Module - III

Context Free Grammar

Formal Defination : Rules to form a string, inturn forms a language

CFG : (V, T, P, S)

finite of Non-Terminals (captial)

→ finite set of Terminals (lower) $(V \cap T = \phi)$

→ Production rules. $\alpha \rightarrow \beta$

L.H.S R.H.S

Type 2 { only 1 NT $(V \cup T)^*$

→ start symbol. (in left hand side)

Ex: $S \rightarrow 0S1$

$S \rightarrow \epsilon$

start symbol = {S}

T = {0, 1} P = 2 (01) 1

Derivation Tree: {V, T, P, S}

$G : E \rightarrow E+E \,/\, E*E \,/\, E=E \,/\, id$

$W : id + id * id$

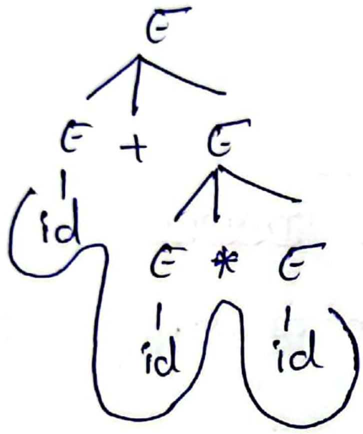
whether a word belong to Grammar or not

Sol:

$$
\begin{aligned}
E &\rightarrow E + E \\
&\rightarrow E + E * E \\
&\rightarrow id + id * id
\end{aligned}
\Bigg\} \text{ Derivation.}
$$

$w \in G$

Pictorial representation is called as Derivation tree / Parse tree



id + id * id.

Sentential form:

* Derivations from the start symbol are called Sentential forms
* given CFG

$G = (V, T, P, S)$ if $S \Rightarrow \alpha$, where $\alpha \in (V \cup T)^*$ then α is a sentential form

* if $S \underset{lm}{\overset{*}{\Rightarrow}} \alpha$ where $\alpha \in (V \cup T)^*$, then α is left-sentential form
* if $S \underset{rm}{\overset{*}{\Rightarrow}} \alpha$ where $\alpha \in (V \cup T)^*$, then α is Right sentential form

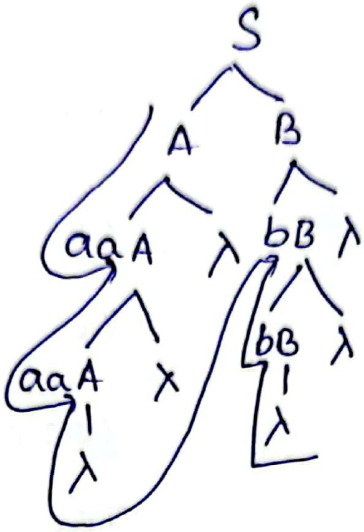
Derivation $\begin{cases} LM \\ RM \end{cases}$

$S \to AB$

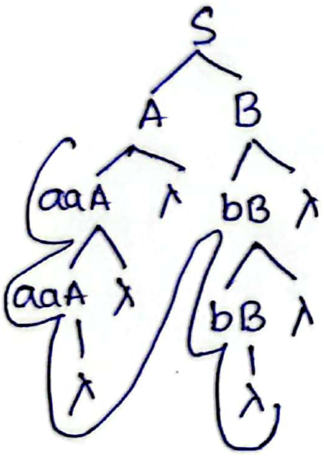
$A \to aaA \mid \lambda$

$B \to bB \mid \lambda$

Check wheather $\omega = aaaabb \in L(G)$



you take only terminal so the string is aaaabb



Types of Derivation Tree

LMDT

Derivation will be done by LM Non-Terminal in PR

RMDT

Derivation will be done from Right most Non-terminal in PR.

Ex: $E \to E+E$
$E \to E*E$
$E \to a \mid b \mid c$

string $a+b*c$

$a+b*c$

E
$E+E$
$a+E$
$a+E*E$
$a+b*E$
$a+b*c$

E
$E+E$
$E+E*E$
$E+E*c$
$E+b*c$
$a+b*c$

(2) $S \rightarrow T00T$

$T \rightarrow 0T$

$T \rightarrow 1T$

$T \rightarrow \epsilon$

String = 1000111

S
$\downarrow$
$T00T$
$\downarrow$
$1T00T$
$\downarrow$
$10T00T$
$\downarrow$
ϵ
$10\epsilon00T$
$1000T$
$1000 1T$
$1000 1T$
$100011T$
$1000111T$
$\downarrow$
ϵ
1000111

S
$\downarrow$
$T00T$
$\downarrow$
$T000T$
$\downarrow$
$T0001T$
$\downarrow$
$T00011T$
$\downarrow$
$T000111T$
$\downarrow$
$T000111\epsilon$
$\downarrow$
$1T000111$
$\downarrow$
ϵ
1000111

Application

$G_2, G_3, G_4, G_5, G_7, G_9$ N_1, N_0, N_6, N_7, N_8

$H_0, H_1, H_2, H_4, H_8, H_9$ $P_1, 3, 5, 6, 8$

$J_0, J_2, J_3, J_6, J_7, J_8, J_9$ L_e 13, 14, 15, 16, 17, 18,

K_2, K_3, K_4, K_6, K_9

L_0, L_4, L_6, L_8,

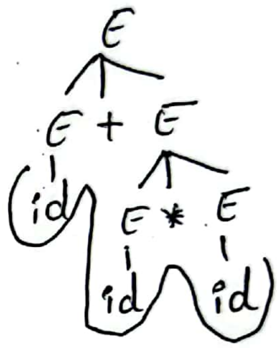
M_0, M_1, M_3, M_6, M_9

Ambiguity in CFG

① $E \rightarrow E+E$
$E \rightarrow E*E$ $\quad\quad$ $id + id * id.$
$E \rightarrow id$

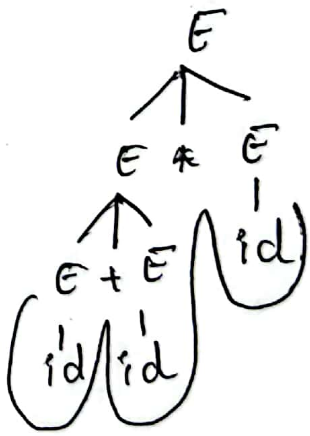
LMD	RMD
$E \Rightarrow E+E$	$E \Rightarrow E+E$
$\Rightarrow id + E$	$\Rightarrow E + E*E$
$\Rightarrow id + E*E$	$\Rightarrow E + E*id$
$\Rightarrow id + id*E$	$\Rightarrow E + id*id$
$\Rightarrow id + id * id$	$\Rightarrow id + id * id$

Derivation tree



$id+id*id$

LMD	RMD
$E \Rightarrow E * E$	$E \Rightarrow E*E$
$\Rightarrow E + E * E$	$\Rightarrow E * id$
$\Rightarrow id + E * E$	$\Rightarrow E+E * id$
$\Rightarrow id + id * E$	$\Rightarrow E + id * id$
$\Rightarrow id + id * id$	$\Rightarrow id + id * id$



* For any Grammar if we derive more than one LMD or RMD or parse tree then the Grammar is said to be Ambig

Ambiguity in Context free grammars:

* "If a string is derived from one leftmost derivation and one right most derivation then that Particular Grammar is called ambigoas grammar."

→ More then one left Most Parse tree.
→ More than one Right Most parse tree.
→ More than one parse tree.

$S \rightarrow ABA$
$A \rightarrow aA / \epsilon$
$B \rightarrow bB / \epsilon$

String "aa"

$S \rightarrow ABA$
$\rightarrow aA\ BA$
$\rightarrow aaABA$
$\rightarrow aa\epsilon BA$
$\rightarrow aa\epsilon A$
$\rightarrow aa\epsilon$
$\rightarrow \boxed{aa}$

Left Most

$S \rightarrow ABA$
$\rightarrow \epsilon BA$
$\rightarrow \epsilon \epsilon BA$
$\rightarrow \epsilon A$
$\rightarrow aA$
$\rightarrow aa\epsilon$
$\rightarrow \boxed{aa}$

If we have ambiguity Grammar will not be used to construct Parser.

(Q) $E \rightarrow E+E$
$E \rightarrow E*E$
$E \rightarrow id$

string id + id * id.

Sol: Right Most derivation

$E \rightarrow E+E$
$\rightarrow E+E*E$
$\rightarrow E+E*id$
$\rightarrow E+id*id$
$\rightarrow id+id*id$

$E \rightarrow E*E$
$\rightarrow E*id$
$\rightarrow E+E*id$
$\rightarrow E+id*id$
$\rightarrow id+id*id.$

∴ the given grammar is ambiguity Grammar

5

Simplification of Context free Grammar: (Reduction of CFG)

In CFG, sometimes all the productions rules and symbols are not needed for the derivation of strings. Besides this, there may also be some Null Productions and UNIT Productions. Elimination of these Productions and symbols is called simplification of CFG

Simplification of context free Grammar of the following steps:

① Reduction of CFG

Phase I

Derivation of an Equivalent Grammar G' from the CFG G such that Each variable derives some terminal string

Derivation Procedure:

Phase I

Step 1: Include all symbols W_1, that derives some terminal and Initialize $i=1$

Step 2: Include symbols W_{i+1}, that derives W_i

Step 3: Increment i and repeat step 2, until $W_{i+1} = W_i$

Step 4: Include all Production rules that have W_i in it

Phase II

Step 1: Derivation of an Equivalent grammar G'', from the CFG G' such that Each symbol appears in sentential form.

Derivation Procedure:

Step 1: Include the start symbol in y_1 and initialize $i=1$

Step 2: Include all symbols y_{i+1} that can be derived from y_i

CFS 2) $S \to A \mid 0C1$

$A \to B \mid 01 \mid 10$

$C \to \epsilon \mid CD$

12/06/23

$G_5, G_6, H_3, 4, 5, 9$

$J_0, J_1, 4, 5, 7$

$K_0, 1, 2, \cancel{3}, 4, 5, 9,$

$L_2, 3, \cancel{4}, 7, 9,$

$M_0, 6, 7, 9$

$N_2, 3, 5, 6,$

$P_0, \cancel{}, 3, 4, 8,$

step: 1 Elimination of useless symbol.

$S \to AB$

$A \to 01$

B is useless.

if symbol is not producing set of terminals.

2

$S \to A$

$S \to 01$ ✓ $A \to B$ ✗ → terminal

$S \to 0C1$ — $S \to 0\epsilon1$ $A \to 01$ ✓ $C \to \epsilon$ ✓

$S \to 01$ ✓ $A \to 10$ ✓ $C \to CD$ ✗

$\cancel{G_1}, G_2, \cancel{G_9}, \cancel{H_0}, H_2$

$H_4, \cancel{J_3}, J_5, J_8, \cancel{K_1}$

$K_4, L_0, L_5, L_7, \cancel{M_4}$

$M_9, N_4; N_8$

$P_2, \cancel{P_3}, P_7$

15.

After Eliminating useless symbols.

Grammar $S \to A$

$S \to 01 / 0C1$

$A \to 01 \mid 10$

$C \to \epsilon$

step 2: Elimination of epslion production.

$S \to A$

$S \to 0\epsilon / 0C1 / 01$

$A \to 01 \mid 10$.

step 3: Elimination of unit Production.

1 $A \to B$ ✗ $B \to x_1, x_2, \ldots x_n$..

$A \to x_1, x_2 \ldots x_n$

2 $S \to A$

$A \to 01 \mid 10$ $S \to 01 \mid 10 / 0C1$

and include all Production rules that have been applied.

Step 3: Increment i and repeat step 2, until $y_{i+1} = y_i$

Ex: Find a Reduced Grammar Equivalent to the Grammar G, having Production rules

$$P: S \to AC/B$$
$$A \to a$$
$$C \to c/BC$$
$$E \to aA/e$$

ACB

a, c, e

Phase 1: Terminal symbols = $\{a, c, e\}$

$W_1 = \{A, C, E\}$

$W_2 = \{A, C, E, S\}$

$W_3 = \{A, C, E, S\}$

$G' = \{(A, C, E, S)(a, c, e), P, S\}$

$$P: S \to AC$$
$$A \to a$$
$$C \to c$$
$$E \to aA/e$$

Phase 2: $Y_1 = \{S\}$ what are derived from S

$Y_2 = \{A, S, C\}$ what are all derived from A, S, C

$Y_3 = \{S, A, C, a, c\}$

$Y_4 = \{S, A, C, a, c\}$

$G'' = \{(A, C, S)(a\ c)\}P, S\}$

$$P: S \to AC \quad C \to c$$
$$A \to a$$

"Reduced production rule."

8/6/2 CFG CF fol

18/6/23 Unit 3 iare

CFG

Chomsky's Normal Form.

+ CFG is in CNF if all productions rules satisfy one of the following conditions:

* start symbol generating ϵ
  eg: $A \to \epsilon$
* A non-terminal generating two non-terminals.
  Eg: $S \to AB$
* A non-terminal generating a terminal.
  Eg: $S \to a$

Example

$G_1 = S \to AB \to$ Rule 2

$S \to c$
$A \to a$
$B \to b$ } Rule 3

So it is a CNF

$G_2 = \{ S \to aA$
$A \to a$
$B \to c \}$

G_2 does not staisfy the rules specified for CNF as $S \to aA$ contains terminal followed by non-terminal

So G_2 is not in CNF

Greibach Normal Form GNF

A CFG is in GNF if all the production rules satisfy one of the following conditions

* A start symbol Generating ϵ
  Eg: $A \to \epsilon$
* A non-terminal generating terminal
  eg: $A \to a$

GNF

* A non-terminal generating a terminal which followed by number of non terminals.

Eg: $S \rightarrow aASB$

Examples

$G_1 = \{ S \rightarrow aAB / aB$
$A \rightarrow aA / a$
$B \rightarrow bB / b \}$

NOT CNF
But GNF

$G_2 = \{ S \rightarrow aAB / aB$
$A \rightarrow aA / \epsilon$
$B \rightarrow bB / \epsilon \}$

NOT GNF

Context free Grammar

CNF $\xleftarrow{\text{Represents}}$ $\xrightarrow{}$ GNF

CNF (Chomsky Normal form)	GNF (Greibach Normal form)
① $A \rightarrow BC$ (or) $A \rightarrow a$	① $A \rightarrow aX$ where $X \in V^*$, $a \in T$
② No. of steps required to generate string of length 'n' is "$2\|n\| - 1$"	② No. of steps required to form a string of length 'n' is 'n'.
③ Membership Algorithm	③ used to convert CFG to PDA
④ Binary tree always.	④ NOT always
⑤ length of Each Production is Restricted	⑤ NOT Restricted

$A \rightarrow AB$
$A \rightarrow a$
$B \rightarrow b.$

ab

A $2|2| - 1$
↓
AB ① $2 \times 2 - 1$
↓
aB ② $4 - 1$
↓
ab ③   ③

$A \rightarrow aB$
$B \rightarrow b$

$A \rightarrow aB$ ②
$\rightarrow ab.$

Conversion of CFG to GNF

GNF

→ Start symbol can generate ϵ

$S \rightarrow \epsilon$ ✓ GNF

→ Non Terminal generating single terminal

$A \rightarrow a$ ✓ GNF

⇒ Non Terminal generating single terminal followed by any no. of terminals (or) Non Terminals.

$A \rightarrow aABCc$ ✓ GNF.

Steps to be followed in converting CFG to GNF

step 1: Make sure that CFG is in CNF.

step 2: Rename Non Terminals with numeric variables in ascending order in which they appeared.

$A \rightarrow BC \longrightarrow A_1 \rightarrow A_2 A_3$

step 3: Consider $A_i \rightarrow A_j$ $i < j$ $i > j$ $i = 0$.

① ② ③

substitution " left Recursion

step 4: Remove left recursion.

step 5: Make Productions following GNF Rules.

Example.

$S \rightarrow CA \mid BB$

$B \rightarrow b \mid SB$

$C \rightarrow b$

$A \rightarrow a$

CNF Rules

$S \rightarrow AB$ ✓

$S \rightarrow a$ ✓

$S \rightarrow \epsilon$ ✓

step 1: satisfied

step 2: $S \rightarrow A_1$ $C \rightarrow A_2$ $A \rightarrow A_3$ $B \rightarrow A_4$

Removal of left Recursion.

Recursion

- Normal: $S \to ASB$
- left: $S \to Sa$
- Right: $S \to aS$

Parsing — Top-down → left Recursion is not suitable.

Example

$A \to A\alpha$

$A \to Aa/b$. left Recursion

$A \to bA'$

$A' \to aA'/\epsilon$

Examples:

$A \to ABd/Aa/a$

$B \to Bc/b$

sol: ① $A \to ABd/Aa/a$

$A \to aA'$

$A' \to BdA'/aA'/\epsilon$

② $B \to Be/b$

$B \to bB'$

$B' \to eB'/\epsilon$

② $E \to E+T/T$

$T \to T*F/F$

$F \to id$

sol: ① $E \to E+T/T$

$E \to TE'$

$E' \to +TE'/\epsilon$

② $T \to T*F/F$

$T \to FT'$

$T' \to *FT'/\epsilon$

③ $A \to Ba/Aa/c$

$B \to Bb/Ab/d$

sol: ① $A \to Ba/Aa/c$

$\cancel{A \to cA'}$

$\cancel{A' \to aA'/aB'/\epsilon}$

$A \to BaA'/cA'$

$A' \to aA'/\epsilon$

② $B \to Bb/Ab/d$.

$\cancel{B \to AbB'/dB'}$

$\cancel{B' \to dB'/\epsilon}$

sub 'A' Production in A

$B \to Bb/BaA'b/cA'b/d$.

	for records		
6	Travel (if some visit is planned)	-	
7	Course Material, stationery and consumables	-	
8	Any other		5
9	**TOTAL**		5
	The Amount of Advance required conducting the program: Rs. T Thousand Rupees.		

$$B \rightarrow cA'B' \,/\, dB'$$

$$B' \rightarrow bB' \,/\, aA'bB' \,/\, \epsilon$$

Example.

$S \to CA \mid BB$
$B \to b \mid SB$
$C \to b$
$A \to a$

sol: $S \to A_1$ $\quad C \to A_2$ $\quad A \to A_3$ $\quad B \to A_4$

$$\left.\begin{aligned} &A_1 \to A_2A_3 \mid A_4A_4 \\ &A_4 \to b \mid A_1A_4 \\ &A_2 \to b \\ &A_3 \to a \end{aligned}\right\} \text{step 2.}$$

step 3: $i > j$

$\cancel{A_4} > 1$ $\quad A_4 \to b \mid A_1A_4$

sub A_1 in A_4

$A_4 \to b \mid A_2A_3A_4 \mid A_4A_4A_4$

$4 > 2$

sub A_2 in A_4

$A_4 \to b \mid bA_3A_4 \mid A_4A_4A_4$

$4 = 4$

left Recursion.

$A_4 \to bZ \mid bA_3A_4Z$

$Z \to A_4A_4Z \mid \epsilon$.

Remove Null production

$Z \to \epsilon$

$\therefore$ $A_4 \to bZ \mid bA_3A_4Z \mid b \mid bA_3A_4$ ✓ GNF

$Z \to A_4A_4Z \mid A_4A_4$

sub A_4 in Z

$Z \to \underbrace{bZ}_{A_4}A_4Z \mid \underbrace{bA_3A_4Z}_{A_4}A_4Z \mid \underbrace{b}_{A_4}A_4Z \mid \underbrace{bA_3A_4}_{A_4}A_4Z \mid$

$A_4 \rightarrow bZ / bA_3A_4Z / b / bA_3A_4 \Leftrightarrow$ GNF

$Z \rightarrow A_4A_4Z / A_4A_4$

Now sub A_4 in Z

$\Leftrightarrow Z \rightarrow \underbrace{bZ}_{A_4}A_4Z / \underbrace{bA_3A_4Z}_{A_4}A_4Z / \underbrace{b}_{A_4}A_4Z / \underbrace{bA_3A_4}_{A_4}A_4Z / \underbrace{bZ}_{A_4}A_4 / \underbrace{bA_3A_4Z}_{A_4}A_4$

$/ \underbrace{b}_{A_4}A_4 / \underbrace{bA_3A_4}_{A_4}A_4 \Leftrightarrow$ GNF

Consider $A_1 \rightarrow A_2A_3 / A_4A_4$

sub A_2 in A_1

$A_1 \rightarrow bA_3 / A_4A_4 \Leftrightarrow$ NOT in GNT

so again sub A_4 in A_1

$A_1 \rightarrow bA_3 / \underbrace{bZ}_{A_4}A_4 / \underbrace{bA_3A_4Z}_{A_4}A_4 / \underbrace{b}_{A_4}A_4 / \underbrace{bA_3A_4}_{A_4}A_4 \Leftrightarrow$ GNF

$A_2 \rightarrow b \Leftrightarrow$ GNF

$A_3 \rightarrow a \Leftrightarrow$ GNF

$\therefore$ we converted given CNF to GNF.

$S \rightarrow XB / AA$

$A \rightarrow a / SA$

$B \rightarrow b$

$X \rightarrow a$

$S \to 0A / 1B / C$

$A \to 0S / 00$ unit Productions.

$B \to 1 / A$

$C \to 01$

Sol: $S \to 0A$ $A \to 0S$ $B \to 1$ $C \to 01$

$S \to 1B$ $A \to 00$ $B \to A$

$S \to C$

$S \to 0A / 1B / 01$

$B \to 1 / 0S / 00$

$S \to 0A / 1B / 01$

$A \to 0S / 00$

$B \to 1 / 0S / 00$

$C \to 01$

Recursion

- Normal: $A \to ASA$, $S \to ASB$
- Left: $S \to Sa$
- Right: $S \to aS$

Parsing
- Top down → left Recursion is not suitable
- Bottom up

① $A \rightarrow Aa / b$ → left Recursion

$\left\{ \begin{array}{l} A' \rightarrow bA' \\ A' \rightarrow aA' / \epsilon \end{array} \right\}$ Process to Remove left Recursion

② $A \rightarrow ABd / Aa / a$

$B \rightarrow Be / b.$

Sol: $A \rightarrow ABd / Aa / a$

① $A' \rightarrow aA'$

$A' \rightarrow BdA' / aA' / \epsilon$

② $B \rightarrow Be / b$

$B' \rightarrow bB'$

$B' \rightarrow BeB' / \epsilon$

③ $E \rightarrow E+T / T$

$T \rightarrow T*F / F$

$F \rightarrow id.$

sol: $E \rightarrow E+T / T$

$E' \rightarrow TE'$

$E' \rightarrow +TE' / \epsilon$

$T \rightarrow T*F / F$

$T' \rightarrow FT'$

$T' \rightarrow *FT' / \epsilon$

$S \rightarrow CA / BB$

$B \rightarrow b / SB$

$C \rightarrow b$

$A \rightarrow a$

step 1: CNF

step 2: $S - A_1$

$C \rightarrow A_2$

$A \rightarrow A_3$

$B \rightarrow A_4$

step 3: $A_1 \rightarrow A_2A_3 / A_4A_4$

④ $A_4 \rightarrow b / A_1A_4$

② $A_2 \rightarrow b$

③ $A_3 \rightarrow a$

step 4: $i > j$

$4 > 1$ (2nd Production)

step 5:

$A_4 \rightarrow b / A_1A_4$

sub A_1 in A_4

$A_4 \rightarrow b / A_2A_3A_4 / A_4A_4A_4$

again $4 > 2$

Now sub A_2 in A_4

$A_4 \rightarrow b \,/\, bA_3A_4 \,/\, A_4A_4A_4$

$4 = 4 \quad i = j \Longleftrightarrow$ left Recursion

* Remove left Recursion.

$A_4 \rightarrow bZ \,/\, bA_3A_4Z$

$Z \rightarrow A_4A_4Z \,/\, \epsilon$

Now Remove Null production and sub ϵ in Z. $Z \rightarrow \epsilon$

$Z \rightarrow A_4A_4Z$

$A_4 \rightarrow bZ \,/\, bA_3A_4Z \,/\, b \,/\, bA_3A_4$ ✓ GNF

$Z \rightarrow A_4A_4Z \,/\, A_4A_4$

sub A_4 in Z to make it GNF.

(3) $Z \rightarrow \underbrace{bZ}_{A_4}A_4Z \,/\, \underbrace{bA_3A_4Z}_{A_4}A_4Z \,/\, \underbrace{b}_{A_4}A_4Z \,/\, \underbrace{bA_3A_4}_{A_4}A_4Z$

$/\, \underbrace{bZ}_{A}A_4 \,/\, \underbrace{bA_3A_4Z}_{A_4}A_4 \,/\, \underbrace{b}_{A_4}A_4 \,/\, \underbrace{bA_3A_4}_{A_4}A_4$ ✓ GNF

Consider

$A_1 \rightarrow A_2A_3 \,/\, A_4A_4$

sub A_2 in A_1

$A_1 \rightarrow bA_3 \,/\, A_4A_4$

sub A_4 in A_1.

(1) $A_1 \rightarrow bA_3 \,/\, bZA_4 \,/\, bA_3 \,/\, bA_3A_4ZA_4 \,/\, bA_4 \,/\, bA_3A_4A_4$ ✓ GNF

(9) $S \to AA/0$
$A \to SS/1$

Sol: Minimized form so, Proceed to steps:

step 1: check for CNF

Yes ✓

step 2: Numerical Variables.

$S \to A_1 \quad A_1 \to A_2$

$A_1 \to A_2A_2/0$

$A_2 \to A_1A_1/1$

step 3: ① $A_1 \to A_2A_2/0 \to ①$

$i=1 \quad j=2$

$i<j \quad 1<2.$

② $A_2 \to A_1A_1/1 \quad 2>1$

sub A_1 in A_2

$A_2 \to A_2A_2A_1/\cancel{A_2A_2}\,0A_1/1$

Make A_2 in GNF Form.

$i=2 \quad j=2$

left Recursion

Eliminate left Recursion

$A_2 \to 0A_1B_2/1B_2$

$B_2 \to A_2A_1B_2/\epsilon$

step 4: Remove Null Productions.

$A_2 \to 0A_1B_2/1B_2/0A_1/1 \Rightarrow \text{GNF} \to ②$

$B_2 \to A_2A_1B_2/A_2A_1 \to ③$

step5: $A_1 \to A_2A_2/0$

Now sub A_2 in A_1

$A_1 \to 0A_1B_2A_2/1B_2A_2/0A_1A_2/1A_2/0 \Rightarrow \text{GNF}$

Step6: Now Make B_2 as GNF

$B_2 \to A_2A_1B_2/A_2A_1$

for that sub A_2 in B_2

$B_2 \to 0A_1B_2A_1B_2/1B_2A_1B_2/0A_1A_1B_2$
$/1A_1B_2/0A_1B_2A_1/1B_2A_1$
$/0A_1A_1/1A_1 \Rightarrow \text{GNF}$

Pumping Lemma for Context Free Language

⇒ Let 'L' be context free language, let 'n' be a integer constant select a string 'z' from 'L' such that $|z| \geq n$.

Divide the string z into 5 parts

uvwxyz such that $|vwx| \leq n$ $|vx| \geq 1$

for $i \geq 0$, uv^iwx^iy is in 'n'

Ex: Show that $L = \{a^n b^n c^n \mid n \geq 1\}$ is not a CFL

sol: Let 'L' is a CFL

let n = 3

L = {abc, aabbcc, aaabbbccc - - - -}

z = aaabbbccc

$|z| \geq n$

|z| = aaabbbccc ≥ n

$9 \geq 3$

divid aaabbbccc into 5 parts
u v w x

u = aa v = a w = b x = b y = bccc

① $|vwx| \leq n \Rightarrow |a.b.b| \leq 3$

$3 \leq 3$ ✓

② $|vx| \geq 1 \Rightarrow |a.b| \geq 1$

$2 \geq 1$ ✓

③ for $i \geq 0$, $uv^iwx^iy \in n$

if $i = 0$ $uv^iwx^iy \in n$

$= aa.a^0.b.b^0bccc$

= aa.b.bccc ∉ 'n' ∴ the L is not a CFL.

Go, G1, G4, G5, G6, G9

Ho, ~~H~~1, 3, 5, 9

J1, 3, ~~4~~, 5, ~~7~~, ~~8~~,

Ko, 4, 5, 9

Lo, 1, 2, 5, 9

Mo, 1, 4, 7,

N1, 2, 3, 5, 6, 8,

Po, 2, 3, 4, 5, 9

LC 13, 14, 15

closure of CFL (Context free Language)

Union ✓

Concatenation ✓

Kleen closure ✓

Intersection ✗

Compliment ✗

$L_1 \to G_1$

$L_2 \to G_2$

Let say $S_1 \to aS_1b / \lambda$

$S_2 \to bS_2c / \lambda$

$S \to S_1 / S_2$ union.

$L_1^* \to G_1 \to S_1$

$S \to S_1 S / \lambda$

$a^* = 0, a, aa, aaa....$

$S \to S_1 \cdot S_2$ Concatination

S_1 when we concatinate both S_1 & S_2 the obtained string should in CFL

↓

1 comparison

Intersection

$L_1 \cap L_2$

always not closed.

$a^n b^n c^m \cap a^m b^n c^n$

$\boxed{a^n b^n c^n} \to$ CSL

↓

no. of comparision

Compliment

$\overline{\overline{L_1} \cup \overline{L_2}}$

CFL is closed under. Compliment

$\overline{CFL \cup CFL}$

$= \overline{CFL} \Rightarrow$ closed.