05/11/2024

Intro to Grammar:-

↓.

It is a 4 tuple representation, given by

$$\boxed{G = (V, T, P, S)}$$

where V ← finite set of variables (or) Non-terminals (uppercase Alphabets)

T ← Terminal symbols (finite set). (lowercase alphabets, Σ)

S ← starting symbol.

Imp P ← production rules.

Production rules format for regular grammar:-

Non-Terminal ⟶ Terminal. Non-Terminal

Non-Terminal ⟶ Terminal | ε.

Ex: $S \rightarrow aA$ {read as S derives a A}.

$A \rightarrow a \mid \varepsilon$

Language generated (L) is

$S \longrightarrow aa$ $\{\because A \rightarrow a\}$.

$S \longrightarrow a$ $\{\because A \rightarrow \varepsilon\}$.

$\therefore L = \{a, aa\}$.

Conversion of Grammer into DFA :-

Given production rule, (*) $q_i \longrightarrow a q_j$

$\Rightarrow$ $q_i \xrightarrow{a} q_j$

*). $q_k \longrightarrow a$.

$\Rightarrow$ $q_k \xrightarrow{a} q_f$

Imp *

*). $A \longrightarrow aB \mid a$

here, $\boxed{\begin{array}{l} A \rightarrow aB \\ A \rightarrow a \end{array}}$

$\Rightarrow$ $A \xrightarrow{a} B$

---

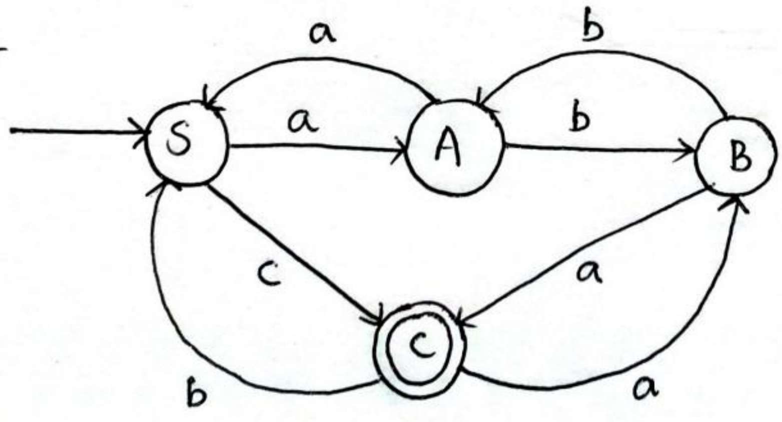
1 *). Given, $S \longrightarrow aA \mid bC \mid b$

$A \longrightarrow aS \mid bB$

$B \longrightarrow aC \mid bA \mid a$

$C \longrightarrow aB \mid bS$.

Sol:-

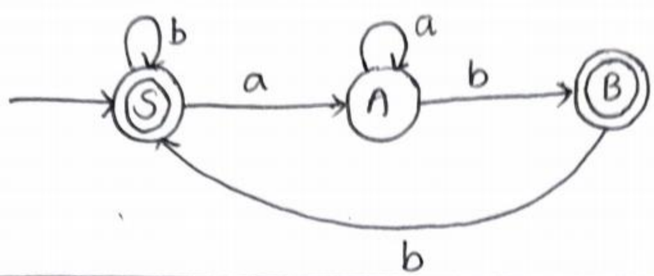


*). Given, $S \longrightarrow bS \mid aA \mid \varepsilon$

$A \longrightarrow aA \mid bB \mid b$

$B \longrightarrow bS.$

Sol:

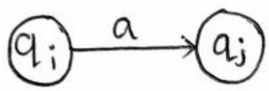


---

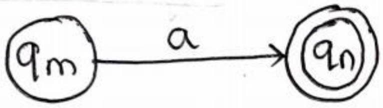
v. Imp.

Note:- $\boxed{S \longrightarrow \varepsilon, \text{ It means } S \text{ is one of the final states.}}$

Conversion of DFA into regular grammar:-

*) 

$\Rightarrow q_i \longrightarrow aq_j$

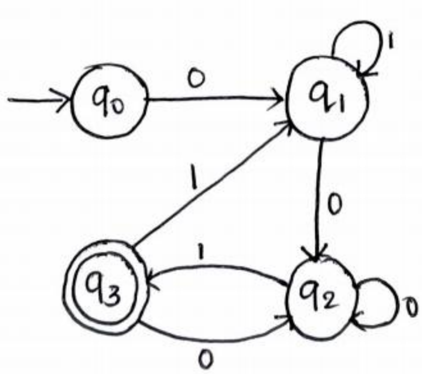
*). 

$\Rightarrow q_m \longrightarrow aq_n$ (or) $q_m \longrightarrow aq_n \mid a.$

$q_m \longrightarrow a$

---

1). Given,



Sol:-

$q_0 \rightarrow 0q_1$

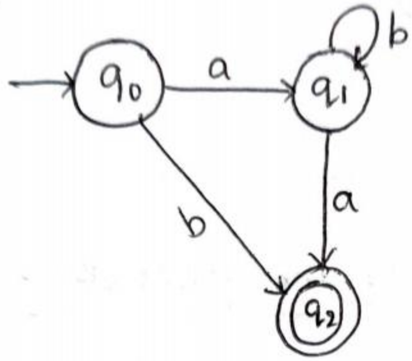
$q_1 \rightarrow 1q_1 | 0q_2$

$q_2 \rightarrow 0q_2 | 1q_3 | 1$

$q_3 \rightarrow 0q_2 | 1q_1$

---

2). Given.



Sol:-

$q_0 \rightarrow aq_1 | bq_2 | b$

$q_1 \rightarrow aq_2 | bq_1 | a$

Note:-

Regular grammar is also called as ~~watte~~ "right linear grammar"

06/11/2024

MODULE-3

06/11/2024

Context Free Grammer:-

It is a 4 tuple representation, given by

$$\boxed{G_{CFG} = (V, T, P, S)}$$

where, $V \rightarrow$ set of finite variables (or) non-terminals

$T \rightarrow$ finite set of terminals.

$S \rightarrow$ start symbol

$P \rightarrow$ Production rules,

such that, if

$$\alpha \longrightarrow \beta$$

then, $\alpha \in V$ and

$$\beta \in (V \cup T)^*$$

---

1). Given,

L = strings having any no of a's over $\Sigma = \{a\}$.

Sol: R.E = a^* * {if possible, write R.E, else not required}

$L = \{\varepsilon, a, aa, aaa, aaaa, \ldots\}$

here $S \longrightarrow \varepsilon$

$S \longrightarrow aS$ $\Rightarrow$ $\boxed{S \longrightarrow aS \mid \varepsilon}$ is the reqd. CFG.

where, $V = \{S\}$

$T = \{a\}$

$S = S$

$P = \{S \longrightarrow aS$

$S \longrightarrow \varepsilon\}$

Consider aaaa,

$\Rightarrow S \quad [S \rightarrow aS]$
$\Rightarrow aS \quad [S \rightarrow aS]$
$\Rightarrow aaS \quad [S \rightarrow aS]$
$\Rightarrow aaaS \quad [S \rightarrow aS]$
$\Rightarrow aaaa \quad [S \rightarrow \varepsilon]$

is called as derivation.

---

2). Given,

L = strings having any no of a's (or) b's over $\Sigma = \{a, b\}$

Sol: $L = \{\varepsilon, a, b, aa, bb, ab, ba, aaa, aab, aba, bbb, bab, bba, \ldots\}$

R.E = $(a+b)^*$ (or) $a^* b^*$

$S \rightarrow \varepsilon$
$S \rightarrow aS$ $\Rightarrow$ $\boxed{S \rightarrow aS | bS | \varepsilon}$ is the reqd CFG.
$S \rightarrow bS$

where $V = \{S\}$
$T = \{a, b\}$
$S = S$
$P = \{S \rightarrow aS$
$S \rightarrow bS$
$S \rightarrow \varepsilon\}$.

Consider baaba,

$\Rightarrow S$
$\Rightarrow bS \quad [S \rightarrow bS]$
$\Rightarrow baS \quad [S \rightarrow aS]$
$\Rightarrow baaS \quad [S \rightarrow aS]$
$\Rightarrow baabS \quad [S \rightarrow bS]$
$\Rightarrow baabaS \quad [S \rightarrow aS]$
$\Rightarrow baaba \quad [S \rightarrow \varepsilon]$

3). Given.

L = strings having atleast two a's over $\Sigma = \{a, b\}$

Sol: $L = \{aa, baa, aaa, baba, baaa, \ldots\}$

R.E = $(a+b)^* a (a+b)^* a (a+b)^*$

$$\boxed{\begin{array}{l} S \rightarrow AaAaA \\ A \rightarrow aA \mid bA \mid \varepsilon \end{array}}$$ is the reqd CFG.

here $V = \{S, A\}$

$T = \{a, b\}$

$S = S$

$P = \{ S \rightarrow AaAaA$

$A \rightarrow aA$

$A \rightarrow bA$

$A \rightarrow \varepsilon \}$

consider baaa.

$\Rightarrow S$

$\Rightarrow AaAaA \quad [S \rightarrow AaAaA]$

$\Rightarrow bAaAaA \quad [A \rightarrow bA]$

$\Rightarrow baAaA \quad [A \rightarrow \varepsilon]$

$\Rightarrow baaA \quad [A \rightarrow \varepsilon]$

$\Rightarrow baaaA \quad [A \rightarrow aA]$

$\Rightarrow baaa \quad [A \rightarrow \varepsilon]$

=.

consider baba, (another string).

$\Rightarrow S.$

$\Rightarrow AaAaA.$ [$S \rightarrow AaAaA$]

$\Rightarrow bAaAaA$ [$A \rightarrow bA$]

$\Rightarrow baAaA$ [$A \rightarrow \varepsilon$]

$\Rightarrow babAaA$ [$A \rightarrow bA$]

$\Rightarrow babaA$ [$A \rightarrow \varepsilon$]

$\Rightarrow baba$ [$A \rightarrow \varepsilon$]

---

4) Given,

L = strings having atleast one occurence of 000 over $\Sigma = \{0,1\}$

Soln: R.E = $(0+1)^* (000)^+ (0+1)^*$

$L = \{000, 00000, 1000, 0001, 00001, 10001, \ldots\}$

$$\boxed{\begin{array}{l} S \rightarrow A000A \\ A \rightarrow 0A \mid 1A \mid \varepsilon \end{array}}$$ is the reqd CFG.

where $V = \{S, A\}$

$T = \{0,1\}$

$S = S$

$P = \{S \rightarrow A000A$

$A \rightarrow 0A$

$A \rightarrow 1A$

$A \rightarrow \varepsilon\}$

consider, 10001

$\Rightarrow S$

$\Rightarrow A000A$ [$S \to A000A$]

$\Rightarrow 1A000A$ [$A \to 1A$]

$\Rightarrow 1000A$ [$A \to \varepsilon$]

$\Rightarrow 10001A$ [$A \to 1A$]

$\Rightarrow 10001$ [$A \to \varepsilon$]

---

5) Given,

L = strings having equal no of a's and b's.

(or)

$L = \{w : n_a(w) = n_b(w)\}$.

Sol:- $L = \{\varepsilon, ab, aabb, aaabbb, \text{-----}\}$.

~~R.E= (ab)* (ba)*~~ no regular expression possible.

$\boxed{S \to aSb \mid bSa \mid SS \mid \varepsilon}$ is the reqd. CFG.

where $V = \{S\}$

$T = \{a, b\}$

$S = S$

$P = \{S \to aSb$

$S \to bSa$

$S \to SS$

$S \to \varepsilon\}$

---

consider abbaab

$\Rightarrow S$

$\Rightarrow aSb$ [$S \to aSb$]

$\Rightarrow abSab$ [$S \to bSa$]

$\Rightarrow abbSaab$ [$S \rightarrow bSa$]

$\Rightarrow abbaab$ [$S \rightarrow \varepsilon$]

Consider abba,

$\Rightarrow S$

$\Rightarrow SS$ [$S \rightarrow SS$]

$\Rightarrow aSbS$ [$S \rightarrow aSb$]

$\Rightarrow abS$ [$S \rightarrow \varepsilon$]

$\Rightarrow abbSa$ [$S \rightarrow bSa$]

$\Rightarrow abba$ [$S \rightarrow \varepsilon$]

Consider ababba.

~~$\Rightarrow S$~~

~~$\Rightarrow aSb$ [$S \rightarrow aSb$]~~

~~$\Rightarrow abSab$ [$S \rightarrow bSa$]~~

$\Rightarrow S$

$\Rightarrow SS$

$\Rightarrow aSbS$ [$S \rightarrow aSb$]

$\Rightarrow abSabS$ [$S \rightarrow bSa$]

$\Rightarrow ababS$ [$S \rightarrow \varepsilon$]

$\Rightarrow ababbSa$ [$S \rightarrow bSa$]

$\Rightarrow ababba$ [$S \rightarrow \varepsilon$]

07/11/2024

6) Given,

$L = \{w : w$ has substring $ab\}$.

Sol: $L = \{ab, aab, abab, abbb, aaab, \ldots\}$.

R.E $= (a+b)^* ab (a+b)^*$

$\boxed{\begin{array}{l} S \rightarrow AabA \\ A \rightarrow aA \mid bA \mid \varepsilon \end{array}}$ is the reqd. CFG.

where $V = \{S, A\}$

$T = \{a, b\}$

$S = S$

$P = \{ S \rightarrow AabA$

$A \rightarrow aA$

$A \rightarrow bA$

$A \rightarrow \varepsilon \}$

consider, aaabbb

$\Rightarrow S$

$\Rightarrow AabA$ $[S \rightarrow AabA]$

$\Rightarrow aAabA$ $[S \rightarrow aA]$

$\Rightarrow aaAabA$ $[S \rightarrow aA]$

$\Rightarrow aaabA$ $[S \rightarrow \varepsilon]$

$\Rightarrow aaabbA$ $[S \rightarrow bA]$

$\Rightarrow aaabbbA$ $[S \rightarrow bA]$

$\Rightarrow aaabbb$ $[S \rightarrow \varepsilon]$

7). Given,

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

(or)

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

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

$\boxed{S \longrightarrow aSb \mid ab}$ is the reqd CFG.

where $V = \{S\}$

$T = \{a, b\}$

$S = S$

$P = \{ S \rightarrow aSb$

$S \rightarrow ab \}$

---

8). Given,

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

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

$\boxed{S \rightarrow aSb \mid \varepsilon}$ is the reqd CFG

where, $V = \{S\}$

$T = \{a, b\}$

$S = S$

$P = \{ S \rightarrow aSb$

$S \rightarrow \varepsilon \}$

a) Given,

$L = \{a^n b^m : m > n\}$

Sol:- since, $m > n$

$\Rightarrow$ Let $m = n + x$, $x > 0$.

i.e., $a^n b^m = a^n b^{n+x}$

$$= \underbrace{a^n . b^n}_{A} . \underbrace{b^x}_{B}$$

$$\boxed{\begin{aligned} S &\rightarrow A.B \\ A &\rightarrow aAb \mid ab \\ B &\rightarrow bB \mid b \end{aligned}}$$

is the reqd CFG.

where $V = \{S, A, B\}$

$T = \{a, b\}$

$S = S$

$P = \{ S \rightarrow AB$

$A \rightarrow aAb$

$A \rightarrow ab$

$B \rightarrow bB$

$B \rightarrow b \}$

10) Given,

$L = \{a^n b^m : m < n\}$

Sol: $\because m < n$

$\Rightarrow$ Let $n = m + x$, $x > 0$

i.e., $a^n . b^m = a^{m+x} . b^m$

$= \cancel{\underbrace{a^m . b^m}_{A} . \underbrace{a^x}_{B}}$ $\quad \underbrace{a^x}_{A} . \underbrace{a^m . b^m}_{B}$

$$\boxed{\begin{aligned} S &\rightarrow A.B \\ A &\rightarrow aA \mid a \\ B &\rightarrow aBb \mid ab \end{aligned}}$$

is the reqd CFG.

where $V = \{S, A, B\}$

$T = \{a, b\}$

$S = S$.

$P = \{S \rightarrow A.B$

$A \rightarrow aA$

$A \rightarrow a$

$B \rightarrow aBb$

$B \rightarrow ab\}$.

11) Given,

$L = \{a^n b^m : m \neq n\}$

Sol: $\because m \neq n \Rightarrow m > n$ (or) $m < n$.

$$\Rightarrow \boxed{\begin{array}{l} S \rightarrow S1 \mid S2 \\ S1 \rightarrow AB \\ A \rightarrow aAb \mid ab \\ B \rightarrow bB \mid b \\ S2 \rightarrow CD \\ C \rightarrow aC \mid a \\ D \rightarrow aDb \mid ab \end{array}}$$ is the reqd CFG.

where $V = \{S, S1, A, B, S2, C, D\}$

$T = \{a, b\}$

$S = S$

$P = \{ S \rightarrow S1 \mid S2$

$S1 \rightarrow AB$

$A \rightarrow aAb$

$A \rightarrow ab$

$B \rightarrow bB$

$B \rightarrow b$

$S2 \rightarrow CD$

$C \rightarrow aC$

$C \rightarrow a$

$D \rightarrow aDb$

$D \rightarrow ab \}$

12). Given,

$L = \{a^n b^{2n} : n \geq 1\}$.

Sol:- $L = \{abb, aabbbb, \ldots\}$.

$\boxed{S \to aSbb \mid abb}$ is the reqd CFG.

where $V = \{S\}$

$T = \{a, b\}$

$S = S$

$P = \{S \to aSbb$

$S \to abb\}$

Imp.

13). Given,

$L = \{a^i b^j c^k : k = i + j, (i, j \geq 0)\}$

Note:-

{If $i, j \geq 1$ then replace $\varepsilon \to bc$}.

Sol:- $a^i b^j c^k = a^i . b^j . c^{i+j}$

$= a^i b^j . c^i . c^j$

$= a^i b^j c^j c^i$

same no

$$\boxed{\begin{array}{l} S \to aSc \mid T \\ T \to bTc \mid \varepsilon \end{array}}$$

is the reqd. CFG.

where $V = \{S, T\}$

$T = \{a, b, c\}$

$S = S$

$P = \{S \to aSc$

$S \to T$

$T \to bTc$

$T \to \varepsilon\}$

Imp
14). Given, 11/11/2024

$L = \{w : w \in (a,b)^*$ and w is palendrome of even length$\}$
$\hookrightarrow (ww^R)$

Sol:- $L = \{aa, bb, ababbaba, \ldots\}$

$\boxed{S \rightarrow aSa \mid bSb \mid \epsilon}$ is the reqd. CFG.

Consider, (abbbba)

$\Rightarrow S$
$\Rightarrow aSa \quad [S \rightarrow aSa]$
$\Rightarrow abSba \quad [S \rightarrow bSb]$
$\Rightarrow abbSbba \quad [S \rightarrow bSb]$
$\Rightarrow abbbba \quad [S \rightarrow \epsilon]$

---

15). Given,

$L = \{w : w \in (a,b)^*$ and w is palendrome of odd length$\}$.

Sol:- $L = \{aaa, aba, ababa, \ldots\}$

$\boxed{S \rightarrow aSa \mid bSb \mid a \mid b}$ is the reqd CFG.

where $V = \{S\}$
$T = \{a, b\}$
$S = S$
$P = \{S \rightarrow aSa$
$S \rightarrow bSb$
$S \rightarrow a$
$S \rightarrow b\}$

Consider, abbabba

$\Rightarrow S$

$\Rightarrow aSa \quad [S \to aSa]$

$\Rightarrow abSba \quad [S \to bSb]$

$\Rightarrow abbSbba \quad [S \to bSb]$

$\Rightarrow abbabba \quad [S \to a]$

16) Given,

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

Solⁿ: $L = \{aca, bcb, abcba, \ldots\}$

$\boxed{S \to aSa \mid bSb \mid c}$ is the reqd CFG.

where $V = \{S\}$

$T = \{a, b, c\}$

$S = S$

$P = \{S \to aSa,$

$S \to bSb,$

$S \to c\}$

consider, abcba

$\Rightarrow S$

$\Rightarrow aSa \quad [S \to aSa]$

$\Rightarrow abSba \quad [S \to bSb]$

$\Rightarrow abcba \quad [S \to c]$

17). Given,

$L = \{w : w \in (a,b)^* \text{ and } w \text{ is palindrome}\}$

Sol:- $L = \{aba, aabaa, aabbaa, \ldots\}$

$\boxed{S \rightarrow aSa \mid bSb \mid a \mid b \mid \varepsilon}$ is the reqd CFG.

where, $V = \{S\}$

$T = \{a, b\}$

$S = S$

$P = \{S \rightarrow aSa$

$S \rightarrow bSb$

$S \rightarrow a$

$S \rightarrow b$

$S \rightarrow \varepsilon\}$

consider, abbba

$\Rightarrow S$

$\Rightarrow aSa \quad [S \rightarrow aSa]$

$\Rightarrow abSba \quad [S \rightarrow bSb]$

$\Rightarrow abbba \quad [S \rightarrow b]$

---

18). Given,

$L = \{w : w \in (a,b)^* \text{ and } w \text{ is palindrome of even length and } |w| > 0\}$

Sol:- $L = \{aa, bb, aabbaa, \ldots\}$

$\boxed{S \rightarrow aSa \mid bSb \mid aa \mid bb}$ is the reqd. CFG where, $V = \{S\}$

$T = \{a, b\}$

$S = S$

$P = \{ S \to aSa$

$S \to bSb$

$S \to aa$

$S \to bb \}$

consider, aaabbbbbbaaa

$\Rightarrow \quad S$

$\Rightarrow \quad aSa \qquad [S \to aSa]$

$\Rightarrow \quad aaSaa \qquad [S \to aSa]$

$\Rightarrow \quad aaaSaaa \qquad [S \to aSa]$

$\Rightarrow aaabSbaaa \qquad [S \to bSb]$

$\Rightarrow aaabbSbbaa \qquad [S \to bSb]$

$\Rightarrow asabbbbaa \qquad [S \to bb]$

---

19). Given,

$L = \{a^m b^n : m = 2n+3\}$ $\quad$ {same for $n \geq 0$, $n \geq 1$}.

Sol: $\because m = 2n+3$

$\Rightarrow a^m b^n = a^{2n+3} \cdot b^n$

$= a^{2n} \cdot a^3 \cdot b^n$

$= a^3 \cdot a^{2n} b^n$

$$\boxed{\begin{aligned} S &\to aaSb \mid A \\ A &\to aaa \end{aligned}}$$ is the reqd CFG.

where $V = \{S, A\}$, $T = \{a, b\}$, $S = S$, $P = \{S \to aaSb$

$S \to A$

$A \to aaa\}$

20) Given,

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

Sol: $L = \{abc, abbc, aabbbcc, \ldots\}$

$$\boxed{\begin{array}{l} S \rightarrow aSc \mid aBc \\ B \rightarrow bB \mid b \end{array}}$$ is the reqd CFG.

where, $V = \{S, B\}$

$T = \{a, b, c\}$

$S = S$

$P = \{ S \rightarrow aSc$

$S \rightarrow aBc$

$B \rightarrow bB$

$B \rightarrow b \}$

Consider aabbbcc

$\Rightarrow$ S.

$\Rightarrow$ aSc [$S \rightarrow aSc$]

$\Rightarrow$ aaBac [$S \rightarrow aBc$]

$\Rightarrow$ aabBac [$S \rightarrow bB$]

$\Rightarrow$ aabbBac [$S \rightarrow bB$]

$\Rightarrow$ aabbbac [$S \rightarrow b$]

21). Given,

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

Sol: $L = \{abcd, aabcdd, abbccd, \ldots\}$

$$\boxed{\begin{array}{l} S \rightarrow aSd \mid aAd \\ A \rightarrow bAc \mid bc \end{array}}$$

is the reqd. CFG.

where, $V = \{S, A\}$

$T = \{a, b, c, d\}$

$S = S$

$P = \{ S \rightarrow aSd$

$S \rightarrow aAd$

$A \rightarrow bAc$

$A \rightarrow bc \}$

---

22). Given,

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

Soln $L = \{abcd, aabbcd, abccdd, \ldots\}$.

Wrong

$$\boxed{\begin{array}{l} S \rightarrow aSbA \mid abA \\ A \rightarrow cAd \mid cd \end{array}}$$

is the reqd CFG

aSbA

aabAbA

cd

where, $V = \{S, A\}$

$T = \{a, b, c, d\}$

$S = S$

$P = \{S \rightarrow aSbA$

$S \rightarrow ab$

$A \rightarrow cAd$

$A \rightarrow cd \}$

~~(or)~~

$$\boxed{\begin{array}{l} S \longrightarrow A.B \\ A \rightarrow aAb \mid ab \\ B \rightarrow cBd \mid cd \end{array}}$$ is the reqd CFG.

where $V = \{S, A, B\}$

$T = \{a, b, c, d\}$

$S = S$

$P = \{S \rightarrow A.B$

$A \rightarrow aAb$

$A \rightarrow ab$

$B \rightarrow cBd$

$B \rightarrow cd\}$.

---

23). Given,

$0^* 1 (0+1)^*$.

Sol:- $L = \{1, 01, 010, 011, 00111, \ldots\}$.

$$\boxed{\begin{array}{l} S \longrightarrow A1B \\ A \longrightarrow 0A \mid \varepsilon \\ B \rightarrow 0B \mid 1B \mid \varepsilon \end{array}}$$ is the reqd CFG.

where $V = \{S, A, B\}$

$T = \{0, 1\}$

$S = S$

$P = \{S \rightarrow A1B$

$A \rightarrow 0A$

$A \rightarrow \varepsilon$

$B \rightarrow 0B$

$B \rightarrow 1B$

$B \rightarrow \varepsilon\}$

24). Given,

$L = \{0^i 1^j 2^k : i=j \text{ (or) } j=k, i,j,k \geq 0\}$.

Soln: Consider,

$L_1 = \{0^i 1^j 2^k : i=j \text{ and } i,j,k \geq 0\}$.

$L_2 = \{0^i 1^j 2^k : j=k \text{ and } i,j,k \geq 0\}$.

Such that $L = L_1 \cup L_2$.

Now,

$S \longrightarrow A \mid B$.
$A \longrightarrow C.D$
$C \rightarrow 0C1 \mid \varepsilon$
$D \rightarrow 2D \mid \varepsilon$
$B \longrightarrow E.F$
$E \longrightarrow 0E \mid \varepsilon$
$F \rightarrow 1F2 \mid \varepsilon$

(or)

$S \rightarrow A.B \mid C.D$
$A \rightarrow 0A1 \mid \varepsilon$
$B \rightarrow 2B \mid \varepsilon$
$C \rightarrow 0C \mid \varepsilon$
$D \rightarrow 1D2 \mid \varepsilon$

is the reqd CFG.

where, $V = \{S, A, B, C, D, E, F\}$.

$T = \{0, 1, 2\}$

$S = S$

$P = \{ S \rightarrow A$

$S \rightarrow B$

$A \rightarrow C.D$

$C \rightarrow 0C1$

$C \rightarrow \varepsilon$

$D \rightarrow 2D$

$D \rightarrow \varepsilon$

$B \rightarrow E.F$

$E \rightarrow 0E$

$E \rightarrow \varepsilon$

$F \rightarrow 1F2$

$F \rightarrow \varepsilon \}$

25) Write a CFG for balanced parenthesis.

Sol: $\boxed{S \to (S) \mid () \mid SS}$ is the reqd. CFG.

where, $V = \{S\}$

$T = \{(,)\}$.

$S = S$

$P = \{S \to (S)$

$S \to ()$

$S \to SS\}$.

Note: $()() \to SS$

$((())) \to (S)$

---

26) Given,

$L = \{a^i b^j c^k : i \neq j \text{ and } i, j, k \geq 1\}$.

Sol: Consider,

$L_1 = \{a^i b^j c^k : i \neq j \text{ and } i \ngtr j \text{ and } i, j, k \geq 1\}$

$L_2 = \{a^i b^j c^k : i \neq j \text{ and } i \nless j \text{ and } i, j, k \geq 1\}$

such that $L_1 \cup L_2$.

$$\boxed{\begin{aligned} &S \to ABC \mid DEE \\ &A \to aA \mid a \\ &B \to aBb \mid ab \\ &C \to cB \mid c \\ &D \to aDb \mid ab \\ &E \to bE \mid b \end{aligned}}$$

is the reqd. CFG.

Derivation and Parse Tree:- (6-12m).

Derivation:- The process of deriving a given string from the start symbol.

Parse Tree:-

*) Also known as derivation tree

*) It is the graphical representation of the derivation.

Properties of parse tree:-

*) Root node should be the start symbol of the grammer.

*) Leaf node is made up of terminals.

*) Internal node is made up of variables.

*) Parse tree should be read from left to right.

Ex:- Given, production rules

$S \rightarrow AB$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

Now, consider string aabb

Derivation:- $\Rightarrow S$

$\Rightarrow AB \quad (S \rightarrow AB)$

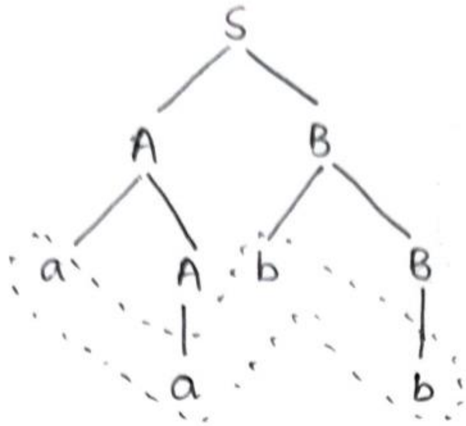
$\Rightarrow aAB \quad (A \rightarrow aA)$

$\Rightarrow aaB \quad (A \rightarrow a)$

$\Rightarrow aabB \quad (B \rightarrow bB)$

$\Rightarrow aabb \quad (B \rightarrow b)$

Parse tree:-



aabb ~~is~~ are the leaf nodes

Imp.

1) Given,

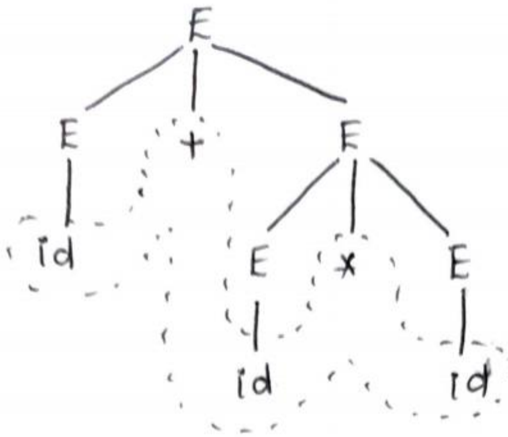
$E \longrightarrow E+E \mid E*E \mid id$

string : id + id * id (required).

Sol:-

Derivation	Applicable rule.
E.	
E+E	$E \longrightarrow E+E$
id+E	$E \longrightarrow id$
id + E*E	$E \longrightarrow E*E$
id + id * E	$E \longrightarrow id$
id + id * id	$E \longrightarrow id$

Parse tree:-



$\Rightarrow$ id+id*id

---

2) Given,

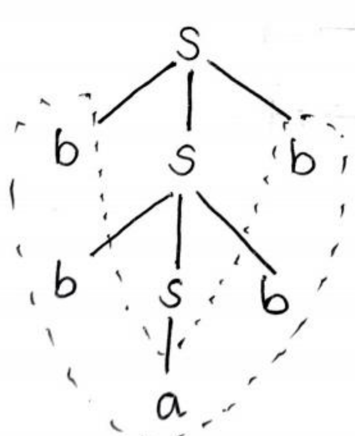
$S \rightarrow bSb \mid a \mid b$

String $\rightarrow$ bbabb.

Sol:-

Derivation	Applicable rule.
S	
b S b	$S \rightarrow bSb$.
b bS bb	$S \rightarrow bSb$
b babb	$S \rightarrow a$

Parse tree:-



$\Rightarrow$ bbabb

3) Given,

$S \to AB \mid \varepsilon$

$A \to aB$

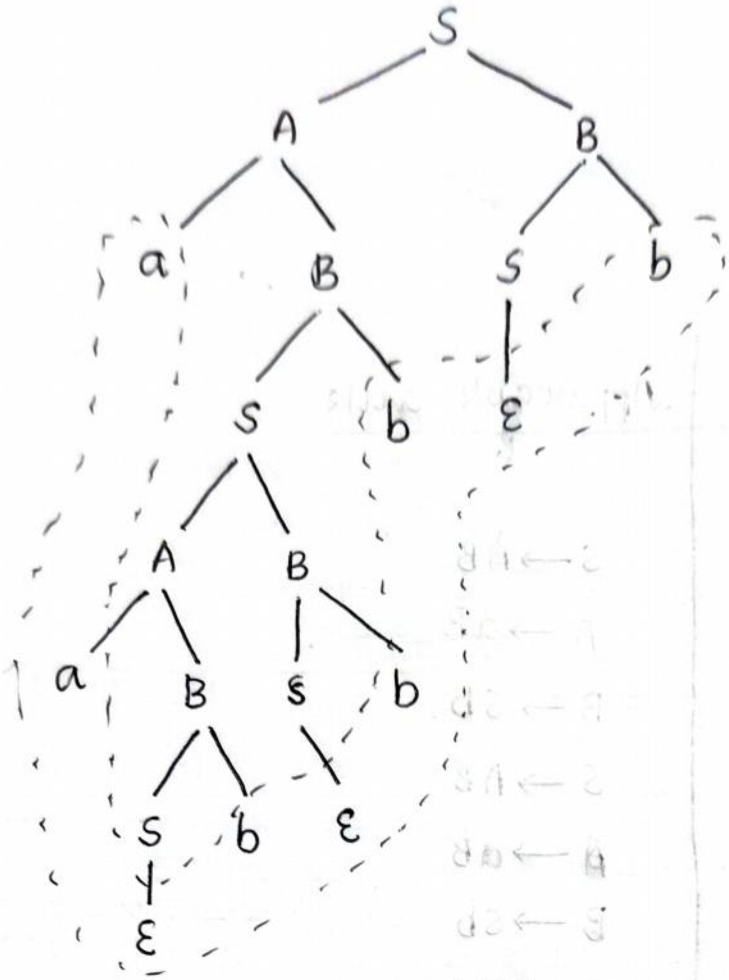
$B \to Sb$

String $\to$ aabbbb

Sol:-

Derivation	Applicable rule
S	
AB	$S \to AB$
aBB	$A \to aB$
$aSbB$	$B \to Sb$
$aABbB$	$S \to AB$
$aaBBbB$	$A \to aB$
$aaSbBbB$	$B \to Sb$
$aabBbB$	$S \to \varepsilon$
$aabSbbB$	$B \to Sb$
$aabbbB$	$S \to \varepsilon$
$aabbbSb$	$B \to Sb$
$aabbbb$	$S \to \varepsilon$

Parse tree:-



$\Rightarrow aabbbb$

Leftmost Derivation (LMD) & Rightmost Derivation (RMD):-

LMD:- In a derivation process, if you are replacing leftmost variable in each step, then it is LMD.

RMD:- In a derivation process, if you are replacing rightmost variable in each step, then it is RMD.

---

1). Given,

$E \longrightarrow E+E \mid E*E \mid E-E \mid id$

String: $id - id * id$

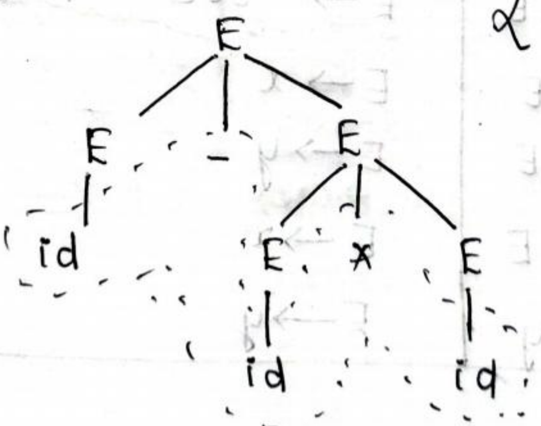
Sol:- LMD:-

Derivation	Applicable rule
E	
E-E	E → E-E
id-E	E → id
id-E*E	E → E*E
id-id*E	E → id
id-id*id	E → id

RMD:-

Derivation	Applicable rule
E	
E-E	E → E-E
E-E*E	E → E*E
E-E*id	E → id
E-id*id	E → id
id-id*id	E → id

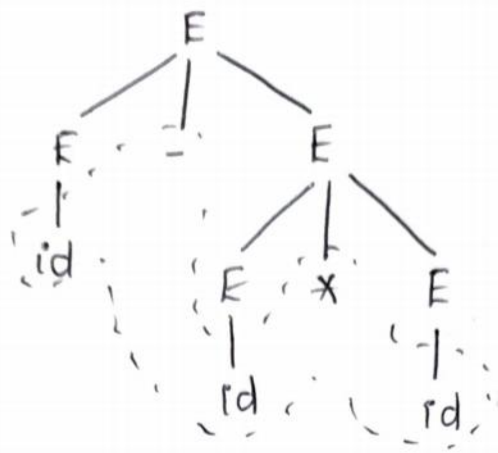
LMD parse tree:



{anyone LMD/RMD parse tree is enough}.

id-id*id

RMD parse tree:



id-id*id

2). Given,

$E \rightarrow +EE \mid -EE \mid *EE \mid x \mid y$

String → +*-xyxy.

Sol:-

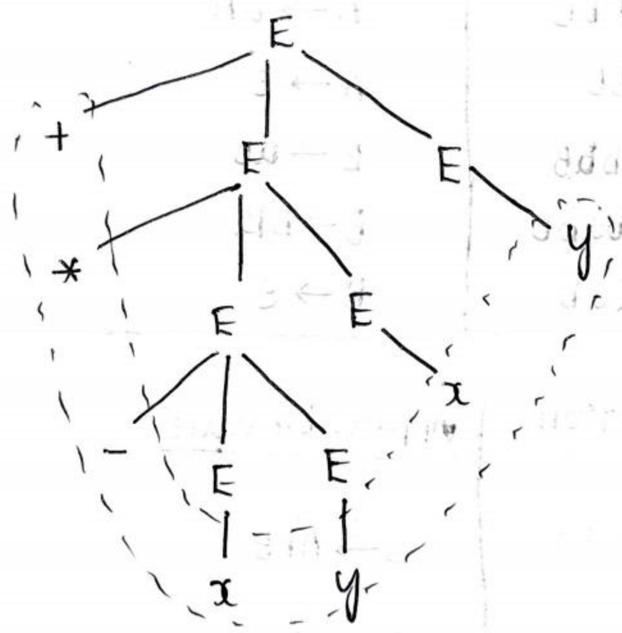
LMD:-

Derivation	Applicable rule
E	
+EE	E→ +EE
+*EEE	E→ *EE
+*-EEEE	E→ -EEE
+*-xEEE	E→ x
+*-xyEE	E→ y
+*-xyxE	E→ x
+*-xyxy	E→ y

RMD:-

Derivation	Applicable rule.
E	
+EE	E → +EE
+Ey	E → y
+*EEy	E → *EE
+*Exy	E → x
+*-EExy	E → -EE
+*-Eyxy	E → y
+*-xyxy	E → x

LMD sparse tree:



⇒ +*-xyxy

12/11/2024

3). Given,

$S \rightarrow AbB$

$A \rightarrow aA | \varepsilon$

$B \rightarrow aB | bB | \varepsilon$

String: aaabab.

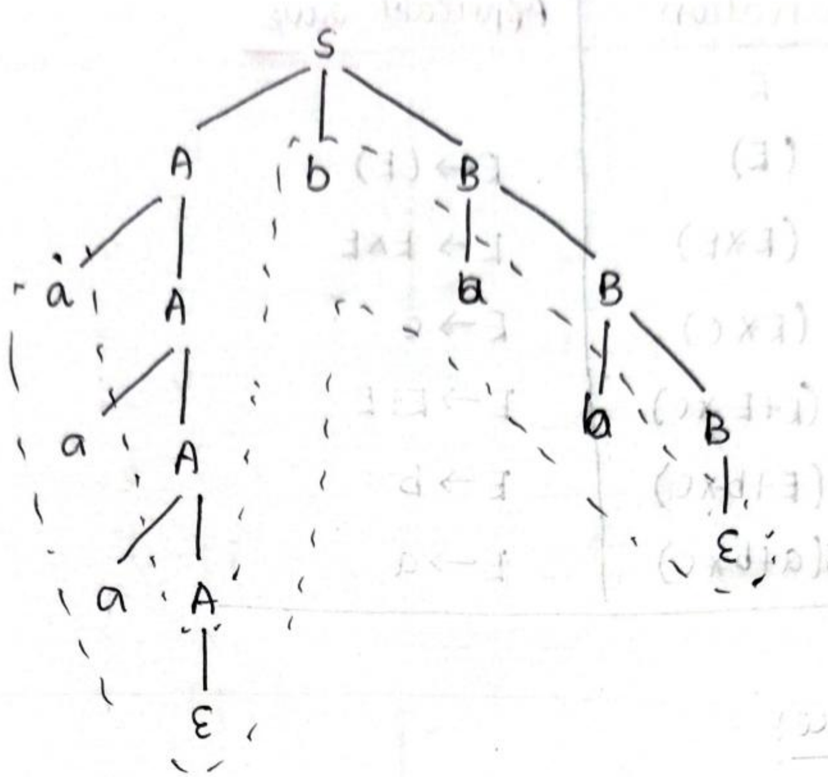
Sol:- LMD:-

Derivation	Applicable rule.
S	
AbB	$S \rightarrow AbB$
aAbB	$A \rightarrow aA$
aaAbB	$A \rightarrow aA$
aaaAbB	$A \rightarrow aA$
aaabB	$A \rightarrow \varepsilon$
aaabaB	$B \rightarrow aB$
aaababB	$B \rightarrow bB$
aaabab	$B \rightarrow \varepsilon$

RMD:-

Derivation	Applicable rule
S	
AbB	$S \rightarrow AbB$
AbaB	$B \rightarrow aB$
AbabB	$B \rightarrow bB$
Abab	$B \rightarrow \varepsilon$
aAbab	$A \rightarrow aA$
aaAbab	$A \rightarrow aA$
aaaAbab	$A \rightarrow aA$
aaabab	$A \rightarrow \varepsilon$

LMD parse tree:



⇒ aaabab

4). Given,

E → E+E | E-E

E → E*E | E/E

E → (E)

E → a | b | c

String :- (a+b*c).

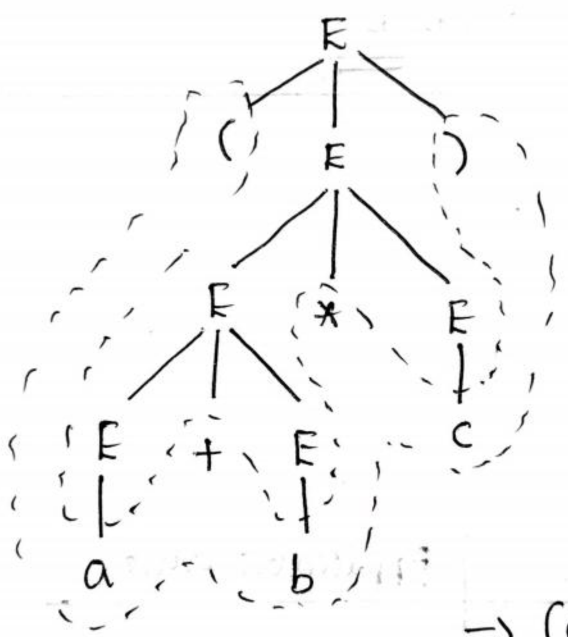
Sol:- LMD:-

Derivation	Applicable rule.
E	
(E)	E → (E)
(E+E)	E → E+E
(a+E)	E → a
(a+E*E)	E → E*E
(a+b*E)	E → b
(a+b*c)	E → c

RMD:-

Derivation	Applicable rule
E	
(E)	E→(E)
(E*E)	E→E*E
(E*c)	E→c
(E+E*c)	E→E+E
(E+b*c)	E→b
(a+b*c)	E→a

RMD parse tree:-



⇒ (a+b*c)

5). Given,

E→T*E|T

T→F-T|F

F→(E)|0|1

String:- 0-((1*0)-0)

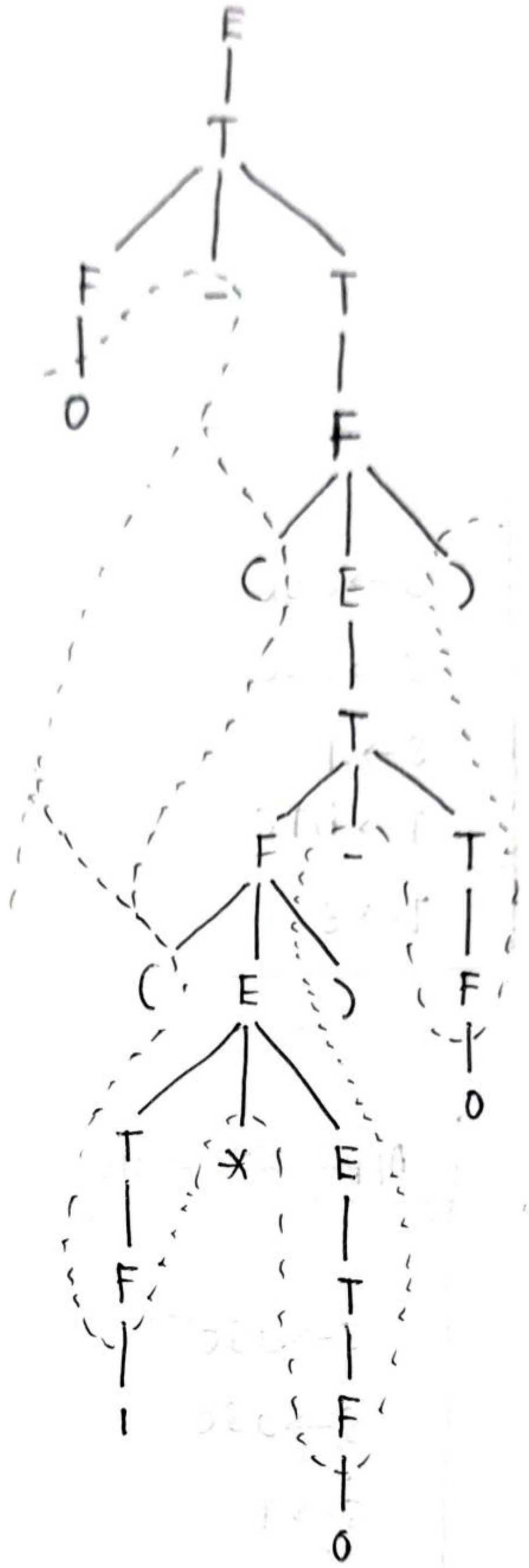
Sol: LMD:-

Derivation	Applicable rule.
E	
T	E → T
F - T	T → F - T
0 - T	F → 0
0 - F	T → F
0 - (E)	F → (E)
0 - (T)	E → T
~~0 - ((E))~~	
0 - (F - T)	T → F - T
0 - ((E) - T)	F → (E)
0 - ((T * E) - T)	E → T * E
0 - ((F * E) - T)	T → F
0 - ((1 * E) - T)	F → 1
0 - ((1 * T) - T)	E → T
0 - ((1 * F) - T)	T → F
0 - ((1 * 0) - T)	F → 0
0 - ((1 * 0) - F)	T → F
0 - ((1 * 0) - 0)	F → 0

RMD:-

Derivation	Applicable rule
E	
T	E → T
F - T	T → F - T
F - F	T → F
F - (E)	F → (E)
F - (T)	E → T
F - (F - T)	T → F - T
F - (F - F)	T → F
F - (F - 0)	F → 0
F - ((E) - 0)	F → (E)
F - ((T * E) - 0)	E → T * E
F - ((T * T) - 0)	E → T
F - ((T * F) - 0)	T → F
F - ((T * 0) - 0)	F → 0
F - ((F * 0) - 0)	T → F
F - ((1 * 0) - 0)	F → 1
0 - ((1 * 0) - 0)	F → 0

RMD parse tree



$\Rightarrow$ 0 - ((1*0) - 0)

6) Given.

$S \rightarrow aSc \mid T$

$T \rightarrow bTc \mid \varepsilon$

String:- aabccc

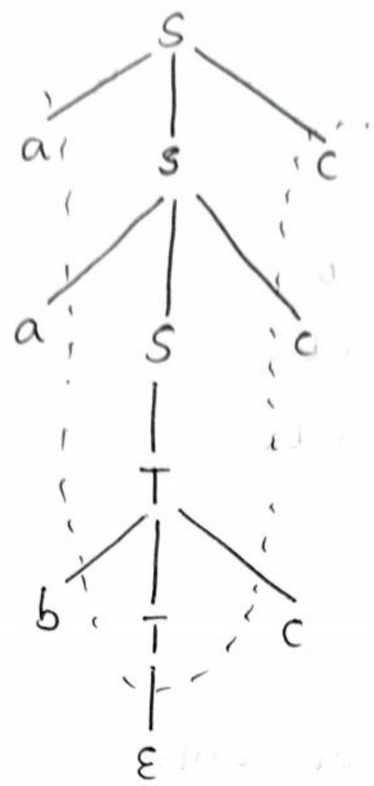
Sol:- LMD:-

Derivation	Applicable rule
S	
aSc	$S \rightarrow aSc$
aaScc	$S \rightarrow aSc$
aaTcc	$S \rightarrow T$
aabTccc	$T \rightarrow bTc$
aabccc	$T \rightarrow \varepsilon$

RMD:-

Derivation	Applicable rule
S	
aSc	$S \rightarrow aSc$
aaScc	$S \rightarrow aSc$
aaTcc	$S \rightarrow T$
aabTccc	$T \rightarrow bTc$
aabccc	$T \rightarrow \varepsilon$

LMD parse tree:



$\Rightarrow$ aabccc

7) Given, $S \rightarrow A | B$

$A \rightarrow CaA | CaC$

$B \rightarrow CbB | CbC$

$C \rightarrow aCbC | bCaC | \varepsilon$

Note:- {grammer for unequal number of a's & b's}.

String :- abbaaba

Sol:- LMD:-

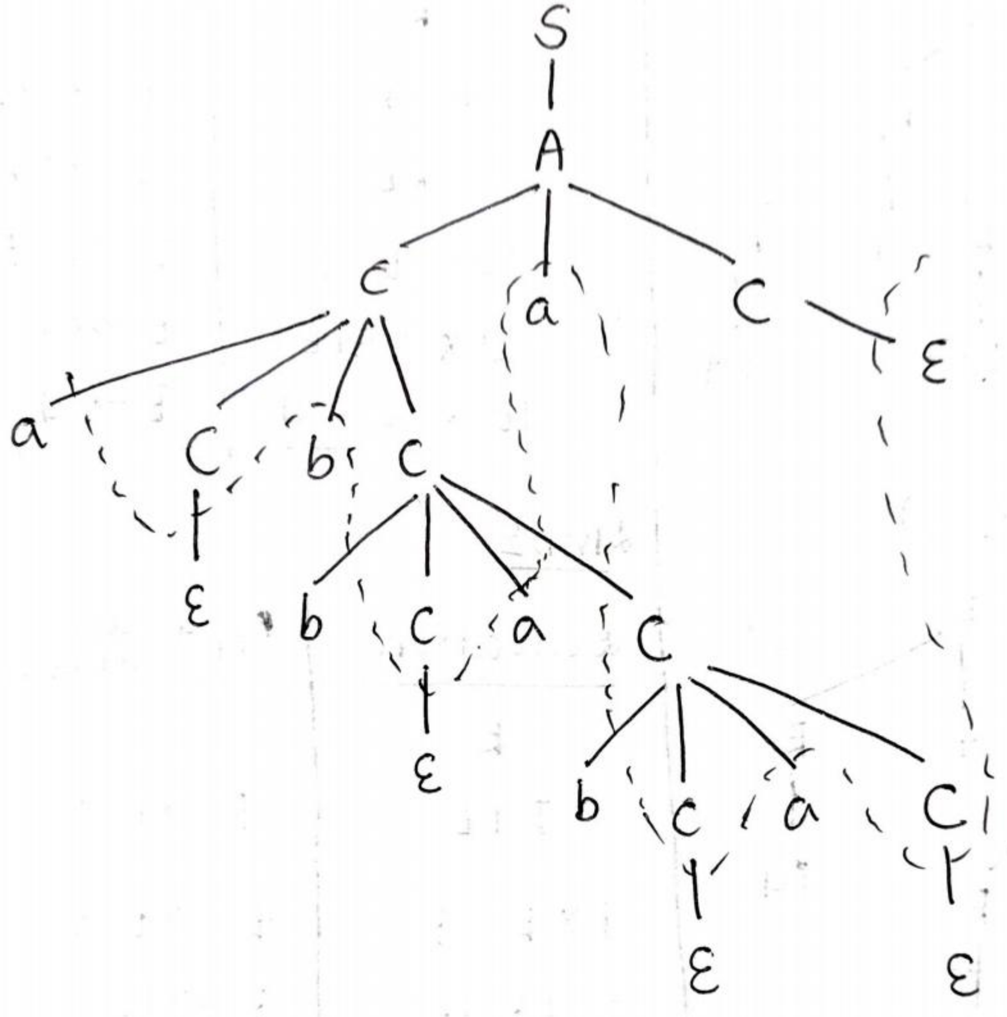
Derivation	Applicable rule
S	
A	$S \rightarrow A$
CaC	$A \rightarrow CaC$
$aCbCaC$	$C \rightarrow aCbC$
$abCaC$	$C \rightarrow \varepsilon$
$abbCaCaC$	$C \rightarrow bCaC$
$abbaCaC$	$C \rightarrow \varepsilon$
$abbaaCbCaC$	$C \rightarrow aCbC$
$abbaabCaC$	$C \rightarrow \varepsilon$
$abbaabaC$	$C \rightarrow \varepsilon$
$abbaaba$	$C \rightarrow \varepsilon$

RMD:-

Derivation	Applicable rule
S	
A	$S \rightarrow A$
CaC	$A \rightarrow CaC$
Ca	$C \rightarrow \varepsilon$
$aCbCa$	$C \rightarrow aCbC$
$aCba$	$C \rightarrow \varepsilon$
$abCaCba$	$C \rightarrow bCaC$
$abCaba$	$C \rightarrow \varepsilon$

abbCaCaba	$C \rightarrow bCaC$
abbCaaba	$C \rightarrow \varepsilon$
abbaaba	$C \rightarrow \varepsilon$

LMD parse tree:-



$\Rightarrow a\varepsilon bb\varepsilon aab\varepsilon a\varepsilon\varepsilon$

abbaaba

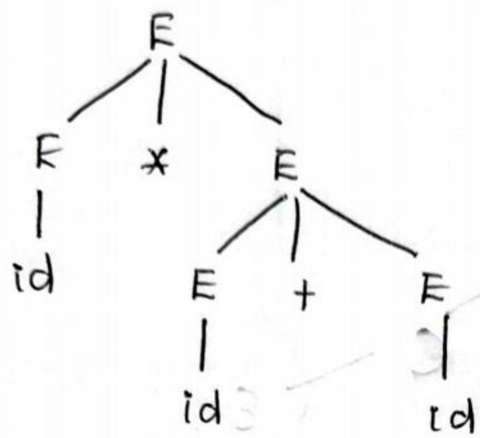
Ambiguity in Grammer:-

*) A grammer is said to be ambigious, if there exists 2 LMD's (or) 2 RMD's (or) 2 diff parse tree's for it.

Ex:- $E \rightarrow E*E \mid E+E \mid id$

String $\rightarrow$ $id*id+id$.

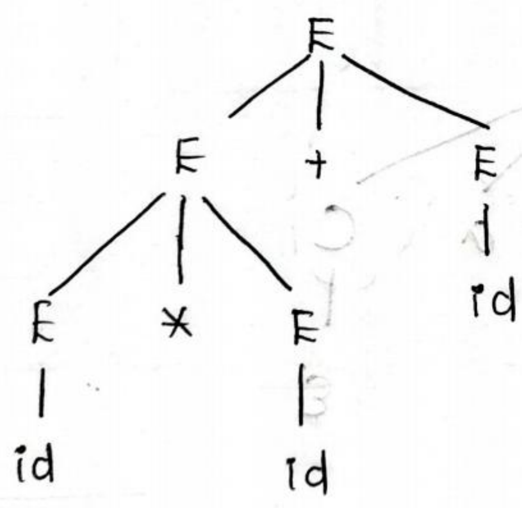
Sol: LMD1:- (Parse tree)



LMD1:-

Derivation	Applicable rule
E	
E*E	E → E*E
id*E	E → id
id*E+E	E → E+E
id*id+E	E → id
id*id+id	E → id

LMD2:- (Parse tree)



LMD2:-

Derivation	Applicable rule
E	
E+E	E → E+E
E*E+E	E → E*E
id*E+E	E → id
id*id+E	E → id
id*id+id	E → id

RMD1:-

Derivation	Application rule
E	
E*E	E→E*E
E*E+E	E→E+E
E*E+id	E→id
E*id+id	E→id
id*id+id	E→id

RMD2:-

Derivation	Application rule
E	
E+E	E→E+E
E+id	E→id
E*E+id	E→E*E
E*id+id	E→id
id*id+id	E→id

Note:- The ambiguity is due to violation of the presendence and associativity in the grammer (*, /, +, -).

1) Given,

S→iCtS

S→iCtSeS

S→a

C→b

String:- ibtibtibtaea

Note:- if b then

if b then

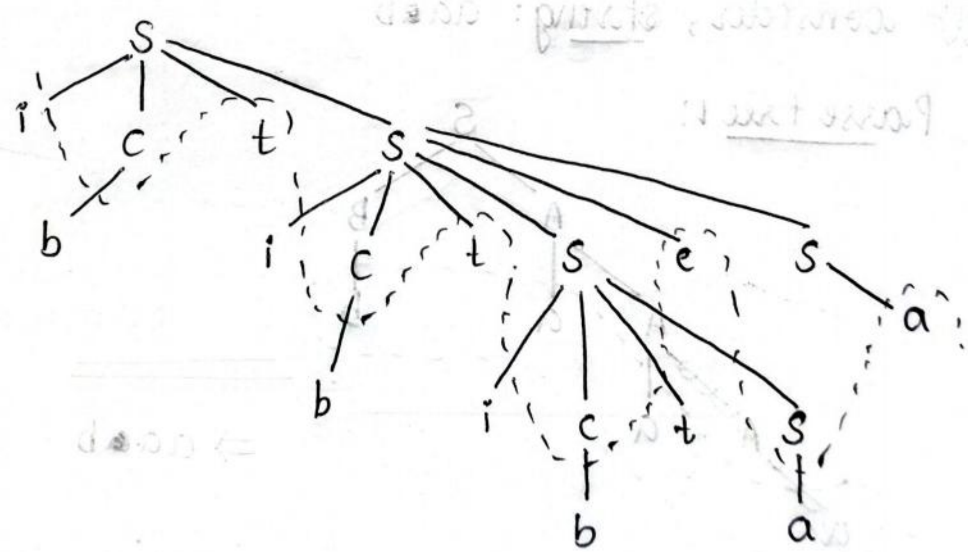
if b then

a

else

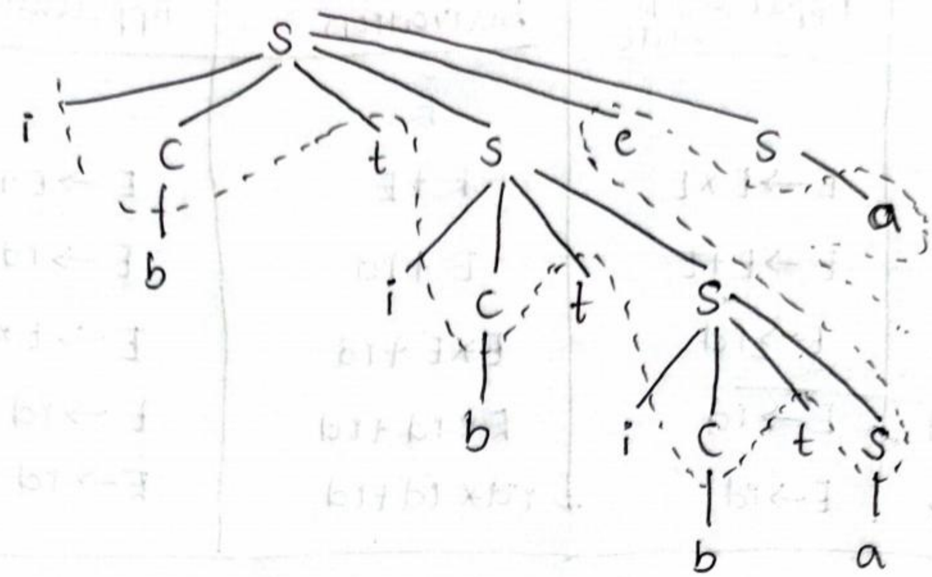
a

Sol:- Parse tree1:-



ibtibtibtaea

Parse tree 2:-



ibtibtibtaea

∵ there are 2 parse trees, the given grammer is ambiguious.

2) Check the ambiguity of the grammer $\boxed{13/11/2024}$

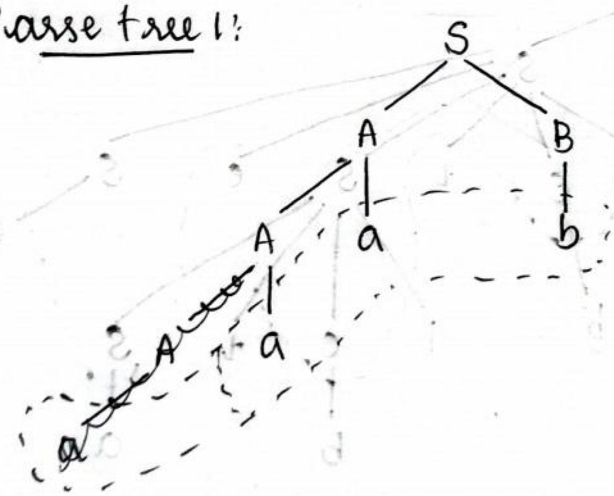
$S \longrightarrow AB \mid aaB$

$A \rightarrow a \mid Aa$

$B \longrightarrow b$

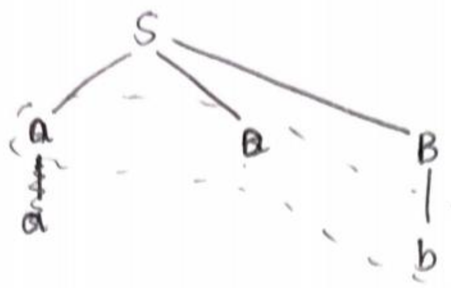
Sol:- consider, string: aaab

Parse tree 1:



$\Rightarrow$ aaab

Parse tree 2:



$\Rightarrow$ aab

3). Given,

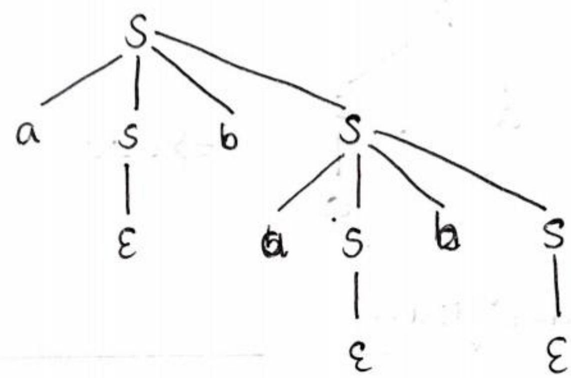
$S \rightarrow aSbS$

$S \rightarrow bSaS$

$S \rightarrow \varepsilon$

Sol: Consider string:- abab

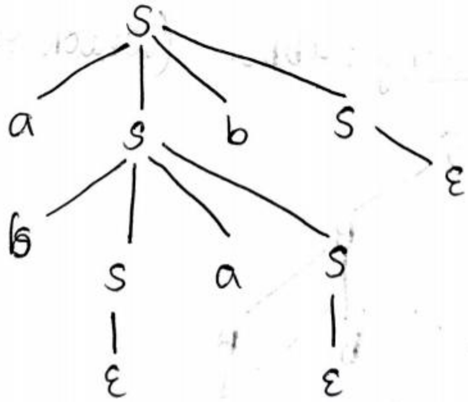
Parse tree 1:



$\Rightarrow a\varepsilon ba\varepsilon b\varepsilon$

$= abab$

Parse tree 2:



$\Rightarrow ab\varepsilon a\varepsilon b\varepsilon$

$= abab$

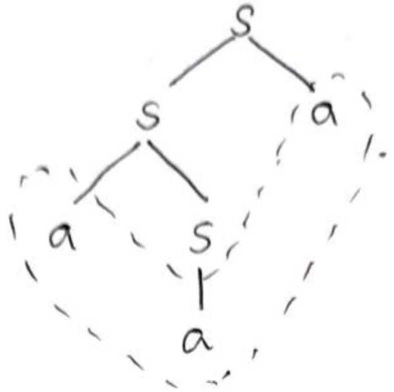
$\Rightarrow$ they are ambiguous as there are 2 parse trees

4) Given,

$S \rightarrow Sa|aS|a$

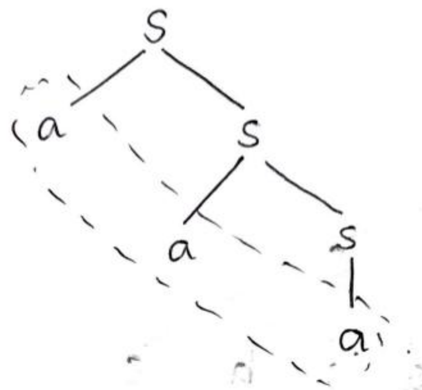
Sol:- Consider, string: aaa,

PT1:-



$\Rightarrow$ aaa

PT2:



$\Rightarrow$ aaa

$\Rightarrow$ They are ambigious.

Imp

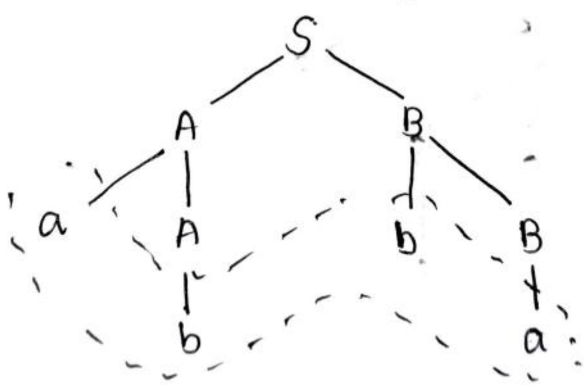
5) Given,

$S \rightarrow AB$

$A \rightarrow aA|b$

$B \rightarrow bB|a$ and String: abba (check the ambiguity).

Sol:- PT1:-



$\Rightarrow$ abba

PT2 :: there is only 1 Parse tree

⇒ the given grammer is unambigious

6) Check the ambiguity

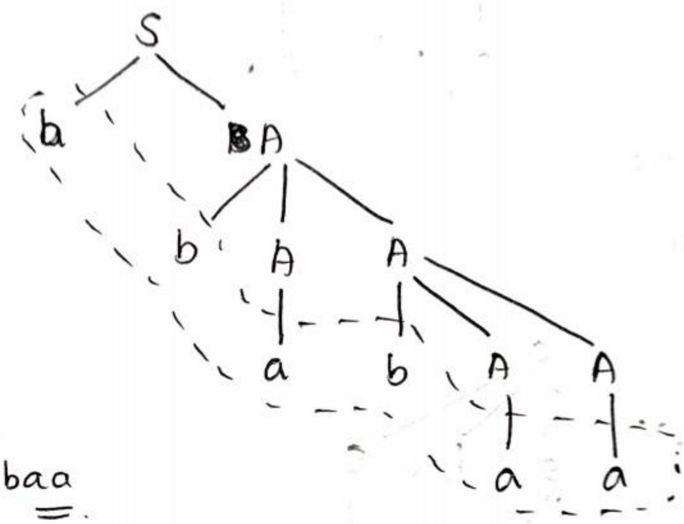
$S \rightarrow aB \mid bA$

$A \rightarrow a \mid aS \mid bAA$

$B \rightarrow b \mid bS \mid aBB$

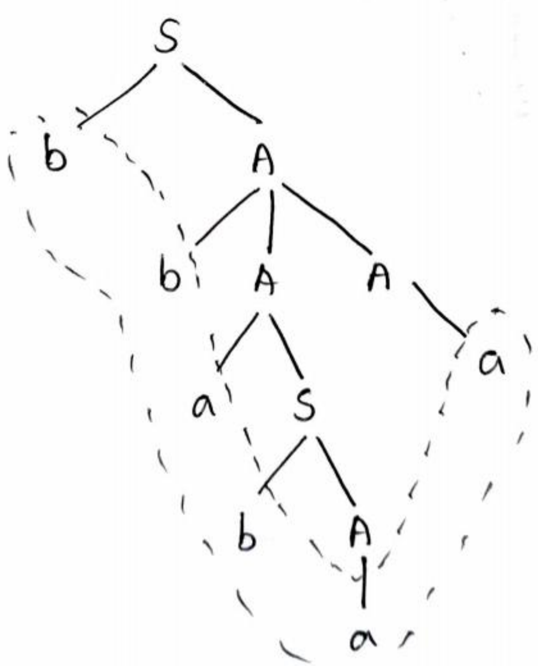
String: bbabaa

Sol:- PT1:-



⇒ bbabaa

PT2:



⇒ bbabaa

⇒ ~~They are~~ ambigious

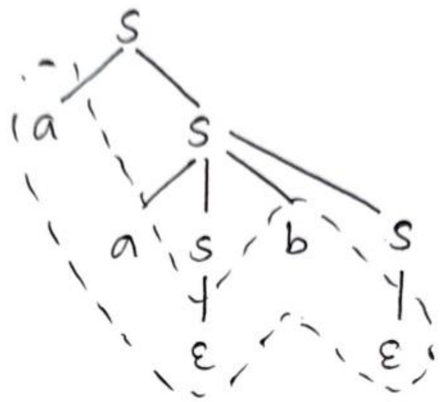
7) Check the ambiguity

$S \rightarrow aS \mid \varepsilon$

$S \rightarrow aSbS$

String:- aab

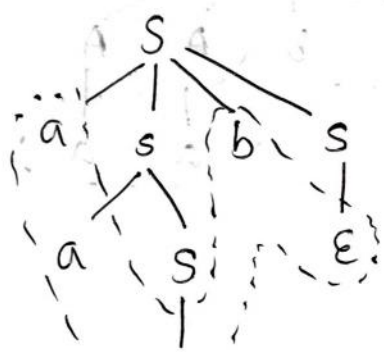
Sol:- PT1:-



$\Rightarrow aa\varepsilon b\varepsilon$

$= aab$

PT2:



$\Rightarrow aa\varepsilon b\varepsilon$

$= aab$

$\Rightarrow$ Ambigeous

8) $S \rightarrow AB$

$A \rightarrow AA$

$A \rightarrow a$

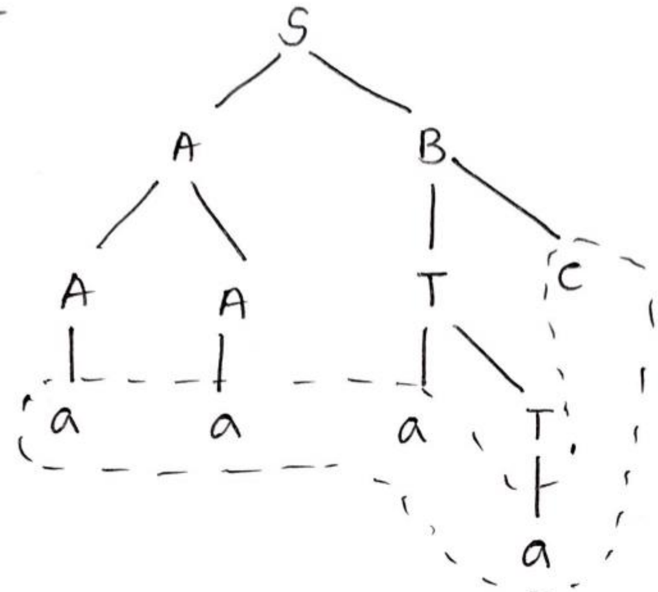
$B \rightarrow Tc$

$T \rightarrow aT$

$T \rightarrow a$

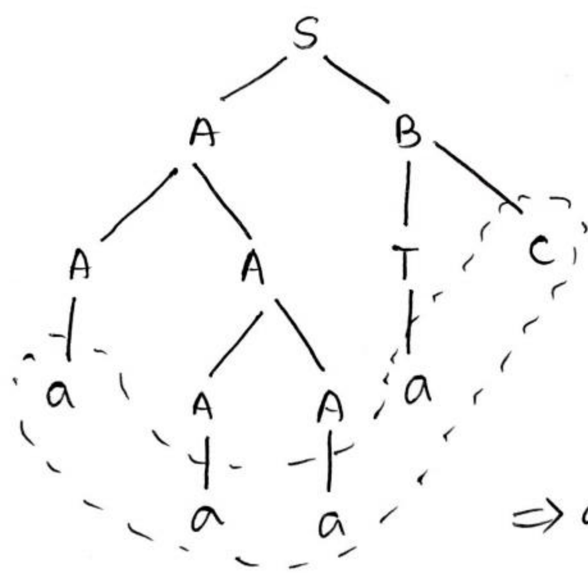
String: aaaac

Sol: PT1:-



$\Rightarrow$ aaaac

PT2:-



$\Rightarrow$ aaac

$\Rightarrow$ Ambegious

14/11/2024

1). Given,

i) $E \rightarrow E*E \mid E \wedge E \mid (E) \mid id$ ii) $E \rightarrow E+E \mid E-E \mid (E) \mid id$

Strings:-

i) $id*id \wedge id$ ii) $(id-id+id)$.

Convert into unambigious statements.

Sol:- i) Given,

$E \rightarrow E*E \mid E \wedge E \mid (E) \mid id$

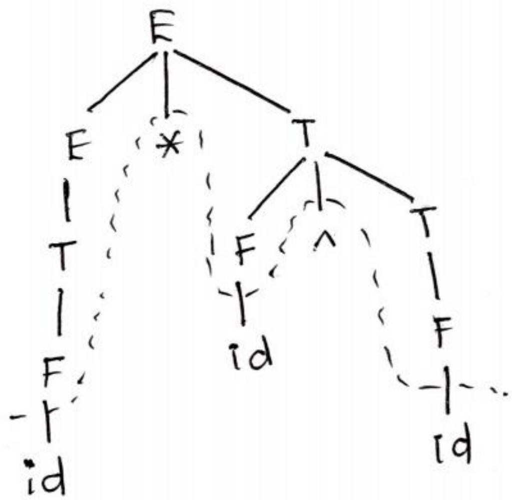
we convert it into

$E \rightarrow E*T \mid T$

$T \rightarrow F \wedge T \mid F$

$F \rightarrow (E) \mid id$ $\Rightarrow$ it is unambigious.

Parse tree:-



$\Rightarrow id*id \wedge id$

and

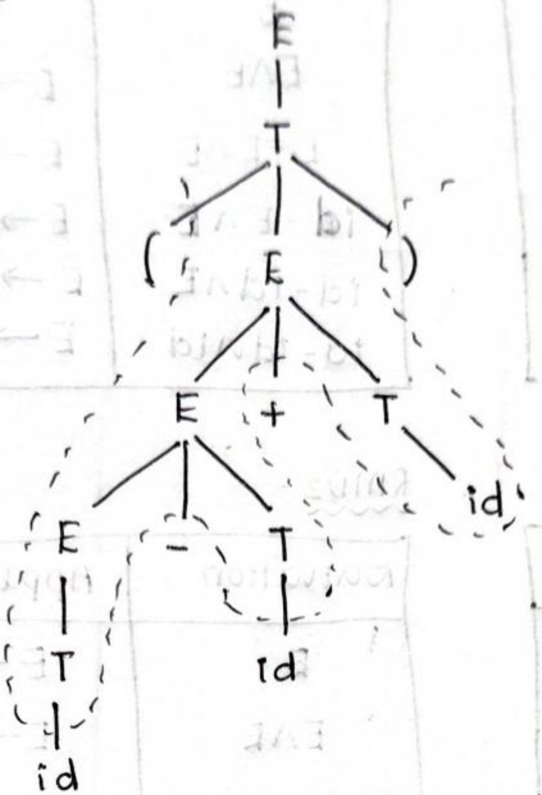
ii) $E \rightarrow E+E \mid E-E \mid (E) \mid id$

we convert it into

$E \rightarrow E+T \mid E-T \mid T$

$T \rightarrow (E) \mid id$

Parse tree:-



$\Rightarrow (id-id+id)$

2) Prove that the following given grammer is ~~un~~ambigious. Write unambigious grammer for the given. Write LMD, RMD and parse tree of the grammer. And define ambigious grammer.

Given, $E \rightarrow E-E \mid E \wedge E \mid E+E \mid id$

String:- $id-id \wedge id$

Sol:- Def:- A grammer is said to be ambigious, if there exists 2 LMD's (or) 2 RMD's (or) 2 parse trees.

Given, $E \rightarrow E-E \mid E \wedge E \mid E+E \mid id$

LMD1:-

Derivation	Applicable rule
E	
E - E	E → E - E
id - E	E → id
id - E∧E	E → E∧E
id - id∧E	E → id
id - id∧id	E → id

LMD2:-

Derivation	Applicable rule
E	
E∧E	E → E∧E
E - E∧E	E → E - E
id - E∧E	E → id
id - id∧E	E → id
id - id∧id	E → id

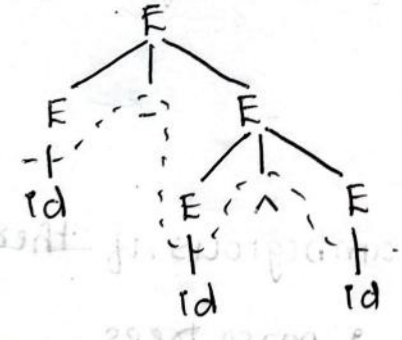
RMD1:-

Derivation	Applicable rule
E	
E-E	E → E-E
E - E∧E	E → E∧E
E - E∧id	E → id
E - id∧id	E → id
id - id∧id	E → id

RMD2:-

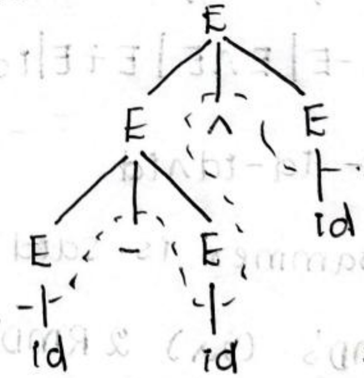
Derivation	Applicable rule
E	E → E∧E
E∧E	E → id
E∧id	E → E-E
E - E∧id	E → id
E - id∧id	E → id
id - id∧id	E → id

PT1:-



⇒ id - id∧id

PT2:-



⇒ id - id∧id

The given grammer is ambigious (By the definition).

Removal of ambiguity:-

we convert the given grammer into

$E \to T \wedge E \mid T$

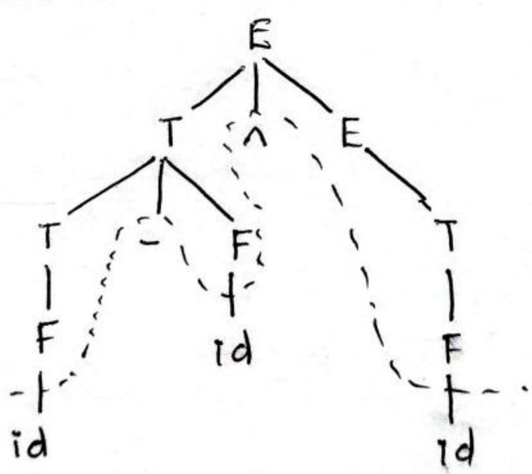
$T \to T - F \mid T + F \mid F$

$F \to id$ and string:- $id - id \wedge id$

⇒ LMD:-

Derivation	Applicable rule
E	
$T \wedge E$	$E \to T \wedge E$
$T - F \wedge E$	$T \to T - F$
$F - F \wedge E$	$T \to F$
$id - F \wedge E$	$F \to id$
$id - id \wedge E$	$F \to id$
$id - id \wedge T$	$E \to T$
$id - id \wedge F$	$T \to F$
$id - id \wedge id$	$F \to id$

Parse tree:-



$\Rightarrow id - id \wedge id$

⇒ ∵ it has only one PT, ⇒ It is unambiguous

Note:-

(or)

we can convert it into

Given, $\boxed{E \to E-E \mid E \wedge E \mid E+E \mid id}$

$$\left.\begin{aligned} &E \to E-T \mid E+T \mid T \\ &T \to F \wedge T \mid F \\ &F \to id \end{aligned}\right\} \text{unambigeous.}$$

---

3). Given,

$S \to iCtS$

$S \to iCtSeS$

$S \to a_1 \mid a_2$

$C \to b_1 \mid b_2$.

String:- $ib_1tib_2tib_1ta_1ea_2$ / $ib_1tib_2tib_1ta_1ea_2$

Sol:- M → matched statement : → if else

U → unmatched statement : → if | if ---- if

we convert the given into;

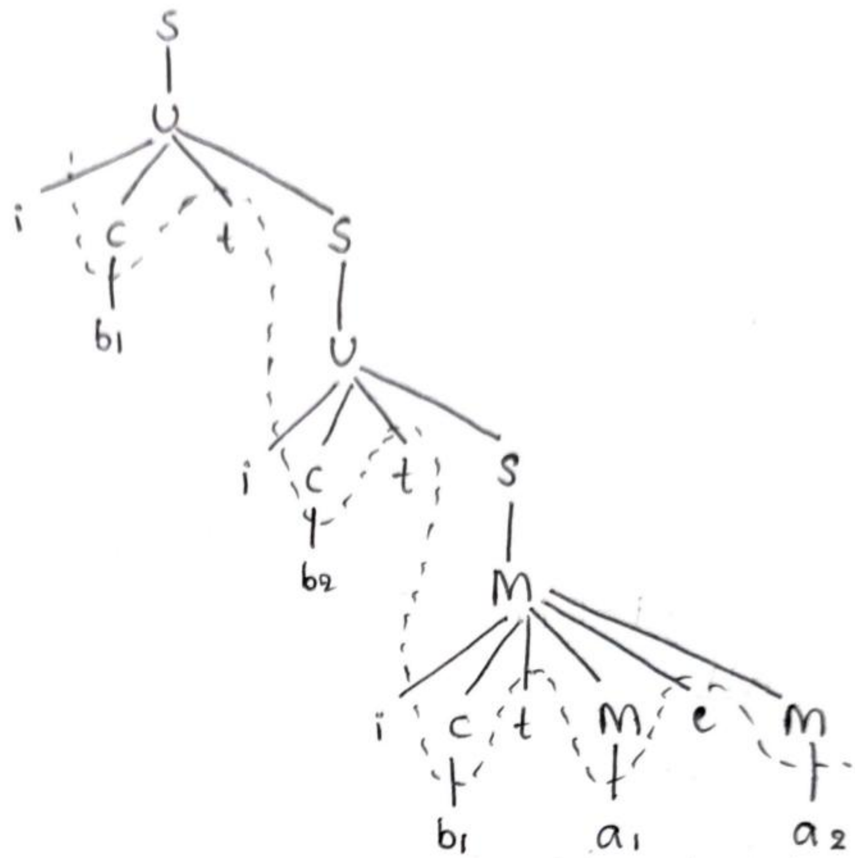
$S \to M \mid U$

$M \to ictMeM \mid a_1 \mid a_2$

$U \to ictS \mid ictMeU$

$C \to b_1 \mid b_2$

Parse tree:-



$\Rightarrow ib_1tib_2tib_1ta_1ea_2$

$\Rightarrow$ unambigious.

Removal of ambiguity by eliminating left recursion:-

1). $A \rightarrow A\alpha \mid \beta$.

$\Rightarrow A \rightarrow \alpha A'$

$A' \rightarrow \beta A' \mid \varepsilon$.

2). $E \rightarrow E \overset{①}{+} T \mid E \overset{②}{-} T \mid T$

$T \rightarrow id$.

$\Rightarrow E \rightarrow +TE' \mid -TE'$

$E' \rightarrow TE' \mid \varepsilon$

$T \rightarrow id$

3) $S \to AB$

$A \to Aa \mid a$

$B \to b.$

$\Rightarrow$ $S \to AB$

$A \to aA'$

$A' \to aA' \mid \varepsilon$

$B \to b$

4). $E \to E-F \mid F-E \mid F$

$F \to a \mid b.$

$\Rightarrow$ $E \to -FE'$

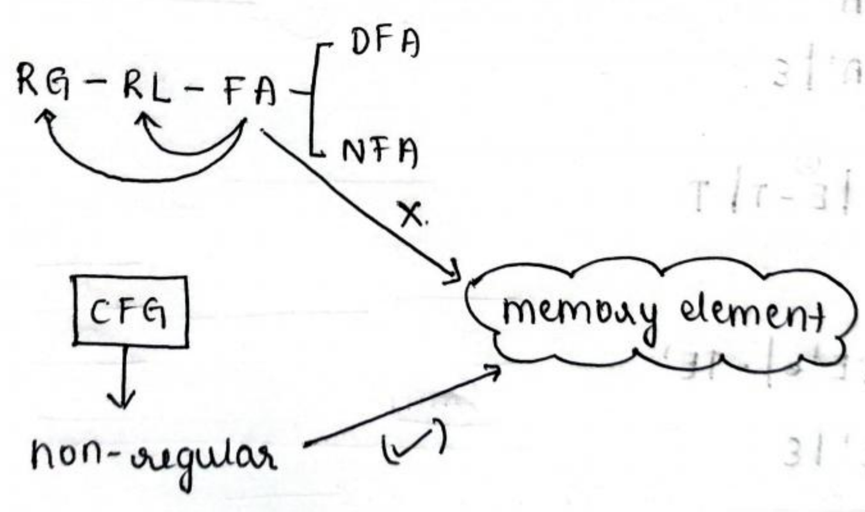
$E' \to F-EE' \mid FE' \mid \varepsilon$

$F \to a \mid b$

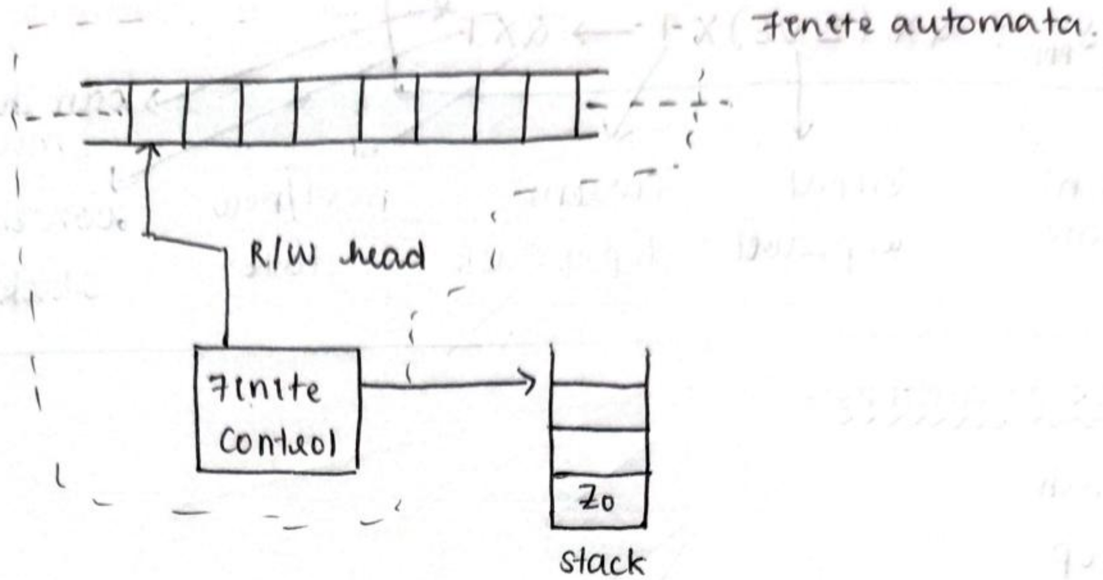
Note:- stack $\to$ zero(0) addressing memory element.

16|11|2024

Pushdown Automata (PDA):-



Working principle:-



$$\boxed{PDA = FA + stack}$$

$Q, \Sigma, q_0, \delta, F$ $\qquad$ Z_0, Γ

where Z_0: initial stack symbol, $Z_0 \in \Gamma$

Γ: set of all elements in stack.

- PDA is a 7 tuple representation.

$$\boxed{M_{PDA} = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)}$$

where $q_0 \in Q$ and $Z_0 \in \Gamma$

Q: finite set of all states.

Σ: alphabet set

Γ: finite set of stack symbol

q_0: start state

Z_0: initial stack symbol

F: set of all final states

δ: transition function

δ is defined as

$$\boxed{\delta_{PDA} : Q \times (\Sigma \cup \varepsilon) \times T \rightarrow Q \times T^*}$$

current state — Input alphabet — Current top of stack — next/new state — content of stack

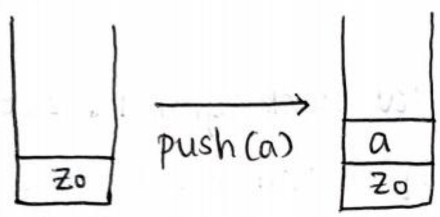
T^* → can have multiple symbols.

Stack operations:-

* Push
* Pop
* No change.

Push:-

Consider,



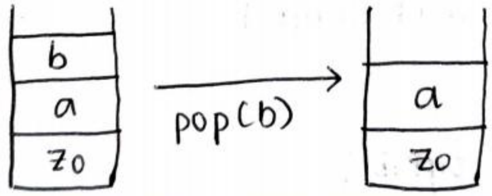
(insert a)

$$\boxed{\delta(q_i, a, z_0) = (q_j, a z_0)}$$

$$q_i \xrightarrow{a,\, z_0 \mid a z_0} q_j$$

Pop:-

Consider,



$$\boxed{\delta(q_j, a, b) \longrightarrow (q_k, \varepsilon)} \Rightarrow q_i \xrightarrow{a,\, b \mid \varepsilon} q_j$$

No change:-

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

$$q_i \xrightarrow{a,a \mid a} q_j$$

(10m)
***
When the question is asked for PDA, explain with operation & example

If only deterministic is asked, then no-need

Representation of PDA:-

PDA can be represented in 3-ways:

a) State transition function

b) Instantaneous description (ID)

c) State diagram

State transition diagram:-

It is a 3 tuple representation

Input: (q, w, z_0)

$q \rightarrow$ current state

$w \rightarrow$ current input string (symbol)

$z_0 \rightarrow$ top of stack (initial)

output: (q, γ)

$q \rightarrow$ new state

$\gamma \rightarrow$ stack content

$$\Rightarrow \boxed{\delta(q, w, z_0) = (q, \gamma)}$$

Instantaneous description (ID):-

*) It is a step by step procedure to show the acception (or) rejection of string by given PDA.

*) For this we have to make use of triple representation with turn stile symbol ($\vdash$) → (fallen T)

ie., $(q_i, w, \gamma) \vdash (q_j, w_1, \gamma_1)$ ---- n

- q_i → current state
- w → input string to be processed
- γ → stack content
- q_j → new state
- w_1 → Remaining part of input string to be processed
- γ_1 → new stack content.

* $(q_i, w, \gamma) \overset{*}{\vdash} (q_f, \underbrace{\varepsilon, z_0}_{})$

$\overset{*}{\vdash}$ → more than 1 step derivation

ε, z_0 → end of string and empty stack.

State diagram:-

○ → state   →○ → start state   ◎ → final state

$\rightarrow (q_i) \xrightarrow{xs|\gamma} (q_j)$

Transition from q_i to q_j

X → pushed symbol

S → top of stack.

If x: current scan input

s: current top of stack

γ: stack content,

then if γ is

$\varepsilon \longrightarrow$ pop operation

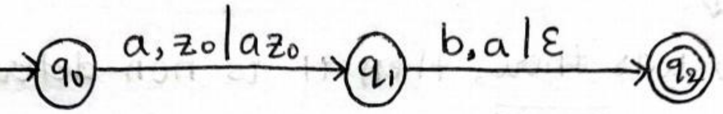
$s \longrightarrow$ no operation

$\gamma s \longrightarrow$ push operation.

---

1) Construct PDA to accept 'ab' over $\Sigma = \{a, b\}$

Sol:- State transition diagram:-



State transition function:- (STF)

$$\delta(q_0, a, z_0) = (q_1, az_0)$$
$$\delta(q_1, b, a) = (q_2, \varepsilon)$$

Instantaneous description:- check this in STF.

$$(q_0, ab, z_0) \vdash (q_1, b, az_0) \vdash (q_2, \varepsilon, z_0)$$

(accepted.)

consider aab,

$$(q_0, aab, z_0) \vdash (q_1, ab, az_0)$$

(rejected.)

***

Imp. (5m) Acceptance of language by PDA.

a) Acceptance by empty stack.

b) Acceptance by final state.

Example:-

Consider aab,

$\delta(q_1, a, z_0) = \delta(q_j, az_0) \rightarrow$ push
$\delta(q_k, a, z_0) = \delta(q_k, \varepsilon) \rightarrow$ pop

deterministic PDA.

if b was there, then it is non-deterministic PDA.

Acceptance of empty stack:-

*) At the end of input string, if the stack content is empty irrespective of weather you have reached final state or not, then we say that language is accepted.

$L = \{ w : (q_1, w, z_0) \vdash^* (q_j, \varepsilon, \varepsilon), q \in Q \}$

Acceptance of final state:-

At the end of input string if you have reached final state, then we say language is accepted.

$L = \{ w : (q_i, w, z_0) \vdash^* (q, \varepsilon, \gamma), q \in F \}$

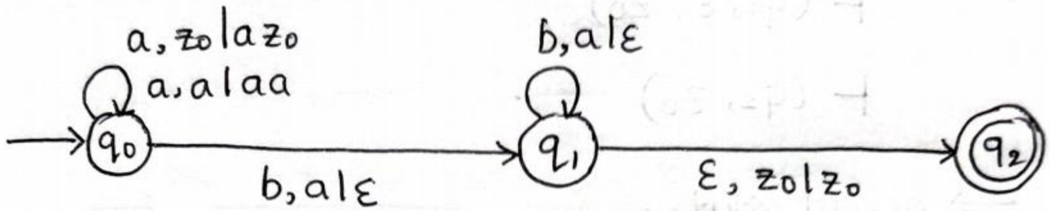
19/11/2024

1). Given, $L = \{a^n b^n : n \geq 1\}$

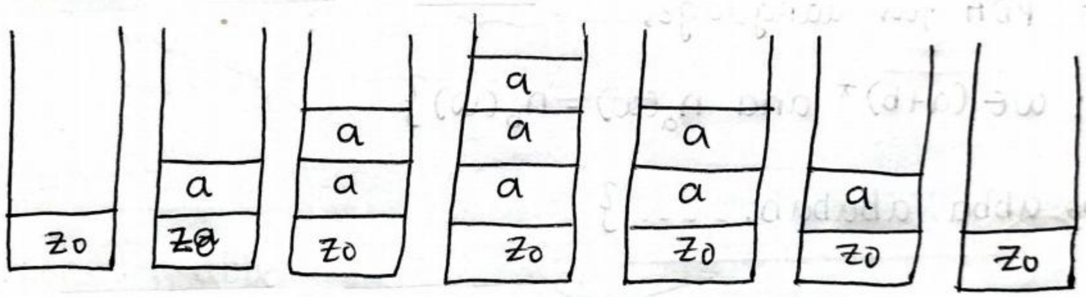
Sol:- Given, $a^n b^n$ ⟹ equal no of a's followed by equal no of b's.

Logic:- We start pushing a's into stack until 'b' is encountered as input symbol. Once 'b' is encountered, then will pop a's. If we get z_0 at the end of the pop, then the string is accepted, else rejected.

Transition diagram:-



Transition function:-



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

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

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

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

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

Instantaneous description (ID):-

Consider, aaabbb

$\delta(q_0, aaabbb, z_0) \vdash (q_0, aabbb, az_0)$

$\vdash (q_0, abbb, aaz_0)$

$\vdash (q_0, bbb, aaaz_0)$

$\vdash (q_1, bb, aaz_0)$

$\vdash (q_1, b, az_0)$

$\vdash (q_1, \varepsilon, z_0)$

$\vdash (q_2, z_0)$

$\Longrightarrow$ accept state.

---

2) Construct PDA for language,

$L = \{w : w \in (a+b)^* \text{ and } n_a(w) = n_b(w)\}$

Sol:- $L = \{ab, abba, ababab, \ldots\}$

Logic:- If top is z_0, and

scanned symbol is $a/b \Rightarrow$ push

If top is 'a', then

if scanned symbol is $a \Rightarrow$ push

$b \Rightarrow$ pop

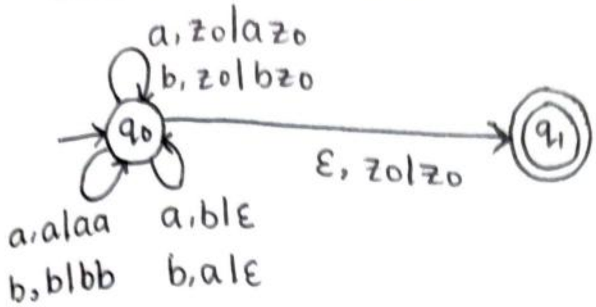
If top is 'b', then

if scanned symbol is $b \Rightarrow$ push

$a \Rightarrow$ pop

When string is 'ε' and top is 'z_0' $\Rightarrow$ go to final state.

Transition diagram:-



Transition function:-

$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, b, z_0) = (q_0, bz_0)$

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

$\delta(q_0, b, b) = (q_0, bb)$

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

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

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

Instantaneous description:-

Consider, abba

$\delta(q_0, abba, z_0) \vdash (q_0, bba, az_0)$

$\vdash (q_0, ba, z_0)$

$\vdash (q_0, a, bz_0)$

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

$\vdash (q_1, z_0)$

$\Rightarrow$ accept state.

3) Given,

$L = \{w : w \in (a+b)^* \text{ and } n_a(w) < n_b(w)\}$

Sol:- $L = \{ab, abb, bab, bbab, \ldots\}$

Logic:- If top is z_0, then

if scanned symbol is a/b ⇒ push

If top is 'a', then if

scanned symbol is a ⇒ push

b ⇒ pop

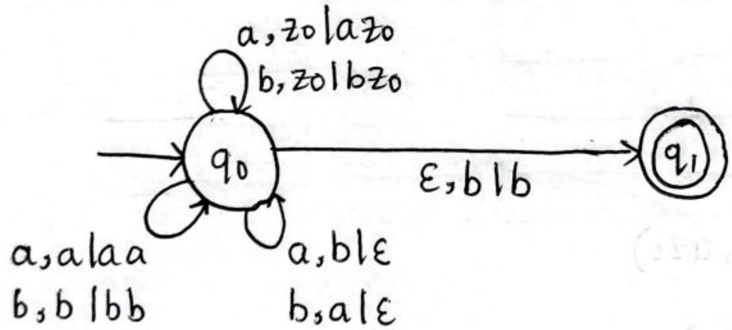
If top is 'b', then if

scanned symbol is b ⇒ push

a ⇒ pop

If top is 'b' and symbol (string) is 'ε' ⇒ go to final state.

Transition diagram:-



Transition function:-

$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, b, z_0) = (q_0, bz_0)$

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

$\delta(q_0, b, b) = (q_0, bb)$

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

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

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

Instantaneous description:-

Consider, bbaab

$\delta(q_0, bbaab, z_0) \vdash (q_0, baab, bz_0)$

$\vdash (q_0, aab, bbz_0)$

$\vdash (q_0, ab, bz_0)$

$\vdash (q_0, b, z_0)$

$\vdash (q_0, \varepsilon, bz_0)$

$\vdash (q_0, bz_0)$

$\Rightarrow$ accept state.

---

4). Given,

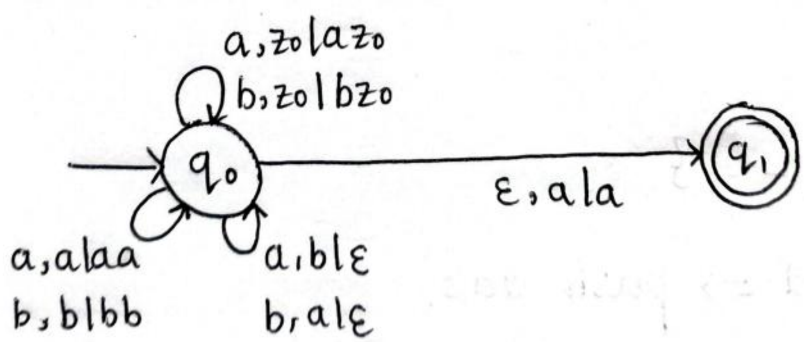
$L = \{w : w \in (a+b)^* \text{ and } n_a(w) > n_b(w)\}$

Sol: $L = \{a, aab, aba, aabba, \ldots\}$

Logic:- 1st three steps same as previous problem

If string is 'ε' and top is 'a' ⇒ go to final state.

Transition diagram:-



Transition function:-

$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, b, z_0) = (q_0, bz_0)$

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

$\delta(q_0, b, b) = (q_0, bb)$

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

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

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

Instantaneous description:-

Consider, aabba.

$\delta(q_0, aabba, z_0) \vdash (q_0, abba, az_0)$

$\vdash (q_0, bba, aaz_0)$

$\vdash (q_0, ba, az_0)$

$\vdash (q_0, a, z_0)$

$\vdash (q_0, \varepsilon, az_0)$

$\vdash (q_1, az_0)$

$\Rightarrow$ accept state.

Note:- Push :- we can push multiple symbols

Pop :- we can only pop one symbol at a time.

---

5). Given,

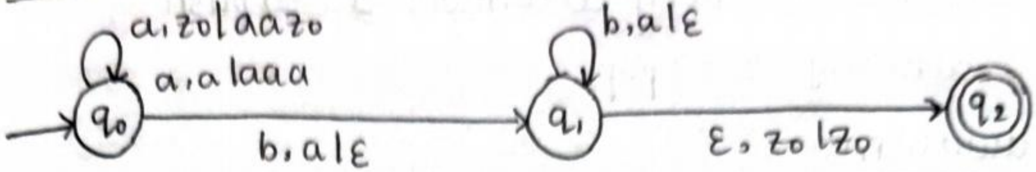
$L = \{a^n b^{2n} : n \geq 1\}$

Sol:- $L = \{abb, aabbbb, \ldots\}$

Logic:- If 'a' is scanned $\Rightarrow$ push 2a's.

'b' is scanned $\Rightarrow$ pop 'a'

Transition diagram:-



Transition function:

$\delta(q_0, a, Z_0) = (q_0, aaZ_0)$

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

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

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

$\delta(q_1, \varepsilon, Z_0) = (q_2, Z_0)$

Instantaneous description:-

consider, aabbbb

$\delta(q_0, aabbbb, Z_0) \vdash (q_0, abbbb, aaZ_0)$

$\vdash (q_0, bbbb, aaaaZ_0)$

$\vdash (q_0, bbb, aaaZ_0)$

$\vdash (q_0, bb, aaZ_0)$

$\vdash (q_0, b, aZ_0)$

$\vdash (q_1, \varepsilon, Z_0)$

$\vdash (q_2, Z_0)$

$\Rightarrow$ accept state.

Imp.

6). Given,

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

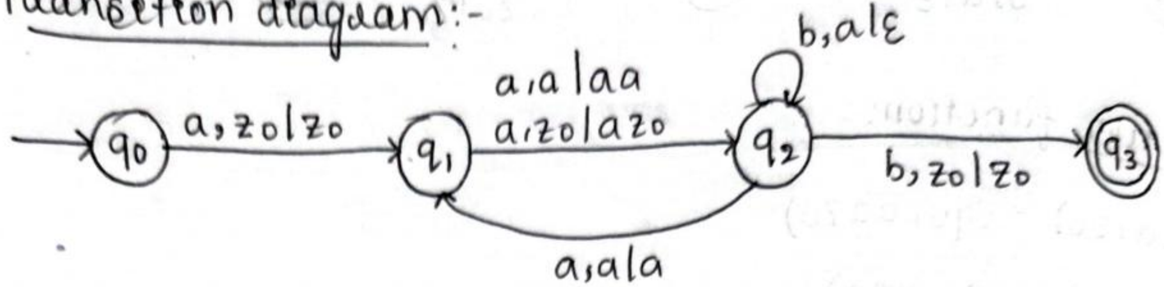
Sol: $L = \{aab, aaaabb, \ldots\}$.

Logic: read first 'a' ⇒ don't do any operation

read second 'a' & every alternate a's ⇒ push

while scanning 'b' ⇒ pop.

Transition diagram:-



Transition function:-

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

$\delta(q_1, a, a) = (q_2, aa)$

$\delta(q_1, a, z_0) = (q_2, az_0)$

$\delta(q_1, b, a) = (q_2, \varepsilon)$

$\delta(q_2, a, a) = (q_1, a)$

$\delta(q_2, \varepsilon, z_0) = (q_3, z_0)$

Instantaneous description:- Consider, aaaabb

$\delta(q_0, aaaabb) \vdash (q_1, aaabb, z_0)$

$\vdash (q_2, aabb, az_0)$

$\vdash (q_1, abb, az_0)$

$\vdash (q_2, bb, aaz_0)$

$\vdash (q_2, b, az_0)$

$\vdash (q_2, \varepsilon, z_0)$

$\vdash (q_3, z_0)$

⇒ accept state.

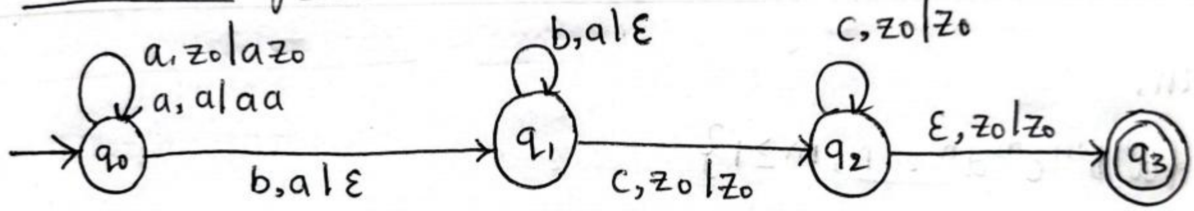
7). $L = \{a^n b^n c^m : n \geq 1, m \geq 1\}$ 25/11/2024

Sol:- a's & b's are equal in number and m no of c's.

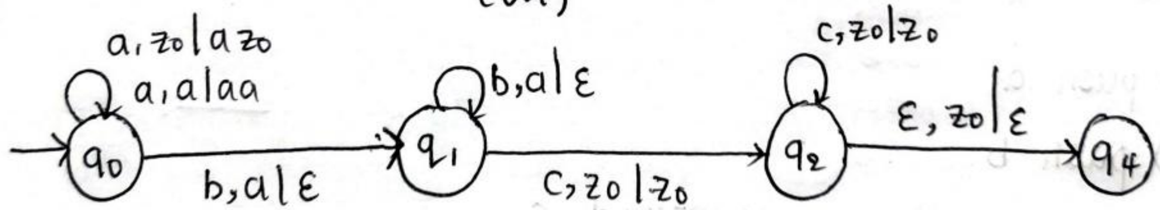
$L = \{abc, aabbc, aabbcc, aabbccc, \ldots\}$

Logic:- ∵ we should have equal no of a's and b's, we push all a's to the stack, once 'b' is encountered, we pop it. If we get z_0 as the top of stack, then there are equal no of a's & b's, and then we will take 'c' as input and remain in the same state.

Transition diagram:-



(or)



⇓

in this case, no need to take final state as we are popping the top of stack (z_0).

Transition function:-

$\delta(q_0, a, z_0) = \delta(q_0, az_0)$ $\quad\quad$ $\delta(q_2, \varepsilon, z_0) = \delta(q_3, z_0)$

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

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

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

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

Instantaneous description:-

consider, aabbc,

$\delta(q_0, aabbc, z_0) \vdash (q_0, abbc, az_0)$

$\vdash (q_0, bbc, aaz_0)$

$\vdash (q_1, bc, az_0)$

$\vdash (q_1, c, z_0)$

$\vdash (q_2, \varepsilon, z_0)$

$\vdash (q_3, z_0)$

$\Rightarrow$ accept state.

---

8). Given,

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

Sol:- $\Rightarrow$ equal no of a's and d's and equal no of b's and c's.

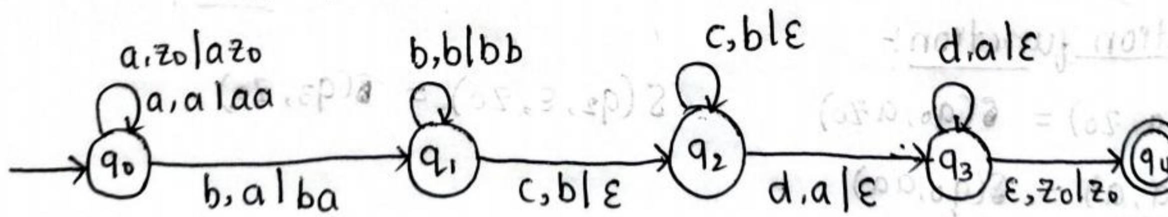
Steps:- i) push a

ii) push b.

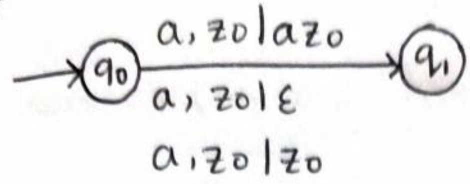
iii) pop b when encountered c

iv) pop d when encountered d.

Transition diagram:



Note:-



same input, therefore it is non-determenistic PDA (with different operation).

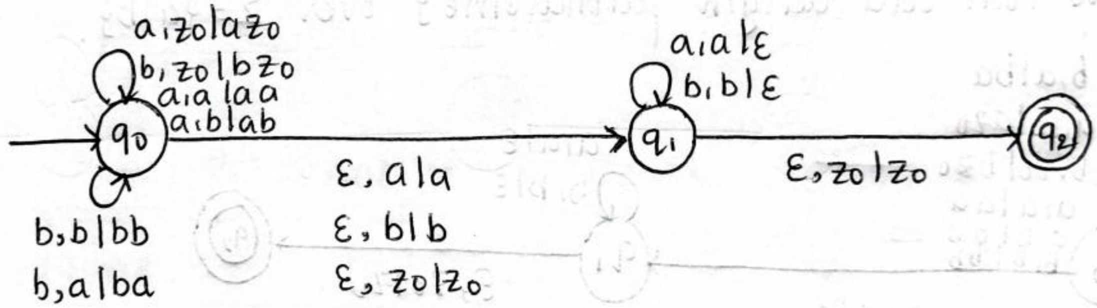
---

q). Given,

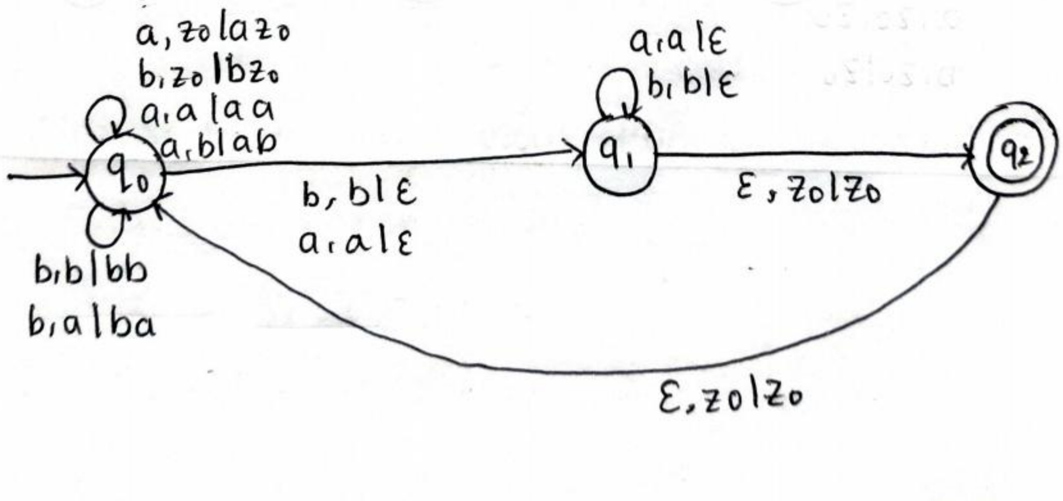
$L = \{ww^R : w \in (a+b)^*\}$

Sol:- $L = \{\varepsilon, abba, abbbba, ---\}$.

Transition ~~function~~ diagram:- (Deterministic PDA).



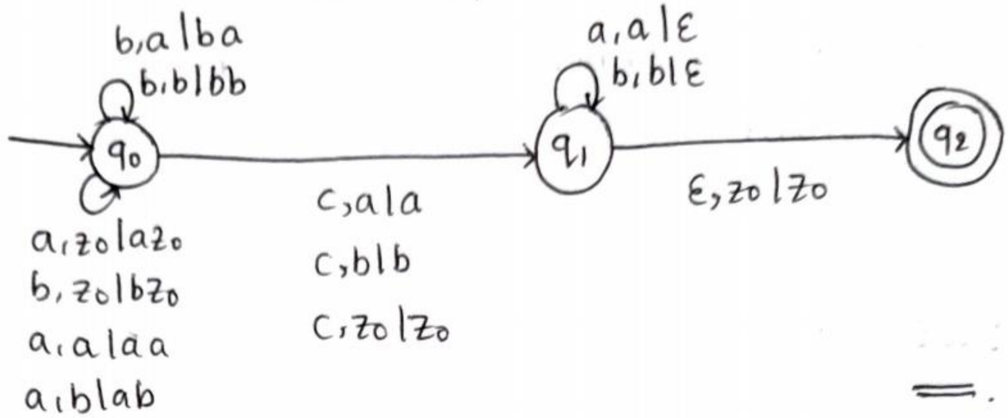
Transition diagram:- (Non-deterministic PDA)



10) Given,

$L = \{wcw^R\}$ over $\Sigma = \{a, b\}$.

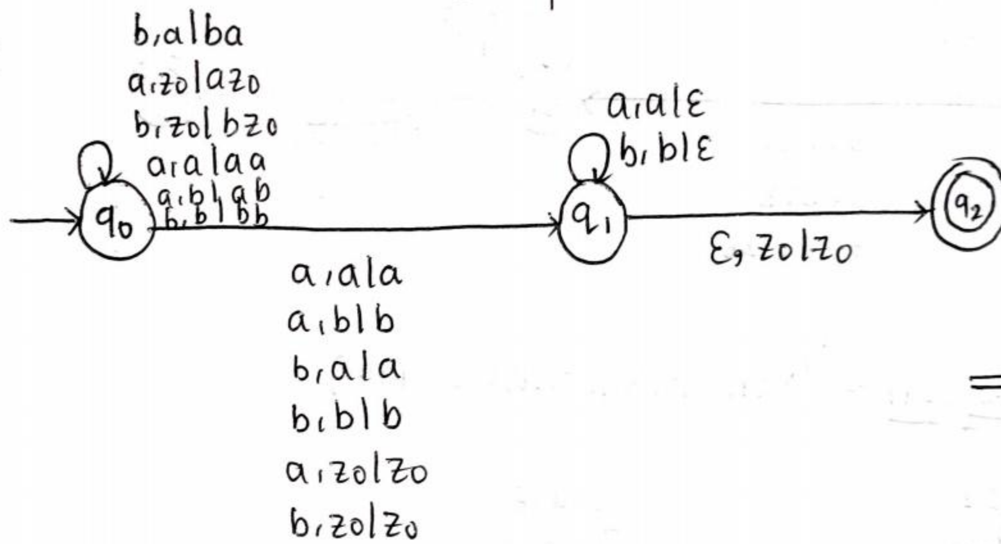
Sol: <u>Transition diagram:-</u>



11). Given,

$L = \{ww^R$: an odd length palindrome$\}$ over $\Sigma = \{a, b\}$

Sol:



26|11|2024

Equivalence of CFG and PDA:-

Equivalence of CFG and PDA can be proved by converting CFG into PDA (or) PDA into CFG.

conversion of CFG into PDA:-

Let $G = \{V, T, P, S\}$ be the given CFG.

The PDA that accepts $L(G)$ by empty stack is represented as

$$\boxed{P = (\{q\}, V, V \cup T, q, S, F, \delta)}$$

where the 'δ' is described as below

*) for every non-terminal $A \rightarrow \beta$, introduce the function

$$\boxed{\delta(q, \varepsilon, A) = (q, \beta)}$$

*) for every terminal ie, a/b, introduce the function

$$\boxed{\delta(q, a, a) = (q, \varepsilon)}$$

Ex:- Consider the grammer,

$S \rightarrow SA|a$

$A \rightarrow AA|b$

convert it into equivalent PDA.

Sol:- Consider the non-terminal "S"

$\Rightarrow \delta(q, \varepsilon, S) = (q, SA)$

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

consider the non-terminal "A"

$\Rightarrow \delta(q, \varepsilon, A) = (q, AA)$

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

Consider the terminal "a"

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

consider the terminal "b"

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

The required PDA is

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

2) $\delta(q, \epsilon, S) = (q, a)$

3) $\delta(q, \epsilon, A) = (q, AA)$

4) $\delta(q, \epsilon, A) = (q, b)$

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

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

consider string "abbb"

$$
\begin{aligned}
\delta(q, abbb, S) &\vdash (q, abbb, SA) && (1)\\
&\vdash (q, abbb, aA) && (2)\\
&\vdash (q, bbb, A) && (5)\\
&\vdash (q, bbb, AA) && (3)\\
&\vdash (q, bbb, bA) && (4)\\
&\vdash (q, bb, A) && (6)\\
&\vdash (q, bb, AA) && (3)\\
&\vdash (q, bb, bA) && (4)\\
&\vdash (q, b, A) && (6)\\
&\vdash (q, b, b) && (4)\\
&\vdash (q, \epsilon) && (6)
\end{aligned}
$$

1). Given,

$S \rightarrow AA \mid a$

$A \rightarrow SA \mid b$

Sol:-

Production	Transition
$S \rightarrow AA$	$\delta(q, \varepsilon, S) = (q, AA)$
$S \rightarrow a$	$\delta(q, \varepsilon, S) = (q, a)$
$A \rightarrow SA$	$\delta(q, \varepsilon, A) = (q, SA)$
$A \rightarrow b$	$\delta(q, \varepsilon, A) = (q, b)$
a	$\delta(q, a, a) = (q, \varepsilon)$
b	$\delta(q, b, b) = (q, \varepsilon)$

The required PDA is

1) $\delta(q, \varepsilon, S) = (q, AA)$

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

3) $\delta(q, \varepsilon, A) = (q, SA)$

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

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

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

- Consider string "abbabb"

$\delta(q, abbabb, S) \vdash (q, abbabb, AA)$ (1)

$\vdash (q, abbabb, SAA)$ (3)

$\vdash (q, abbabb, aAA)$ (2)

$\vdash (q, bbabb, AA)$ (5)

$\vdash (q, bbabb, bA)$ (4)

$\vdash (q, babb, A)$ (6)

$\vdash (q, babb, SA)$ (3)

$\vdash (q, babb, AAA)$ (1)

$\vdash (q, babb, bAA)$ (4)

$\vdash (q, abb, AA)$ (6)

$\vdash (q, abb, SAA)$ (3)

$\vdash (q, abb, aAA)$ (2)

$\vdash (q, bb, AA)$ (5)

$\vdash (q, bb, bA)$ (4)

$\vdash (q, b, A)$ (6)

$\vdash (q, b, b)$ (4)

$\vdash (q, \varepsilon)$ (6)

---

2). Given,

$S \rightarrow aABC$

$A \rightarrow aB \mid a$

$B \rightarrow bA \mid b$

$C \rightarrow a$

convert it into PDA and show that "aabba" gets accepted.

Sol:-

Production.	Transition.
$S \rightarrow aABC$	$\delta(q, \varepsilon, S) = (q, aABC)$
$A \rightarrow aB$	$\delta(q, \varepsilon, A) = (q, aB)$
$A \rightarrow a$	$\delta(q, \varepsilon, A) = (q, a)$
$B \rightarrow bA$	$\delta(q, \varepsilon, B) = (q, bA)$
$B \rightarrow b$	$\delta(q, \varepsilon, B) = (q, b)$
$C \rightarrow a$	$\delta(q, \varepsilon, C) = (q, a)$

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

The required PDA is

1) $\delta(q, \varepsilon, S) = (q, aABC)$ (1)
2) $\delta(q, \varepsilon, A) = (q, aB)$ (2)
3) $\delta(q, \varepsilon, A) = (q, a)$ (3)
4) $\delta(q, \varepsilon, B) = (q, bA)$
5) $\delta(q, \varepsilon, B) = (q, b)$
6) $\delta(q, \varepsilon, C) = (q, a)$
7) $\delta(q, a, a) = (q, \varepsilon)$
8) $\delta(q, b, b) = (q, \varepsilon)$

Consider, "aabba"

$$
\begin{aligned}
\delta(q, aabba, S) &\vdash (q, aabba, aABC) && (1)\\
&\vdash (q, abba, ABC) && (7)\\
&\vdash (q, abba, aBBC) && (2)\\
&\vdash (q, bba, BBC) && (7)\\
&\vdash (q, bba, bBC) && (5)\\
&\vdash (q, ba, BC) && (8)\\
&\vdash (q, ba, bC) && (5)\\
&\vdash (q, a, C) && (8)\\
&\vdash (q, a, a) && (6)\\
&\vdash (q, \varepsilon) && (7).
\end{aligned}
$$

3). Given,

$S \rightarrow aSa \mid bSb \mid c$

Convert the CFG into PDA, also write ID for string "abbcbba".

Sol:

Production	Transition
$S \rightarrow aSa$	$\delta(q, \varepsilon, S) = (q, aSa)$
$S \rightarrow bSb$	$\delta(q, \varepsilon, S) = (q, bSb)$
$S \rightarrow c$	$\delta(q, \varepsilon, S) = (q, c)$
a	$\delta(q, a, a) = (q, \varepsilon)$
b	$\delta(q, b, b) = (q, \varepsilon)$
c	$\delta(q, c, c) = (q, c)$.

The required PDA is

1) $\delta(q, \varepsilon, S) = (q, aSa)$
2) $\delta(q, \varepsilon, S) = (q, bSb)$
3) $\delta(q, \varepsilon, S) = (q, c)$
4) $\delta(q, a, a) = (q, \varepsilon)$
5) $\delta(q, b, b) = (q, \varepsilon)$
6) $\delta(q, c, c) = (q, \varepsilon)$

consider, "abbcbba"

$\delta(q, abbcbba, S) \vdash (q, abbcbba, aSa)$ (1)

$\vdash (q, bbcbba, Sa)$ (4)

$\vdash (q, bbcbba, bSba)$ (2)

$\vdash (q, bcbba, Sba)$ (5)

$\vdash (q, bcbba, bSbba)$ (2)

$\vdash (q, cbba, Sbba)$ (5)

$\vdash (q, cbba, cbba)$ (3)

$\vdash (q, bba, bba)$ (6)

$\vdash (q, ba, ba)$ (5)

$\vdash (q, a, a)$ (5)

$\vdash (q, \varepsilon)$ (4)

---

4). Given, $S \rightarrow aSb | ab$

and string : "aaabbb"

Sol:-

Production	Transition
$S \rightarrow aSb$	$\delta(q, \varepsilon, S) = (q, aSb)$
$S \rightarrow ab$	$\delta(q, \varepsilon, S) = (q, ab)$
a	$\delta(q, a, a) = (q, \varepsilon)$
b	$\delta(q, b, b) = (q, \varepsilon)$

The required PDA is

1) $\delta(q, \varepsilon, S) = (q, aSb)$
2) $\delta(q, \varepsilon, S) = (q, ab)$
3) $\delta(q, a, a) = (q, \varepsilon)$
4) $\delta(q, b, b) = (q, \varepsilon)$

consider, "aaabbb"

$\delta(q, aaabbb, S) \vdash (q, aaabbb, aSb)$ (1)

$\vdash (q, aabbb, Sb)$ (3)

$\vdash (q, aabbb, aSbb)$ (1)

$\vdash (q, abbb, Sbb)$ (3)

$\vdash (q, abbb, abbb)$ (2)

$\vdash (q, bbb, bbb)$ (3)

$\vdash (q, bb, bb)$ (4)

$\vdash (q, b, b)$ (4)

$\vdash (q, \varepsilon)$ (4)

5) $S \rightarrow aS \mid aA$

$A \rightarrow bA \mid b$

String: "aabb"

Sol^n:

Production	Transition
$S \rightarrow aS$	$\delta(q, \varepsilon, S) = (q, aS)$
$S \rightarrow aA$	$\delta(q, \varepsilon, S) = (q, aA)$
$A \rightarrow bA$	$\delta(q, \varepsilon, A) = (q, bA)$
$A \rightarrow b$	$\delta(q, \varepsilon, A) = (q, b)$
a	$\delta(q, a, a) = (q, \varepsilon)$
b	$\delta(q, b, b) = (q, \varepsilon)$

The required PDA is

1) $\delta(q, \varepsilon, S) = (q, aS)$
2) $\delta(q, \varepsilon, S) = (q, aA)$
3) $\delta(q, \varepsilon, A) = (q, bA)$
4) $\delta(q, \varepsilon, A) = (q, b)$
5) $\delta(q, a, a) = (q, \varepsilon)$
6) $\delta(q, b, b) = (q, \varepsilon)$

consider, "aabb"

$\delta(q, aabb, S) \vdash (q, aabb, aS)$ (1)

$\vdash (q, abb, S)$ (5)

$\vdash (q, abb, aA)$ (2)

$\vdash (q, bb, A)$ (5)

$\vdash (q, bb, bA)$ (3)

$\vdash (q, b, A)$ (6)

$\vdash (q, b, b)$ (4)

$\vdash (q, \varepsilon, \varepsilon)$ (6)

---

6). Given,

$S \longrightarrow 0CC$

$C \longrightarrow 0S \mid 1S \mid 0$.

String: "010000"

Sol:

Production	Transition
$S \rightarrow 0CC$	$\delta(q, \varepsilon, S) = (q, 0CC)$
$C \rightarrow 0S$	$\delta(q, \varepsilon, C) = (q, 0S)$
$C \rightarrow 1S$	$\delta(q, \varepsilon, C) = (q, 1S)$
$C \rightarrow 0$	$\delta(q, \varepsilon, C) = (q, 0)$
0	$\delta(q, 0, 0) = (q, \varepsilon)$
1	$\delta(q, 1, 1) = (q, \varepsilon)$

The required PDA is

1) $\delta(q, \varepsilon, S) = (q, 0CC)$

2) $\delta(q, \varepsilon, C) = (q, 0S)$

3) $\delta(q, \varepsilon, C) = (q, 1S)$

4) $\delta(q, \varepsilon, C) = (q, 0)$

5) $\delta(q, 0, 0) = (q, \varepsilon)$

6) $\delta(q, 1, 1) = (q, \varepsilon)$

consider, <u>010000</u>.

$\delta(q, 010000, S)$	$\vdash (q, 010000, 0CC)$	(1)
	$\vdash (q, 10000, CC)$	(5)
	$\vdash (q, 10000, 1SC)$	(3)
	$\vdash (q, 0000, SC)$	(6)
	$\vdash (q, 0000, 0CCC)$	(1)
	$\vdash (q, 000, CCC)$	(5)
	$\vdash (q, 000, 0CC)$	(4)
	$\vdash (q, 00, CC)$	(5)
	$\vdash (q, 00, 0C)$	(4)
	$\vdash (q, 0, C)$	(5)
	$\vdash (q, 0, 0)$	(4)
	$\vdash (q, \varepsilon)$	(5)

7) Given,

$E \rightarrow aAB \mid d$

$A \rightarrow BA \mid a$

$B \rightarrow Ead \mid c$

String: "acadad"

Sol^n:

Production	Transition
$E \rightarrow aAB$	$\delta(q, \varepsilon, E) = (q, aAB)$
$E \rightarrow d$	$\delta(q, \varepsilon, E) = (q, d)$
$A \rightarrow BA$	$\delta(q, \varepsilon, A) = (q, BA)$
$A \rightarrow a$	$\delta(q, \varepsilon, A) = (q, a)$
$B \rightarrow Ead$	$\delta(q, \varepsilon, B) = (q, Ead)$
$B \rightarrow c$	$\delta(q, \varepsilon, B) = (q, c)$
a	$\delta(q, a, a) = (q, \varepsilon)$
c	$\delta(q, c, c) = (q, \varepsilon)$
d	$\delta(q, d, d) = (q, d)$

The required PDA is:

1) $\delta(q, \varepsilon, E) = (q, aAB)$
2) $\delta(q, \varepsilon, E) = (q, d)$
3) $\delta(q, \varepsilon, A) = (q, BA)$
4) $\delta(q, \varepsilon, A) = (q, a)$
5) $\delta(q, \varepsilon, B) = (q, Ead)$
6) $\delta(q, \varepsilon, B) = (q, c)$
7) $\delta(q, a, a) = (q, \varepsilon)$
8) $\delta(q, c, c) = (q, \varepsilon)$
9) $\delta(q, d, d) = (q, \varepsilon)$

consider, acadad

$\delta(q, acadad, E) \vdash (q, acadad, aAB)$ (1)

$\vdash (q, cadad, AB)$ (7)

$\vdash (q, cadad, BAB)$ (3)

$\vdash (q, cadad, CAB)$ (6)

$\vdash (q, adad, AB)$ (8)

$\vdash (q, adad, aB)$ (4)

$\vdash (q, dad, B)$ (7)

$\vdash (q, dad, Ead)$ (5)

$\vdash (q, dad, dad)$ (2)

$\vdash (q, ad, ad)$ (9)

$\vdash (q, d, d)$ (7)

$\vdash (q, \varepsilon)$ (9)