Unit 2

Types of Grammar

Type 3, 2, 1, 0.

Right Linear Grammar

$A \to \alpha B \mid \beta$

$A, B \in V$

$\alpha, \beta \in T^*$

Left Linear Grammar

$A \to B\alpha \mid \beta$

$A, B \in V$

$\alpha, \beta \in T^*$

- Finite State Automaton or Regular Exp.
- Regular Language.

eg: $A \to aB \mid a$

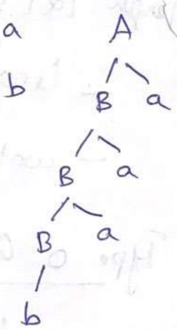
$B \to aB \mid b$

$A \to Ba \mid a$

$B \to Ba \mid b$

baaa

ba



Type 2 Grammar

↳ Context free grammar

$\alpha \to \beta$

$\alpha \in V \;,\; |\alpha| = 1$

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

- Push down Automata
- context free language

eg) $S \rightarrow aS \mid Sa \mid a$

$S \Rightarrow aS$

$\Rightarrow aSa$

$\Rightarrow aa\cancel{S}a$

$\Rightarrow aaaa$

Type 1 Grammar

↳ Context Sensitive Grammar

$\alpha \rightarrow \beta$ ϵ - Not allowed

$\alpha \in (V \cup T)^+$ eg) $\overset{\alpha}{aA} \rightarrow \overset{\beta}{abB}$

$\beta \in (V \cup T)^*$ $S \rightarrow aA$

$\rightarrow abB$

length $|\alpha| \leq \beta$

- Linear bound automata
- Context Sensitive Language

Type 0 Grammar

↳ unrestricted Grammar.

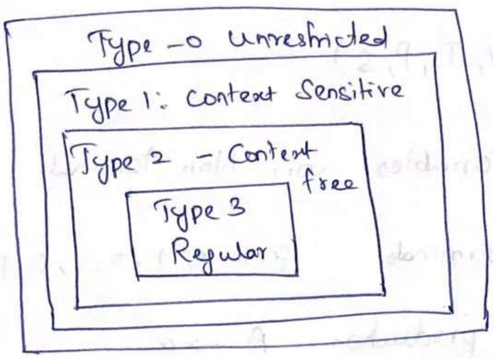
$\alpha \rightarrow \beta$

$\alpha \in (V \cup T)^+$ eg : $aA \rightarrow abcB$

$\beta \in (V \cup T)^*$ $S \rightarrow \epsilon$

- Turing machine
- Recursively Enumerable Language

Chomsky Hierarchy



Context free Language

CFL $\rightarrow$ PDA

$G = \{V, \Sigma, S, P\}$.

V- Set of variable / non terminal Symbol

Σ - Set of terminal Symbol

S- start

P- prod. Rule.

CFG. has prod. rule.

$A \rightarrow \alpha$

where $\alpha = \{V \cup \Sigma\}^*$ & $A \in V$

(eg) For generating language that generates equal no. of a's & b's in the form $a^n b^n$, CFG is defined as $G = \{(S, A), (a, b), (S \rightarrow aAb, A \rightarrow aAb | \epsilon)\}$

$S \rightarrow aAb$

$\rightarrow aaAbb$

$\rightarrow aaaAbbb$

$\rightarrow aaa\epsilon bbb$

$\rightarrow aaabbb$

$\rightarrow a^3b^3 \Rightarrow a^n b^n$

(eg) Context free Grammar (CFG)

CFG ; $G = (V, T, P, S)$

V - Set of Variables (or) Non Terminal.

T - Set of terminals $a, b, *, +, 0, 1, (\ )$

P - Set of production $A \to \alpha$

S - Start Symbol

$A \in V$

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

eg, $E \to E + E$

P = $E \to E * E$ CFG, $G = (\{E, I\}, \{+, *, a\}, P, E\})$

$E \to I$

$I \to a$

$a + a * a$

Derivation $\Rightarrow$ $\overset{*}{\Rightarrow}$ $\underset{lm}{\Rightarrow}$ $\underset{lm}{\overset{*}{\Rightarrow}}$ $\underset{rm}{\Rightarrow}$ $\underset{rm}{\overset{*}{\Rightarrow}}$

1. Left Most derivation (LMD)

2. Right " " (RMD)

LMD: $a + a * a$

$E \underset{lm}{\Rightarrow} E + E$

$\underset{lm}{\Rightarrow} I + E$

$\underset{lm}{\Rightarrow} a + E$

$\underset{lm}{\Rightarrow} a + E * E$

$\underset{lm}{\Rightarrow} a + I * I$

$\underset{lm}{\Rightarrow} a + a * a$

RMD: $a + a * a$

$E \underset{rm}{\Rightarrow} E + E$

$\underset{rm}{\Rightarrow} E + E * E$

$\underset{rm}{\Rightarrow} E + E * I$

$\underset{rm}{\Rightarrow} E + E * a$

$\underset{rm}{\Rightarrow} E + I * a$

$\underset{rm}{\Rightarrow} E + a * a$

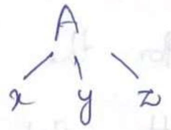
$\underset{rm}{\Rightarrow} I + a * a$

$\underset{rm}{\Rightarrow} a + a * a$

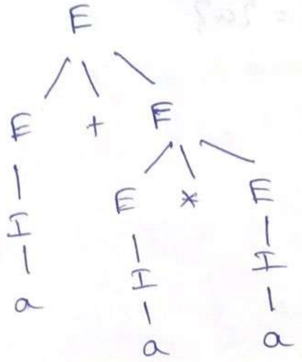
$L(G) = \{ w \text{ in } T^* \mid S \overset{*}{\Rightarrow} w \}$

Parse Tree:

$A \rightarrow xyz$



LMD



a+ a*a

Ambiguity:

CFG, G (V, T, P, S) is ambiguous, if there is atleast one string w in T^*, for which we can find

(i) more than one LMD (different)

(ii) " " " RMD (different)

(iii) " " " parse tree (different)

$E \rightarrow E+E$

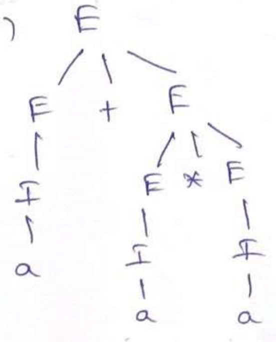
$E \rightarrow E*E$

$E \rightarrow I$

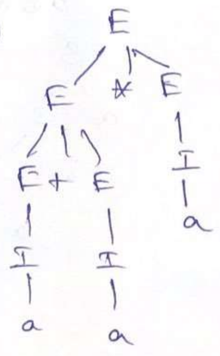
$I \rightarrow a$

a+a*a

(i) LMD



ii) RMD



ambiguous grammar

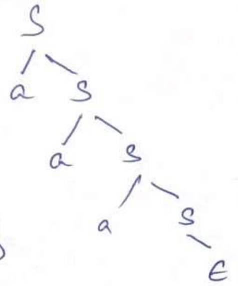
Construction of CFG

1. Construct CFG for the language having any no. of a's over the set $\Sigma = \{a\}$

$L = \{\epsilon, a, aa, aaa \dots\}$

$\{P\}$ $S \to aS$

$S \to \epsilon$



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

$= (\{S\}, \{a\}, \{P\}, \{S\})$

2. Const. a Language for any no. of a & b.

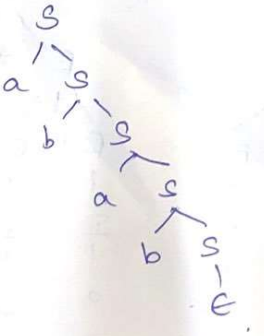
$RE = (a+b)^*$

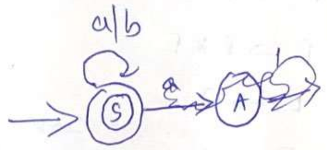
$L = \{\epsilon, a, b, aa, ab, ba \dots\}$

$S \to aS \mid bS \mid \epsilon$

abab

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





3) Atleast 2 a's

RE

$(a+b)^* a (a+b)^* a (a+b)^*$

$S \to AaAaA$

$A \to aA \mid bA \mid \epsilon$

4. One ocurrence of 000

$S \to A\,000\,A$

$A \to 0A \mid 1A \mid \epsilon$

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

$L = \{ab, aabb \dots\}$

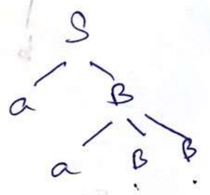
$S \to aSb$

$S \to ab$

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

$S \to aSb$

$S \to abb$.



Derivation

Q1 aabbabba.

$S \to aB \mid bA$

$A \to a \mid aS \mid bAA$

$B \to b \mid bS \mid aBB$

LMD

$S \underset{lm}{\Rightarrow} aB$

$\Rightarrow aaBB$

$\Rightarrow aabB$

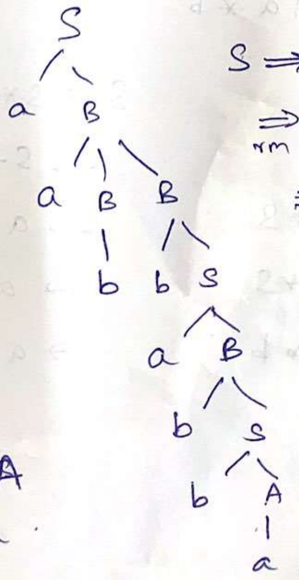
$\Rightarrow aabbS$

$\Rightarrow aabbaB$

$\Rightarrow aabbabS$

$\Rightarrow aabbabbA$

$\Rightarrow aabbabba$.



RMD

$S \Rightarrow aB$

$\underset{rm}{\Rightarrow} aaBB$

$\underset{rm}{\Rightarrow} aaBbS$

$\Rightarrow aaBbbA$

$\Rightarrow aaBbba$

$\Rightarrow aabSbba$

$\Rightarrow aabbAbba$

$\Rightarrow aabbabba$

Ambiguous

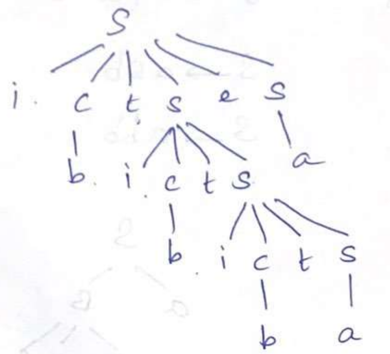
1. Check whether the gn grammar is ambiguous or not for string "ibtibtibtaea"

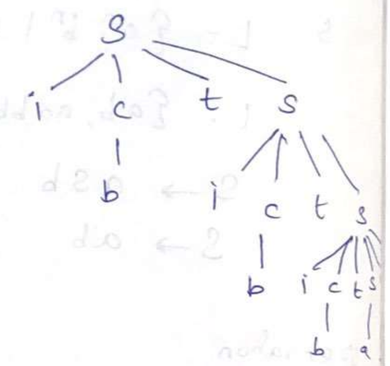
$S \rightarrow ictS$

$S \rightarrow ictSeS$

$S \rightarrow a$

$c \rightarrow b$





It is ambiguous

2. $G = (\{S\}, \{a+b, +, *\}, P, S\}$

$S \rightarrow S+S \mid S*S \mid a \mid b$

String $a+a*b$

$S \rightarrow S+S$
$\rightarrow a+S$
$\rightarrow a+S*S$
$\rightarrow a+a*S$
$\rightarrow a+a*b$

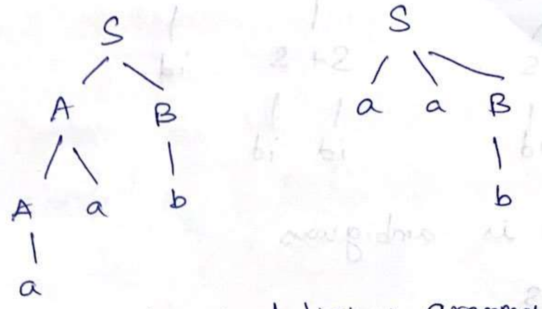
$S \rightarrow S*S$
$\rightarrow S+S*S$
$\rightarrow a+S*S$
$\rightarrow a+a*S$
$\rightarrow a+a*b$

3. $S \Rightarrow AB \mid aaB$

$A \Rightarrow a \mid Aa$

$B \Rightarrow b$.

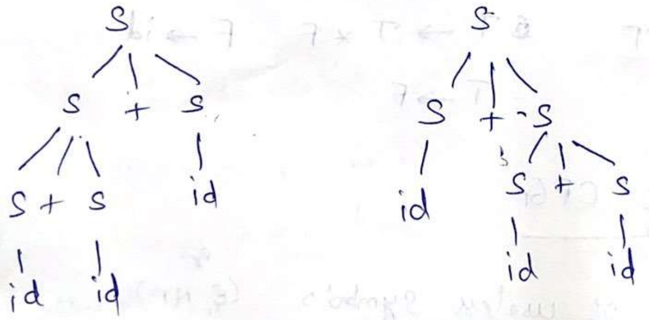
aab → string.



Convert to unambigious grammar.

(4) $S \rightarrow S + id$

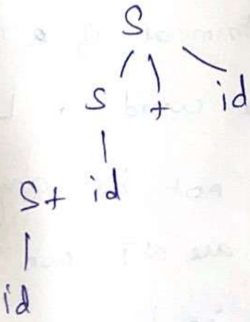
$S \rightarrow id$. String $id + id + id$



It is ambiguous grammar.

$S \rightarrow S + id$

$S \rightarrow id$

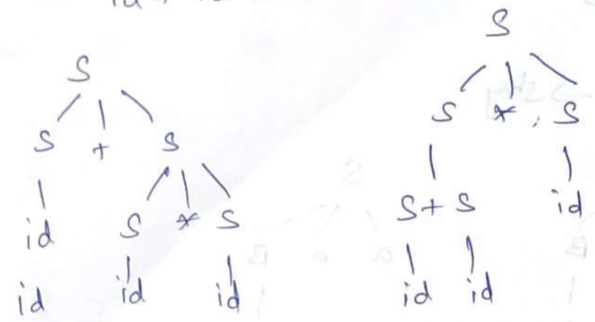


Only one Parse tree

So unambiguous

⑤ $S \rightarrow S+S \mid S*S \mid id$

$id + id * id$



It is ambiguous

$S \rightarrow S+S$

$S \rightarrow S*S$

$S \rightarrow id$

unambiguous grammar.

$S \rightarrow S+T$ $\quad$ $T \rightarrow T*F$ $\quad$ $F \rightarrow id$

$S \rightarrow T$ $\quad$ $T \rightarrow F$

Simplification of CFG

↳ removal of useless symbols (Σ, NT)

properties

1. Each of NT & terminal of a G appear in derivation of some word in L.
2. There should not be any production as $x \rightarrow y$, x & y are NT (non Terminals)
3. If ϵ is not in L, then there need not be a production $x \rightarrow \epsilon$

Reduce grammar.

removal of useless symbols | elimination of ϵ production | removal of unit production

Removal of useless Symbols

A Symbol can be useless if it doesn't appear on the RHS of production or it doesn't take part in the derivation of any string.

① eg, $T \rightarrow aaB \mid abA \mid aaT$

$A \rightarrow aA$

$B \rightarrow ab \mid b$

$C \rightarrow ad$

C is useless symbols

$T \rightarrow aaB$ ✓

$\rightarrow aaab \mid aab$

$T \rightarrow abA$ ×

$\rightarrow abaA$

$\rightarrow abaaA$

$T \rightarrow aaT$ ✓

$\rightarrow aaaaB$

$\rightarrow aaaaab \mid aaaab$

after elimination

$T \rightarrow aaB \mid aaT$

$B \rightarrow ab \mid b$

② $S \rightarrow AB \mid a$

$A \rightarrow BC \mid b$

$B \rightarrow aB \mid C$

$C \rightarrow aC \mid B$

$S \rightarrow a$

3. $S \to a \mid bXY$

$A \to Bad \mid bSX \mid a$ ✓

$B \to aSB \mid bBX$

$X \to SBD \mid aBX \mid ad$ ✓

$Y \to SBa \mid XYb$. ✗

Soln

i) Non generating symbols: B & Y

$S \to a$

$A \to bSX \mid a$

$X \to ad$.

ii) Remove unreachable symbol.

$S \to a$

(4) $S \to aZ \mid SY \mid XA$

$X \to bSZa$ ✗

$Y \to aSY \mid bYZ$ ✗

$Z \to aYZ \mid ad$ ✓

Soln $A \to ab \mid aA$ ✓

(i) Step1

$S \to aZ$

$Z \to ad$

$A \to ab \mid aA$

ii) Step2

$S \to aZ$

$Z \to ad$

Elimination of ϵ production

procedure

Step 1: To remove $A \to \epsilon$ look all prod. whose right side contain A.

Step 2: replace each occurrences of 'A' in each of these prod. with ϵ

Step 3: Add the resultant prod. of grammar.

1) Remove ϵ prod. from grammar.

$S \to ABAC \quad A \to aA \mid \epsilon \quad B \to bB \mid \epsilon \,, \; C \to c$

Soln. Eliminate ϵ

$S \to ABAC$

$S \to ABC \quad (A \to \epsilon)$

$S \to BAC \quad (A \to \epsilon)$

$S \to BC \quad (A \to \epsilon)$

$S \to AAC \quad (B \to \epsilon)$

$S \to C \quad (A \mid B \to \epsilon)$

$A \to aA \quad A \to a$

$B \to bB \quad B \to b$

$C \to c$

$S \to ABAC \mid ABC \mid BAC \mid BC \mid AAC \mid C$

$A \to aA \mid a$

$B \to bB \mid b$

$C \to c$

(2) $S \rightarrow aAB$

$A \rightarrow aAA \mid \epsilon$

$B \rightarrow bBB \mid \epsilon$

$S \rightarrow aAB$ $\quad$ $A \rightarrow aAA$ $\quad$ $B \rightarrow bBB$

$S \rightarrow aB$ $\quad$ $A \rightarrow aA$ $\quad$ $B \rightarrow bB$

$S \rightarrow aA$ $\quad$ $A \rightarrow a$ $\quad$ $B \rightarrow b$

$S \rightarrow a$

$S \rightarrow aAB \mid aB \mid aA \mid a$

$A \rightarrow aAA \mid aA \mid a$

$B \rightarrow bBB \mid bB \mid b$

(3) $S \rightarrow AB$

$A \rightarrow aAA \mid \epsilon$

$B \rightarrow bBB \mid \epsilon$

Removal of unit production

$A \rightarrow B \quad A, B \in NT.$

procedure

1. Remove $A \rightarrow B$, add prod. $A \rightarrow x$ to the grammar rule whenever $B \rightarrow x$ occurs in the grammar.
2. delete $A \rightarrow B$. from G.
3. Repeat Step 1.

1) $P: S \to XY$

$X \to a$

$Y \to Z | b$

$M \to N$

$N \to a$

$Z \to M$

Soln

$Y \to Z$, $Z \to M$, $M \to N$ — NT symbol. $\to$ unit prod.

i) Since $N \to a$

$M \to a$

$P: S \to XY$ $\quad Y \to Z|b$ $\quad M \to a$

$X \to a$ $\quad Z \to M$ $\quad N \to a$.

$P: Z \to a \quad \because M \to a$

$P: Y \to a|b \quad \because Z \to a$.

$P: S \to XY$ $\quad Y \to a|b$ $\quad N \to a$ x

$X \to a$ $\quad M \to a$ x $\quad Z \to a$. x

Removal all unreachable states

$P: S \to XY$ $\quad Y \to a|b$ $\quad X \to a$

2) $S \to Aa|B$ $\quad B \to A|bb$ $\quad A \to a|bc|B$

$S \to B$, $\quad B \to A$ $\quad A \to B$.

$S \to bb$ $\quad B \to a$ $\quad A \to bb$

$B \to bc$

$S \to Aa|bb$

$\boxed{B \to a|bc|bb}$ x

$A \to a|bc|bb$

Chomsky Normal form

In CNF we have restrication of length of RHS, (ie) RHS either two variable (or) a Terminal.

$A \rightarrow a$ $A, B, C \rightarrow NT$

$A \rightarrow BC$ $a \rightarrow T$

CFG to CNF

step 1: If Start symbol S occurs on some right side Create S' new prod. $S' \rightarrow S$

step 2: Remove Null prod.

step 3: Remove unit production

step 4: Replace $A \rightarrow B_1 \cdots B_n$ $n > 2$

$A \rightarrow B_1 C$ $C \rightarrow B_2 \cdots B_n$

Repeat unit two or more symbol in RHS

step 5. If RHS $A \rightarrow aB$

replace $A \rightarrow XB$

$X \rightarrow a$.

eg) $S \rightarrow AAC \mid AC \mid ac \mid a$.

$A \rightarrow aAb \mid ab$

$C \rightarrow ac \mid a$

$A \to XB$
$X \to a$

(i) $P \to a$ length > 2

ii) Break with length $3 \Rightarrow 2$

If body $\Rightarrow k$ $k-2$ cascade symbol needed.

$S \to ABCD$ 2

$S \to AC_1$

$C_1 \to BC_2$

$C_2 \to CD$

(Q) Convert the following CFG into CNF

$S \to aA \mid aBB$

$A \to aAA \mid \epsilon$

$B \to bB \mid bbC$

$C \to B$

(i) ϵ elimination

$S \to aA \mid aBB \mid a$

~~B → bB | b~~ $A \to aAA \mid aA \mid a$

$B \to bB \mid bbC$

$C \to B$

ii) Eliminate unit produ.

$C \to B$

$C \to bbC$

$S \to aA \mid aBB \mid a$ $B \to bB \mid bbC$

$A \to aAA \mid aA \mid a$ $C \to bbC$

iii) Eliminate useless sym.

$S \to aA \mid a$

$A \to aAA \mid aA \mid a$

Reachable symbol = $\{S, a, A\}$

iv) Introduce terminal symbol

$P \to a$

$S \to PA \mid a$

$A \to PAA \mid PA \mid a$

v) CNF

$P \to a$ $\qquad S \to PA \mid a$

Cascade. $\underset{K}{\overset{\text{length}}{3}} - 2 = 1$

$P \to a$

$S \to PA \mid a$

$A \to PC_1 \mid PA \mid a$

$C_1 \to AA$

(2) CFG - CNF

$S \to a \mid aA \mid B$

$A \to aBB \mid \epsilon$

$B \to Aa \mid b$

Create new prod

$S_1 \to S$.

$S \to a \mid aA \mid B$

$A \to aBB \mid \epsilon$

$B \to Aa \mid b$

(i) ϵ elimination.

$S_1 \to S$

$S \to a \mid aA \mid B$

$A \to aBB$

$B \to Aa \mid b \mid a$

v) CNF

$S \to EF \mid AF \mid EB \mid AB$

$X \to AY \mid BY \mid a \mid b$

$Y \to AY \mid BY \mid a \mid b \mid c$

$E \to AX$

$F \to BX$

$A \to a \quad B \to b$

(4) CFG - CNF

$S \to AbA$

$A \to Aa \mid \epsilon$

(i) $S \to AbA \mid bA \mid Ab \mid b$

$A \to Aa \mid a$

ii) $S \to AbA \mid bA \mid Ab$

$A \to Aa.$

iii) $T \to Ab$

$S \to TA \mid bA \mid Ab \mid b$

$A \to Aa \mid a$

$T \to Ab$

iv) $S \to TA \mid BA \mid AB \mid b$

$A \to AC \mid a$

$T \to AB$

$B \to b$

$C \to a$

Greiback Normal form (GNF)

A grammar $G = (V, T, P, S)$ is said to be GNF if every prod. rule is of form

$$X \rightarrow a\alpha$$

where $a \in T$, $X \in V$

$\alpha \in V^*$

RHS of every prod. starts with a terminal followed by a string of variable of zero/more length

(1) Convert CFG to GNF

$S \rightarrow ABA$

$A \rightarrow aA \mid \epsilon$

$B \rightarrow bB \mid \epsilon$

Soln

Simply CFG, eliminate ϵ prod.

$S \rightarrow ABA \mid BA \mid AA \mid AB \mid A \mid B$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

Eliminate unit prod.

$S \rightarrow ABA \mid BA \mid \cancel{BA} \mid AA \mid AB \mid aA \mid a \mid bB \mid b$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b.$

Apply GNF rules.

* $S \rightarrow ABA$

$S \rightarrow aABA \mid aBA$

* $S \rightarrow BA$

$S \rightarrow bBA \mid bA$

* $S \rightarrow AA$

$S \rightarrow aAA \mid aA$

* $S \rightarrow AB$

$S \rightarrow aAB \mid aB$

Resultant Prod.

$S \rightarrow aABA \mid aBA \mid bBA \mid bA \mid aAA \mid aA \mid aAB \mid aB \mid a \mid b$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

② CFG - GNF where $V = \{S, A\}$, $T = \{0, 1\}$ and P is

$S \rightarrow AA \mid 0$

$A \rightarrow SS \mid 1$

Soln. step 1 Gn grammar is in CNF

Step 2: $A_1 \rightarrow S$ $\quad A_2 \rightarrow A$ $\quad\quad A_i \rightarrow A_j B$

$A_1 \rightarrow A_2 A_2 \mid 0$

$A_2 \rightarrow A_1 A_1 \mid 1$

Step 3: i) $A_1 \rightarrow A_2 A_2 \mid 0$ $\quad i = 1 \quad j = 2 \quad i < j$ — ①

ii) $A_2 \rightarrow A_1 A_1 \mid 1$ $\quad i = 2 \quad j = 1 \quad i > j$

$A_2 \rightarrow A_2 A_2 A_1 \mid 0 A_1 \mid 1$

Not in GNF $\quad i = 2 \quad j = 2 \quad i = j$ $\quad A_i \rightarrow A_j \beta \mid \alpha$

Eliminate left recursion. $\quad B \rightarrow \beta \mid \beta B$

$A_i \rightarrow \alpha \mid \alpha B$

$\overset{A_1}{\cancel{A_2}}\overset{\beta}{A_2 A_1}$ $\quad B_2 \rightarrow A_2 A_1 \mid A_2 A_1 B_2$ — ②

$A_2 \rightarrow 0A_1 \mid 1 \mid 0A_1B_2 \mid 1B_2$ — ③

Subs ③ in ①

$A_1 \rightarrow 0A_1A_2 \mid 1A_2 \mid 0A_1B_2A_2 \mid 1B_2A_2 \mid 0$ — ④

Subs ③ in ②

$B_2 \rightarrow 0A_1A_1 \mid 1A_1 \mid 0A_1B_2A_1 \mid 1B_2A_1 \mid 0A_1A_1B_2 \mid$
$1A_1B_2 \mid 0A_1B_2A_1B_2 \mid 1B_2A_1B_2$ — ⑤

GNF $\rightarrow A_1, A_2, B_2$
④ ③ ⑤

(3) CFG - GNF

$S \to AB$

$A \to a$

$B \to CA$

$C \to AB \mid b$

Soln:

Step 1: Gvn grammar is in CNF

Step 2: $S = A_1$, $A \to A_2$, $B \to A_3$, $C \to A_4$

$A_1 \to AB$ $\quad$ $A_1 \to A_2 A_3$

$A_2 \to a$ $\quad$ $A_2 \to a$

$A_3 \to CA$ $\quad$ $A_3 \to A_4 A_2$

$A_4 \to AB \mid b$ $\quad$ $A_4 \to A_2 A_3 \mid b$

Step 3: in $A_1 \to A_2 A_3$ $\quad$ $i=1$ $\quad$ $j=2$ $\quad$ $i<j$

leave prod. $\because i<j$

$\therefore A_1 \to A_2 A_3$ — (1)

ii) $A_2 \to a$ — (2)

iii) $A_3 \to A_4 A_2$ $\quad$ $i=3$ $\quad$ $j=4$ $\quad$ $i<j$

└ (3)

→ always take 1st variable (A_2)

iv) $A_4 \to A_2 A_3$ $\quad$ $i=4$ $\quad$ $j=2$ $\quad$ $i>j$

$i>j$ so replace A_2 by prod.

$A_4 \to aA_3 \mid b$ — (4)

(2) (3) (4) are already in GNF format.

Subs (2) in (1) $\quad$ $A_1 \to aA_3$ — (5)

Subs (4) in (3) $\quad$ $A_3 \to aA_3A_2 \mid bA_2$ — (6)

∴ GNF $\quad$ $A_1 \to aA_3$ $\quad$ $A_3 \to aA_3A_2 \mid bA_2$

$A_2 \to a$ $\quad$ $A_4 \to aA_3 \mid b$