Context Free Grammar :- A context free grammar is denoted as $G = (V, T, P, S)$

where V is finite set of variables
T is finite set of terminals
P is a finite set of Productions.
S is a starting symbol of Grammar.

where P is defined :-

Each Production is of the form $\boxed{A \to \alpha}$

where A is a variable and $\alpha \in (V \cup T)^*$

A Language generated by context Free Grammar, is called context Free Language.

A Language generated by Grammar is a string is in $L(G)$ if :

1.) The string consists of terminals.
2.) The string can be derived from S.

So we call L a context-Free Language (CFL) if it is $L(G)$ for some CFG G.

* We define Grammars G_1 & G_2 to be equivalent if $L(G_1) = L(G_2)$

Example :- The Grammar $G = (\{S\}, \{a, b\}, P, S)$, with Productions $S \to aSa \mid bSb \mid \epsilon$. is a CFG.

$\therefore$ $L(G) = \{ ww^R : w \in \{a,b\}^* \}$ is a CFL.

* We categorize the derivation of context Free Grammars in two categories.

1. left most derivation (LMD).
2. Right most derivation (RMD).

LMD (left most derivation) :- In left Most derivation, left most variable of the Right hand side always expanded to generate string.

RMD (Right most derivation) :- In Right most derivation, Right most variable of the Right hand side always expanded to generate string.

* Parse Tree / derivation Tree :- Parse Tree is constructed to generate valid string of the grammar. It has following properties.

① Root of the tree will always be start variable of the grammar.
② All Internal node must be variable of the grammar.
③ All leaf nodes are the terminal of the grammar.
④ If ϵ is a leaf node then it should not have any sibling.

No. of derivation Tree $\geq$ No. of words generated by it.

Q1) Consider the CFG whose Productions are:

$S \to aAS / a$

$A \to ba$

For the string abaa, find
① The left most derivation ② The Right most derivation
③ The derivation Tree

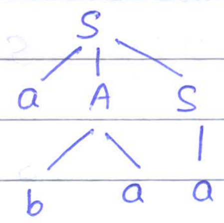
Solution :-

① LMD :-

$S \to a\underline{A}S$

$S \to aba\underline{S}$

$S \to abaa$



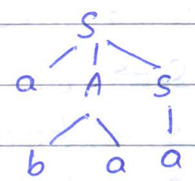
derivation Tree

② RMD :-

$S \to aA\underline{S}$

$S \to a\underline{A}a$

$S \to abaa$



derivation Tree

Q2) Consider the CFG G whose Productions are

$S \to aAS \mid a$

$A \to SbA \mid SS \mid ba$

Show that $S \to aabbaa$ and construct a derivation Tree.

Solution:-

LMD:-

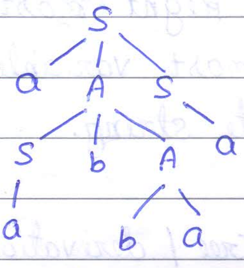
$S \to aAS$

$S \to aSbAS$

$S \to aabAS$

$S \to aabbaS$

$S \to aabbaa$



derivation Tree.

RMD:-

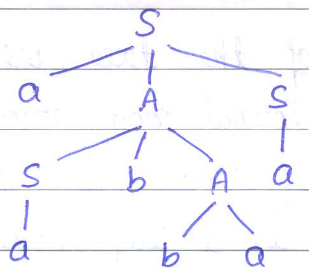
$S \to aAS$

$S \to aAa$

$S \to aSbAa$

$S \to aSbbaa$

$S \to aabbaa$



derivation Tree.

Q3) Consider the CFG G whose Productions are:-

$S \to bB \mid aA$

$A \to b \mid bS \mid aAA$

$B \to a \mid aS \mid bBB$

For the string bbaababa find LMD, RMD, Parse Tree.

Solution:-

LMD:-

$S \to bB$

$S \to bbBB$

$S \to bbaB$

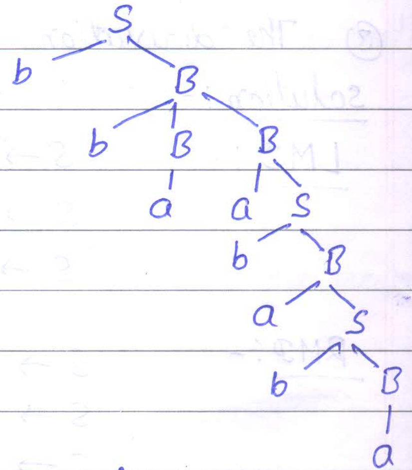
$S \to bbaaS$

$S \to bbaabB$

$S \to bbaabaS$

$S \to bbaababB$

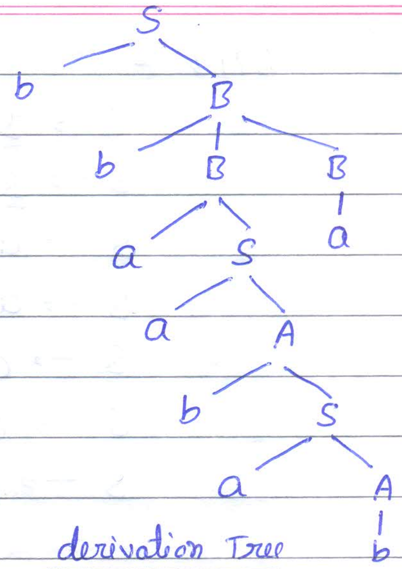
$S \to bbaababa$



derivation Tree

<u>RMD</u> :-

$S \rightarrow bB$
$S \rightarrow bbB\underline{B}$
$S \rightarrow bb\underline{B}a$
$S \rightarrow bba\underline{S}a$
$S \rightarrow bbaa\underline{A}a$
$S \rightarrow bbaab\underline{S}a$
$S \rightarrow bbaaba\underline{A}a$
$S \rightarrow bbaababa$



<u>derivation Tree</u>

⇒ Ambiguity in context Free Grammar :-

The same terminal string can be generated more than one derivation Tree so the Grammar is ambiguous.

<u>or</u>

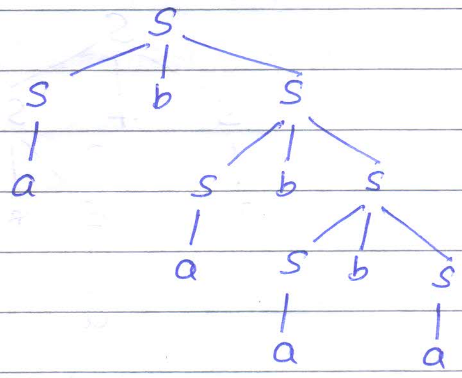
A Terminal string $w \in L(G)$ is ambiguous if there exist more than one derivation Trees for w.

Q1) If G is a Grammar $S \rightarrow SbS/a$, show that G is ambiguous.

<u>Solution</u>:- consider string is ababab a.

<u>LMD</u>:-

$S \rightarrow \underline{S}bS$
$S \rightarrow abS$
$S \rightarrow ab\underline{S}bS$
$S \rightarrow abab\underline{S}$
$S \rightarrow abab\underline{S}bS$
$S \rightarrow ababab\underline{S}$
$S \rightarrow abababa$



Derivation Tree for abababa

LMD:-

$S \rightarrow \underline{S}bS$

$S \rightarrow \underline{S}bSbS$

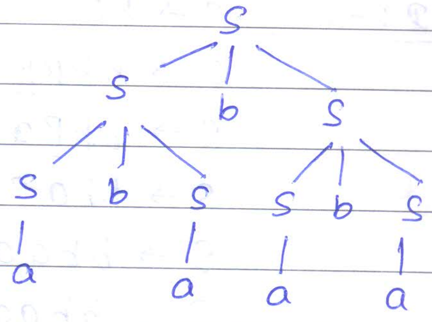
$S \rightarrow ab\underline{S}bS$

$S \rightarrow abab\underline{S}$

$S \rightarrow abab\underline{S}bS$

$S \rightarrow ababab\underline{S}$

$S \rightarrow abababa$



Derivation Tree for abababa

Here more than one derivation Tree for string abababa. Thus Grammar is ambiguous.

Q2) If G is the Grammar $S \rightarrow S+S \mid S*S \mid a \mid b$. Show that G is ambiguous for string $a+a*b$.

Solution:-

LMD:-

$S \rightarrow \underline{S}+S$

$S \rightarrow a+\underline{S}$

$S \rightarrow a+\underline{S}*S$

$S \rightarrow a+a*\underline{S}$

$S \rightarrow a+a*b$

LMD:-

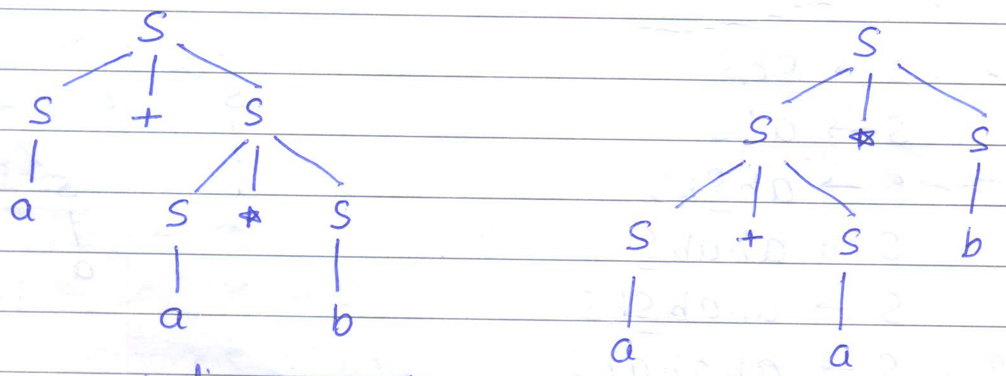
$S \rightarrow \underline{S}*S$

$S \rightarrow \underline{S}+S*S$

$S \rightarrow a+\underline{S}*S$

$S \rightarrow a+a*\underline{S}$

$S \rightarrow a+a*b$



Derivation Tree for string $a+a*b$.

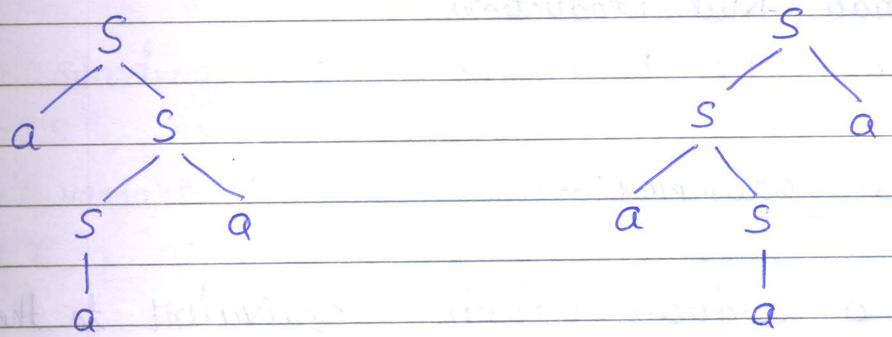
Here more than one derivation Tree for string $a+a*b$. Thus Grammar G is ambiguous.

Q3) If G is the Grammar $S \to aS / Sa / a$. show that G is ambiguous for string aaa.

Solution:-

LMD:- $S \to a\underline{S}$
$S \to a\underline{S}a$
$S \to aaa$

LMD:- $S \to \underline{S}a$
$S \to a\underline{S}a$
$S \to aaa$



Derivation Tree for string aaa.

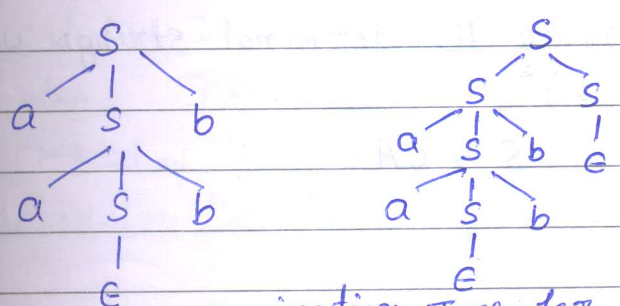
Here more than one derivation Tree for string aaa. Thus Grammar G is ambiguous.

Q4) If G is a Grammar $S \to aSb / SS / \epsilon$. Show that G is ambiguous.

Solution:- Take a string aabb. (Generated by G)

LMD:- $S \to a\underline{S}b$
$S \to aa\underline{S}bb$
$S \to aa\epsilon bb$
$S \to aabb$

LMD:- $S \to \underline{S}S$
$S \to a\underline{S}bS$
$S \to aaSbbS$
$S \to aa\epsilon bbS$
$S \to aabb\underline{S}$
$S \to aabb\epsilon$
$S \to aabb$



Derivation Tree for string aabb.

Here more than one derivation Tree for string aabb. Thus Grammar G is ambiguous.

⇒ Simplification of context Free Grammar.

For simplification we Perform:-

1. Eliminate variables not deriving terminal strings.
2. Eliminate symbols not appearing in any sentential form.
3. Eliminate Null Productions.
4. Eliminate unit Productions.

⇒ Reduced Grammar:-

Q1) Find a reduced Grammar equivalent to the Grammar G whose Productions are:-

$S \rightarrow AB / CA$

$B \rightarrow BC / AB$

$A \rightarrow a$

$C \rightarrow aB / b$

Solution:- Assume given Productions of Grammar is P.

Step 1:- Eliminate the variables not deriving terminal strings.
(Find set of all Productions that generate terminal strings)

Construction of P_1 from P.

① Productions which generate the terminal strings Direct.

$A \rightarrow a$ and $C \rightarrow b$

Here, $V_1 = \{A, C\}$

② Productions which generate the terminal strings with the help of $(\Sigma \cup \{A, C\})^*$

$A \rightarrow a$, $C \rightarrow b$, $S \rightarrow CA$

Here $V_2 = \{A, C, S\}$

③ Productions which generate terminal string with the help of $(\Sigma \cup \{A, C, S\})^*$

$A \rightarrow a$, $C \rightarrow b$, $S \rightarrow CA$

Production rules are same, so stop the Procedure.

so $P_1 = \{A \rightarrow a, C \rightarrow b, S \rightarrow CA\}$

step 2:- Eliminate symbols not appearing in any sentential form. (construction of P_2 from P_1).

The starting symbol is S.

① for S, as we have $S \rightarrow CA$
New variable is C and A.

② for C, as we have $C \rightarrow b$

③ for A, as we have $A \rightarrow a$

No new variable occurs, so stop the Procedure. so

$P_2 = \{S \rightarrow CA, C \rightarrow b, A \rightarrow a\}$

Thus the Reduced Grammar $G = (\{S, A, C\}, \{a, b\}, P_2, S)$ where P_2 is defined by $\{S \rightarrow CA, C \rightarrow b, A \rightarrow a\}$.

Q2) Find a reduced grammar equivalent to the grammar G whose Productions are:-

$S \rightarrow aAa$

$A \rightarrow Sb \mid bCC \mid DaA$

$C \rightarrow abb \mid DD$

$E \rightarrow aC$

$D \rightarrow aDA$.

Solution:- Assume given Productions of Grammar is P.

step 1:- Eliminate the variables not deriving terminal strings.

Find set of all Productions that generate terminal strings.

(construction of P_1 from P.)

(1) Productions which generate the terminal strings direct.

$C \rightarrow abb$

Here $V_1 = \{C\}$

(2) Productions which generate the terminal strings with the help of $(\Sigma \cup \{C\})^*$.

$C \rightarrow abb$, $E \rightarrow aC$, $A \rightarrow bCC$

Here $V_2 = \{A, E, C\}$

(3) Productions which generate the terminal strings with the help of $(\Sigma \cup \{A, E, C\})^*$.

$C \rightarrow abb$, $E \rightarrow aC$, $A \rightarrow bCC$, $S \rightarrow aAa$

Here $V_3 = \{S, A, E, C\}$

(4) Productions which generate the terminal strings with the help of $(\Sigma \cup \{S, A, E, C\})^*$.

$C \rightarrow abb$, $E \rightarrow aC$, $A \rightarrow bCC$, $S \rightarrow aAa$, $A \rightarrow Sb$.

So Productions $P_1 = \{S \rightarrow aAa, A \rightarrow Sb, A \rightarrow bCC,$
$C \rightarrow abb, E \rightarrow ac\}$

Step 2:- Eliminate symbols not appearing in any sentential form. (construction of P_2 from P_1)

The starting symbol is S.

(1) for S, as we have $S \rightarrow aAa$.
New Variable is A.

(2) for A, as we have $A \rightarrow Sb$, $A \rightarrow bCC$
New Variable is C.

(3) for C, as we have $C \rightarrow abb$

Thus Productions $P_2 = \{S \rightarrow aAa, A \rightarrow Sb, A \rightarrow bCC, C \rightarrow abb\}$

① Productions which generate the terminal strings direct.

$C \to abb$

Here $V_1 = \{C\}$

② Productions which generate the terminal strings with the help of $(\Sigma \cup \{C\})^*$.

$C \to abb$, $E \to aC$, $A \to bCC$

Here $V_2 = \{A, E, C\}$

③ Productions which generate the terminal strings with the help of $(\Sigma \cup \{A, E, C\})^*$.

$C \to abb$, $E \to aC$, $A \to bCC$, $S \to aAa$

Here $V_3 = \{S, A, E, C\}$

④ Productions which generate the terminal strings with the help of $(\Sigma \cup \{S, A, E, C\})^*$.

$C \to abb$, $E \to aC$, $A \to bCC$, $S \to aAa$, $A \to Sb$.

So Productions $P_1 = \{ S \to aAa, A \to Sb, A \to bCC, C \to abb, E \to aC \}$

Step 2:- Eliminate symbols not appearing in any sentential form. (construction of P_2 from P_1)

The starting symbol is S.

① for S, as we have $S \to aAa$.
New variable is A.

② for A, as we have $A \to Sb$, $A \to bCC$
New variable is C.

③ for C, as we have $C \to abb$

Thus Productions $P_2 = \{ S \to aAa, A \to Sb, A \to bCC, C \to abb \}$

Q2) Eliminate ε-Productions from CFG.

$S \rightarrow ABCD$
$A \rightarrow CDa$
$C \rightarrow a / \epsilon$
$D \rightarrow bD / \epsilon$

Solution:- Step 1:- construction of the set of Nullable variables.
$V = \{ C, D \}$.

Step 2:- construction of P_1 (Productions)

$S \rightarrow ABCD / ABD / ABC / AB$
$A \rightarrow CDa / Ca / Da / a$
$C \rightarrow a$
$D \rightarrow bD / b$

Thus the Grammar $G = (\{A, B, S, C, D\}, \{a, b\}, P_1, S)$
where P_1 is defined above.

⇒ Eliminate unit Productions.

Q1) Remove all unit Productions from

$S \rightarrow Aa / B$
$B \rightarrow A / bb$
$A \rightarrow a / bc / B$

Solution:-

Step 1:- Seperate Unit Productions and Non-unit Production.

① original Non-unit production.

$S \rightarrow Aa$
$B \rightarrow bb$
$A \rightarrow a / bC$

(2) Unit Production

$S \rightarrow B$
$B \rightarrow A$
$A \rightarrow B$

Step 2:- find set of all unit Production:-

$S \rightarrow B$, $B \rightarrow A$, $A \rightarrow B$, $S \xrightarrow{*} A$.

Step 3:- convert the set of all Unit Production into Non-unit Production.

$S \rightarrow B$ gives $S \rightarrow bb$
$B \rightarrow A$ gives $B \rightarrow a/bc$
$S \xrightarrow{*} A$ gives $S \rightarrow a/bc$
$A \rightarrow B$ gives $A \rightarrow bb$

so converted Non-unit Productions are

$S \rightarrow a/bb/bc$
$A \rightarrow bb$
$B \rightarrow a/bc$

Step 4:- combine origional & converted Non-Unit Production.

$S \rightarrow Aa/a/bb/bc$
$A \rightarrow a/bc/bb$
$B \rightarrow bb/a/bc$

So the Grammar $G = (\{S,A,B\}, \{a,b,c\}, P_2, S)$ where P_2 is defined as

$S \rightarrow Aa/a/bb/bc$
$A \rightarrow a/bc/bb$
$B \rightarrow bb/a/bc$.

Q2) let G be $S \to AB$, $A \to a$, $B \to C/b$, $C \to D$, $D \to E$ and $E \to a$.
Eliminate unit Productions and get an equivalent Grammar.

Solution:-

Step 1:- seperate unit Production and Non-unit Production.

① original Non-unit Production

$S \to AB$, $A \to a$, $B \to b$, $E \to a$

② unit Production

$B \to C$, $C \to D$, $D \to E$.

Step 2:- Find the set of all unit Production:-

$B \to C$, $C \to D$, $D \to E$, $B \xrightarrow{*} D$,
$B \xrightarrow{*} E$, $C \xrightarrow{*} E$.

Step 3:- Convert the set of all unit Production into Non-unit Production.

$B \to C$, $C \to D$, $B \xrightarrow{*} D$ Not gives any productions.

$D \to E$ gives $D \to a$

$B \xrightarrow{*} E$ gives $B \to a$

$C \xrightarrow{*} E$ gives $C \to a$

so converted Non-unit Productions are

$D \to a$, $B \to a$, $C \to a$.

Step 4:- Combine origional & converted Non-unit Productions.

$S \to AB$

$A \to a$

$B \to a/b$

$C \to a$, $D \to a$, $E \to a$

So the grammar $G = (\{S, A, B, C, D, E\}, \{a, b\}, P_2, S)$
where P_2 is defined as $\{S \to AB, A \to a, B \to a/b,$
$C \to a, D \to a, E \to a\}$.