→ The construction goes as follows:
given PDA (M), we will construct a grammar G such that $L(G) = L(M)$

Rule-1:

→ The Productions for start symbol S are given by $S \rightarrow [q_0, z_0, q]$ for each state q in Q

Rule-2:

→ Each move that pops a symbol from the stack with transition as $\delta(q, a, z_i) = (q_1, \epsilon)$ includes a production as $[q, z_i, q_1] \rightarrow a$ for q_1 in

Rule-3:

→ Each move that does not pop symbol from stack with transition as $\delta(q, a, z_0) = (q_1, z_1 z_2 z_3 \ldots)$

$$[q, z_0, q_m] \rightarrow a[q_1 z_1 q_2][q_2 z_2 q_3][q_3 z_3 q_4] \ldots [q_{m-1} z_m q_m]$$

eg:1) Give the equivalance CFG for the following PDA, $M = \{q_0, q_1\}, \{a, b\}, \{z, z_0\}, \delta, q_0, z_0, \phi\}$

where δ is defined by

- $\delta(q_0, b, z_0) = (q_0, z z_0)$
- $\delta(q_0, \epsilon, z_0) = (q_0, \epsilon)$
- $\delta(q_0, b, z) = (q_0, zz)$
- $\delta(q_0, a, z) = (q_1, z)$
- $\delta(q_1, b, z) = (q_1, \epsilon)$
- $\delta(q_1, a, z_0) = (q_0, z_0)$

sol)

$S \rightarrow [q_0, z_0, q_0] / [q_0, z_0, q_1]$

(i) $[q_0, z_0, q_0] \rightarrow b[q_0\ z\ q_0]\ [q_0\ z\ q_0]$

$\qquad b[q_0\ z\ q_1]\ [q_1\ z\ q_0]$

$[q_0, z_0, q_1] \rightarrow b[q_0\ z_0\ q_0]\ [q_0\ z_0\ q_1]$

$\qquad b[q_0\ z_0\ q_1]\ [q_1\ z_0 q_1]$

(ii) $[q_0, z_0, q_0] \rightarrow \epsilon$

(iii) $[q_0, z, q_0] \rightarrow b[q_0, z, q_0]\ [q_0, z, q_0]$

$\qquad b[q_0, z, q_1]\ [q_1, z, q_0]$

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

$\qquad b[q_0, z, q_1]\ [q_1, z, q_1]$

(iv) $[q_0, z, q_0] \rightarrow a[q_1, z, q_0]$

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

(v) $[q_1, z_1, q_1] \rightarrow b$

(vi) $[q_1, z_0, q_0] \rightarrow a[q_0, z_0, q_0]$

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

Q.) Convert the PDA to CFG. PDA is given by

$\{ \{P, q\}, \{0, 1\}, \{x, z\}, \delta, q, z, \phi \}$

$\delta(q, 1, z) = \{(q, xz)\}$

$\delta(q, 1, x) = \{(q, xx)\}$

$\delta(q, \epsilon, x) = \{(q, \epsilon)\}$

$\delta(q, 0, x) = \{(p, x)\}$

$\delta(p, 1, x) = \{(p, \epsilon)\}$

$\delta(p, 0, z) = \{(q, z)\}$

sol)

$S \rightarrow [q, z, p] \; [q, z, q]$

(i) $[q, z, p] \rightarrow 1[q\,x\,q]\,[q\,z\,p]$

$\quad 1[q\,x\,p]\,[p\,z\,p]$

$[q, z, q] \rightarrow 1[q\,x\,q]\,[q\,z\,q]$

$\quad 1[q\,x\,p]\,[p\,z\,q]$

(ii) $[q, x, q] \rightarrow 1[q\,x\,q]\,[q\,x\,q]$

$\quad 1[q\,x\,p]\,[p\,x\,q]$

$[q, x, p] \rightarrow 1[q\,x\,q]\,[q\,x\,p]$

$\quad 1[q\,x\,p]\,[p\,x\,p]$

(iii) $[q, x, q] \rightarrow \epsilon$

(iv) $[q, x, q] \rightarrow 0[p\,x\,q]$

$[q, x, p] \rightarrow 0[p\,x\,p]$

(v) $[p, x, p] \rightarrow 1\,\cancel{[p \quad]}\;\epsilon$

~~$[p, x, p] \rightarrow$~~ $1[p \quad]$

(vii) $[P,3,q] \rightarrow 0[q,3,q]$ ~~[ ]~~

~~$0[\quad][\quad]$~~

$[P,3,P] \rightarrow 0[q,3,P]$ ~~[ ]~~

~~$0[\quad][\quad]$~~

$S \rightarrow A/B$

$A \rightarrow 1EA / 1FC$

$B \rightarrow 1EB / 1FD$

$E \rightarrow 1EE / 1FH$

$F \rightarrow 1EF / 1FG$

$E \rightarrow \epsilon / 0H$

$F \rightarrow 0G$

$G \rightarrow 1$

$C \rightarrow 0A$

$D \rightarrow 0B$

Final Grammar:

$E \rightarrow 1EE$

$F \rightarrow 1EF / 1FG$

$E \rightarrow \epsilon$

$F \rightarrow 0G$

$G \rightarrow 1$

• Simplifying the grammar:

In the above grammar 1st identify the non terminals that are not defined and eliminate the productions that refers to these productions. Similarly, use these procedure of eliminating the useless symbols and useless production and then complete grammar is a minized grammar (By removing useless products)

First example simplifying:

$[q_0, z_0, q_0] = A$

$[q_0, z_0, q_1] = B$

$[q_0, z, q_0] = C$

$[q_0, z, q_1] = D$

$[q_1, z, q_1] = E$

$[q_1, z_0, q_1] = F$

$[q_1, z_0, q_0] = G$

$[q_1, z, q_0] = H$

$S \rightarrow A/B$

$A \rightarrow bCA / bDG$

$B \rightarrow bCB / bDF$

$A \rightarrow \epsilon$

$C \rightarrow bCC / bDH$

$D \rightarrow aE / bCD / bDE$

$E \rightarrow b$

$F \rightarrow aE$

$F \rightarrow aB$

$G \rightarrow aA$ $\quad H \rightarrow aB$

(vii) $[P, 3, q] \rightarrow 0[q, 3, q]$ ~~[ ]~~

~~0[ ] [ ]~~

$[P, 3, P] \rightarrow 0[q, 3, P]$ ~~[ ]~~

~~0[ ] [ ]~~

$S \rightarrow A / B$
$A \rightarrow 1CA / 1FC$
$B \rightarrow 1CB / 1FD$
$E \rightarrow 1EE / 1FH$
$F \rightarrow 1EF / 1FG$
$E \rightarrow E / OH$
$F \rightarrow OG$
$G \rightarrow 1$
$C \rightarrow OA$
$D \rightarrow OB$

Final Grammar:

$E \rightarrow 1EE$
$F \rightarrow 1EF / 1FG$
$E \rightarrow E$
$F \rightarrow OG$
$G \rightarrow 1$

- Simplifying the grammar:

In the above grammar 1st identify the non terminals that are not defined and eliminate the productions that refers to these productions. Similarly, use these procedure of eliminating the useless symbols and useless production and then complete grammar is a minimized grammar (By removing useless products)

First example simplifying:

$[q_0, z_0, q_0] = A$
$[q_0, z_0, q_1] = B$
$[q_0, z, q_0] = C$
$[q_0, z, q_1] = D$
$[q_1, z, q_1] = E$
$[q_1, z_0, q_1] = F$
$[q_1, z_0, q_0] = G$
$[q_1, z, q_0] = H$

$S \rightarrow A / B$
$A \rightarrow bCA / bDG$
$B \rightarrow bCB / bDF$
$A \rightarrow \epsilon$
$C \rightarrow bCC / bDH$
$D \rightarrow aC / bCD / bDE$
$E \rightarrow b$
$F \rightarrow aE$
$F \rightarrow aB$
$G \rightarrow aA$ $H \rightarrow aB$

(vi) $[P, 3, q] \rightarrow 0[q, 3, q]$ ~~[ ]~~

~~0[ ] [ ]~~

$[P, 3, P] \rightarrow 0[q, 3, P]$ ~~[ ]~~

~~0[ ] [ ]~~

$S \rightarrow A / B$

$A \rightarrow 1CA / 1FC$

$B \rightarrow 1CB / 1FD$

$E \rightarrow 1EE / 1FH$

$F \rightarrow 1EF / 1FG$

$E \rightarrow e / 0H$

$F \rightarrow 0G$

$G \rightarrow 1$

$C \rightarrow 0A$.

$D \rightarrow 0B$

Final Grammar:

$E \rightarrow 1EE$

$F \rightarrow 1EF / 1FG$

$E \rightarrow E$

$F \rightarrow 0G$

$G \rightarrow 1$

- Simplifying the grammar:

In the above grammar 1st identify the non terminals that are not defined and eliminate the productions that refers to these productions. Similarly, use these procedure of eliminating the useless symbols and useless production and then complete grammar is a minimized grammar (By removing useless products)

First example simplifying:

$[q_0, z_0, q_0] = A$

$[q_0, z_0, q_1] = B$

$[q_0, z, q_0] = C$

$[q_0, z, q_1] = D$

$[q_1, z, q_1] = E$

$[q_1, z_0, q_1] = F$

$[q_1, z_0, q_0] = G$

$[q_1, z, q_0] = H$

$S \rightarrow A / B$

$A \rightarrow bCA / bDG$

$B \rightarrow bCB / bDF$

$A \rightarrow e$

$C \rightarrow bCC / bDH$

$D \rightarrow aE / bCD / bDE$

$E \rightarrow b$

$F \rightarrow aE$

$F \rightarrow aB$

$G \rightarrow aA$

$H \rightarrow aB$

Q) $M = \{\{q_0, q_1\}, \{0,1\}, \{0,1,z_0\}, \delta, q_0, z_0, \phi\}$

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

$\delta(q_0, 0, z_0) = (q_0, 0z_0)$

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

$\delta(q_0, 1, 0) = (q_0, 10)$

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

$\delta(q_0, 0, 1) = (q_1, \epsilon)$

$\delta(q_1, 0, 1) = (q_1, \epsilon)$

$\delta(q_1, 0, 1) = (q_1, \epsilon)$

$\delta(q_1, 0, 0) = (q, \epsilon)$

$\delta(q, \epsilon, z_0) = (q, \epsilon)$

$S \rightarrow [q_0, z_0, q_0] \,/\, [q_0, z_0, q_1]$

$\rightarrow [q_0, z_0, q_1] \rightarrow \epsilon$

$[q_0, z_0, q_0] \rightarrow 0[q_0, 0, q_0][q_0, z_0, q_0] \,/$
$0[q_0, 0, q_1][q_1, z_0, q_0]$

$[q_0, z_0, q_1] \rightarrow 0[q_0, 0, q_0][q_0, z_0, q_0] \,/$
$0[q_0, 0, q_1][q_1, z_0, q_1]$

$[q_0, 0, q_0] \rightarrow 0[q_0, 0, q_0][q_0, 0, q_0] \,/$
$0[q_0, 0, q_1][q_1, 0, q_0]$

$[q_0, 0, q_1] \rightarrow 0[q_0, 0, q_0][q_0, 0, q_1]$
$0[q_0, 0, q_1][q_1, 0, q_1]$

$[q_0, 0, q_0] \rightarrow 1[q_0, 1, q_0][q_0, 0, q_0] \,/$
$1[q_0, 1, q_1][q_1, 0, q_0]$

$[q_0, 0, q_1] \rightarrow 1[q_0, 1, q_0][q_0, 0, q_1] \,/$
$1[q_0, 1, q_1][q_1, 0, q_1]$

$[q_0, 1, q_0] \rightarrow 1[q_0, 1, q_0][q_0, 1, q_0] \,/$
$1[q_0, 1, q_1][q_1, 1, q_0]$

$[q_0, 1, q_0] \rightarrow 1 [q_0, 1, q_0] [q_0, 1, q_1] /$
$1 [q_0, 1, q_1] [q_1, 1, q_1]$

$\rightarrow [q_0, 1, q_1] \rightarrow 0$
$[q_1, 1, q_1] \rightarrow 0$
$[q_1, 0, q_1] \rightarrow 0$
$[q_1, z_0, q_1] \rightarrow \epsilon$

$[q_0, z_0, q_0]$ - A
$[q_0, z_0, q_1]$ - B
$[q_0, 0, q_0]$ - C
$[q_0, 0, q_1]$ - D
$[q_0, 1, q_0]$ - E
$[q_0, 1, q_1]$ - F
$[q_1, 1, q_1]$ - G
$[q_1, 0, q_1]$ - H
$[q_1, z_0, q_1]$ - I
$[q_1, z_0, q_0]$ - J
$[q_1, 0, q_0]$ - K
$[q_1, 1, q_0]$ - L

$S \rightarrow A / B$
$B \rightarrow \epsilon$
$A \rightarrow 0CA / 0DJ$
$B \rightarrow 0CB / 0DI$
$C \rightarrow 1EC / 1FK$
$D \rightarrow 1ED / 1FG$
$E \rightarrow 1EE / 1FL$
$F \rightarrow 1EF / 1FG$
$F \rightarrow 0$
$G \rightarrow 0$
$H \rightarrow 0$
$I \rightarrow \epsilon$

Q) $r = \{\{q\}, \{i, e\}, \{x, z\}, \delta, q, z, \phi\}$
$\delta(q, i, z) = \{(q, x, z)\}$
$\delta(q, e, x) = \{(q, \epsilon)\}$
$\delta(q, \epsilon, z) = \{(q, \epsilon)\}$
$S \rightarrow [q, z, q]$
$[q, z, q] \rightarrow i [q, x, q], [q, z, q]$
$[q, x, q] \rightarrow e$
$[q, z, q] \rightarrow \epsilon$

$S \rightarrow A$

$A \rightarrow iBA$

$B \rightarrow \epsilon$

$A \rightarrow \epsilon$

All are careful grammar.