Unit - III

Context free languages

→ Context-free language (CFL) is a larger class of language than regular language.

→ These languages have a natural recursive notation called context-free grammars (CFG).

→ Major application of CFG is in programming language and compiler design.

Ex: Some of the context-free languages are

1) $L = \{ a^n b^n \mid n \geq 1 \}$
2) $L = \{ ww^R \mid w \in \{a,b\}^* \}$
3) $L = \{ a^n b^{2n} \mid n \geq 1 \}$
4) $L = \{$ no of a's in the string = no of b's in the string $\}$
5) $L = \{ a^n b^n c^m \mid n,m \geq 1 \}$
6) $L = \{ a^n b^m c^n \mid n,m \geq 1 \}$
7) $L = \{ a^n a^m b^m c^n \mid n,m \geq 1 \}$
8) $L = \{ a^n b^n b^m c^m \mid n,m \geq 1 \}$
9) $L = \{$ set of all strings which are all palindromes over $\{a,b\} \}$

Context-free grammar (CFG) : (Type 2)

A grammar $G = (V, T, P, S)$ is said to be CFG if the productions of G are of the form:

$$A \rightarrow \alpha$$

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

The right hand side of a CFG is not restricted and it may be null (or) a combination of variables and terminals. The possible length of right hand sentential form ranges from 0 to ∞ $(0 \leq |\alpha| \leq \infty)$

Ex: 1) $S \rightarrow aSa / aSb / \epsilon$

2) $S \rightarrow 0A / \epsilon$

$A \rightarrow 1S$

3) $S \rightarrow aCa$

$C \rightarrow aCa / b$

1) Construct CFG for the language

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

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

$S \rightarrow aSb \mid ab$

2) obtain CFG for the language

$$L = \{w \mid n_a(w) = n_b(w)\}$$

Solution: $L = \{\epsilon, ab, ba, abab, baba, abba, baab, \ldots\}$

$S \rightarrow SS \mid \epsilon$

$S \rightarrow aSb \mid bSa$

3) Construct CFG for the language L which has all the strings which are all palindromes over $\{a, b\}$.

Solution: $L = \{\epsilon, a, b, aa, aba, bab, abba, ababa, \ldots\}$

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

4) Construct a CFG for the language $L = \{a^n b^{2n} \mid n \geq 1\}$

Solution: $L = \{abb, aabbbb, \ldots\}$

$S \rightarrow aSbb \mid abb$

5) Construct a CFG for the language $L = \{a^n b^{2n} \mid n \geq 0\}$

Solution: $L = \{\epsilon, abb, aabbbb, \ldots\}$

$S \rightarrow aSbb \mid \epsilon$

6) Construct a CFG for the language $L = \{a^n b^n \mid n \geq 0\}$

Solution: $L = \{\epsilon, ab, abab, ababab, \ldots\}$

$S \rightarrow aSb \mid \epsilon$

7) Construct a CFG for the language

$L = \{ww^R \mid w \in \{a,b\}^*\}$

Solution: $L = \{\epsilon, aa, bb, abba, baab, \ldots\}$

$S \rightarrow aSa \mid bSb \mid \epsilon$

8) Construct a CFG for the language

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

Solution: $L = \{aba, aabaa, aaabaaa, \dots \}$

$S \rightarrow aAa$

$A \rightarrow aAa \mid b$

9) Construct a CFG for the language

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

Solution: $L = \{abc, aabbc, abcc, abccc, \dots \}$

$S \rightarrow AB$

$A \rightarrow aAb \mid ab$

$B \rightarrow cB \mid c$

10) Construct a CFG for the language

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

Solution:

$S \rightarrow aSb \mid aAb$

$A \rightarrow cA \mid c$

11) Construct a CFG for the language

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

Solution:-

$S \rightarrow AB$

$A \rightarrow aAb \mid ab$

$B \rightarrow cBd \mid cd$

12) Construct a CFG for the language

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

Solution:-

$S \rightarrow aSd \mid aAd$

$A \rightarrow bAc \mid bc$

13) Construct a CFG for the language

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

Solution:-

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

$S \rightarrow AB$

$A \rightarrow aAb \mid ab$

$B \rightarrow bAc \mid bc$

14) Construct a CFG for the language

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

Solution:-

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

$S \rightarrow aSc \mid aAc$

$A \rightarrow aAb \mid ab$

15) Construct a CFG for the language

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

Solution:-

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

$S \rightarrow aSc \mid aAc$

$A \rightarrow bAc \mid bc$

16) Construct a CFG for the language

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

Solution:-

$S \rightarrow BA \mid AC$

$A \rightarrow aAb \mid \epsilon$

$B \rightarrow aB \mid a$

$C \rightarrow bC \mid b$

7) construct a CFG for the language
$$L = \{a^n b^{n+1} \mid n \geq 0\}$$

Solution:

$S \rightarrow Ab$

$A \rightarrow aAb \mid \epsilon$

Derivation of strings:

Leftmost derivation:- A derivation is said to be leftmost if in each step the leftmost variable in the sentential form is replaced.

Rightmost derivation:- A derivation is said to be rightmost if in each step the rightmost variable in the sentential form is replaced.

Derivation tree (or) parse tree:-

Let $G = (V, T, P, S)$ is a CFG. Each production of G is represented with a tree satisfying the following conditions:

1) If $A \rightarrow x_1 x_2 x_3 \ldots x_n$ is a production in G, then A becomes the parent of nodes labeled $x_1, x_2, x_3 \ldots x_n$

2) The collection of children from left to right yields $x_1 x_2 x_3 \ldots x_n$

If $w \in L(G)$ then it is represented by a tree called derivation tree (or) parse tree satisfying the following conditions:

1) The root has label S (start symbol)
2) The all internal nodes are labeled with variables.
3) The leaves (or) terminal nodes are labeled with ϵ (or) terminal symbols.

1) Consider a CFG

$$S \rightarrow SS / aSb / bSa / \epsilon$$

Find the
1) Leftmost derivation
2) Rightmost derivation
3) Derivation tree for the string (w) = ababab

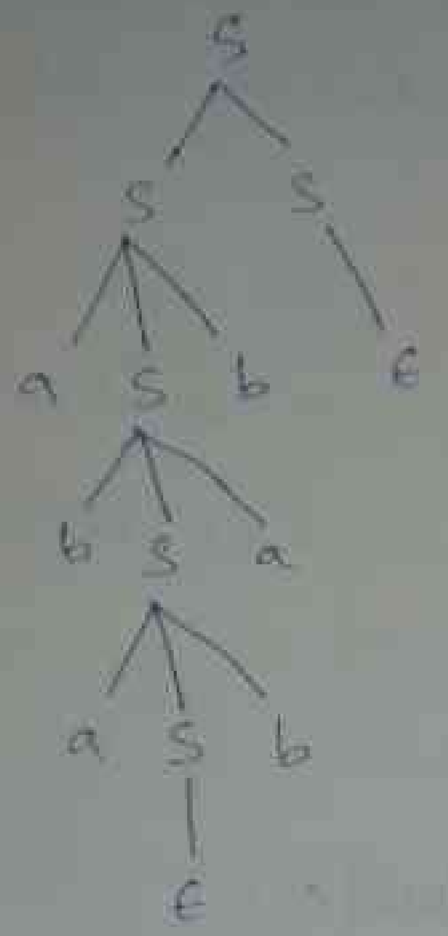
Solution: Given string (w) = ababab.

Leftmost derivation:

$$\left.\begin{aligned} S &\Rightarrow SS \\ &\Rightarrow aSbS \\ &\Rightarrow abSabS \\ &\Rightarrow abaSbabS \\ &\Rightarrow aba\epsilon babS \\ &\Rightarrow ababab S \\ &\Rightarrow ababab\epsilon \end{aligned}\right\} \text{Sentential forms}$$

$$\Rightarrow ababab$$

Derivation tree:



yield of a tree = ababab

Rightmost derivation: String(w) = ababab

$S \Rightarrow SS$

$\Rightarrow SaSb$

$\Rightarrow SabSab$

$\Rightarrow SabaSbab$

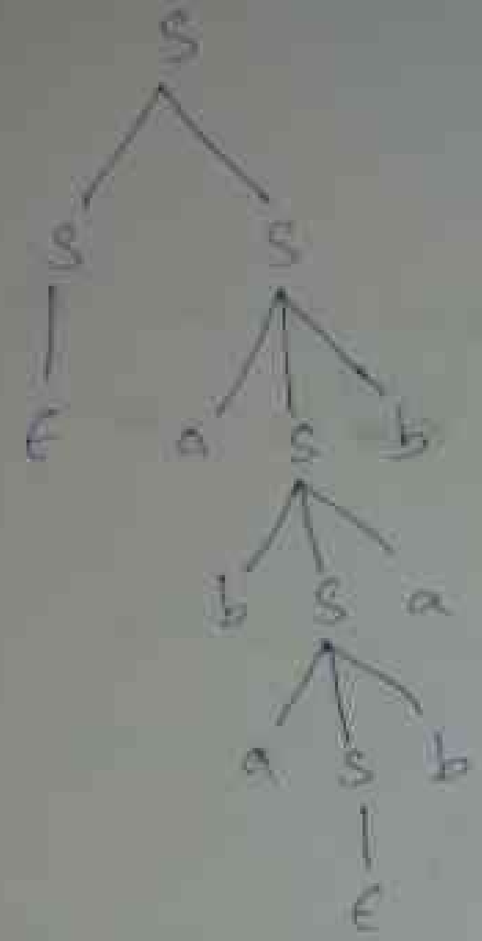
$\Rightarrow Saba\epsilon bab$

$\Rightarrow \epsilon ababab$

$\Rightarrow ababab$

Sentential forms

Derivation tree:



Sentential forms:- Any string of variables and/or terminals derived from the start symbol is called a sentential form.

$$S \Rightarrow \alpha$$

α is a ~~sentent~~ sentential form.

Ex:

$$E \Rightarrow E+T$$
$$\Rightarrow E+id$$
$$\Rightarrow T+id$$
$$\Rightarrow id+id$$

E+id, T+id are sentential forms.

2) Consider a CFG

$S \rightarrow bA \mid aB$

$A \rightarrow aS \mid aAA \mid a$

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

Find the leftmost and rightmost derivations for the string(w) = aaabbabbba. Construct the derivation tree also.

Solution:

Leftmost derivation: string(w) = aaabbabbba

$S \Rightarrow aB$

$\Rightarrow aaBB$

$\Rightarrow aaaBBB$

$\Rightarrow aaabBB$

$\Rightarrow aaabbB$

$\Rightarrow aaabbaBB$

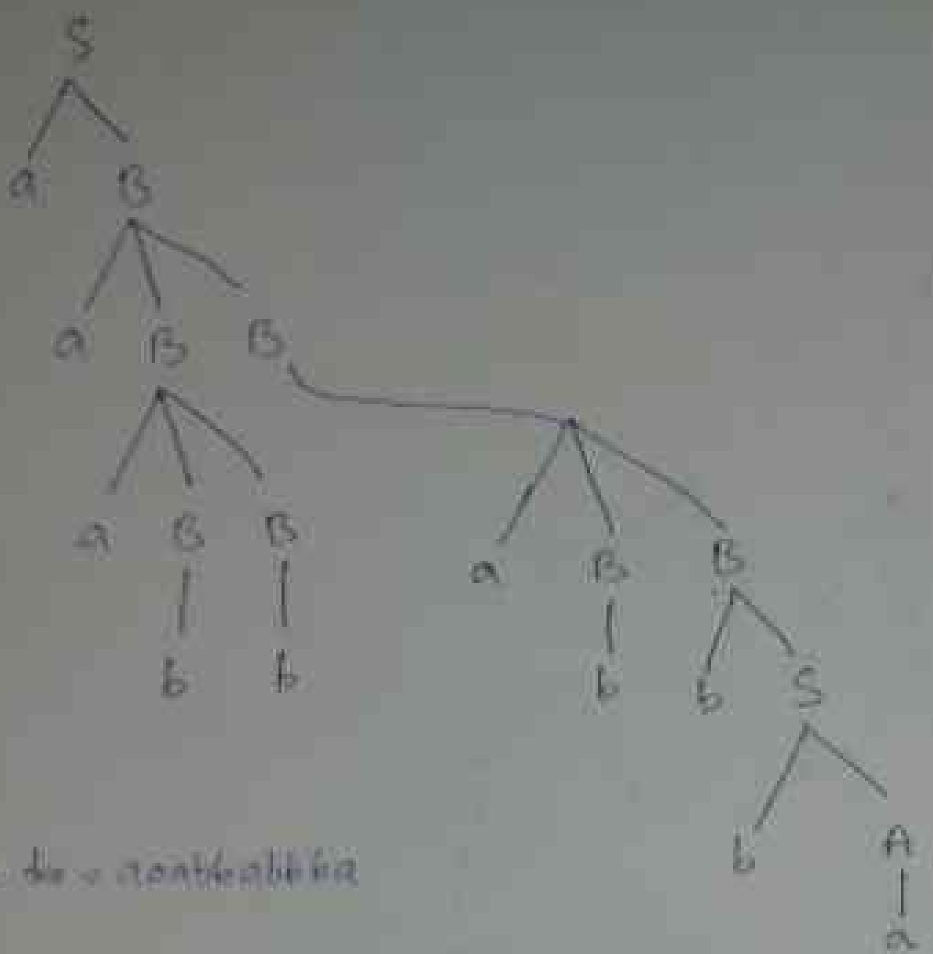
$\Rightarrow aaabbabB$

$\Rightarrow aaabbabbS$

$\Rightarrow aaabbabbbA$

$\Rightarrow aaabbabbba$

Derivation tree:



yield of a tree = aaabbabbba

Rightmost derivation:-

$S \Rightarrow aB$

$\Rightarrow aaBB$

$\Rightarrow aaBbS$

$\Rightarrow aaBbbA$

$\Rightarrow aaBbba$

$\Rightarrow aaaBBbba$

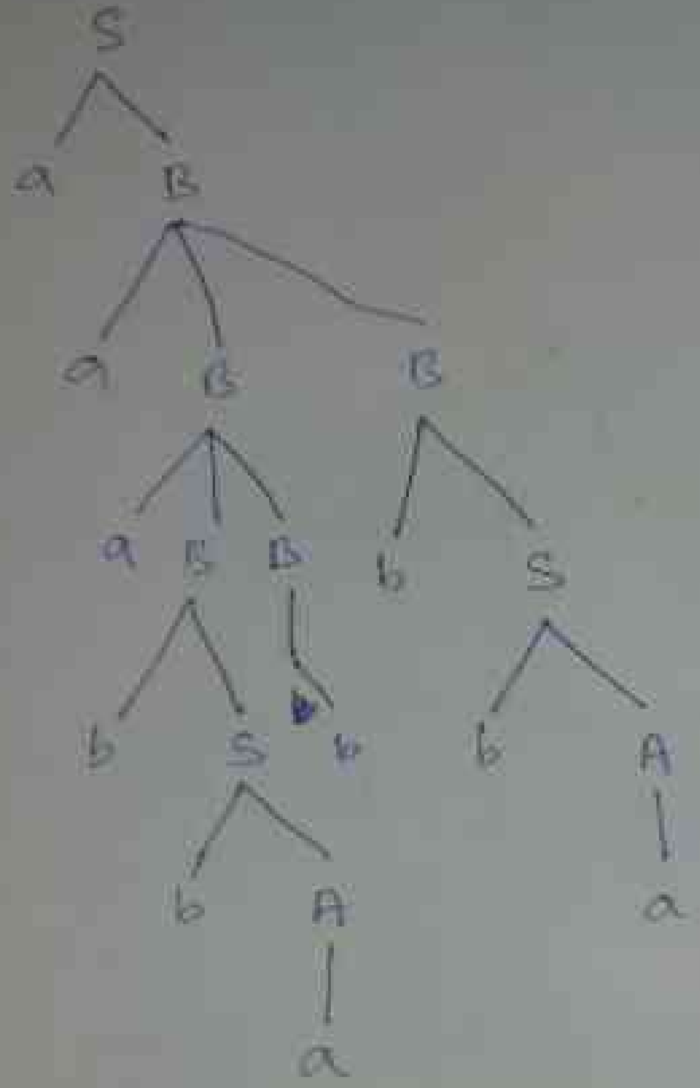
$\Rightarrow aaaBbbba$

$\Rightarrow aaabSbbba$

$\Rightarrow aaabbAbbba$

$\Rightarrow aaabbabbba$

Derivation tree:



3) Consider the CFG

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

Find the leftmost and rightmost derivation for string(w) = a*a+b.

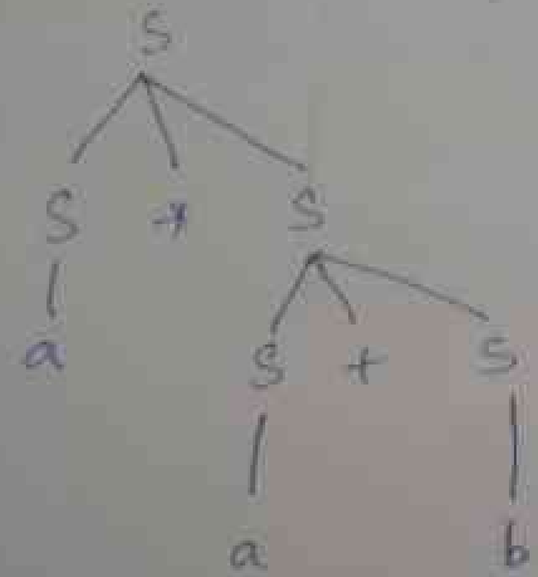
Also construct the derivation (or) parse tree.

<u>Solution:-</u>

<u>Leftmost derivation:-</u> string(w) = a*a+b.

$$\begin{aligned} S &\Rightarrow S*S \\ &\Rightarrow a*S \\ &\Rightarrow a*S+S \\ &\Rightarrow a*a+S \\ &\Rightarrow a*a+b \end{aligned}$$

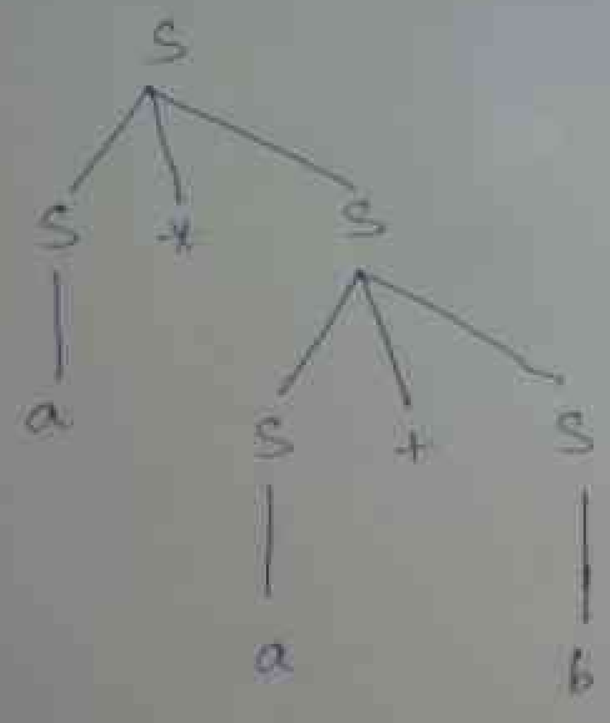
<u>parse tree:</u>



Rightmost derivation:- string(w) = a*a+b

$$S \Rightarrow S * S$$
$$\Rightarrow S * S + S$$
$$\Rightarrow S * S + b$$
$$\Rightarrow S * a + b$$
$$\Rightarrow a * a + b$$

parse tree:



4) Consider the CFG

$S \rightarrow aAS \mid a$

$A \rightarrow SbA \mid ba$

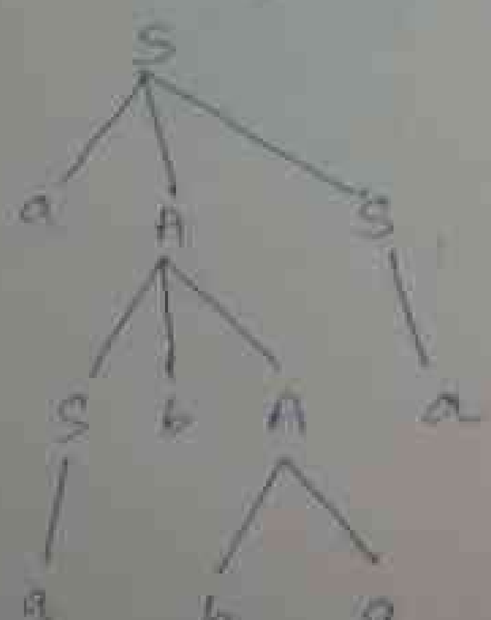
Find the leftmost derivation and rightmost derivation for the string(w) = aabbaa and also draw the derivation tree.

Solution:

Leftmost derivation: string(w) = aabbaa

$$\begin{aligned} S &\Rightarrow aAS \\ &\Rightarrow aSbAS \\ &\Rightarrow aabAS \\ &\Rightarrow aabbaS \\ &\Rightarrow aabbaa \end{aligned}$$

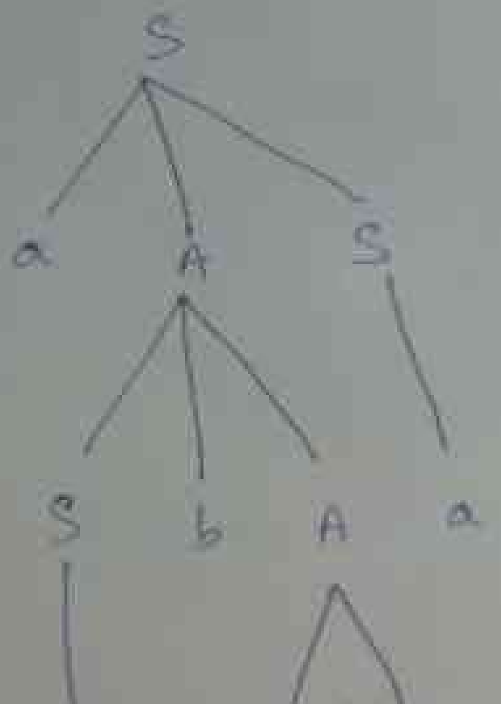
Derivation (or) parse tree:



Rightmost derivation: string(w) = aabbaa

$$S \Rightarrow aAS$$
$$\Rightarrow aAa$$
$$\Rightarrow aSbAa$$
$$\Rightarrow aSbbaa$$
$$\Rightarrow aabbaa$$

Parse tree:



5) Consider the CFG

$S \rightarrow 0B \mid 1A$

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

$B \rightarrow 1 \mid 1S \mid 0BB$

Find the leftmost and rightmost derivation for the string $w = 00110101$ and also draw the derivation tree.

Solution:-

Leftmost derivation:- string (w) = 00110101

$S \Rightarrow 0B$

$\Rightarrow 00BB$

$\Rightarrow 001B$

$\Rightarrow 0011S$

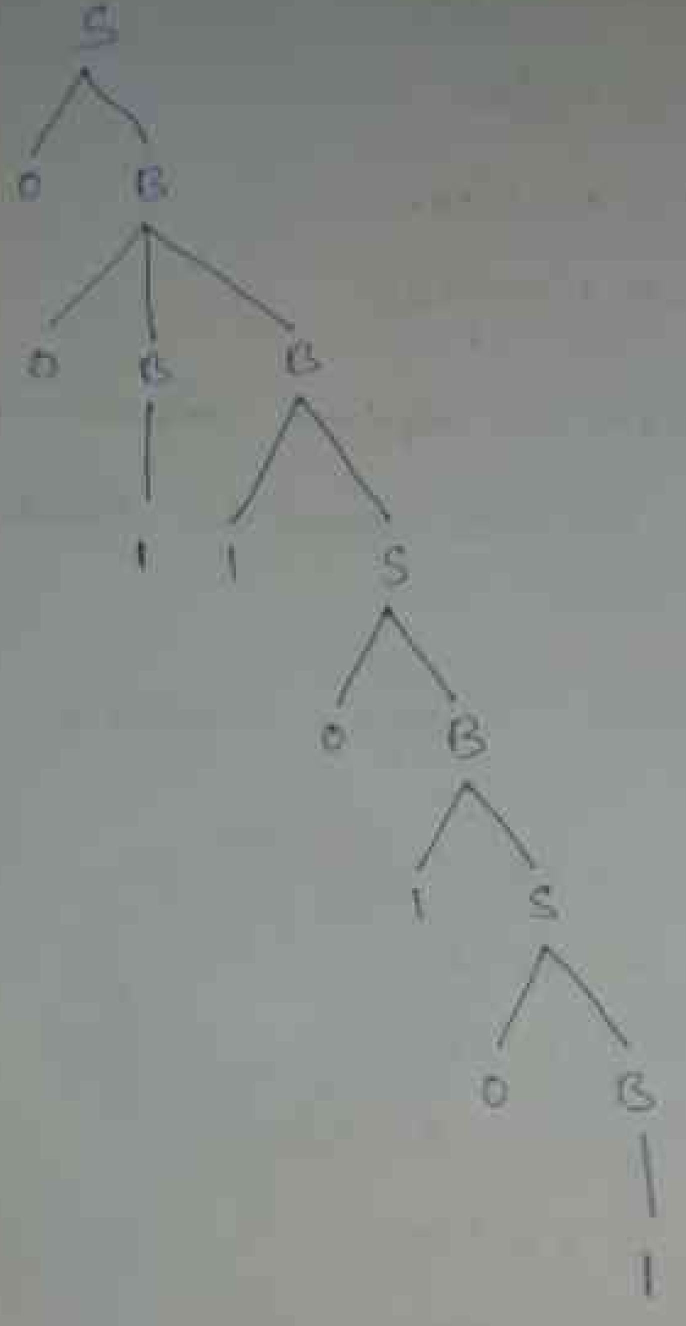
$\Rightarrow 00110B$

$\Rightarrow 001101S$

$\Rightarrow 0011010B$

$\Rightarrow 00110101$

parse tree:

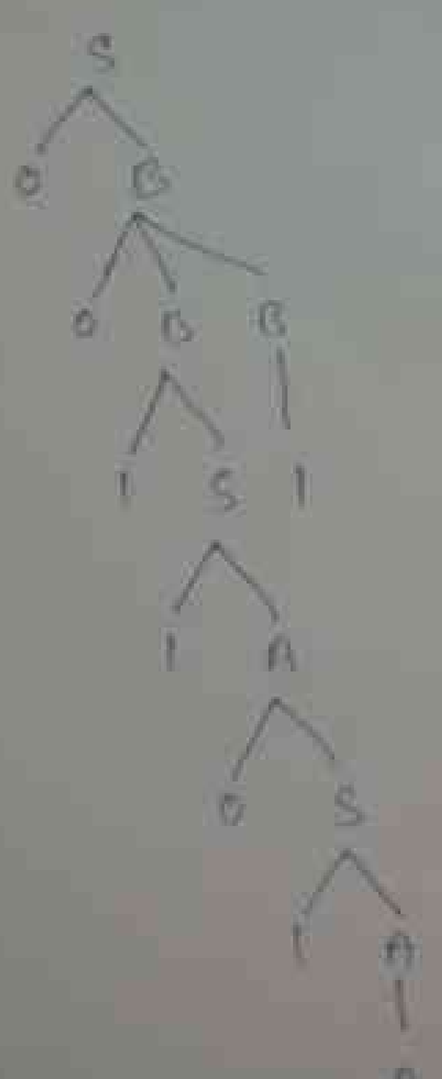


yield of a tree = 00110101

Rightmost derivation:-

$$
\begin{aligned}
S &\Rightarrow 0B \\
&\Rightarrow 00BB \\
&\Rightarrow 00B1 \\
&\Rightarrow 001S1 \\
&\Rightarrow 0011A1 \\
&\Rightarrow 00110S1 \\
&\Rightarrow 001101A1 \\
&\Rightarrow 00110101
\end{aligned}
$$

parse tree:-



6) Consider the CFG

$$S \rightarrow iCtS \mid iCtSeS \mid a$$

$$C \rightarrow b$$

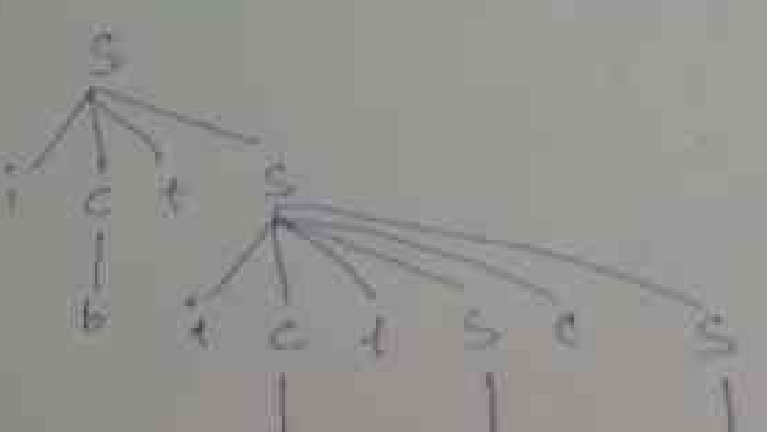
Find the leftmost and rightmost derivation for the string $w = ibtibtaea$ and also draw the derivation tree.

Solution:

Leftmost derivation:- String (w) = ibtibtaea

$$
\begin{aligned}
S &\Rightarrow iCtS \\
&\Rightarrow ibtS \\
&\Rightarrow ibtiCtSeS \\
&\Rightarrow ibtibtSeS \\
&\Rightarrow ibtibtaeS \\
&\Rightarrow ibtibtaea
\end{aligned}
$$

parse tree:

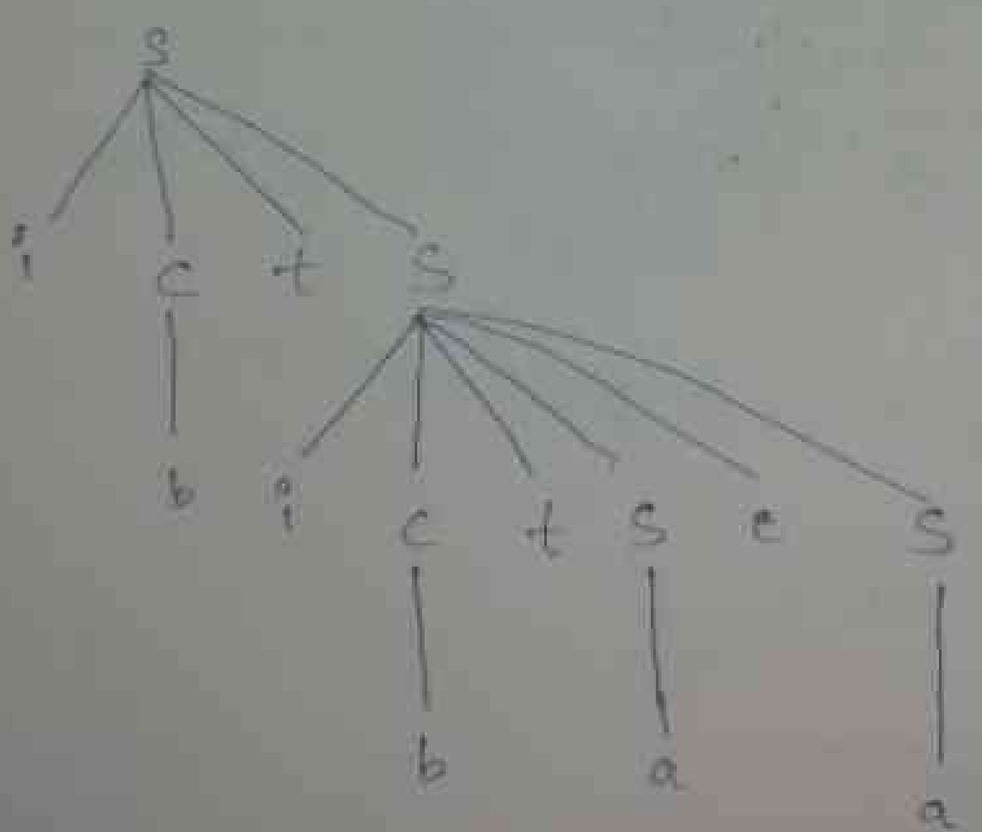


yield of a tree = ibtibtaea

Rightmost derivation:-

$$
\begin{aligned}
S &\Rightarrow ictS \\
&\Rightarrow ictictSeS \\
&\Rightarrow ictictSea \\
&\Rightarrow ictictaea \\
&\Rightarrow ictibtaea \\
&\Rightarrow ibtibtaea
\end{aligned}
$$

parse tree:



Ambiguity in CFG:- A grammar G is ambiguous if there exists some string $w \in L(G)$ for which there are two (or) more distinct leftmost derivations, or there are two (or) more distinct rightmost derivations, or there are two (or) more distinct derivation trees.

1) Show that the following grammar is ambiguous with respect to the string aaabbabbba.

$$S \rightarrow aB \mid bA$$
$$A \rightarrow aS \mid bAA \mid a$$
$$B \rightarrow bS \mid aBB \mid b$$

Solution:-

First rightmost derivation:- String(w) = aaabbabbba

$S \Rightarrow aB$
$\Rightarrow aaBB$
$\Rightarrow aaBbS$
$\Rightarrow aaBbbA$
$\Rightarrow aaBbba$
$\Rightarrow aaaBBbba$
$\Rightarrow aaaBbbba$
$\Rightarrow aaabSbbba$
$\Rightarrow aaabbAbbba$
$\Rightarrow aaabbabbba$

Second rightmost derivation: String(w) = aaabbabbba

$S \Rightarrow aB$

$\Rightarrow aaBB$

$\Rightarrow aaBaBB$

$\Rightarrow aaBaBbS$

$\Rightarrow aaBaBbbA$

$\Rightarrow aaBaBbba$

$\Rightarrow aaBabbba$

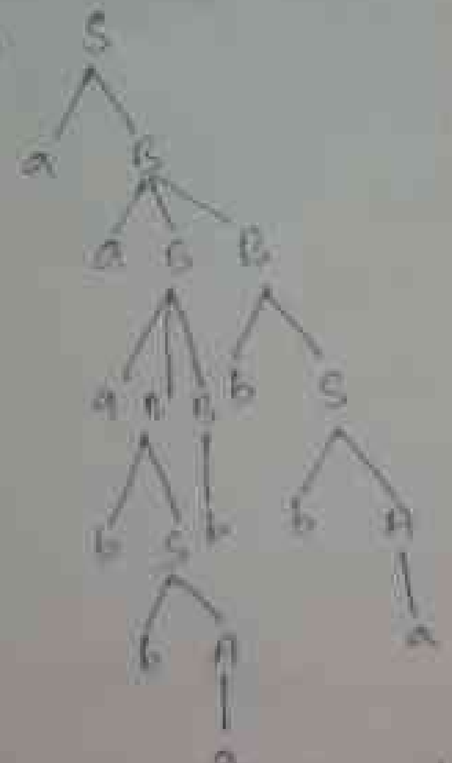
$\Rightarrow aaaBBabbba$

$\Rightarrow aaaBbabbba$

$\Rightarrow aaabbabbba$

For a given grammar (G) and given string(w) = aaabbabbba, we get two distinct rightmost derivations. So, given grammar (G) is ambiguous.

parse tree



parse tree

Since there are two distinct parse trees for string(w) = aaabbabbba, given grammar (G) is ambiguous.

2) Show that the following grammar is ambiguous with respect to the string ibtibtaea

$$S \rightarrow iCtS \mid iCtSeS \mid a$$

$$C \rightarrow b$$

<u>Solution:</u>

<u>First leftmost derivation:</u> string(w) = ibtibtaea

$S \Rightarrow iCtS$

$\Rightarrow ibtS$

$\Rightarrow ibtiCtSeS$

$\Rightarrow ibtibtSeS$

$\Rightarrow ibtibtaeS$

$\Rightarrow ibtibtaea$

<u>Second leftmost derivation:</u> string(w) = ibtibtaea

$S \Rightarrow iCtSeS$

$\Rightarrow ibtSeS$

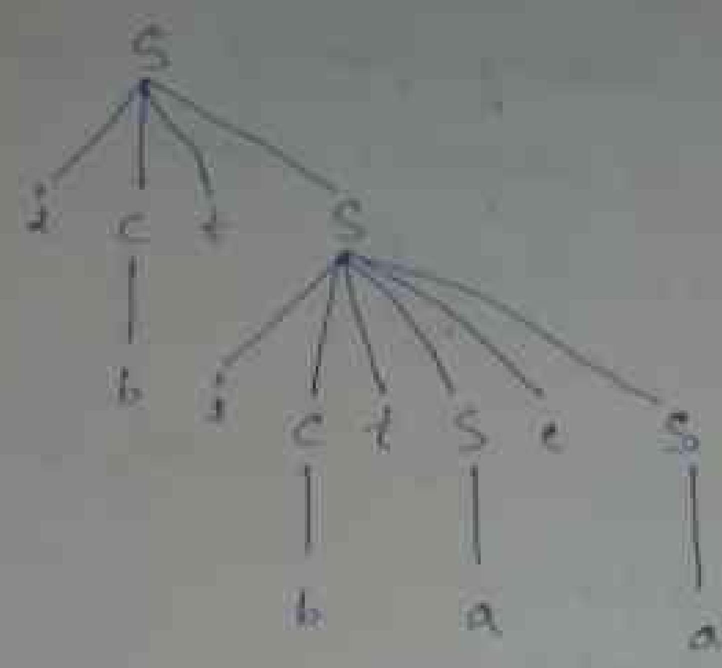
$\Rightarrow ibtiCtSeS$

$\Rightarrow ibtibtSeS$

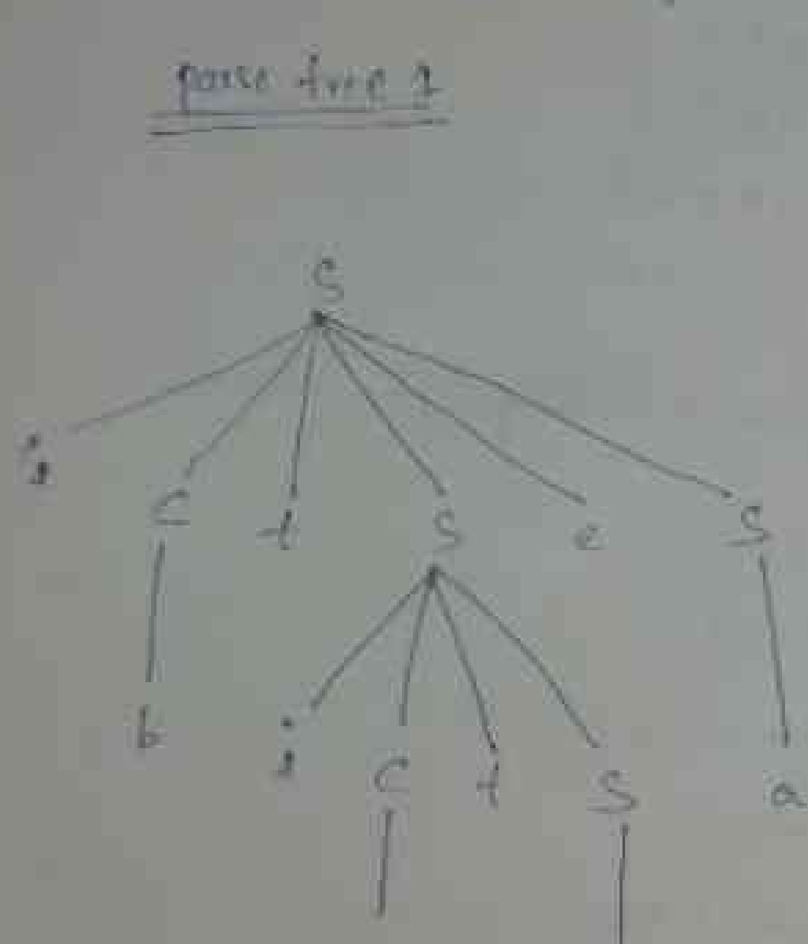
$\Rightarrow ibtibtaeS$

$\Rightarrow ibtibtaea$

Since, there are two distinct leftmost derivations for the string(w) = ibtibtaea, the given grammar is ambiguous.



parse tree 1



parse tree 2

Since, there are two distinct parse trees for the string $w = ibtibtaea$, the given grammar is ambiguous.

3) Show that the following grammar is ambiguous with respect to the string(w) = a*a+b.

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

<u>Solution:</u>

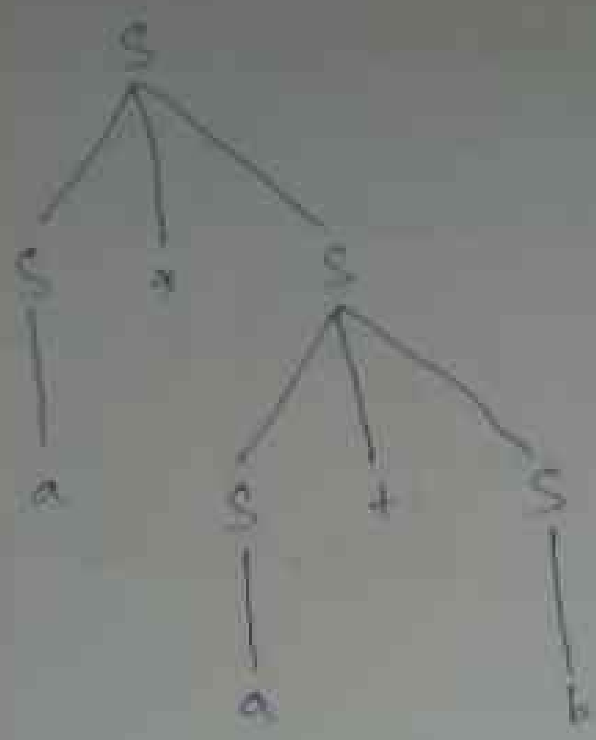
<u>First leftmost derivation:</u> string(w) = a*a+b

$$\begin{aligned} S &\Rightarrow S*S \\ &\Rightarrow a*S \\ &\Rightarrow a*S+S \\ &\Rightarrow a*a+S \\ &\Rightarrow a*a+b \end{aligned}$$

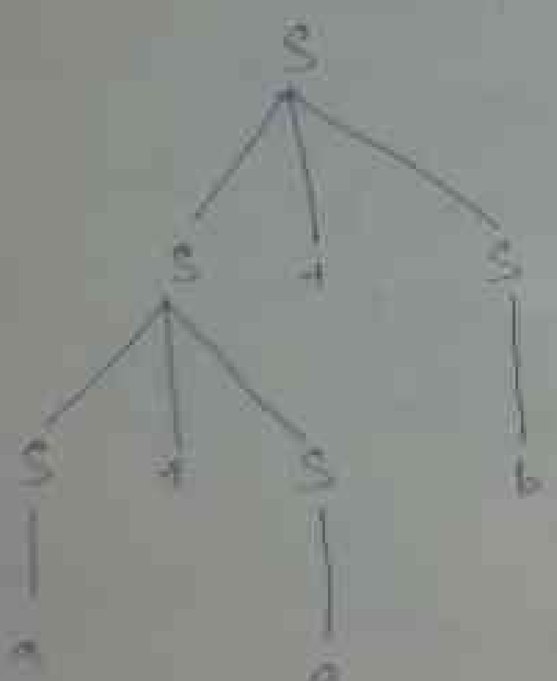
<u>Second leftmost derivation:</u> string(w) = a*a+b

$$\begin{aligned} S &\Rightarrow S+S \\ &\Rightarrow S*S+S \\ &\Rightarrow a*S+S \\ &\Rightarrow a*a+S \\ &\Rightarrow a*a+b \end{aligned}$$

Since there are two distinct leftmost derivations for the string(w) = a*a+b, the given grammar G is ambiguous.



parse tree 1



parse tree 2

Since there are two distinct parse trees for string(w) = a*a+b

the given grammar G is ambiguous.

Removal of Ambiguity:-

Left Recursion:- A grammar can be changed from one form to another accepting the same language. If a grammar has left recursive property, it is undesirable and left recursion should be eliminated. The left recursion is defined as follows:-

Definition: A grammar G is said to be left recursive if there is some non-terminal A such that $A \Rightarrow A\alpha$. In other words, in the derivation process starting from any non-terminal A, if a sentential form starts with the same non-terminal A, then we say that the grammar is having left recursion.

Elimination of left recursion:- The left recursion in a grammar G can be eliminated as shown below.

Consider the A-production of the form,

$$A \rightarrow A\alpha_1 | A\alpha_2 | A\alpha_3 \ldots A\alpha_n | \beta_1 | \beta_2 | \beta_3 \ldots \beta_m$$

Then the A productions can be replaced by

$$A \rightarrow \beta_1 A' | \beta_2 A' | \beta_3 A' | \ldots | \beta_m A'$$

$$A' \rightarrow \alpha_1 A' | \alpha_2 A' | \alpha_3 A' | \ldots | \alpha_n A' | \epsilon$$

1) Eliminate the left recursion from the following grammar

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

<u>Solution:</u> In the given grammar two productions are left recursive

$$S \to S+S$$

$$S \to S*S$$

1) $S \to S+S \mid a \mid b$

$A = S$

$\alpha = +S$

$\beta_1 = a$, $\beta_2 = b$

$$\boxed{\begin{array}{l} S \to aS' \mid bS' \\ S' \to +SS' \mid \epsilon \end{array}}$$

$$\boxed{\begin{array}{c} A \to A\alpha \mid \beta_1 \mid \beta_2 \\ \Downarrow \\ A \to \beta_1 A' \mid \beta_2 A' \\ A' \to \alpha A' \mid \epsilon \end{array}}$$

2) $S \to S*S \mid a \mid b$

$A = S$

$\alpha = *S$, $\beta_1 = a$, $\beta_2 = b$

$$\boxed{\begin{array}{l} S \to aS' \mid bS' \\ S' \to *SS' \mid \epsilon \end{array}}$$

After eliminating the left recursion, the final grammar is

$$\boxed{\begin{aligned} S &\rightarrow aS' \mid bS' \\ S' &\rightarrow +SS' \mid *SS' \mid \epsilon \end{aligned}}$$

Let us drive a string(w) = a*a+b.

<u>Leftmost derivation:</u> $S \Rightarrow aS'$

$\Rightarrow a*SS'$

$\Rightarrow a*aS'S'$

$\Rightarrow a*a+SS'S'$

$\Rightarrow a*a+bS'S'S'$

$\Rightarrow a*a+b\epsilon S'S'$

$\Rightarrow a*a+b\epsilon S'$

$\Rightarrow a*a+b\epsilon$

$\Rightarrow a*a+b$

we have only one leftmost derivation for the string = a*a+b.

So, removal of left recursion helps in removal of ambiguity.

2) Eliminate left recursion from the following grammar:

$S \rightarrow Ab \mid a$

$A \rightarrow Ab \mid Sa$

Solution: The given grammar can be written as

$S \rightarrow Ab \mid a$

$A \rightarrow Ab \mid Aba \mid aa$

Now A production has left recursion, Eliminate it.

$A \rightarrow Ab \mid Aba \mid aa$

$A = A$

$\alpha_1 = b,\ \alpha_2 = ba,\ \beta = aa$

$$\boxed{\begin{array}{l} A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \beta \\ \quad\Downarrow \\ A \rightarrow \beta A' \\ A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \epsilon \end{array}}$$

$$\boxed{\begin{array}{l} A \rightarrow aaA' \\ A' \rightarrow bA' \mid baA' \mid \epsilon \end{array}}$$

The grammar obtained after eliminating left recursion is

$$\boxed{\begin{array}{l} S \rightarrow Ab \mid a \\ A \rightarrow aaA' \\ A' \rightarrow bA' \mid baA' \mid \epsilon \end{array}}$$

3) Eliminate the left recursion from the following grammar.

$$E \rightarrow E+T \mid T$$
$$T \rightarrow T*F \mid F$$
$$F \rightarrow (E) \mid id$$

Solution: E and T productions have left recursion.

$$A \rightarrow A\alpha \mid \beta$$
$$\Downarrow$$
$$A \rightarrow \beta A'$$
$$A' \rightarrow \alpha A' \mid \epsilon$$

1) $E \rightarrow E+T / T$

$A = E$

$\alpha = +T,\ \beta = T$

$$\boxed{\begin{array}{l} E \rightarrow TE' \\ E' \rightarrow +TE' \mid \epsilon \end{array}}$$

2) $T \rightarrow T*F / F$

$A = T$

$\alpha = *F$

$\beta = F$

$$\boxed{\begin{array}{l} T \rightarrow FT' \\ T' \rightarrow *FT' / \epsilon \end{array}}$$

The grammar obtained after eliminating left recursion is

$$E \rightarrow TE'$$
$$E' \rightarrow +TE' \mid \epsilon$$
$$T \rightarrow FT'$$
$$T' \rightarrow *FT' \mid \epsilon$$
$$F \rightarrow (E) \mid id$$

4) Eliminate the left recursion from the following grammar.

$$S \rightarrow S0S1S \mid 01$$

<u>Solution:-</u>

$$S \rightarrow S0S1S \mid 01$$

$A = S$

$\alpha = 0S1S$

$\beta = 01$

$$\boxed{\begin{array}{c} A \rightarrow A\alpha \mid \beta \\ \Downarrow \\ A \rightarrow \beta A' \\ A' \rightarrow \alpha A' \mid \epsilon \end{array}}$$

$$\boxed{\begin{array}{l} S \rightarrow 01S' \\ S' \rightarrow 0S1SS' \mid \epsilon \end{array}}$$

$\downarrow$

The grammar obtained after eliminating left recursion.

Left factoring in CFG's:-

Definition:- Two (or) more productions of a variable A of the grammar $G = (V, T, P, S)$ are said to have left factoring if A-productions are of form

$$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \alpha\beta_3 \mid \cdots \mid \alpha\beta_n$$

All these A-productions have common left-factor α.

Elimination of left factoring:-

$$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \alpha\beta_3 \mid \cdots \mid \alpha\beta_n$$

$$\Downarrow$$

$$A \rightarrow \alpha A'$$

$$A' \rightarrow \beta_1 \mid \beta_2 \mid \beta_3 \mid \cdots \mid \beta_n$$

1) Eliminate the left factoring from the following grammar:

$$S \rightarrow ictS \mid ictSeS \mid a$$

$$C \rightarrow b$$

Solution:

$$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2$$

$$\Downarrow$$

$$A \rightarrow \alpha A'$$

$$A' \rightarrow \beta_1 \mid \beta_2$$

S production has ~~left~~ factoring.

$$S \rightarrow ictS \mid ictSeS$$

$A = S$

$\alpha = ictS$

$\beta_1 = \epsilon$

$\beta_2 = eS$

$$S \rightarrow ictSS'$$

$$S' \rightarrow \epsilon \mid eS$$

The grammar obtained after eliminating left factoring is

$$S \rightarrow ictSS' \mid a$$

$$S' \rightarrow \epsilon \mid eS$$

$$C \rightarrow b$$

2) Eliminate the left factoring from the following grammar:

$S \rightarrow aSa \mid aa$.

Solution:

$S \rightarrow aSa \mid aa$

$A = S$

$\alpha = a$

$\beta_1 = Sa$

$\beta_2 = a$

$$\boxed{\begin{aligned} & A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \\ & \quad\Downarrow \\ & A \rightarrow \alpha A' \\ & A' \rightarrow \beta_1 \mid \beta_2 \end{aligned}}$$

$$\boxed{\begin{aligned} & S \rightarrow aS' \\ & S' \rightarrow Sa \mid a \end{aligned}}$$

$\downarrow$

The grammar obtained after eliminating left factoring.

3) Eliminate the left factoring from the grammar:

$$S \rightarrow aSSbS \mid aSaSb \mid abb \mid b$$

Solution:

$$S \rightarrow aSSbS \mid aSaSb \mid abb$$

$$\boxed{\begin{array}{c} A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \alpha\beta_3 \\ \Downarrow \\ A \rightarrow \alpha A' \\ A' \rightarrow \beta_1 \mid \beta_2 \mid \beta_3 \end{array}}$$

$A = S$

$\alpha = a$

$\beta_1 = SSbS$

$\beta_2 = SaSb$

$\beta_3 = bb$

$$\boxed{\begin{array}{l} S \rightarrow aS' \\ S' \rightarrow SSbS \mid SaSb \mid bb \end{array}}$$

Again S' production has left factoring.

$$S' \rightarrow SSbS \mid SaSb$$

$$\boxed{\begin{array}{c} A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \\ \Downarrow \\ A \rightarrow \alpha A' \\ A' \rightarrow \beta_1 \mid \beta_2 \end{array}}$$

$A = S'$

$\alpha = S$

$\beta_1 = SbS$

$\beta_2 = aSb$

$$\boxed{\begin{array}{l} S' \rightarrow SS'' \\ S'' \rightarrow SbS \mid aSb \end{array}}$$

The grammar obtained after eliminating left factoring is

$$\boxed{\begin{array}{l} S \rightarrow aS' \mid b \\ S' \rightarrow SS'' \mid bb \\ S'' \rightarrow SbS \mid aSb \end{array}}$$

4) Eliminate the left factoring from the grammar:

$$S \rightarrow bSSaaS \mid bSSaSb \mid bSb \mid a$$

<u>Solution:</u>

$$S \rightarrow bSSaaS \mid bSSaSb \mid bSb$$

$A = S,\ \alpha = bS,\ \beta_1 = SaaS,\ \beta_2 = SaSb,\ \beta_3 = b$.

$$\boxed{\begin{array}{c} A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \alpha\beta_3 \\ \Downarrow \\ A \rightarrow \alpha A' \\ A' \rightarrow \beta_1 \mid \beta_2 \mid \beta_3 \end{array}}$$

$$\boxed{\begin{array}{l} S \rightarrow bSS' \\ S' \rightarrow SaaS \mid SaSb \mid b \end{array}}$$

Again S' has left factoring.

$$S' \rightarrow SaaS \mid SaSb$$

$A = S',\ \alpha = Sa,\ \beta_1 = aS,\ \beta_2 = Sb$

$$\boxed{\begin{array}{c} A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \\ \Downarrow \\ A \rightarrow \alpha A' \\ A' \rightarrow \beta_1 \mid \beta_2 \end{array}}$$

$$\boxed{\begin{array}{l} S' \rightarrow SaS'' \\ S'' \rightarrow aS \mid Sb \end{array}}$$

The <u>grammar obtained</u> after eliminating left factoring is

$$\boxed{\begin{array}{l} S \rightarrow bSS' \mid a \\ S' \rightarrow SaS'' \mid b \\ S'' \rightarrow aS \mid Sb \end{array}}$$

Simplification of CFG:-

In a CFG, it may happen that all the production rules and symbols are not needed for the derivation of strings. Besides, there may be epsilon productions and unit productions. Elimination of these productions and symbols is called simplification of CFG.

Simplification of CFG comprises the following steps:

1) Elimination of epsilon (or) null productions.
2) Elimination of unit productions
3) Elimination of useless symbols

By simplifying CFG's we remove all these redundant productions from a grammar, while keeping the transformed grammars equivalent to the original grammar. Two grammars are equivalent if they produce same language.

1) Elimination of Epsilon (or) Null productions:

Any production of a CFG of the form $A \to \epsilon$ is called epsilon production. Here the variable A is called Nullable variable.

Note: For a start symbol (S) if there is epsilon(ϵ) production, don't eliminate.

1) Eliminate ϵ-productions from the following grammar.

$$S \to aSb \mid aAb$$
$$A \to \epsilon$$

Solution: In the given grammar, $A \to \epsilon$ is ϵ-productions. A is nullable variable.

Now write ~~the~~ with A and without A in the remaining productions wherever A is present in the grammar.

$S \to aAb$ (with A)

$S \to ab$ (without A)

Resulting grammar,

$$S \to aSb \mid aAb \mid ab$$

In the resulting grammar, A is not reachable from start symbol (S). So, remove $S \to aAb$ productions.

The grammar obtained after eliminating ϵ-productions are

$$\boxed{S \rightarrow aSb \mid ab}$$

2) Eliminate all ϵ-productions from the grammar:

$S \rightarrow AB$

$A \rightarrow aAA \mid \epsilon$

$B \rightarrow bBB \mid \epsilon$

Solution:- In the given grammar,

$A \rightarrow \epsilon$

$B \rightarrow \epsilon$ are two ϵ-productions.

A and B are Nullable variables. So, write with it and without it.

$$\boxed{\begin{array}{l} S \rightarrow AB \mid A \mid B \mid \epsilon \\ A \rightarrow aAA \mid aA \mid a \\ B \rightarrow bBB \mid bB \mid b \end{array}}$$

The grammar after eliminating ϵ-productions.

3) Eliminate ϵ-productions from the grammar:

$S \rightarrow ABCa \mid bD$

$A \rightarrow BC \mid b$

$B \rightarrow b \mid \epsilon$

$C \rightarrow c \mid \epsilon$

$D \rightarrow d$

Solution: $B \rightarrow \epsilon$ and $C \rightarrow \epsilon$ are two ϵ productions.

$A \rightarrow BC$

$A \rightarrow \epsilon$

So, A, B, C are nullable variables. Now write with it and without it in the remaining productions.

$$\boxed{\begin{array}{l} S \rightarrow ABCa \mid BCa \mid ACa \mid ABa \mid Ca \mid Aa \mid Ba \mid a \mid bD \\ A \rightarrow BC \mid B \mid C \mid b \\ B \rightarrow b \\ C \rightarrow c \\ D \rightarrow d \end{array}}$$

$\downarrow$

Grammar after eliminating ϵ-productions.

4) Eliminate ϵ-productions from the following grammar:

$$S \rightarrow XYX$$
$$X \rightarrow 0X \mid \epsilon$$
$$Y \rightarrow 1Y \mid \epsilon$$

<u>Solution</u>: $\{X, Y\}$ are nullable variables.

$$\boxed{\begin{aligned} S &\rightarrow XYX \mid XY \mid YX \mid XX \mid X \mid Y \mid \epsilon \\ X &\rightarrow 0X \mid 0 \\ Y &\rightarrow 1Y \mid 1 \end{aligned}}$$

5) Eliminate ϵ-productions from the grammar:

$$S \rightarrow BAAB$$
$$A \rightarrow 0A2 \mid 2A0 \mid \epsilon$$
$$B \rightarrow AB \mid 1B \mid \epsilon$$

<u>Solution</u>: A, B are nullable variables.

$$\boxed{\begin{aligned} S &\rightarrow BAAB \mid AAB \mid BAB \mid BAA \mid AB \mid BB \mid BA \mid AA \mid A \mid B \mid \epsilon \\ A &\rightarrow 0A2 \mid 02 \mid 2A0 \mid 20 \\ B &\rightarrow AB \mid A \mid 1B \mid 1 \end{aligned}}$$

6) Eliminate ϵ-productions from the grammar.

$$S \rightarrow ABAC$$
$$A \rightarrow aA \mid \epsilon$$
$$B \rightarrow bB \mid \epsilon$$
$$C \rightarrow c$$

Solution:- A, B are nullable variables.

$$\boxed{\begin{aligned} &S \rightarrow ABAC \mid ABC \mid BAC \mid AAC \mid AC \mid BC \mid C \\ &A \rightarrow aA \mid a \\ &B \rightarrow bB \mid b \\ &C \rightarrow c \end{aligned}}$$

Note:- By eliminating the ϵ-productions from the grammar, unit productions are added.

Elimination of unit productions:-

Any production in CFG of the form

$A \rightarrow B$ is a unit production.

where $A, B \in V$ (variables)

1) Reduce the following grammar such that there are no unit productions.

~~Solution~~

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

Solution:

$S \rightarrow B$

$B \rightarrow A$

$A \rightarrow B$ are unit productions.

$A \rightarrow B \rightarrow bb \Rightarrow A \rightarrow bb$

$B \rightarrow A \rightarrow a \mid bc \Rightarrow B \rightarrow a \mid bc$

$S \rightarrow B \rightarrow A \rightarrow a \mid bc \Rightarrow S \rightarrow a \mid bc \mid bb$

$\rightarrow bb$

$$\boxed{\begin{array}{l} S \rightarrow Aa \mid a \mid bc \mid bb \\ B \rightarrow a \mid bc \mid bb \\ A \rightarrow a \mid bc \mid bb \end{array}}$$

B is not reachable from start symbol (S). So, delete all B productions.

$$\boxed{\begin{array}{l} S \rightarrow Aa \mid a \mid bc \mid bb \\ A \rightarrow a \mid bc \mid bb \end{array}}$$

2) Reduce the following grammar such that there are no unit productions.

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow C \mid b$

$C \rightarrow D$

$D \rightarrow E$

$E \rightarrow a$

Solution: $B \rightarrow C$, $C \rightarrow D$, $D \rightarrow E$ are three unit productions.

$D \rightarrow E \rightarrow a \Rightarrow \boxed{D \rightarrow a}$

$C \rightarrow D \rightarrow E \rightarrow a \Rightarrow \boxed{C \rightarrow a}$

$B \rightarrow C \rightarrow D \rightarrow E \rightarrow a \Rightarrow \boxed{B \rightarrow a}$

The grammar after eliminating unit productions is

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow a|b$

$C \rightarrow a$

$D \rightarrow a$

$E \rightarrow a$.

But the variables C, D, E are not reachable from start symbol. So, delete C, D, E productions.

$$\boxed{\begin{array}{l} S \rightarrow AB \\ A \rightarrow a \\ B \rightarrow a|b \end{array}}$$

3) Eliminate the unit productions from the given grammar:

$$S \rightarrow 0A \mid 1B \mid C$$
$$A \rightarrow 0S \mid 00$$
$$B \rightarrow 1 \mid A$$
$$C \rightarrow 01$$

Solution:- In the given grammar, there are two unit productions

$S \rightarrow C$ and $B \rightarrow A$

$S \rightarrow C \rightarrow 01 \Rightarrow \boxed{S \rightarrow 01}$

$B \rightarrow A \rightarrow 0S$, $\rightarrow 00$ $\Rightarrow \boxed{B \rightarrow 0S \mid 00}$

The grammar after eliminating unit productions are

$$\boxed{\begin{aligned} S &\rightarrow 0A \mid 1B \mid 01 \\ A &\rightarrow 0S \mid 00 \\ B &\rightarrow 1 \mid 0S \mid 00 \\ C &\rightarrow 01 \end{aligned}}$$

4) Reduce the following grammar such that there are no unit productions:

$$S \rightarrow AA$$
$$A \rightarrow B \mid BB$$
$$B \rightarrow abB \mid b \mid bb$$

<u>Solution:</u> In the given grammar,

$A \rightarrow B$ is a unit production.

$A \rightarrow B \rightarrow abB \mid b \mid bb \Rightarrow \boxed{A \rightarrow abB \mid b \mid bb}$

The grammar after eliminating unit productions is

~~S→AA~~

~~A→B|BB~~

$$\boxed{\begin{array}{l} S \rightarrow AA \\ A \rightarrow abB \mid b \mid bb \mid BB \\ B \rightarrow abB \mid b \mid bb \end{array}}$$

Elimination of useless symbols:-

1) Eliminate useless symbols in the grammar

$S \rightarrow AB \mid a$

$A \rightarrow BC \mid b$

$B \rightarrow aB$

$C \rightarrow aC$

Solution:-

Step-1: First find useful symbols. Always terminals are useful symbols.

useful symbols = $\{a, b\}$.

Step-2: If there is any productions with only terminals in right hand side, then add their left hand side variables to the useful symbols.

$S \rightarrow a$

$A \rightarrow b$.

useful symbols = $\{a, b, S, A\}$

Step-3: If there is any productions in which RHS of that production contains only useful symbols, then add their LHS variables to the useful symbols.

useful symbols = $\{a, b, S, A\}$.

Step 4:

So, {B, C} are useless symbols. So, delete all the productions with the variables {B, C}.

And also delete all the productions where {B, C} present in RHS of remaining productions.

$S \rightarrow a$

$A \rightarrow b$

But A is not reachable from start symbol S. So, delete A production.

$\boxed{S \rightarrow a}$

The grammar after eliminating useless symbols.

2) Eliminate the useless symbols in the grammar

$S \rightarrow aA \mid bB$

$A \rightarrow aA \mid a$

$B \rightarrow bB$

$D \rightarrow ab \mid Ea$

$E \rightarrow aC \mid d$

Solution: Step-1: useful symbols = {a, b, d}

Step-2: $A \to a$

$D \to ab$

$E \to d$

useful symbols = {a, b, d, A, D, E}

Step-3: $S \to aA$

useful symbols = {a, b, d, A, D, E, S}

Step-4: Variable B & C is useless symbols. So, delete all B & C productions and wherever B & C present in RHS of remaining productions, delete those productions also.

$S \to aA$

$A \to aA \mid a$

$D \to ab \mid Ea$

$E \to d$

But the variables D and E are not reachable from start symbol S.

So, delete D and E productions.

The grammar after eliminating useless symbols is

$$\boxed{\begin{array}{l} S \to aA \\ A \to aA \mid a \end{array}}$$

3) Eliminate the useless symbols in the grammar

$S \rightarrow aS | A | C$

$A \rightarrow a$

$B \rightarrow aa$

$C \rightarrow aCb$

Solution:

Step-1: useful symbols = $\{a, b\}$

Step-2: $A \rightarrow a$

$B \rightarrow aa$

useful symbols = $\{a, b, A, B\}$

Step-3: $S \rightarrow A$

useful symbols = $\{a, b, A, B, S\}$

Step-4: C is useless symbol. So, delete all C productions from the grammar.

$S \rightarrow aS | A$

$A \rightarrow a$

$B \rightarrow aa$

But B is not reachable from start symbol 'S'. So, delete B production

$$\boxed{\begin{array}{l} S \rightarrow aS | A \\ A \rightarrow a \end{array}}$$

4) Eliminate useless symbols from the grammar

$$S \rightarrow ABC \mid BaB$$
$$A \rightarrow aA \mid BaC \mid aaa$$
$$B \rightarrow bBb \mid a$$
$$C \rightarrow CA \mid AC$$

Solution:

Step-1: useful symbols = $\{a, b\}$

Step-2: $A \rightarrow aaa$

$B \rightarrow a$

useful symbols = $\{a, b, A, B\}$

Step-3: $S \rightarrow BaB$

useful symbols = $\{a, b, A, B, S\}$

Step-4: Variable C is useless symbol. So delete all C productions from the grammar

$$S \rightarrow BaB$$
$$A \rightarrow aA \mid aaa$$
$$B \rightarrow bBb \mid a$$

But variable A is not reachable from start symbol 'S' so, delete A productions.

$$\boxed{\begin{array}{l} S \rightarrow BaB \\ B \rightarrow bBb \mid a \end{array}}$$

5) Eliminate useless symbols from the grammar:

$S \rightarrow AB \mid AC$

$A \rightarrow aAb \mid bAa \mid a$

$B \rightarrow bbA \mid aaB \mid AB$

$C \rightarrow abCA \mid aDb$

$D \rightarrow bD \mid aC$

Solution:-

Step-1: useful symbols = $\{a, b\}$

Step-2: $A \rightarrow a$

useful symbols = $\{a, b, A\}$

Step-3: $B \rightarrow bbA$

useful symbols = $\{a, b, A, B\}$

$S \rightarrow AB$

useful symbols = $\{a, b, A, B, S\}$

Step-4: variables $\{C, D\}$ are useless symbols. So, delete all productions of C and D variables.

$$\boxed{\begin{array}{l} S \rightarrow AB \\ A \rightarrow aAb \mid bAa \mid a \\ B \rightarrow bbA \mid aaB \mid AB \end{array}}$$

Normal forms:

1) In a CFG, there is no restriction on the ~~other~~ right hand side of a production.

2) Normal form of a grammar impose restrictions on the right hand side of productions.

Two of the most useful normal forms are:

1) Chomsky Normal form (CNF)
2) Greibach Normal form (GNF)

Chomsky Normal form (CNF):-

A context free grammar (CFG) is said to be in CNF if all productions are of the form

$$\boxed{\begin{array}{c} A \to BC \\ \text{(or)} \\ A \to a \end{array}}$$

where $A, B, C \in V$ (Variables)

$a \in T$ (Terminal)

Example:- $S \to AS | a$

$A \to SA | b$

Steps to convert a given CFG to CNF:-

1) If the start symbol S occurs on some right side of the production, create a new start symbol S' and a new production $S' \to S$

2) Reduce the grammar such that there are no ε-productions, unit productions and useless symbols in the grammar.

3) Replace each production $A \to B_1 B_2 \cdots B_n$ where $n > 2$, with $A \to B_1 C$ where $C \to B_2 \cdots B_n$. Repeat this step for all productions having more than two variables on the right side.

4) If the right side of any production is in the form of $A \to aB$ where a is the terminal and A & B are variables, then production is replaced by $A \to XB$ and $X \to a$. Repeat this step for every production which is in the form of $A \to aB$.

1) Convert the following grammar to CNF

$S \rightarrow ASA \mid aB$

$A \rightarrow B \mid S$

$B \rightarrow b \mid \epsilon$

<u>Solution:-</u>

1) Since start symbol S appears in RHS, create a new start symbol S^1 and new production $S^1 \rightarrow S$.

$S^1 \rightarrow S$

$S \rightarrow ASA \mid aB$

$A \rightarrow B \mid S$

$B \rightarrow b \mid \epsilon$

2) Remove ϵ-productions $B \rightarrow \epsilon$ and $A \rightarrow \epsilon$.

After removing ϵ-productions,

$S^1 \rightarrow S$

$S \rightarrow ASA \mid AS \mid SA \mid aB \mid a$

$A \rightarrow B \mid S$

$B \rightarrow b$

3) Remove unit productions

$S' \to S$, $A \to B$ and $A \to S$

After removing unit productions,

$S' \to ASA | AS | SA | aB | a$

$S \to ASA | AS | SA | aB | a$

$A \to b | ASA | AS | SA | aB | a$

$B \to b$

4) The resulting grammar does not have any useless symbols.

5) Now find out the productions that has more than two variables in RHS.

$S' \to ASA$, $S \to ASA$, $A \to ASA$

After simplifying,

$S' \to AC | AS | SA | aB | a$

$S \to AC | AS | SA | aB | a$

$A \to b | AC | AS | SA | aB | a$

$B \to b$

$C \to SA$

6) Now find out the productions which are in the form of $A \to aB$.

$S' \to aB$, $S \to aB$ and $A \to aB$

The grammar which is in CNF is

$$\boxed{\begin{aligned} S' &\to AC \mid AS \mid SA \mid DB \mid a \\ S &\to AC \mid AS \mid SA \mid DB \mid a \\ A &\to b \mid AC \mid AS \mid SA \mid DB \mid a \\ B &\to b \\ C &\to SA \\ D &\to a \end{aligned}}$$

8) Convert the following grammar to CNF:

$S \rightarrow ABA$

$A \rightarrow aA | \epsilon$

$B \rightarrow bB | \epsilon$

Solution:

1) Eliminate ϵ-productions $A \rightarrow \epsilon$ and $B \rightarrow \epsilon$

After eliminating ϵ-productions,

$S \rightarrow ABA | AB | BA | B | AA | \epsilon$

$A \rightarrow aA | a$

$B \rightarrow bB | b$

2) Eliminate unit-production $S \rightarrow B$.

After Eliminating unit-production,

$S \rightarrow ABA | AB | BA | AA | bB | b | \epsilon$

$A \rightarrow aA | a$

$B \rightarrow bB | b$

3) Now find out the productions that has more than two variables in RHS.

$S \rightarrow ABA$

substitute $C \rightarrow AB$

After simplifying,

$S \rightarrow CA|AB|BA|AA|bB|b|\epsilon$

$A \rightarrow aA|a$

$B \rightarrow bB|b$

$C \rightarrow AB$

4) Now find out the productions which are in the form of $A \rightarrow aB$.

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

substitute $D \rightarrow a$; $E \rightarrow b$

The grammar which is in CNF is

$$\boxed{\begin{array}{l} S \rightarrow CA|AB|BA|AA|EB|b|\epsilon \\ A \rightarrow DA|a \\ B \rightarrow EB|b \\ C \rightarrow AB \\ D \rightarrow a \\ E \rightarrow b \end{array}}$$

3) Convert the following CFG to CNF:

$S \rightarrow aaaaS \mid aaaa$

Solution:-

1) Since start symbol S appears in RHS, create a new start symbol S' and new production $S' \rightarrow S$.

$S' \rightarrow S$

$S \rightarrow aaaaS \mid aaaa$

5) Remove unit production $S' \rightarrow S$

$S' \rightarrow aaaaS \mid aaaa$

$S \rightarrow aaaaS \mid aaaa$

substitute $A \rightarrow a$.

$S' \rightarrow AAAAS \mid AAAA$

$S \rightarrow AAAAS \mid AAAA$

$A \rightarrow a$

substitute $B \rightarrow AAAA$

$S' \rightarrow BS \mid AAAA$

$S \rightarrow BS \mid AAAA$

$B \rightarrow AAAA$

$A \rightarrow a$

Substitute $C \rightarrow AAA$

$S' \rightarrow BS \mid CA$

$S \rightarrow BS \mid CA$

$B \rightarrow CA$

$C \rightarrow AAA$

$A \rightarrow a$

Substitute $D \rightarrow AA$

$$\boxed{\begin{array}{l} S' \rightarrow BS \mid CA \\ S \rightarrow BS \mid CA \\ B \rightarrow CA \\ C \rightarrow DA \\ D \rightarrow AA \\ A \rightarrow a \end{array}}$$

The grammar is in CNF.

4) Convert the following CFG to CNF

$$S \rightarrow 0S0 \mid 1S1 \mid \epsilon$$

Solution:

1) Since start symbol S appears in RHS, create a new start symbol S' and new production $S' \rightarrow S$

$$S' \rightarrow S$$
$$S \rightarrow 0S0 \mid 1S1 \mid \epsilon$$

2) Eliminate ϵ-production $S \rightarrow \epsilon$

After eliminating ϵ-productions,

$$S' \rightarrow S \mid \epsilon$$
$$S \rightarrow 0S0 \mid 00 \mid 1S1 \mid 11$$

3) Eliminate unit production $S' \rightarrow S$

After eliminating unit-productions,

$$S' \rightarrow 0S0 \mid 00 \mid 1S1 \mid 11 \mid \epsilon$$
$$S \rightarrow 0S0 \mid 00 \mid 1S1 \mid 11$$

substitute $A \rightarrow 0$, $B \rightarrow 1$

$$\boxed{\begin{array}{l} S' \rightarrow ASA \mid AA \mid BSB \mid BB \mid \epsilon \\ S \rightarrow ASA \mid AA \mid BSB \mid BB \\ A \rightarrow 0 \\ A \rightarrow 1 \end{array}}$$

4) Now find out the productions that has more than two variables in RHS

substitute $C \to AS$ and $D \to BS$

The grammar which is in CNF is

$$\boxed{\begin{array}{l} S' \to CA \mid AA \mid DB \mid BB \mid \epsilon \\ S \to CA \mid AA \mid DB \mid BB \\ A \to 0 \\ B \to 1 \\ C \to AS \\ D \to BS \end{array}}$$

5) convert the following CFG to CNF.

$$S \rightarrow 0A \mid 1B$$

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

$$B \rightarrow 1BB \mid 0S \mid 0$$

Solution:

1) Since start symbol S appears in RHS, create a new start symbol S^1 and new production $S^1 \rightarrow S$

$$S^1 \rightarrow S$$

$$S \rightarrow 0A \mid 1B$$

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

$$B \rightarrow 1BB \mid 0S \mid 0$$

2) Eliminate the unit production $S^1 \rightarrow S$.

$$S^1 \rightarrow 0A \mid 1B$$

$$S \rightarrow 0A \mid 1B$$

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

$$B \rightarrow 1BB \mid 0S \mid 0$$

3) Now find out the productions that has more than two variables in RHS.

$A \rightarrow 0AA$ and $B \rightarrow 1BB$

substitute $X \rightarrow AA$ & $Y \rightarrow BB$

$S' \rightarrow 0A | 1B$
$S \rightarrow 0A | 1B$
$A \rightarrow 0X | 1S | 1$
$B \rightarrow 1Y | 0S | 0$
$X \rightarrow AA$
$Y \rightarrow BB$

4) Now find out the productions which are in the form of $A \rightarrow aB$.

substitute $C \rightarrow 0$ and $D \rightarrow 1$

$S' \rightarrow CA | DB$
$S \rightarrow CA | DB$
$A \rightarrow CX | DS | 1$
$B \rightarrow DY | CS | 0$
$X \rightarrow AA$
$Y \rightarrow BB$
$C \rightarrow 0$
$D \rightarrow 1$

Greibach Normal Form (GNF):-

A CFG is said to be in GNF if the productions are in the form of

$$A \rightarrow aB_1B_2 \ldots B_n$$
(or)
$$A \rightarrow a$$

where $A, B_1, B_2, \ldots B_n$ are variables and a is a terminal.

Ex: $A \rightarrow aBCD$
(or)
$A \rightarrow a$

Steps to convert a given CFG to GNF:-

1) check if the given CFG has any ε-productions, unit productions & useless symbols and Remove if there are any.
2) check whether the CFG is already in CNF and convert it to CNF if it is not.
3) change the names of the variables into some A_i in ascending order of i.
4) Alter the rules so that the variables are in ascending order such that, If the production is of the form $A_i \rightarrow A_jX$, then $i < j$.

5) Remove left recursion (if exists).

6) If any production in the grammar is not in GNF form, convert the production into GNF form.

1) Convert the following CFG to GNF

$S \rightarrow CA | BB$

$B \rightarrow b | SB$

$C \rightarrow b$

$A \rightarrow a$

Solution:

1) The given grammar is already in CNF.

2) change the names of the variables into some A_i in ascending order of i.

Replace S with A_1

C with A_2

A with A_3

B with A_4

we get,

$A_1 \rightarrow A_2A_3 | A_4A_4$

$A_4 \rightarrow b | A_1A_4$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

3) If the production is of the form $A_i \rightarrow A_jX$, then $i<j$.

$A_4 \rightarrow b | A_1A_4$

$A_4 \rightarrow b | A_2A_3A_4 | A_4A_4A_4$ (Replace $A_1 \rightarrow A_2A_3 | A_4A_4$).

$A_4 \rightarrow b | bA_3A_4 | A_4A_4A_4$ (Replace $A_2 \rightarrow b$)

4) Eliminate the left recursion.

$A = A_4$, $\beta_1 = b$, $\beta_2 = bA_3A_4$, $\alpha = A_4A_4$

$A_4 \rightarrow bZ | bA_3A_4Z$

$Z \rightarrow A_4A_4Z | \epsilon$

$$\boxed{\begin{array}{c} A \rightarrow A\alpha | \beta_1 | \beta_2 \\ \Downarrow \\ A \rightarrow \beta_1A' | \beta_2A' \\ A' \rightarrow \alpha A' | \epsilon \end{array}}$$

Eliminate ϵ-production $Z \rightarrow \epsilon$

$A_4 \rightarrow bZ | b | bA_3A_4Z | bA_3A_4$

$Z \rightarrow A_4A_4Z | A_4A_4$

Now the grammar is,

$A_1 \rightarrow A_2A_3 \mid A_4A_4$

$A_4 \rightarrow b \mid bA_3A_4 \mid bZ \mid bA_3A_4Z$

$Z \rightarrow A_4A_4 \mid A_4A_4Z$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

5) If any production in the grammar is not in GNF form, convert the production into GNF form.

$A_1 \rightarrow bA_3 \mid bA_4 \mid bA_3A_4A_4 \mid bZA_4 \mid bA_3A_4ZA_4$

$A_4 \rightarrow b \mid bA_3A_4 \mid bZ \mid bA_3A_4Z$

$Z \rightarrow bA_4 \mid bA_3A_4A_4 \mid bZA_4 \mid bA_3A_4ZA_4 \mid bA_4Z \mid bA_3A_4A_4Z \mid$
$bZA_4Z \mid bA_3A_4ZA_4Z$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

2) convert the following grammar to GNF

$A_1 \rightarrow A_2 A_3$

$A_2 \rightarrow A_3 A_1 | b$

$A_3 \rightarrow A_1 A_2 | a$

Solution:

1) The given grammar is already in CNF

2) If the production is of form $A_i \rightarrow A_j X$, then $i < j$

$A_3 \rightarrow A_1 A_2 | a$

$A_3 \rightarrow A_2 A_3 A_2 | a$ (Replace $A_1 \rightarrow A_2 A_3$)

$A_3 \rightarrow A_3 A_1 A_3 A_2 | b A_3 A_2 | a$ (Replace $A_2 \rightarrow A_3 A_1 | b$)

3) Eliminate the left recursion.

~~$A_3 \rightarrow b A_3 A_2 A_1 A_3 A_2$~~

$A_3 \rightarrow b A_3 A_2 X | a X$

$X \rightarrow A_1 A_3 A_2 X | \epsilon$

eliminate ϵ production $X \rightarrow \epsilon$

~~$X \rightarrow A_1 A_3 A_2 X$~~ $A_3 \rightarrow b A_3 A_2 X | b A_3 A_2 | a X | a$

$X \rightarrow A_1 A_3 A_2 X | A_1 A_3 A_2$

Now the grammar is,

$A_1 \rightarrow A_2A_3$

$A_2 \rightarrow A_3A_1 | b$

$A_3 \rightarrow bA_3A_2X | aX | bA_3A_2 | a$

$X \rightarrow A_1A_3A_2 | A_1A_3A_2X$

Substitute A_3 production in A_2 production.

$A_2 \rightarrow bA_3A_2XA_1 | aXA_1 | bA_3A_2A_1 | aA_1 | b$

Now substitute A_2 production in A_1 production.

$A_1 \rightarrow bA_3A_2XA_1A_3 | aXA_1A_3 | bA_3A_2A_1A_3 | aA_1A_3 | bA_3$

Now substitute A_1 production in X production.

The grammar is

$A_1 \rightarrow bA_3A_2XA_1A_3 | aXA_1A_3 | bA_3A_2A_1A_3 | aA_1A_3 | bA_3$

$A_2 \rightarrow bA_3A_2XA_1 | aXA_1 | bA_3A_2A_1 | aA_1 | b$

$A_3 \rightarrow bA_3A_2X | aX | bA_3A_2 | a$

$X \rightarrow bA_3A_2XA_1A_3A_3A_2 | aXA_1A_3A_3A_2 | bA_3A_2A_1A_3A_3A_2 | aA_1A_3A_3A_2 |$
$bA_3A_3A_2 | bA_3A_2XA_1A_3A_3A_2X | aXA_1A_3A_3A_2X | bA_3A_2A_1A_3A_3A_2X |$
$aA_1A_3A_3A_2X | bA_3A_3A_2X$

3) Convert the following grammar to Greibach normal form.

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

$$A \rightarrow aA \mid a$$

$$B \rightarrow bB \mid b$$

Solution:

1) Remove unit productions $S \rightarrow B$

$$S \rightarrow ABA \mid AB \mid BA \mid AA \mid bB \mid b$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow bB \mid b$$

2) Convert the grammar into CNF

$$S \rightarrow XA \mid AB \mid BA \mid AA \mid ZB \mid b$$

$$A \rightarrow YA \mid a$$

$$B \rightarrow ZB \mid b$$

$$X \rightarrow AB$$

$$Y \rightarrow a$$

$$Z \rightarrow b$$

3) change the names of variables into some A_i in ascending order of i;

Replace S with A_1

X with A_2

A with A_3

B with A_4

Z with A_5

Y with A_6

we get,

$A_1 \rightarrow A_2A_5 | A_5A_4 | A_4A_3 | A_3A_5 | A_5A_4 | b$

$A_3 \rightarrow A_3A_5 | a$

$A_4 \rightarrow A_5A_4 | b$

$A_2 \rightarrow A_3A_4$

$A_6 \rightarrow a$

$A_5 \rightarrow b$

4) If any production in the grammar is not in GNF form, convert the production into GNF form.

Substitute $A_6 \rightarrow a$ in A_3

Substitute $A_5 \rightarrow b$ in A_4 & A_1

$A_1 \rightarrow A_2A_3 \mid A_3A_4 \mid A_4A_3 \mid A_3A_3 \mid bA_4 \mid b$

$A_3 \rightarrow aA_3 \mid a$

$A_4 \rightarrow bA_4 \mid b$

$A_2 \rightarrow A_3A_4$

$A_6 \rightarrow a$

$A_5 \rightarrow b$

Substitute A_3 in A_2 & A_1 and A_4 in A_1

$A_1 \rightarrow A_2A_3 \mid aA_3A_4 \mid aA_4 \mid bA_4A_3 \mid bA_3 \mid aA_3A_3 \mid aA_3 \mid bA_4 \mid b$

$A_3 \rightarrow aA_3 \mid a$

$A_4 \rightarrow bA_4 \mid a$

$A_2 \rightarrow aA_3A_4 \mid aA_4$

$A_6 \rightarrow a$

$A_5 \rightarrow b$

Substitute A_2 in A_1

$A_1 \rightarrow aA_3A_4A_3 \mid aA_4A_3 \mid aA_3A_4 \mid aA_4 \mid bA_4A_3 \mid bA_3 \mid aA_3A_3 \mid aA_3 \mid bA_4 \mid b$

$A_3 \rightarrow aA_3 \mid a$

$A_4 \rightarrow bA_4 \mid a$

$A_2 \rightarrow aA_3A_4 \mid aA_4$

$A_6 \rightarrow a$

$A_5 \rightarrow b$

4) Convert the following grammar into GNF

$$S \rightarrow AA \mid a$$

$$A \rightarrow SS \mid b$$

Solution:-

1) The grammar is already in CNF.

2) change the names of the variables into some A_i in ascending order of i.

S with A_1

A with A_2.

we get,

$$A_1 \rightarrow A_2A_2 \mid a$$

$$A_2 \rightarrow A_1A_1 \mid b.$$

3) If the production is of the form $A_i \rightarrow A_jX$, then $i<j$

$$A_2 \rightarrow A_1A_1 \mid b.$$

$$A_2 \rightarrow A_2A_2A_1 \mid aA_1 \mid b \quad (\text{Replace } A_1 \rightarrow A_2A_2 \mid a)$$

4). Eliminate the left recursion.

$$A_2 \rightarrow aA_1B \mid bB$$

$$B \rightarrow A_2A_1B \mid \epsilon$$

Eliminate ε-production $B \to \epsilon$

$$A_2 \to aA_1B \mid aA_1 \mid bB \mid b$$

$$B \to A_2A_1B \mid A_2A_1$$

The grammar is,

$$A_1 \to A_2A_2 \mid a$$

$$A_2 \to aA_1B \mid aA_1 \mid bB \mid b$$

$$B \to A_2A_1B \mid A_2A_1$$

5) If any production in the grammar is not in GNF ~~form~~, convert it to GNF.

Substitute A_2 in A_1 & B.

$\cancel{A_2 \to aA_1BA_2}$

$$A_1 \to aA_1BA_2 \mid aA_1A_2 \mid bBA_2 \mid bA_2 \mid a$$

$$A_2 \to aA_1B \mid aA_1 \mid bB \mid b$$

$$B \to aA_1BA_1B \mid aA_1A_1B \mid bBA_1B \mid bA_1B \mid aA_1BA_1 \mid aA_1A_1 \mid bBA_1 \mid bA_1$$

Pumping lemma for context free languages:-

If L be a CFL and $w \in L$ such that $|w| \geq n$ for some positive integer n, the 'w' can be represented as $w = uvwxy$ such that

(i) $vx \neq \epsilon$

(ii) $|vwx| \leq n$

(iii) for all $i \geq 0$, the string $uv^i w x^i y$ is also in L.

Note:-

1) Every CFL satisfies pumping lemma property.

2) If a language L does not satisfy the pumping lemma property, then L is not CFL.

3) If a language L satisfies the pumping lemma property, then L need not be CFL.

*) prove that the language $L=\{a^n b^n c^n \mid n \geq 1\}$ is not context free language.

Solution

Let $n=3$

String(w) = aaabbbccc

Divide the string(w) into uvwxy.

case(i):

$$w = \underbrace{a}_{u}\underbrace{aa}_{v}\underbrace{bbb}_{w}\underbrace{cc}_{x}\underbrace{c}_{y}$$

Let $i=2$

$$uv^2wx^2y = \underbrace{a}_{u}\underbrace{aaaa}_{v^2}\underbrace{bbb}_{w}\underbrace{cccc}_{x^2}\underbrace{c}_{y}$$

$= aaaaabbbccccc$

$= a^5b^3c^5$

So, the string $a^5b^3c^5$ is not in the given language $L=\{a^n b^n c^n \mid n \geq 1\}$

So, the language $L=\{a^n b^n c^n \mid n \geq 1\}$ is not context free language.

case(ii):

$$w = \underbrace{aa}_{u}\underbrace{ab}_{v}\underbrace{bb}_{w}\underbrace{cc}_{x}\underbrace{c}_{y}$$

uv^2wx^2y = aaababbbccccc

So, the string aaababbbccccc is not in the language $L=\{a^n b^n c^n \mid n \geq 1\}$

So, the language $L=\{a^n b^n c^n \mid n \geq 1\}$ is not CFL

Case (iii):- $w = \underset{u\,v\,w\,x\,y}{aaabbbccc}$

uv^2wx^2y = aaaabbbbccc

$= a^4b^4c^3$

So, the string $a^4b^4c^3$ is not in the given language $L=\{a^nb^nc^n \mid n\geq 1\}$

So, the language $L=\{a^nb^nc^n \mid n\geq 1\}$ is not context free language.

2) Prove that the language $L=\{ww \mid w \text{ is in } (a+b)^*\}$ is not context free language.

Solution:- string = $\underbrace{abab}_{w}\underbrace{abab}_{w}$

Divide the string into uvwxy

$= \underset{u\,v\,w\,x\,y}{abababab}$

uv^2wx^2y = abbabaabab

So, the string abbabaabab is not in the language $L=\{ww \mid w \text{ is in } (a+b)^*\}$.

So, the language $L=\{ww \mid w \text{ is in } (a+b)^*\}$ is not context-free language.

3) prove that the language $L = \{a^n / n \text{ is prime}\}$ is not context free language.

Solution: Let $n = 7$

String(w) = aaaaaaa

Divide the string(w) into uvwxy.

$$w = \underset{u\ v\ w\ x\ y}{aaaaaaa}$$

uv^2wx^2y = aaaaaaaaaa

So, the string aaaaaaaaaa(10) is not in the language $L = \{a^n / n \text{ is prime}\}$

So, the language $L = \{a^n / n \text{ is prime}\}$ is not context free language.

closure properties of context free languages:-

CFL are closed under

1) union
2) concatenation
3) Kleene closure.

<u>Union</u>:- If L_1 and L_2 are two context free languages, then their $L_1 \cup L_2$ will also be context free.

<u>Example</u>: $L_1 = \{a^n b^n / n \geq 1\}$

CFG for L_1 is

$S_1 \to aAb / ab$

$L_2 = \{c^m d^m / m \geq 0\}$

CFG for L_2 is

$S_2 \to cBd / \epsilon$

Union of L_1 and L_2 = $L_1 \cup L_2$

$= \{a^n b^n\} \cup \{c^m d^m\}$

CFG for $L_1 \cup L_2$

$$\boxed{\begin{array}{l} S \to S_1 / S_2 \\ S_1 \to aAb / ab \\ S_2 \to cBd / \epsilon \end{array}}$$

8) <u>Concatenation</u>:- If L_1 and L_2 are two context free languages, then their concatenation $L_1 \cdot L_2$ will also be context free.

<u>Example</u>: $L_1 = \{a^n b^n / n \geq 1\}$

$L_2 = \{c^m d^m / m \geq 1\}$

$L = L_1 \cdot L_2$

$= \{a^n b^n c^m d^m / n \geq 1 \; \& \; m \geq 1\}$ is also context free.

CFG for L is

$S \rightarrow S_1 S_2$

$S_1 \rightarrow aS_1 b / ab$

$S_2 \rightarrow cS_2 d / cd$

3) <u>Kleene closure</u>:- If L is a context free language, then L^* is also a context free.

<u>Example</u>: $L = \{a^n b^n / n \geq 0\}$

CFG for L is $S \rightarrow aSb / \epsilon$

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

CFG for L^* is $S_1 \rightarrow S S_1 / \epsilon$