UNIT-3

Grammar:
- Type 3: Regular Grammar (RG)
- Type 2: Context Free Grammar (CFG)
- Type 1: Context Sensitive Grammar (CSG)
- Type 0: Unrestricted Grammar.

Grammar $G(V, T, P, S)$

Type 3: $x \to y$

$x \in V$, $y \in T^*V$ or VT^* or T^*

$y \notin T^*V + VT^* + T^*$

Type 2: $x \to y$

$x \in V^*$, $y \in (V \cup T)^*$

Type 1: $x \to y$

$|x| \leq |y|$

Type 0: $x \to y$

$x \in (V \cup T)^+$

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

Context Free Grammar CFG :-

A context free grammar is $G = (V, T, P, S)$

where,
- → V is a finite set (non-empty) whose elements are called variables
- → T is a finite set (non-empty) whose elements are called terminals.
- → S is a special symbol call start variable.
- → P is finite set of production or rules whose elements are $A \to \alpha$ where A is a variable and α is a string of symbols from $(V \cup T)^*$.
- → [illegible]

Context Free Language: The language generated by context free Grammar is called context free Language.

Suppose G is a grammar of the form CFG, then every production is of the form $A \to \alpha$

where $A \in V$ and $\alpha \in (V \cup T)^*$.

The language generated by G is denoted by $L(G)$. The language L is called context free Language CFL if $L(G)$ is for CFG.

Exp- construct context free grammar G generating all integers (with sign).

Sol: Let $G = (V, T, P, S)$

where $V = \{ S, \langle sign \rangle, \langle digit \rangle, \langle integer \rangle \}$

$T = \{0, 1, 2, \ldots, 9, +, -\}$

S = start variable

P consist of $S \to \langle sign \rangle \langle integer \rangle$

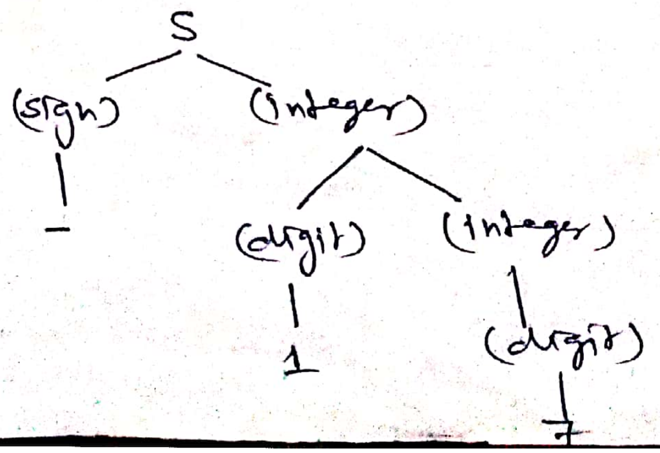
$\langle sign \rangle \to + \mid -$

$\langle integer \rangle \to \langle digit \rangle \langle integer \rangle \mid \langle digit \rangle$

$\langle digit \rangle \to 0 \mid 1 \mid 2 \ldots \mid 9$.

then $L(G)$ = set of all integers.

Exp. Derivation of -17.



Derivation Tree :.

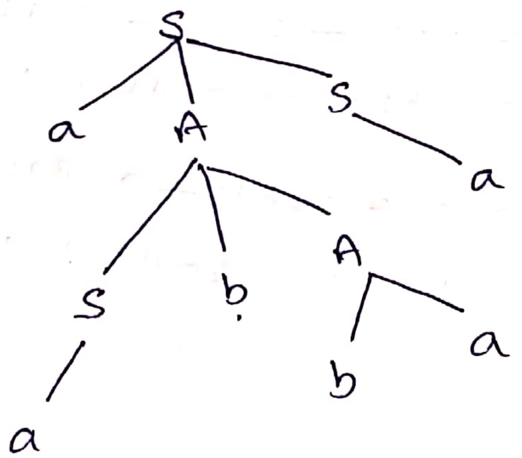
Exp- Consider G whose production are $S \to aAS \mid a$
$A \to SbA \mid SS \mid ba$

show that $S \xrightarrow{*} aabbaa$ and construct a derivation tree whose yield is aabbaa.

Sol. $S \to aAS \to aSbAS \to aabAS \to a^2bbaS \to a^2b^2a^2$
(LM Derivation)

hence $S \xrightarrow{*} a^2b^2a^2$

The derivation tree



Derivation tree with yield aabbaa.

Other derivation of $a^2b^2a^2$ is

$S \to aAS \to aAa \to aSbAa \to aSbbaa \to aabbaa$
(RM Derivation)

other derivation

$S \to aAS \to aSbAS \to aSbAa \to aabAa \to aabbaa$
(Neither Leftmost Right most Derivation)

Definition 1. A derivation $A \xrightarrow{*} w$ is called leftmost derivation if we apply a production only to the leftmost variable, at every steps.

Definition 2: A derivation $A \xrightarrow{*} w$ is called rightmost derivation if we apply a production only to the rightmost variable at every steps.

Q. Let G be the grammar

$S \to 0B \mid 1A$

$A \to 0 \mid 0S \mid 1AA$

$B \to 1 \mid 1S \mid 0BB$.

For the string 00110101, find

(a) the left most derivation

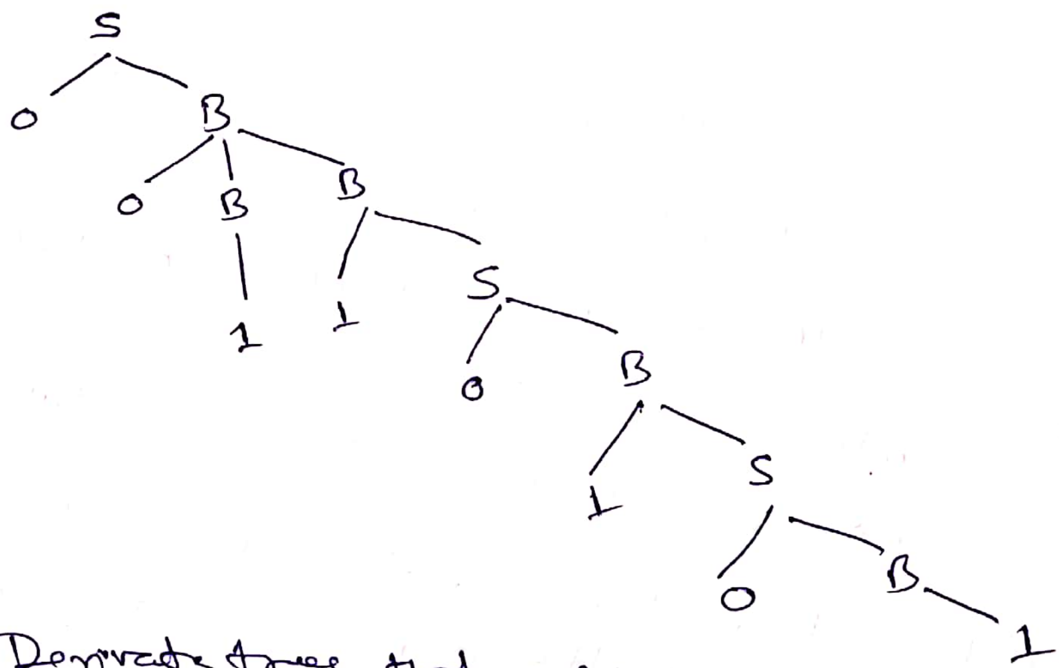
(b) the rightmost derivation

(c) the derivation tree

Sol: (a) $S \to 0B \to 00BB \to 001B \to 0011S \to 0^2 1^2 0B \to 0^2 1^2 01S \to 0^2 1^2 010B \to 00110101$

(b) $S \to 0B \to 00BB \to 00B1S \to 00B10B \to 00B101S \to 00B1010B \to 00B10101 \to 00110101$

(c)



Derivation tree that yield 00110101

Ambiguity in Context Free Grammar :-

The situation in context free language where same ~~sam~~ terminal string may be yield of two derivation trees. So there may be two different leftmost derivation of w. This is the situation of context free language. ambiguous sentences in a context free language.

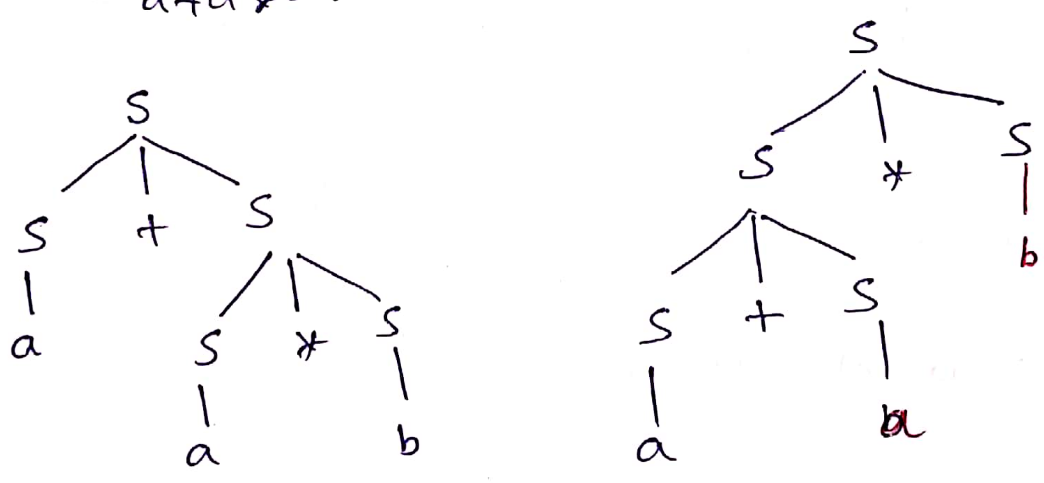
Exp. A terminal string $w \in L(G)$ is ambiguous if there exist two or more derivation tree for w. (or there exist two or more ~~left~~ left most derivation of w).

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

when P consist

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

Now here we have two derivat tree for a+a*b.



Two derivation tree for a+a*b

The leftmost derivation are for a+a*b

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

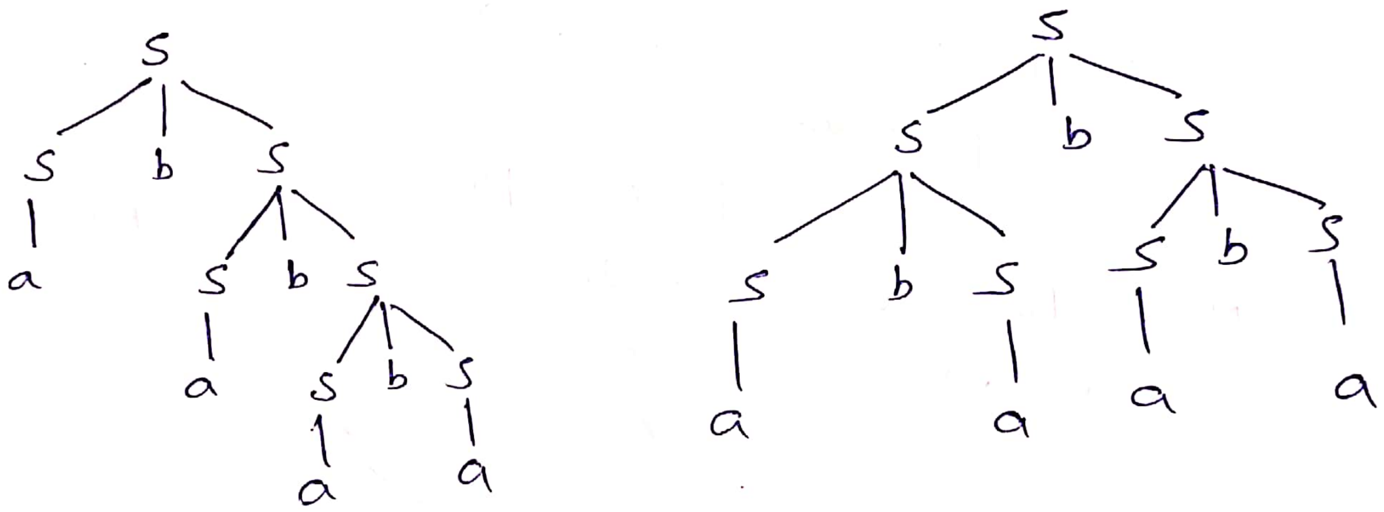
$S \rightarrow S*S \rightarrow S+S*S \rightarrow a+S*S \rightarrow a+a*S \rightarrow a+a*b$

There for a+a*b is ambiguous.

Definition: A context free grammar G is ambiguous if there exist some $w \in L(G)$, which is ambiguous.

Q. If G is the grammar $S \rightarrow SbS \mid a$, show that G is ambiguous.

Sol. To prove that G is ambiguous, we have to find a $w \in L(G)$, which is ambiguous.
Consider $w =$ abababa $\in L(G)$. Then we get two derivation tree for w. Then G is ambiguous.



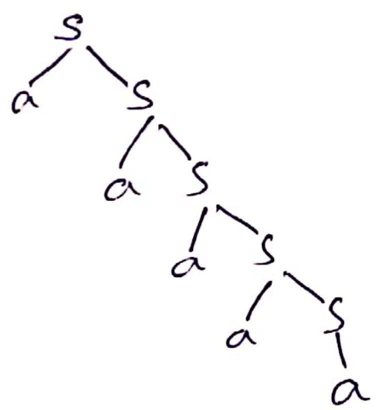
Two derivation tree for abababa.

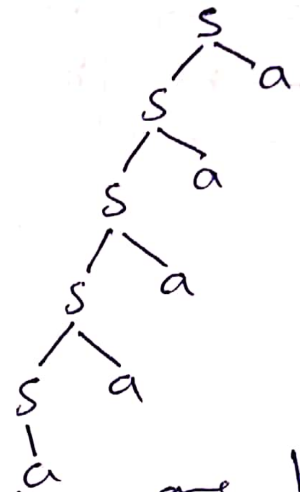
Hence G is ambiguous.

5. Show that $S \rightarrow aS \mid Sa \mid a$ is an ambiguous grammar.

Sol. For the given grammar if there exist more than one parse tree | derivation tree than the grammar is called ambiguous grammar.

For string aaaaa

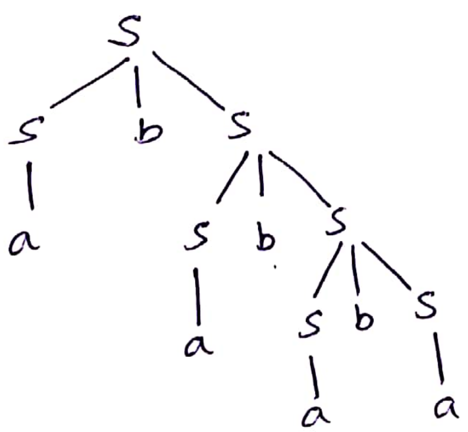




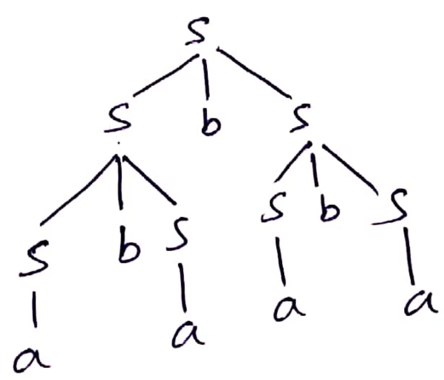
For deriving aaaaa one than one parse tree is there then grammar is ambiguous.

Q. Show that the CFG with production
$S \rightarrow a \mid Sa \mid bSS \mid SSb \mid SbS$ is ambiguous.

Sol. For the string ~~abbaba~~. abababa.



Tree 1



Tree 2

Hence grammar is ambiguous

Inherent Ambiguity :-

A context free grammar is called inherently ambiguous if all the production rules in this grammar are ambiguous.

$S \rightarrow XYZ \mid aaYbb$

$X \rightarrow aaY \mid aa$

$Y \Rightarrow baZ \mid ba$

$Z \rightarrow bZb \mid bb$

Ambiguous to unambiguous CFG:-

For removing ambiguity

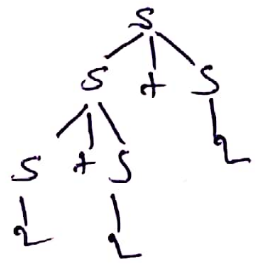
(i) If grammar has left associative operator (+, -, *, /) then induce the left recursion.

(ii) If the grammar has right associative operator (↑) then induce the right recursion.

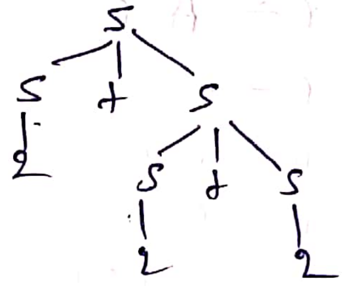
(Exp.)

$S \to S+S$
$S \to 2$

2+2+2



⇒



2+2+2 = 6 | 2+2+2 = 6

$S \to S+S$
$S \to 2$
} is ambiguous

Now the unambiguous form is

$S \to S+2$
$S \to 2$
} is unambiguous.

Exp. $E \to E+E \mid E*E \mid id$

is ambiguous for id+id*id

unambiguous form is

$E \to E+T$
$E \to T$
$T \to T*F$
$T \to F$
$F \to id$
} Unambiguous.

Exp. Show that following grammar is ambiguous.

$E \rightarrow E+E \mid E-E \mid E*E \mid E/E \mid D$

$D \rightarrow a \mid b \mid c \mid d$

Convert to unambiguous grammar

Sol. Consider the string $a+b*c/d$

$E \rightarrow E+A$

$E \rightarrow A$

$A \rightarrow A-B$

$A \rightarrow B$

$B \rightarrow B*C$

$B \rightarrow C$

$C \rightarrow C/\cancel{D}F$

$C \rightarrow \cancel{D}F$

$F \rightarrow D$

$D \rightarrow a \mid b \mid c \mid d$

Unambiguous

Exp. Consider the grammar G with production

$S \rightarrow S+S \mid S*S \mid S\hat{}S \mid a$

(i) Show that grammar is ambiguous

(ii) Write an equivalent unambiguous CFG G_1 which generate same language.

Simplification of Context Free Grammars :-

In a CFG G, it may not be necessary to use all the symbols in $V \cup T$. or all the production in P for deriving sentences; this is called useless symbols.

for examp. $G = (\{S, A, B, C, E\}, \{a, b, c\}, P, S)$

where $P = \{ S \rightarrow AB, A \rightarrow a, B \rightarrow b, B \rightarrow C, E \rightarrow c | \epsilon \}$

Now for $L(G) = \{ab\}$

Let $G' = (\{S, A, B\}, \{a, b\}, P', S\}$.

when $P' = \{ S \rightarrow AB, A \rightarrow a, B \rightarrow b\}$.

$$L(G) = L(G')$$

Here the symbol C, E, c and the production $B \rightarrow C$, $E \rightarrow c | \epsilon$ are eliminated.

Useless symbol/productions :-

The symbol or production that never participated in derivation of the target string w are called useless symbol | production.

In simplification of CFG following points regarding symbols and productions are need to eliminated.

(i) C does not derive any terminal string.
(ii) E and c do not appear in any sentential form.
(iii) $E \rightarrow \Lambda$ is a null production
(iv) $B \rightarrow C$ simply replace B by C.

Removal of Useless symbols:-

A symbol p is useful if there exists some derivation in the following form.

$S \xrightarrow{*} \alpha p \beta$ and $\alpha p \beta \xrightarrow{*} w$.

Then only p will be useful; α and β may be some terminal or non terminal symbol.

Exp- for grammar G

$S \to AB \mid CA$

$B \to BC \mid AB$

$A \to a$

$C \to aB \mid b$

Sol: Here variable are S, A, B, C, in which S, A and C can' be replace by the terminal but B can not replace by any variable which ~~creat~~ create an infinite loop.

Hence B is an ~~usef~~ useless symble.

Now after removing B; Grammar will

$S \to CA$

$A \to a$

$C \to b$.

Exp. $S \to aA \mid bB$

$A \to aA \mid a$

$B \to bB$

$D \to ab \mid Ea$

$E \to aC \mid d$

Sol

$S \to aA$

$A \to aA \mid a$

Steps for Elimination of Null Productions :-

Step 1. Construct set of nullable variable.

(i) $W_1 = \{ A \in V_N \mid A \rightarrow \varepsilon$ is in $P \}$

(ii) $W_{i+1} = W_i \cup \{ A \in V_N \mid$ there exist a production $A \rightarrow \alpha$ with $\alpha \in W_i^* \}$

Step 2. (i) construction of P'.

Any production whose RHS does not have any nullable variable is included in P'.

(ii) If $A \rightarrow X_1 X_2 \cdots X_k$ is in P, the production of the form $A \rightarrow \alpha_1 \alpha_2 \cdots \alpha_k$ are included in P', where $\alpha_j = X_i$ if $X_i \notin W$. $\alpha_i = X_i$ or ε if $X_i \in W$ and $\alpha_1 \alpha_2 \cdots \alpha_k \neq \varepsilon$.

1st gives several production in P'.

Elimination of Null Production:-

A context free grammar may have production of the form $A \to \varepsilon$. The production $A \to \varepsilon$ is just used to erase A. So a production of the form $A \to \varepsilon$, where A is variable, is called null production.

Exp. Consider the grammar G whose productions are

$S \to aS \mid AB$
$A \to \varepsilon$
$B \to \varepsilon$
$D \to b$

construct a grammar G' without null production generating $L(G) - \{\varepsilon\}$.

Sol. Step 1. Construction of the set W of all nullable variables:

$W = \{A, B, S\}$

Construction of Production P':

(i) $D \to b$ is included in P'.

(ii) $S \to aS$ gives rise to $S \to aS$ and $S \to a$.

(iii) $S \to AB$ give rise to $S \to AB$, $S \to A$ and $S \to B$

Hence required grammar without null production is

$G_1 = (\{S, A, B, D\}, \{a, b\}, P', S)$

where P' consists of

$D \to b$
$S \to aS$
$S \to AB$
$S \to a$
$S \to A$
$S \to B$.

Elimination of Unit Productions:

A context free grammar may have production of the form $A \to B$, where $A, B \in V_N$.

Exp - $G: \quad S \to A, A \to B, B \to C, C \to a$

here $L(G) = \{a\}$

The production $S \to A, A \to B, B \to C$ are usefull just to replace S by C. to get a terminal string we need $C \to a$.

If G_1 is $S \to a$ then $L(G_1) = L(G)$.

Step 1. construction of the set of variables derivable from A.

$w_0(A) = \{A\}$

$w_{i+1}(A) = w_i(A) \cup \{B \in V_N \mid C \to B \text{ is in } P \text{ with } C \in w_i(A)\}$

Step 2. Construction of A-production in G_1

The A-production in G_1 are either

(i) the non unit production in G' or

(ii) $A \to \alpha$ whenever $B \to \alpha$ is in G with $B \in W(A)$ and $\alpha \notin V_N$.

Exp. Let G be $S \to AB, A \to a, B \to C \mid b, C \to D, D \to E, E \to a$

Eliminate the unit production and get an equivalent grammar.

Sol. i) $w_0(S) = \{S\} \quad w_1(S) = w_0(S) \cup \phi$

here $w(S) = \{S\}$

$w(A) = \{A\} \quad w(E) = \{E\}$

$w_0(B) = \{B\} \quad w_1(B) = \{B\} \cup \{C\} = \{B, C\}$

$w_2(B) = \{B, C\} \cup \{D\} = \{B\ C\ D\}$

$w_4(B) = \{B, C, D\} \cup \{E\} \Rightarrow w(B) = \{B, C, D, E\}$

~~Exp: Let G be~~

similarly $W(C) = \{C, D, E\}$

$W(D) = \{D, E\}$

(iii) The product in G_1 are

$S \rightarrow AB$ $\quad A \rightarrow a$ $\quad E \rightarrow a$

$B \rightarrow b|a$ $\quad C \rightarrow a$ $\quad D \rightarrow a$

Ans

Normal Forms for Context Free Grammar:-

In a CFG, the RHS of a production can be any string of variable and terminals. When the production in G satisfy certain restrictions then G is said to be in a 'normal form'.

Chomsky Normal form:

In the CNF every production of G is of the form $A \to a$ or $A \to BC$ and $S \to \varepsilon$ is in G if $\varepsilon \in L(G)$. When ε is in $L(G)$, we assume that S do not appear on the RHS of any production.

Exp. Let $S \to AB | \varepsilon$, $A \to a$, $B \to b$ then G in CNF.

Reduction to CNF:-

Let us see an example.

$S \to ABC | aC$, $A \to a$, $B \to b$, $C \to c$

Now, except $S \to ABC | ac$ all other production are in CNF.

(i) In $S \to aC$, terminal a can be replace by an variable D and $D \to a$

ie $S \to DC$, $D \to a$

(ii) $S \to ABC$ is not in required form thus product can be replace by

$S \to AE$ and $E \to BC$

Thus equivalent grammar

$S \to AE | DC$, $E \to BC$, $A \to a$, $B \to b$, $C \to c$, $D \to a$

Note: If the grammar in CNF, the derivation tree has following property: Every node has atmost two decendants, either to internal vertices or a single leaf. Some of the proofs and construction are simpler when grammar in CNF.

Exp. G is $S \to aAD$, $A \to aB \mid bAB$, $B \to b$, $D \to d$.

Sol: step 1. Eliminate all null and unit productions

In this example, no null and unit productions are there.

step 2: Elimination of terminals in RHS.

(i) $B \to b$, $D \to d$ are included in P_1'

(ii) $S \to aAD$ will be $S \to C_a AD$, $C_a \to a$

$A \to aB$ will be $A \to C_a B$

$A \to bAB$ will be $A \to C_b AB$ and $C_b \to b$

$V_N' = \{S, A, B, D, C_a, C_b\}$.

step 3: Restricting the no. of variable in RHS.

P_1 consist

$S \to C_a AD$.
$A \to C_a B \mid C_b AB$
$B \to b$
$D \to d$
$C_a \to a$
$C_b \to b$

Now $A \to C_a B$
$B \to b$
$D \to d$
$C_a \to a$
$C_b \to b$
are added P_2

and $S \to C_a AD$ is replaced by $S \to C_a C_1$ & $C_1 \to AD$

$A \to C_b AB$ is replaced by $A \to C_b C_2$ & $C_2 \to AB$.

Now the CNF of grammar

$G_2 = (\{S, A, B, D, C_a, C_b, C_1, C_2\}, (a, b, d), P_2, S)$

where P_2

$S \to C_a C_1$
$A \to C_a B \mid C_b C_2$
$C_1 \to AD$
$C_2 \to AB$
$B \to b$
$D \to d$

$C_a \to a$
$C_b \to b$

Now G_2 is in CNF and equivalent to G.

Find the grammar in CNF equivalent to (10)

Q. $S \rightarrow aAbB$, $A \rightarrow aA \mid a$, $B \rightarrow bB \mid b$

sol. step 1 ✓

step 2: $A \rightarrow a$, $B \rightarrow b$ added to P_1

$S \rightarrow aAbB$, $A \rightarrow aA$, $B \rightarrow bB$ yield

$S \rightarrow C_aAC_bB$, $A \rightarrow C_aA$, $B \rightarrow C_bB$

$C_a \rightarrow a$, $C_b \rightarrow b$

Now $S \rightarrow C_aAC_bB$ is replaced by

$S \rightarrow C_aC_1$, $C_1 \rightarrow AC_2$, $C_2 \rightarrow C_bB$

New Grammar is CNF is

$G_2 = (\{S, A, B, C_a, C_b, C_1, C_2\}, \{a, b\}, P_2, S)$

where P_2 consists

$S \rightarrow C_aC_1$	$C_a \rightarrow a$
$C_1 \rightarrow AC_2$	$C_b \rightarrow b$
$C_2 \rightarrow C_bB$	$A \rightarrow a$
$A \rightarrow C_aA$	$B \rightarrow b$
$B \rightarrow C_bB$	

Ex. $S \rightarrow \sim S \mid [S \supset S] \mid p \mid q$

find equivalent grammar in CNF.

Greibach Normal form:-

A context free grammar is in Greibach Normal form if every production is of the form
$A \to a\alpha$, where $\alpha \in V_N^*$ and $a \in \Sigma$ (α may be ε).
And $S \to \varepsilon$ is in G if $\varepsilon \in L(G)$, when $\varepsilon \in L(G)$, we assume that S does not appear in RHS of any production.

Note! A CFG is in GNF if productions are of the form $A \to b$
$A \to bC_1C_2 \ldots C_n$

Conversion of CFG to GNF:-

Step1: Elliminate the null and unit production.

Step2: Convert the grammar in CNF.

Step3: Rename the variables as $A_1, A_2, \ldots A_n$ with $S = A_1$.

Step4: ~~Alter the rules~~ Non terminals are in ascending order.

If $A_i \to A_j x$

then $i < j$.

Step5: Remove Left recursion.

Lemma 1: Let $A \to B\gamma$ be an production in P.

Let the B-production be $B \to \beta_1 | \beta_2 | \dots | \beta_s$

Exp. To eliminate B in $A \to B\gamma$, we simply replace B by the right hand size of every B-production. has B as the first symbol on RHS.

Ⓔ $A \to Bab$, $B \to aA | bB | aa | AB$

Now $A \to aAab$,
$A \to bBab$
$A \to aaab$
$A \to ABab$

Lemma 2: Lemma useful to eliminate A from RHS of $A \to A\alpha$.

Let A-production be $A \to A\alpha_1 | \dots A\alpha_r | \beta_1 | \dots | \beta_s$

(β_i's donot start with A)

Now let Z be new variable

(i) The set of A-productions in P_1 are $A \to \beta_1 | \beta_2 | \dots | \beta_s$

$A \to \beta_1 Z | \beta_2 Z | \dots | \beta_s Z$

(ii) The set of Z-products in P_1 are $Z \to \alpha_1 | \alpha_2 | \dots | \alpha_r$

$Z \to \alpha_1 Z | \alpha_2 Z | \dots | \alpha_r Z$

(iii) The production for other variable are in P. ~~...~~

~~...~~ a

Exp- Construct a grammar in GNF equivalent to the grammar $S \to AA \mid a$, $A \to SS \mid b$.

Sol. Step 1. The given grammar is in CNF.
S and A are renamed as A_1 and A_2 respectively. So the productions are

$A_1 \to A_2A_2 \mid a$ and $A_2 \to A_1A_1 \mid b$

As the given grammar has no null productions and is in ~~CFG~~ CNF

Step 2. (i) A_1-productions are in required form. $A_1 \to A_2A_2$
$A_1 \to a$

(ii) $A_2 \to b$ is in required form.

(iii) Apply lemma 1. to $A_2 \to A_1A_1$.

The resulting productions are

$A_2 \to A_2A_2A_1$, $A_2 \to aA_1$

Thus A_2 productions are

$A_2 \to A_2A_2A_1$, $A_2 \to aA_1$, $A_2 \to b$

Step 3. Apply lemma 2 to A_2-productions as we have $A_2 \to A_2A_2A_1$.

Let Z_2 be a new variable, the resulting productions are

$A_2 \to aA_1$, $A_2 \to b$

$A_2 \to aA_1Z_2$, $A_2 \to bZ_2$

$Z_2 \to A_2A_1$, $Z_2 \to A_2A_1Z_2$

Step 4 - (i) The A_2 - productions are

$A_2 \to aA_1 \mid b \mid aA_1Z_2 \mid bZ_2$

(ii) A_1 productions we have
$A_1 \to a$ and eliminate $A_1 \to A_2A_2$

using lemma 1.

$A_1 \to aA_1A_2 \mid bA_2$.

$A_1 \to aA_1Z_2A_2 \mid bZ_2A_2$

Set of all modified A_1 productions

$A_1 \to a \mid aA_1A_2 \mid bA_2 \mid aA_1Z_2A_2 \mid bZ_2A_2$

Step 5 - The Z_2 productions to be modified are
$Z_2 \to A_2A_1$, $Z_2 \to A_2A_1Z_2$, we apply lemma 1.

$Z_2 \to aA_1A_1 \mid bA_1 \mid aA_1Z_2A_1 \mid bZ_2A_1$

$Z_2 \to aA_1A_1Z_2 \mid bA_1Z_2 \mid aA_1Z_2A_1Z_2 \mid bZ_2A_1Z_2$

$Z_2 \to aA_1A_1Z_2 \mid bA_1Z_2$

Pumping Lemma for Context Free Languages:-

Let L be a CFL. The we can find a natural number n such that:

(i) Every $z \in L$ with $|z| \geq n$, can be written as $uvwxy$ for some string u, v, w, x, y.

(ii) $|vx| \geq 1$

(iii) $|vwx| \leq n$

(iv) $uv^k w x^k y \in L$ for all $k \geq 0$

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

Sol. Step 1. Assume L is CFL. Let n be the natural number obtained by using the pumping lemma.

Step 2. Let $z = a^n b^n c^n$

then $|z| = 3n > n$

we can write $z = uvwxy$, when $|vx| \geq 1$

i.e. at least one of the v or x is not ϵ.

Step 3. $uvwxy = a^n b^n c^n$, as $1 \leq vx \leq n$

v or x can not contain all three symbols a, b, c.

(i) v or x is of the form $a^i b^j$ (or $b^i c^j$), $i+j \leq n$

or (ii) v or x is a string formed by repetition of only one symbol among a, b, c.

Now $v = a^i b^j$

$v^2 = a^i b^j a^i b^j$

then uv^2wx^2y will never be of the form $a^m b^m c^m$

So $uv^2wx^2y \notin L$.

Hence for k

$uv^k w x^k y \notin L$.

Prove L is not CFL.