

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

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

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 (i.e.) a combination of variables and terminals. The possible length of right hand side of CFG ranges from 0 to ∞ ($0 \leq |\alpha| \leq \infty$)

Ex: 1) $S \rightarrow aSa | aSh | \epsilon$

2) $S \rightarrow aA | \epsilon$

$A \rightarrow 1S$

3) $S \rightarrow aCa$

$C \rightarrow aCa | b$

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

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

$$S \rightarrow aSb \mid ab$$

2) obtain CFL for the language
 $L = \{w \mid n_a(w) = n_b(w)\}$

Solution: $L = \{\epsilon, ab, ba, aabab, baba, abba, baab, \dots\}$

$$S \rightarrow SS \mid \epsilon$$

$$S \rightarrow aSb \mid bSa$$

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

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

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

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

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

$$S \rightarrow aSbb \mid abb$$

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

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

$$S \rightarrow aSbb \mid \epsilon$$

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

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

$$S \rightarrow aSb \mid \epsilon$$

7) Construct a CFM for the language

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

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

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

8) Construct a CFM 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 CFM for the language

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

Solution: $L = \{ abc, aabbb, aabccc, aabcccc, \dots \}$

$$S \rightarrow AB$$

$$A \rightarrow aAb \mid ab$$

~~$$B \rightarrow cB \mid c$$~~

$$B \rightarrow cB \mid c$$

10) Construct a CFM 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 CFM for the language

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

Solution:-

$$S \rightarrow AB$$

$$A \rightarrow aAb \mid ab$$

$$B \rightarrow cBd \mid cd$$

12) Construct a CFM 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 CFM for the language

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

Solution:-

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

$$S \rightarrow AB$$

$$A \rightarrow aAb \mid ab$$

$$B \rightarrow bAc \mid bc$$

14) Construct a CFL for the language

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

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 CFL for the language

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

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 CFL for the language

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

Solution:-

$$S \rightarrow BA \mid AC$$

$$A \rightarrow aAB \mid \epsilon$$

$$B \rightarrow aB \mid a$$

$$C \rightarrow bC \mid b$$

17) construct a CFG for the language

$$L = \{ a^n b^{n+1} \mid n \geq 0 \}$$

Solution:

$$S \rightarrow AB$$

$$A \rightarrow aA \mid \epsilon$$

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 condition:

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

2) The collection of children from left to right yields

$$x_1 x_2 x_3 \dots x_n$$

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.

Q) Consider a CFG

$$S \rightarrow SS / asb / bsa / \epsilon$$

Find the

- 1) Leftmost derivation
- 2) Rightmost derivation
- 3) Derivation tree for the string $w = ababab$.

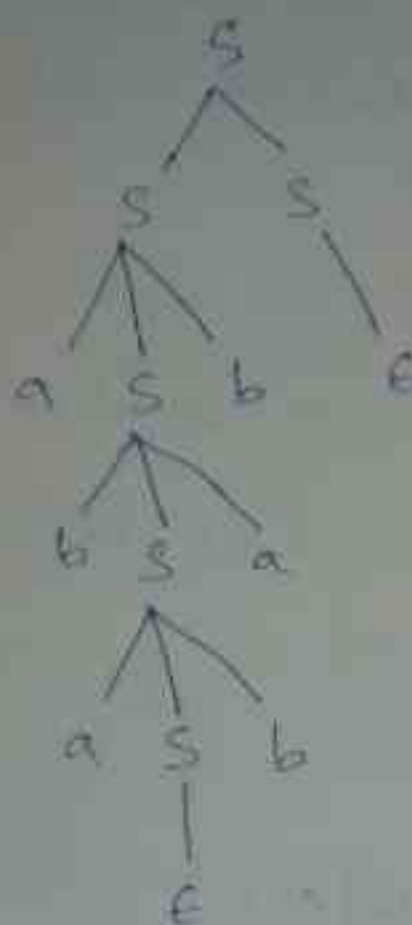
Solution:- Given string $w = ababab$.

Leftmost derivation:

$$\begin{aligned}
 S &\Rightarrow SS \\
 &\Rightarrow asbS \\
 &\Rightarrow abSasbS \\
 &\Rightarrow abasbabsS \\
 &\Rightarrow abaababsS \\
 &\Rightarrow abababS \\
 &\Rightarrow ababab\epsilon \\
 &\Rightarrow ababab
 \end{aligned}$$

Intermediate forms

Derivation tree



Yield of a tree = ababab

Rightmost derivation!

String(w) = ababab

$S \Rightarrow SS$

$\Rightarrow SaSb$

$\Rightarrow SabSab$

$\Rightarrow Sabasbab$

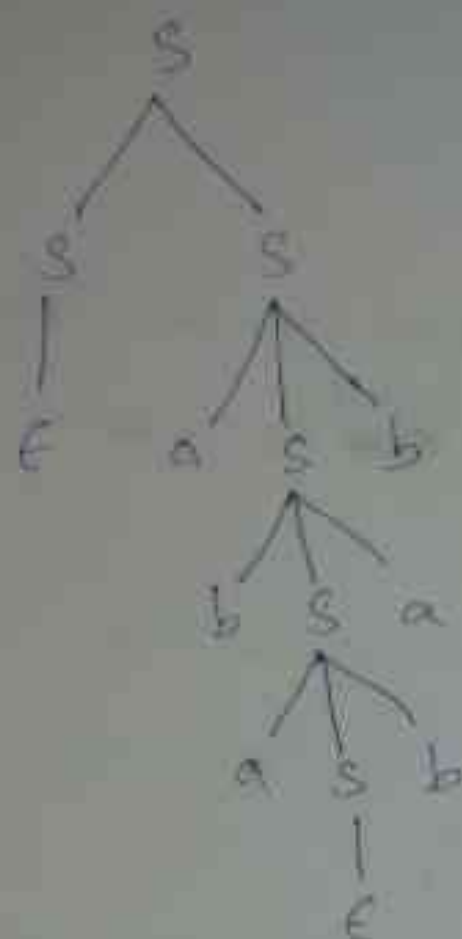
$\Rightarrow Sabaeabab$

$\Rightarrow \epsilon ababab$

$\Rightarrow ababab$

} 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 ~~valid~~ sentential form.

Ex:

$$E \Rightarrow E + T$$

$$\rightarrow E + id$$

$$\rightarrow T + id$$

$$\rightarrow id + id$$

① $E + id, T + id$ 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 $w = aabbabbba$. ~~Construct~~ the derivation tree also.

Solution:-

Leftmost derivation:- string $w = aabbabbba$

$$S \Rightarrow aB$$

$$\Rightarrow aaBB$$

$$\Rightarrow aaabBB$$

$$\Rightarrow aaabbbB$$

$$\Rightarrow aaabbbb$$

$$\Rightarrow aaabbbabbB$$

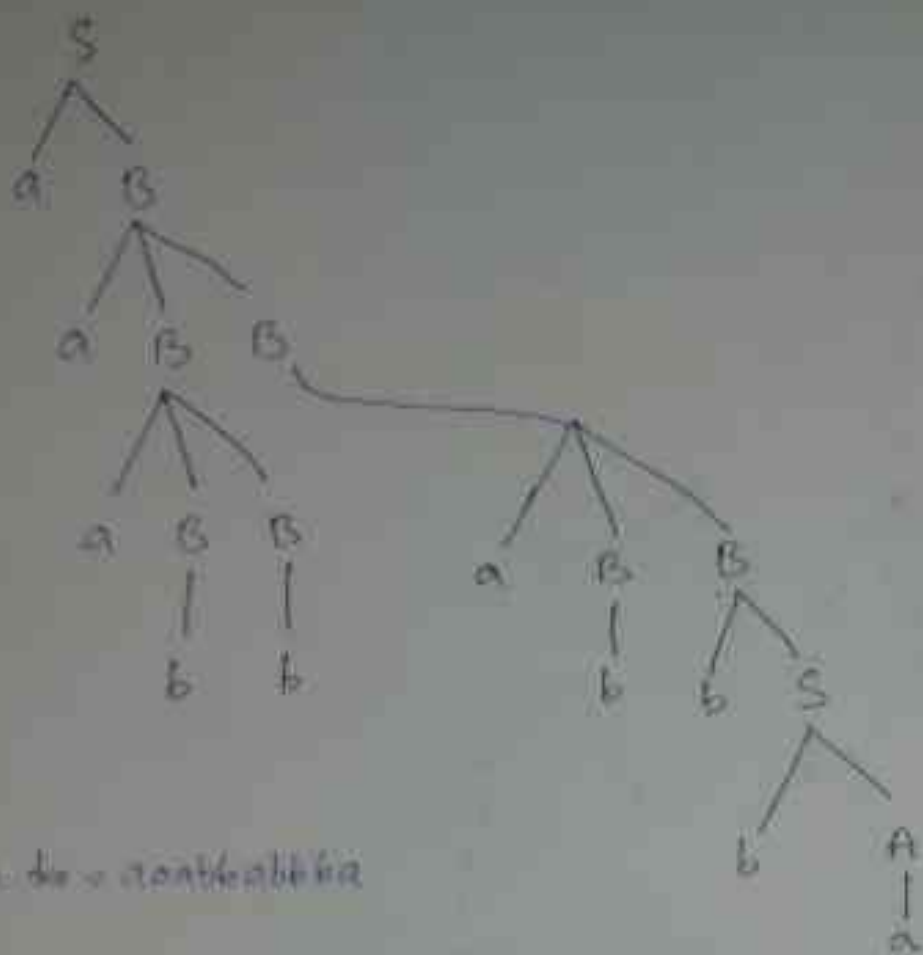
$$\Rightarrow aaabbbabbS$$

$$\Rightarrow aaabbbabbba$$

$$\Rightarrow aaabbbabbba$$

$$\Rightarrow aaabbbabbba$$

Derivation tree



yield of a tree = aabbbaabba

Rightmost derivation:-

$S \Rightarrow aB$
 $\Rightarrow aaB$
 $\Rightarrow aaBS$
 $\Rightarrow aaBbA$
 $\Rightarrow aaBbba$
 $\Rightarrow aaabBbba$
 $\Rightarrow aaabSbba$
 $\Rightarrow aaabAbbba$
 $\Rightarrow aaabbaabba$

Derivation - 1

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.

Solution:-

Leftmost derivation:- string $w = a * a + b$.

$$S \Rightarrow S * S$$

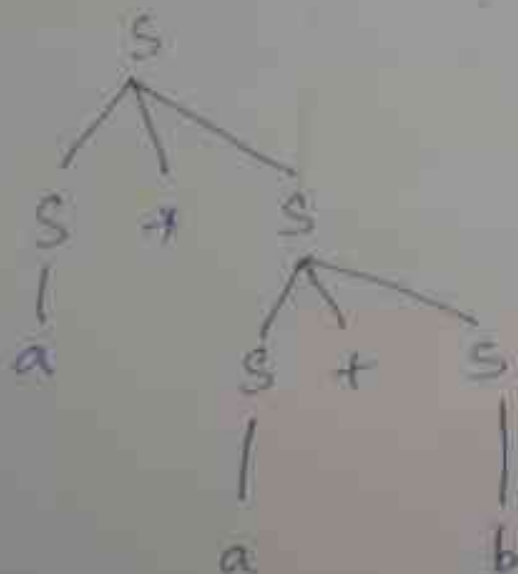
$$\Rightarrow a * S$$

$$\Rightarrow a * S + S$$

$$\Rightarrow a * a + S$$

$$\Rightarrow a * a + b$$

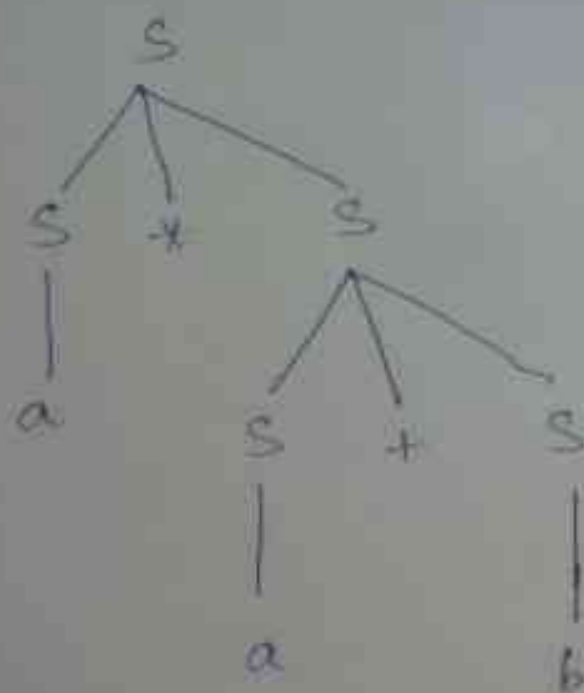
parse tree:-



Rightmost derivation: $\text{string}(w) = a * a + b$

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

parse tree:



4) Consider the CFG

$$S \rightarrow aAS \mid a$$

$$A \rightarrow SbA \mid ba$$

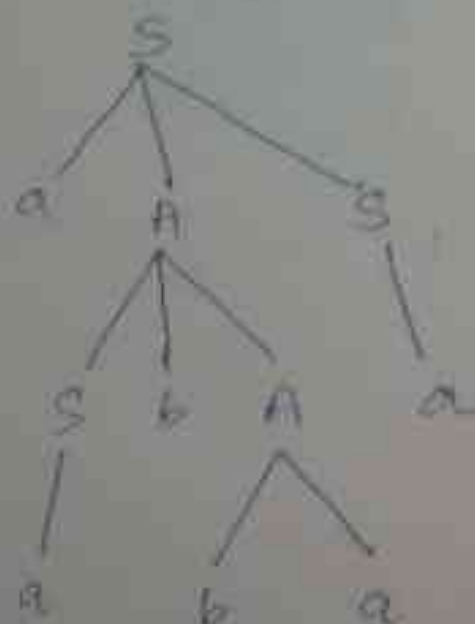
Find the leftmost derivation and rightmost derivation for the string $w = aabbba$ and also draw the derivation tree.

Solution:

Leftmost derivation: $\text{string}(w) = aabbba$

$$\begin{aligned} S &\Rightarrow aAS \\ &\Rightarrow aSbAS \\ &\Rightarrow aabAS \\ &\Rightarrow aabbAS \\ &\Rightarrow aabbba \end{aligned}$$

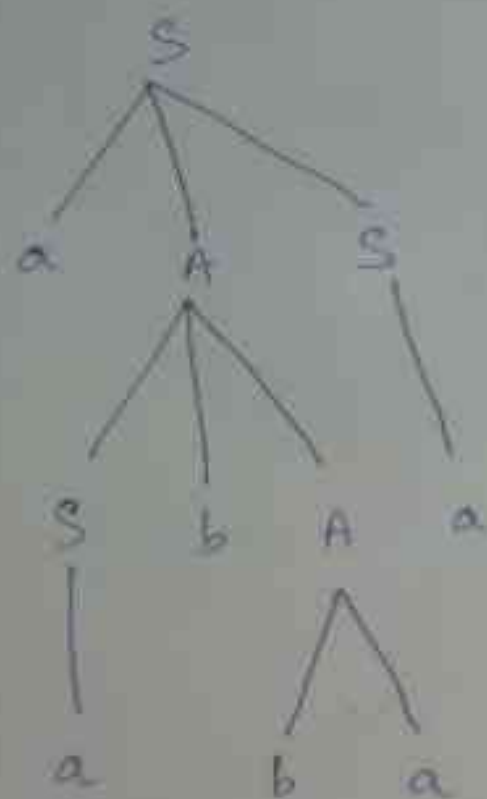
Derivation (or) parse tree:



Rightmost derivation: string = aabbba

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

parse tree:



5) Consider the CFG

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

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

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

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

Solution:-

Leftmost derivation:- $\text{string}(w) = 00110101$

$$S \Rightarrow 0B$$

$$\Rightarrow 00BB$$

$$\Rightarrow 001B$$

$$\Rightarrow 0011S$$

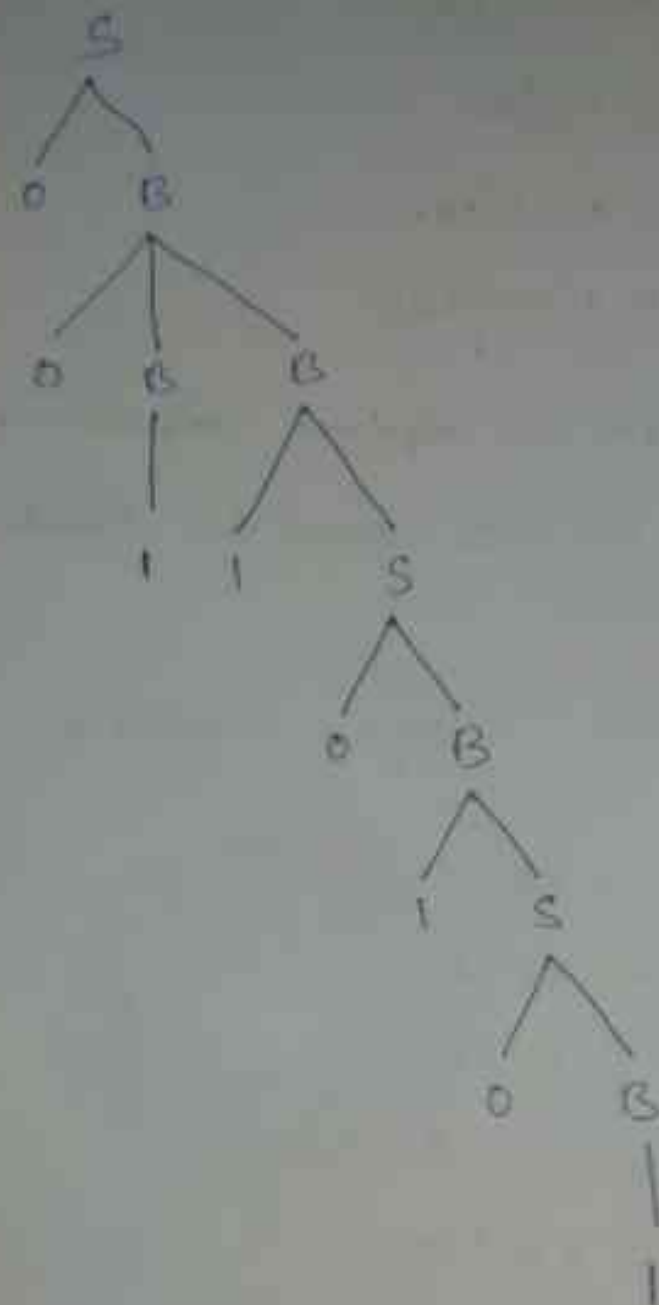
$$\Rightarrow 00110B$$

$$\Rightarrow 001101S$$

$$\Rightarrow 0011010B$$

$$\Rightarrow 00110101$$

parse tree



yield of a tree = 00110101

Rightmost derivation:-

$$S \Rightarrow 0B$$

$$\Rightarrow 00BB$$

$$\Rightarrow 00B1$$

$$\Rightarrow 001S1$$

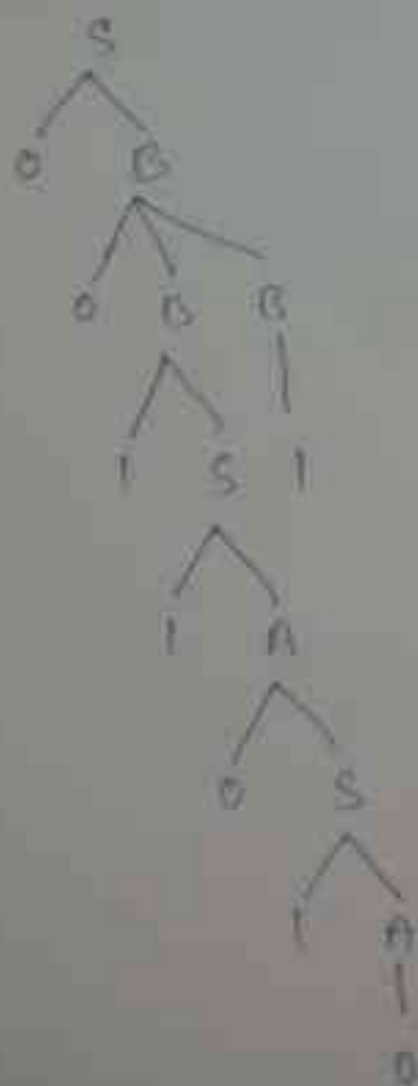
$$\Rightarrow 0011A1$$

$$\Rightarrow 00110S1$$

$$\Rightarrow 001101A1$$

$$\Rightarrow 00110101$$

parse tree:-



6) Consider the CFG

$$S \rightarrow ictS / ictSeS / a$$

$$C \rightarrow b$$

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

Solution:

Leftmost derivation: String $(w) = ibtibtaca$

$$S \Rightarrow ictS$$

$$\Rightarrow ibtS$$

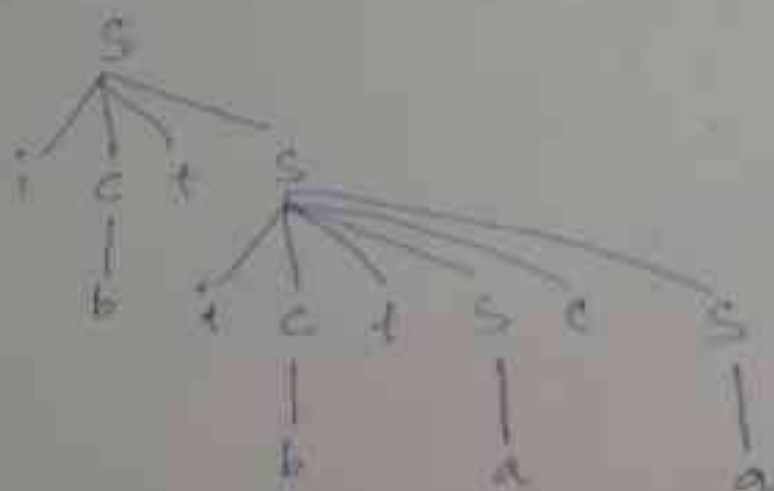
$$\Rightarrow ibtictSeS$$

$$\Rightarrow ibtibtSsS$$

$$\Rightarrow ibtibtacS$$

$$\Rightarrow ibtibtaca$$

parse tree:



yield of a tree = ibtibtaca

Rightmost derivation:

$$S \Rightarrow i c t S$$

$$\Rightarrow i c t i c t S e S$$

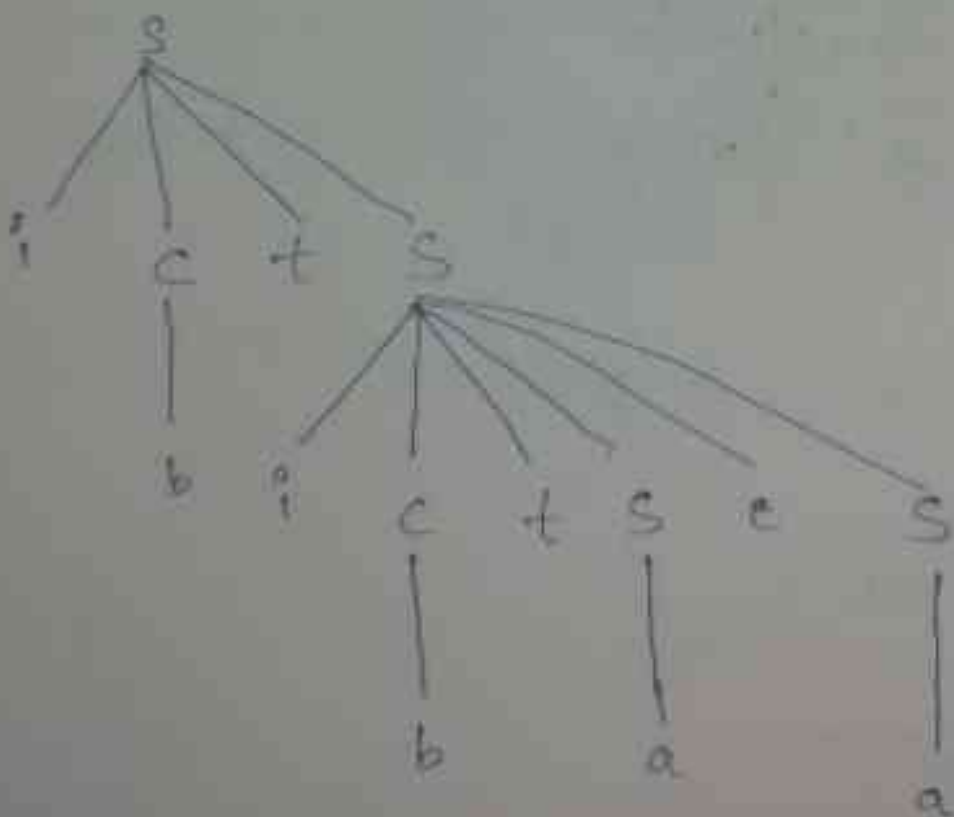
$$\Rightarrow i c t i c t S e a$$

$$\Rightarrow i c t i c t a e a$$

$$\Rightarrow i c t i b t a e a$$

$$\Rightarrow i b t i b t a e a$$

parse tree:



Ambiguity in CFL:- A grammar 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 $aaabbabbba$.

$$S \rightarrow aB | bA$$

$$A \rightarrow aS | bAA | a$$

$$B \rightarrow bS | aBB | b$$

Solution:-

First rightmost derivation: $\text{String}(w) = aaabbabbba$

$$S \Rightarrow aB$$

$$\Rightarrow aaBB$$

$$\Rightarrow aaBbS$$

$$\Rightarrow aaBbbA$$

$$\Rightarrow aaBbbba$$

$$\Rightarrow aaBBbbba$$

$$\Rightarrow aaabbbba$$

$$\Rightarrow aaabSbbba$$

$$\Rightarrow aaabbbaabbba$$

$$\Rightarrow aaabbbaabbba$$

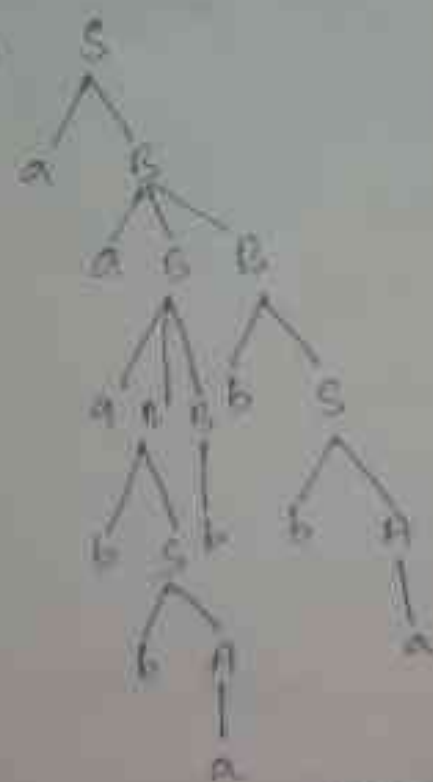
Second rightmost derivation:- $\text{String}(w) = aaababbbba$

$S \Rightarrow aB$
 $\Rightarrow aaBB$
 $\Rightarrow aaBabB$
 $\Rightarrow aaBabbbS$
 $\Rightarrow aaBabbbbaA$
 $\Rightarrow aaBabbbba$
 $\Rightarrow aaBabbbbaa$
 $\Rightarrow aaabBabbbba$
 $\Rightarrow aaabbbabbbba$
 $\Rightarrow aaabbbabbbba$

For a given grammar (G) and given $\text{String}(w) = aaababbbba$, we get two distinct rightmost derivations. So, given grammar (G) is ambiguous.



parse tree



parse tree

Since there are two distinct parse trees for $\text{String}(w) = aaababbbba$, given grammar (G) is ambiguous.

2) Show that the following grammar is ambiguous with respect to the string $ibtibktara$.

$$S \rightarrow ictS | ictSeS | a$$

$$c \rightarrow b$$

Solution:

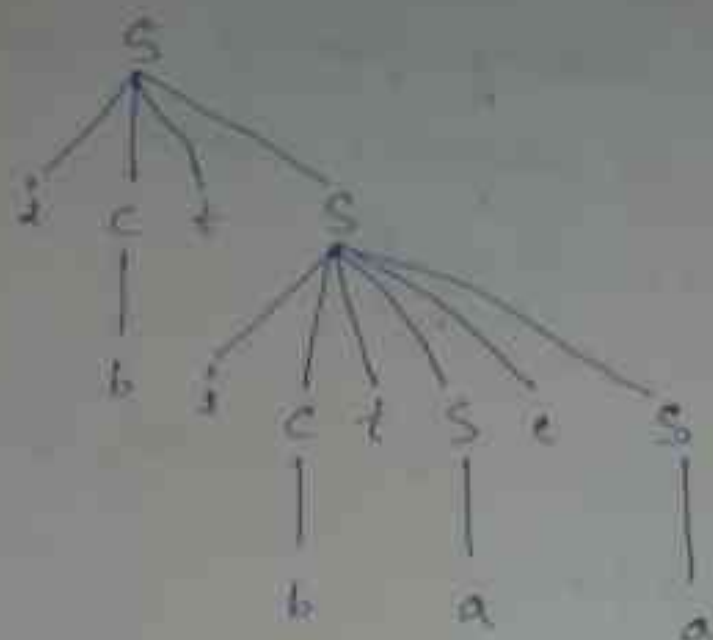
First leftmost derivation: $\text{string}(w) = ibtibktara$

$$\begin{aligned} S &\Rightarrow ictS \\ &\Rightarrow ibtS \\ &\Rightarrow ibtictSeS \\ &\Rightarrow ibtibktSeS \\ &\Rightarrow ibtibktarS \\ &\Rightarrow ibtibktara \end{aligned}$$

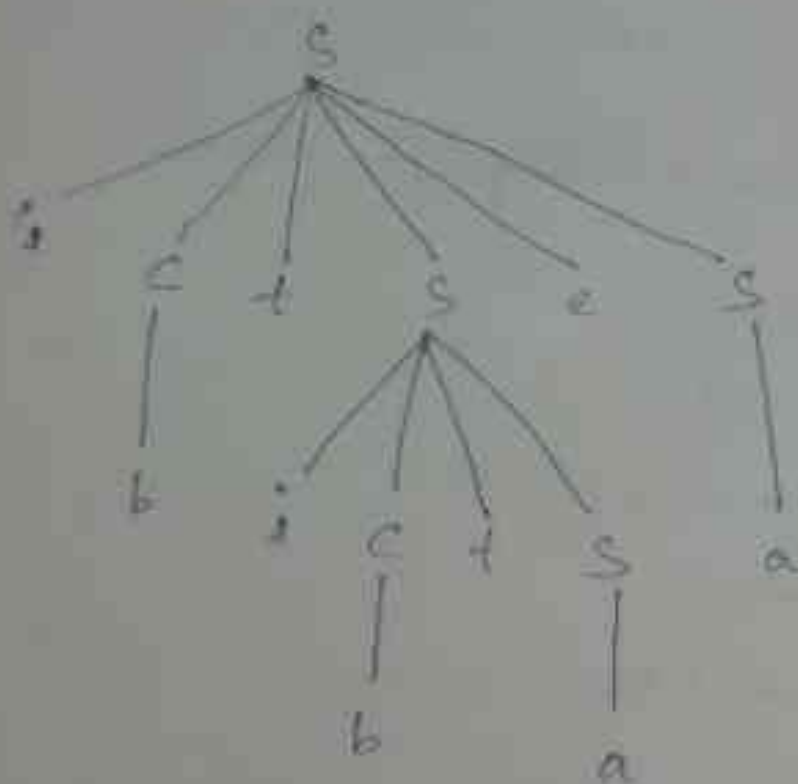
Second leftmost derivation: $\text{string}(w) = ibtibktara$

$$\begin{aligned} S &\Rightarrow ictSeS \\ &\Rightarrow ibtSeS \\ &\Rightarrow ibtictSeS \\ &\Rightarrow ibtibktSeS \\ &\Rightarrow ibtibktarS \\ &\Rightarrow ibtibktara \end{aligned}$$

Since, there are two distinct leftmost derivations for the $\text{string}(w) = ibtibktara$, the given grammar is ambiguous.



parse tree 1



parse tree 2

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

3) Show that the following grammar is ambiguous with respect to the string $w = a \times a + b$.

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

Solution:-

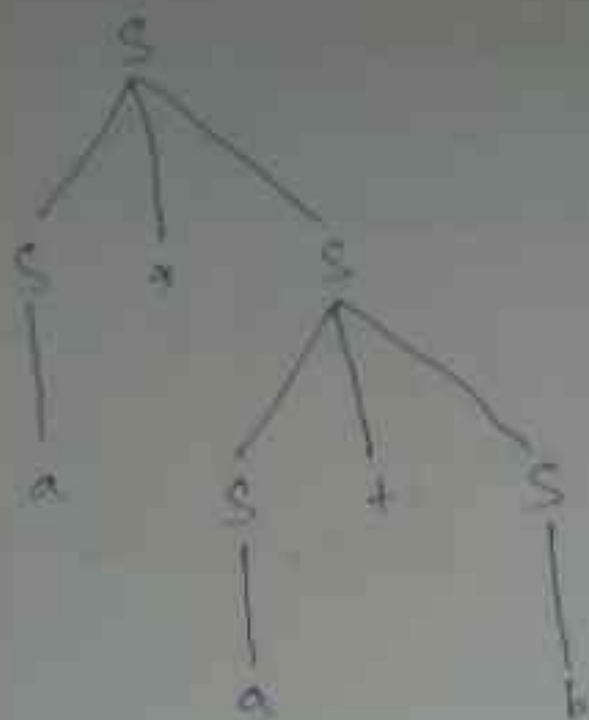
First leftmost derivation:- string $w = a \times a + b$

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

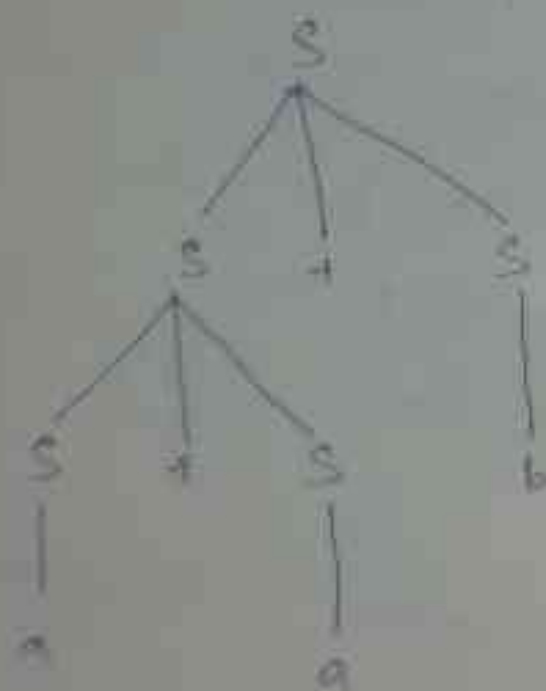
Second leftmost derivation:- string $w = a \times a + b$

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

Since there are two distinct leftmost derivations for the string $w = a \times a + b$, the given grammar is ambiguous.



parse tree 1



parse tree 2

Since there are two distinct parse trees for string $a+ab$ the given grammar 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 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 can be eliminated as shown below:

Consider the A -productions of the form,

$$A \rightarrow A\alpha_1 | A\alpha_2 | A\alpha_3 \dots | A\alpha_n | \beta_1 | \beta_2 | \beta_3 \dots | \beta_n$$

Then the A productions can be replaced by

$$A \rightarrow \beta_1 A' | \beta_2 A' | \beta_3 A' \dots | \beta_n A'$$

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

1) Eliminate the left recursion from the following grammar

$$S \rightarrow S+S | S \times S | a/b$$

Solution: In the given grammar two productions are left recursive

$$S \rightarrow S+S$$

$$S \rightarrow S \times S$$

1) $S \rightarrow S+S | a/b$

$$A = S$$

$$\alpha = +S$$

$$P_1 = a, P_2 = b$$

$$\begin{array}{l} S \rightarrow aS' | bS' \\ S' \rightarrow +SS' | \epsilon \end{array}$$

$$\begin{array}{l} A \rightarrow A\alpha | P_1 | P_2 \\ \Downarrow \\ A \rightarrow P_1A' | P_2A' \\ A' \rightarrow \alpha A' | \epsilon \end{array}$$

2) $S \rightarrow S+S | a/b$

$$A = S$$

$$\alpha = +S, P_1 = a, P_2 = b$$

$$\begin{array}{l} S \rightarrow aS' | bS' \\ S' \rightarrow +SS' | \epsilon \end{array}$$

After eliminating the left recursion, the final grammar is

$$\begin{array}{l} S \rightarrow aS' | bS' \\ S' \rightarrow +SS' | \epsilon \end{array}$$

Let us derive a string $w = a+atb$.

Leftmost derivation:-

$$\begin{aligned} S &\Rightarrow aS' \\ &\Rightarrow a+SS' \\ &\Rightarrow a+aS'S' \\ &\Rightarrow a+a+SS'S' \\ &\Rightarrow a+a+bs'S'S' \\ &\Rightarrow a+a+beS'S' \\ &\Rightarrow a+a+beS' \\ &\Rightarrow a+a+be \\ &\Rightarrow a+atb \end{aligned}$$

we have only one leftmost derivation for the string $w = a+atb$.

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

Q2) Eliminate left recursion from the following grammar:

$$S \rightarrow Ab/a$$

$$A \rightarrow Ab|Sa$$

Solution: The given grammar can be written as

$$S \rightarrow Ab/a$$

$$A \rightarrow Ab|Aba|aa$$

Now A production has left recursion, Eliminate it.

$$A \rightarrow Ab|Aba|aa$$

$$A = A$$

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

$$A \rightarrow aaA'$$

$$A' \rightarrow bA' | baA' | \epsilon$$

$$A \rightarrow A\alpha_1 | A\alpha_2 | \beta$$

$$\Downarrow$$

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha_1 A' | \alpha_2 A' | \epsilon$$

The grammar obtained after eliminating left recursion is

$$S \rightarrow Ab/a$$

$$A \rightarrow aaA'$$

$$A' \rightarrow bA' | baA' | \epsilon$$

3) Eliminate the left recursion from the following grammar.

$$E \rightarrow E+T/T$$

$$T \rightarrow T * F / F$$

$$F \rightarrow (E) / id$$

Solution:- E and T productions have left recursion.

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

$$A = E$$

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

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

$$\boxed{\begin{array}{l} A \rightarrow \alpha A / \beta \\ \Downarrow \\ A \rightarrow \beta A' \\ A' \rightarrow \alpha A' / \epsilon \end{array}}$$

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

$$E \rightarrow TE'$$

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

$$T \rightarrow FT'$$

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

$$F \rightarrow (E) / id$$

4) Eliminate the left recursion from the following grammar.

$$S \rightarrow SOSIS | 01$$

Solution:-

$$S \rightarrow SOSIS | 01$$

$$A = S$$

$$\alpha = OSIS$$

$$\beta = 01$$

$$S \rightarrow 01S'$$

$$S' \rightarrow OSISS' / \epsilon$$



$$\begin{array}{l} A \rightarrow \alpha A \mid \beta \\ \Downarrow \\ A \rightarrow \beta A' \\ A' \rightarrow \alpha A' / \epsilon \end{array}$$

The grammar obtained after eliminating left recursion.

Left factoring in CFGs

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

$$A \rightarrow \alpha \beta_1 | \alpha \beta_2 | \alpha \beta_3 | \dots | \alpha \beta_n$$

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

Elimination of left factoring:

$$A \rightarrow \alpha \beta_1 | \alpha \beta_2 | \alpha \beta_3 | \dots | \alpha \beta_n$$



$$A \rightarrow \alpha A'$$

$$A' \rightarrow \beta_1 | \beta_2 | \beta_3 | \dots | \beta_n$$

1) eliminate the left factoring from the following grammar:

$$S \rightarrow ictS / ictSeS / a$$

$$c \rightarrow b$$

Solution:-

S production has left factoring.

$$S \rightarrow ictS / ictSeS$$

$$A = S$$

$$\alpha = ictS$$

$$\beta_1 = \epsilon$$

$$\beta_2 = \epsilon S$$

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

$$\begin{array}{l} S \rightarrow ictSS' \\ S' \rightarrow \epsilon / \epsilon S \end{array}$$

The grammar obtained after eliminating left factoring is

$$\begin{array}{l} S \rightarrow ictSS' / a \\ S' \rightarrow \epsilon / \epsilon S \\ c \rightarrow b \end{array}$$

2) Eliminate the left factoring from the following grammar.

$$S \rightarrow aSa|aa,$$

Solution:-

$$S \rightarrow aSa|aa$$

$$A = S$$

$$\alpha = a$$

$$\beta_1 = Sa$$

$$\beta_2 = a$$

$$\begin{array}{l} A \rightarrow \alpha\beta_1 | \alpha\beta_2 \\ \Downarrow \\ A \rightarrow \alpha A' \\ A' \rightarrow \beta_1 | \beta_2 \end{array}$$

$$\begin{array}{l} S \rightarrow aS' \\ S' \rightarrow Sa|a \end{array}$$

The grammar obtained after eliminating left factoring.

3) Eliminate the left factoring from the grammar:

$$S \rightarrow a s s b s \mid a s a s b \mid a b b \mid b$$

Solution:-

$$S \rightarrow a s s b s \mid a s a s b \mid a b b$$

$$A = S$$

$$\alpha = a$$

$$P_1 = s s b s$$

$$P_2 = s a s b$$

$$P_3 = b b$$

$$A \rightarrow \alpha P_1 \mid \alpha P_2 \mid \alpha P_3$$

$\Downarrow$

$$A \rightarrow \alpha A'$$

$$A' \rightarrow P_1 \mid P_2 \mid P_3$$

$$S \rightarrow a s'$$

$$s' \rightarrow s s b s \mid s a s b \mid b b$$

Again s' production has left factoring.

$$s' \rightarrow s s b s \mid s a s b$$

$$A = s'$$

$$\alpha = s$$

$$P_1 = s b s$$

$$P_2 = a s b$$

$$A \rightarrow \alpha P_1 \mid \alpha P_2$$

$\Downarrow$

$$A \rightarrow \alpha A'$$

$$A' \rightarrow P_1 \mid P_2$$

$$s' \rightarrow s s''$$

$$s'' \rightarrow s b s \mid a s b$$

The grammar obtained after eliminating left factoring is

$$\begin{aligned} S &\rightarrow as' / b \\ s' &\rightarrow ss'' / bb \\ s'' &\rightarrow sb s / asb \end{aligned}$$

4) Eliminate the left factoring from the grammar:

$$S \rightarrow bssas / bssasb / bsb / a$$

Solution:-

$$S \rightarrow bssas / bssasb / bsb$$

$$A = S, \alpha = bS, \beta_1 = SasS, \beta_2 = Sasb, \beta_3 = b$$

$$\begin{aligned} S &\rightarrow bss' \\ s' &\rightarrow SasS / Sasb / b \end{aligned}$$

$$\begin{aligned} A &\rightarrow \alpha \beta_1 / \alpha \beta_2 / \alpha \beta_3 \\ &\Downarrow \\ A &\rightarrow \alpha A' \\ A' &\rightarrow \beta_1 / \beta_2 / \beta_3 \end{aligned}$$

Again s' has left factoring.

$$s' \rightarrow SasS / Sasb$$

$$A = s', \alpha = Sa, \beta_1 = as, \beta_2 = sb$$

$$\begin{aligned} s' &\rightarrow Sas'' \\ s'' &\rightarrow as / sb \end{aligned}$$

$$\begin{aligned} A &\rightarrow \alpha \beta_1 / \alpha \beta_2 \\ &\Downarrow \\ A &\rightarrow \alpha A' \\ A' &\rightarrow \beta_1 / \beta_2 \end{aligned}$$

The grammar obtained after eliminating left factoring is

$$\begin{aligned} S &\rightarrow bss' / a \\ s' &\rightarrow Sas'' / b \\ s'' &\rightarrow as / sb \end{aligned}$$

Simplification of CFM's

In a CFM, 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 CFM.

Simplification of CFM comprises the following steps:

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

By simplifying CFM'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 $A \rightarrow \epsilon$ is called epsilon production. Here the variable A is called nullable variable.

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

1) Eliminate ϵ -productions from the following grammar.

$$S \rightarrow aSb \mid aAb$$

$$A \rightarrow \epsilon$$

Solution: In the given grammar, $A \rightarrow \epsilon$ is ϵ -production.

A is nullable variable.

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

$$S \rightarrow aAb \quad (\text{with } A)$$

$$S \rightarrow ab \quad (\text{without } A)$$

Resulting grammar,

$$S \rightarrow aSb \mid aAb \mid ab$$

In the resulting grammar, A is not reachable from start symbol (S).

So, remove $S \rightarrow aAb$ productions.

The grammar obtained after eliminating ϵ -productions are

$$S \rightarrow aSb/ab$$

2) Eliminate all ϵ -productions from the grammar.

$$S \rightarrow AB$$

$$A \rightarrow aAA/\epsilon$$

$$B \rightarrow bBB/\epsilon$$

Solution:- In the given grammar,

$$A \rightarrow \epsilon$$

$$B \rightarrow \epsilon \quad \text{are two } \epsilon\text{-productions.}$$

A and B are nullable variables. So, write with ϵ and without ϵ .

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

The grammar after eliminating ϵ -productions.

3) Eliminate ϵ -productions from the grammar:

$$S \rightarrow ABCa | bD$$

$$A \rightarrow BC | b$$

$$B \rightarrow b | \epsilon$$

$$C \rightarrow c | \epsilon$$

$$D \rightarrow d$$

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 ϵ -productions.

Answering

$$S \rightarrow ABCa | BCa | ACa | ABa | Ca | Aa | Ba | a | bD$$

$$A \rightarrow BC | b | c | b$$

$$B \rightarrow b$$

$$C \rightarrow c$$

$$D \rightarrow d$$



Grammar after eliminating ϵ -productions.

4) Eliminate ϵ -productions from the following grammar:

$$S \rightarrow XYX$$

$$X \rightarrow 0X | \epsilon$$

$$Y \rightarrow 1Y | \epsilon$$

Solution:

$\{X, Y\}$ are nullable variables.

$$\begin{array}{l} S \rightarrow XYX | XY | YX | XX | X | Y | \epsilon \\ X \rightarrow 0X | \epsilon \\ Y \rightarrow 1Y | \epsilon \end{array}$$

5) Eliminate ϵ -productions from the grammar:

$$S \rightarrow BAAB$$

$$A \rightarrow 0AA | 2AA | \epsilon$$

$$B \rightarrow AB | 1B | \epsilon$$

Solution:

A, B are nullable variables.

$$\begin{array}{l} S \rightarrow BAAB | BA | AB | BA | AA | AB | BA | AA | A | B | \epsilon \\ A \rightarrow 0AA | 0A | 2AA | 2A \\ B \rightarrow AB | A | 1B | \epsilon \end{array}$$

6) Eliminate ϵ -productions from the grammar.

$$S \rightarrow ABAC$$

$$A \rightarrow aA | \epsilon$$

$$B \rightarrow bB | \epsilon$$

$$C \rightarrow c$$

Solution:- A/B are nullable variables.

$S \rightarrow ABAC ABC BAC AAC AC BC C$
$A \rightarrow aA a$
$B \rightarrow bB b$
$C \rightarrow c$

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 $A, B \in V(\text{variables})$

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

~~Given~~ $S \rightarrow Aa/B$

$$B \rightarrow A/bb$$

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

Solution:-

$$S \rightarrow B$$

$$B \rightarrow A$$

$A \rightarrow B$ are unit productions.

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

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

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

$\rightarrow bb$

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

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

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

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.

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

3) Eliminate the unit productions from the given grammar:

$$S \rightarrow 0A|10|C$$

$$A \rightarrow 0S|00$$

$$B \rightarrow 1|A$$

$$C \rightarrow 01$$

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

$$S \rightarrow C \text{ and } B \rightarrow A$$

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

$$B \rightarrow A \rightarrow \begin{matrix} 0S \\ 00 \end{matrix} \Rightarrow \boxed{B \rightarrow 0S|00}$$

The grammar after eliminating unit productions are

$$\begin{array}{l} S \rightarrow 0A|10|01 \\ A \rightarrow 0S|00 \\ B \rightarrow 1|0S|00 \\ C \rightarrow 01 \end{array}$$

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

$$S \rightarrow AA$$

$$A \rightarrow B|BB$$

$$B \rightarrow abB|b|bb$$

Solution: In the given grammar,

$A \rightarrow B$ is a unit production.

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

The grammar after eliminating unit productions is

~~$$S \rightarrow AA$$~~

~~$$A \rightarrow B|BB$$~~

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

Elimination of useless symbols:-

1) Eliminate useless symbols in the grammar

$$S \rightarrow AC|a$$

$$A \rightarrow BC|b$$

$$B \rightarrow aB$$

$$C \rightarrow aC$$

Solution:-

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

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

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.

$$S \rightarrow a$$

$$A \rightarrow b$$

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

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.

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

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: $A \rightarrow a$

$D \rightarrow ab$

$E \rightarrow d$

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

Step-3: $S \rightarrow aA$

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

Step-4: Variable BFC is useless symbols. So, delete all BFC productions and wherever $\underset{\substack{\uparrow \\ \in C}}{A, E}$ present in RHS of remaining productions, replace those productions also.

$S \rightarrow aA$

$A \rightarrow aA/a$

$D \rightarrow ab/Ea$

$E \rightarrow d$

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

$$\begin{aligned} S &\rightarrow aA \\ A &\rightarrow aA/a \end{aligned}$$

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

$$\begin{aligned} S &\rightarrow aS | A \\ A &\rightarrow a \end{aligned}$$

4) Eliminate useless symbols from the grammar

$$S \rightarrow ABC/BaB$$

$$A \rightarrow aA/BaC/aaa$$

$$B \rightarrow bBb/a$$

$$C \rightarrow CA/AC$$

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

$$S \rightarrow BaB$$

$$A \rightarrow aA/aaa$$

$$B \rightarrow bBb/a$$

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

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

5) Eliminate useless symbols from the grammar:

$$S \rightarrow AB|AC$$

$$A \rightarrow aAb|bAa|a$$

$$B \rightarrow bBA|aAB|AB$$

$$C \rightarrow aBCA|aAb$$

$$D \rightarrow bD|aC$$

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.

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow aAb|bAa|a \\ B &\rightarrow bBA|aAB|AB \end{aligned}$$

Normal forms:-

- 1) In a CFG, there is no restriction on the ~~left~~ 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):-

• Context free grammar (CFG) is

said to be in CNF if all productions are of the form

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

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

$a \in T$ (Terminal)

Example:-

$$\begin{array}{l} S \rightarrow AS | a \\ A \rightarrow SA | b \end{array}$$

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 $S' \rightarrow S$.
- 2) Reduce the grammar such that there are no ϵ -productions, unit productions and useless symbols in the grammar.
- 3) Replace each production $A \rightarrow B_1 B_2 \dots B_n$ where $n > 2$, with $A \rightarrow B_1 C$ where $C \rightarrow B_2 B_n$. 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 $A \rightarrow aB$ where a is the terminal and A, B are variables, then production is replaced by $A \rightarrow XB$ and $X \rightarrow a$. Repeat this step for every production which is in the form of $A \rightarrow aB$.

1) Convert the following grammar to CNF

$$S \rightarrow ASA/aB$$

$$A \rightarrow B/S$$

$$B \rightarrow b/\epsilon$$

Solution:-

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

$$S' \rightarrow S$$

$$S \rightarrow ASA/aB$$

$$A \rightarrow B/S$$

$$B \rightarrow b/\epsilon$$

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

After removing ϵ -productions,

$$S' \rightarrow S$$

$$S \rightarrow ASA/AS/SAB/aB/a$$

$$A \rightarrow B/S$$

$$B \rightarrow b$$

3) Remove unit productions

$$S' \rightarrow S, A \rightarrow B \text{ and } A \rightarrow S$$

After removing unit productions,

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

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

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

$$B \rightarrow b$$

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

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

$$S \rightarrow ASA, S \rightarrow ASA, A \rightarrow ASA$$

After simplifying,

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

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

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

$$B \rightarrow b$$

$$C \rightarrow SA$$

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

$$S' \rightarrow aB, S \rightarrow aB \text{ and } A \rightarrow aB$$

The grammar which is in CNF is

~~$S' \rightarrow aB, S \rightarrow aB, A \rightarrow aB$~~

$$S' \rightarrow AC | AS | SA | DB | a$$

$$S \rightarrow AC | AS | SA | DB | a$$

$$A \rightarrow b | AC | AS | SA | DB | a$$

$$B \rightarrow b$$

$$C \rightarrow SA$$

$$D \rightarrow a$$

Q) 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,

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

$$A \rightarrow aA | a$$

$$B \rightarrow bB | b$$

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,

$$S \rightarrow CA|AB|BA|AA|bA|b|E$$

$$A \rightarrow aA|a$$

$$B \rightarrow bB|b$$

$$C \rightarrow AB$$

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

$$S \rightarrow CA|AB|BA|AA|EB|b|E$$

$$A \rightarrow DA|a$$

$$B \rightarrow EB|b$$

$$C \rightarrow AB$$

$$D \rightarrow a$$

$$E \rightarrow b$$

3) Convert the following CFG to CNF:

$$S \rightarrow aaaaS | aaaa$$

Solution:-

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

$$S' \rightarrow S$$

$$S \rightarrow aaaaS | aaaa$$

2) Remove unit production $S' \rightarrow S$.

$$S' \rightarrow aaaaS | aaaa$$

$$S \rightarrow aaaaS | aaaa$$

substitute $A \rightarrow a$.

$$S' \rightarrow AAAAS | AAAA$$

$$S \rightarrow AAAAS | AAAA$$

$$A \rightarrow a$$

substitute $B \rightarrow AAAA$.

$$S' \rightarrow BS | AAAA$$

$$S \rightarrow BS | AAAA$$

$$B \rightarrow AAAA$$

~~substitute~~

$$A \rightarrow a$$

Substitute $C \rightarrow AAA$

$S' \rightarrow BS|CA$

$S \rightarrow BS|CA$

$B \rightarrow CA$

$C \rightarrow AAA$

$A \rightarrow a$

Substitute $D \rightarrow AA$

$S' \rightarrow BS|CA$

$S \rightarrow BS|CA$

$B \rightarrow CA$

$C \rightarrow DA$

$D \rightarrow AA$

$A \rightarrow a$



The grammar is in CNF.

4) Convert the following CFL 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 $A \rightarrow 0$, $B \rightarrow 1$

$S' \rightarrow ASA \mid AA \mid BSBB \mid BB \mid \epsilon$
$S \rightarrow ASA \mid AA \mid BSBB \mid BB$
$A \rightarrow 0$
$B \rightarrow 1$

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

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

The grammar which is in CNF is

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

5) Convert the following CFG to CNF.

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

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

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

Solution:

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

$$S' \rightarrow S$$

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

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

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

2) Eliminate the unit production $S' \rightarrow S$.

$$S' \rightarrow 0A|1B$$

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

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

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

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

$$A \rightarrow 0AA \text{ and } B \rightarrow 1BB \dots$$

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

$$\begin{array}{l} S' \rightarrow 0A/1B \\ S \rightarrow 0A/1B \\ A \rightarrow 0X/1S/1 \\ B \rightarrow 1Y/0S/0 \\ X \rightarrow AA \\ Y \rightarrow BB \end{array}$$

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

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

$$\begin{array}{l} S' \rightarrow CA/DB \\ S \rightarrow CA/DB \\ A \rightarrow CX/DS/1 \\ B \rightarrow DY/CB/0 \\ X \rightarrow AA \\ Y \rightarrow BB \\ C \rightarrow 0 \\ D \rightarrow 1 \end{array}$$

Kyrenbach Normal Form (KNF):

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

$$A \rightarrow aB_1B_2 \dots B_n$$

(or)

$$A \rightarrow a$$

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

Ex: $A \rightarrow aBCD$

(or)

$$A \rightarrow a$$

Steps to convert a given CFG to KNF:

- 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 $A_i \rightarrow A_j X$, then $i < j$.

5) Remove left recursion (if exists).

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

1) Convert the following CFG to CNF.

$$S \rightarrow CA|BB$$

$$B \rightarrow b|SB$$

$$C \rightarrow b$$

$$A \rightarrow a$$

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,

$$A_1 \rightarrow A_2 A_3 \mid A_4 A_4$$

$$A_4 \rightarrow b \mid A_1 A_3$$

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

3) If the production is of the form $A_i \rightarrow A_j X$, then it's.

$$A_4 \rightarrow b \mid A_1 A_4$$

$$A_4 \rightarrow b \mid A_2 A_3 A_4 \mid A_4 A_4 A_4 \quad (\text{Replace } A_1 \rightarrow A_2 A_3 \mid A_4 A_4)$$

$$A_4 \rightarrow b \mid b A_3 A_4 \mid A_4 A_4 A_4 \quad (\text{Replace } A_2 \rightarrow b)$$

4) Eliminate the left recursion.

$$A = A_4, P_1 = b, P_2 = b A_3 A_4, \alpha = A_4 A_4$$

$$A_4 \rightarrow b z \mid b A_3 A_4 z$$

$$z \rightarrow A_4 A_4 z \mid \epsilon$$

eliminate ϵ -production $z \rightarrow \epsilon$

$$A_4 \rightarrow b z \mid b \mid b A_3 A_4 z \mid b A_3 A_4$$

$$z \rightarrow A_4 A_4 z \mid A_4 A_4$$

$\begin{aligned} A &\rightarrow A \alpha \mid P_1 \mid P_2 \\ &\Downarrow \\ A &\rightarrow P_1 A' \mid P_2 A' \\ A' &\rightarrow \alpha A' \mid \epsilon \end{aligned}$
--

Now the grammar is,

$$A_1 \rightarrow A_2 A_3 | A_4 A_4$$

$$A_4 \rightarrow b | b A_3 A_4 | b z | b A_3 A_4 z$$

$$z \rightarrow A_4 A_4 | A_4 A_4 z$$

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

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

$$A_1 \rightarrow b a a | b a a | b a a A_4 A_4 | b z A_4 | b A_3 A_4 z A_4$$

$$A_4 \rightarrow b | b A_3 A_4 | b z | b A_3 A_4 z$$

$$z \rightarrow b A_4 | b A_3 A_4 A_4 | b z A_4 | b A_3 A_4 z A_4 | b A_4 z | b A_3 A_4 A_4 z | b z A_4 z | b A_3 A_4 z A_4 z$$

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

2) Convert the following grammar to CNF.

$$A_1 \rightarrow A_2 A_3$$

$$A_2 \rightarrow A_3 A_1 | b$$

$$A_3 \rightarrow A_1 A_2 | a$$

Solution:

1) The given grammar is already in CNF.

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

$$A_3 \rightarrow A_1 A_2 | a$$

$$A_3 \rightarrow A_2 A_3 A_2 | a \quad (\text{Replace } A_1 \rightarrow A_2 A_3)$$

$$A_3 \rightarrow A_3 A_1 A_3 A_2 | b A_3 A_2 | a \quad (\text{Replace } A_2 \rightarrow A_3 A_1 | b)$$

3) Eliminate the left recursion.

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

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

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

eliminate ϵ -production $X \rightarrow \epsilon$

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

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

Now the grammar is,

$$A_1 \rightarrow A_2 A_3$$

$$A_2 \rightarrow A_3 A_1 | b$$

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

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

Substitute A_3 production in A_2 production.

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

Now substitute A_3 production in A_1 production.

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

Now substitute A_1 production in X production.

The grammar is

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

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

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

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

$$b A_3 A_2 A_2 | b A_3 A_2 X A_1 A_3 A_2 A_2 X | a X A_1 A_3 A_2 A_2 X | b A_3 A_2 A_1 A_3 A_2 A_2 X |$$

$$a A_1 A_3 A_2 A_2 X | b A_3 A_2 A_2 X$$

3) Convert the following grammar to Greibach Normal form.

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

$$A \rightarrow aA|a$$

$$B \rightarrow bB|b$$

Solution:-

1) Remove unit productions ~~$S \rightarrow B$~~

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

$$A \rightarrow aA|a$$

$$B \rightarrow bB|b$$

2) Convert the grammar into CNF.

$$S \rightarrow XA|AB|BA|AA|bB|b$$

$$A \rightarrow YA|a$$

$$B \rightarrow ZB|b$$

$$X \rightarrow AB$$

$$Y \rightarrow a$$

$$Z \rightarrow b$$

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

x with A_6

we get,

$$A_1 \rightarrow A_2 A_3 \mid A_1 A_4 \mid A_4 A_3 \mid A_1 A_5 \mid A_5 A_4 \mid b$$

$$A_3 \rightarrow A_6 A_3 \mid a$$

$$A_4 \rightarrow A_5 A_4 \mid b$$

$$A_2 \rightarrow A_3 A_4$$

$$A_6 \rightarrow a$$

$$A_5 \rightarrow b$$

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

Substitute $A_6 \rightarrow a$ in A_3

substitute $A_5 \rightarrow b$ in A_4 & A_1

$$A_1 \rightarrow A_0 A_3 | A_3 A_4 | A_4 A_3 | A_3 A_3 | b A_4 | b$$

$$A_3 \rightarrow a A_3 | a$$

$$A_4 \rightarrow b A_4 | b$$

$$A_2 \rightarrow A_3 A_4$$

$$A_6 \rightarrow a$$

$$A_5 \rightarrow b$$

Substitute A_3 in A_2 & A_1 and A_4 in A_1

$$A_1 \rightarrow A_3 A_3 | a A_3 A_4 | a A_4 | b A_4 A_3 | b A_3 | a A_3 A_3 | a A_3 | b A_4 | b$$

$$A_3 \rightarrow a A_3 | a$$

$$A_4 \rightarrow b A_4 | a$$

$$A_2 \rightarrow a A_3 A_4 | a A_4$$

$$A_6 \rightarrow a$$

$$A_5 \rightarrow b$$

Substitute A_2 in A_1

$$A_1 \rightarrow a A_3 A_4 A_3 | a A_4 A_3 | a A_3 A_4 | a A_4 | b A_4 A_3 | b A_3 | a A_3 A_3 | a A_3 | b A_4 | b$$

$$A_3 \rightarrow a A_3 | a$$

$$A_4 \rightarrow b A_4 | a$$

$$A_2 \rightarrow a A_3 A_4 | a A_4$$

$$A_6 \rightarrow a$$

$$A_5 \rightarrow b$$

4) Convert the following grammar into CNF

$$S \rightarrow AA/a$$

$$A \rightarrow SS/b$$

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,

$$A_1 \rightarrow A_2 A_2 / a$$

$$A_2 \rightarrow A_1 A_1 / b.$$

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

$$A_2 \rightarrow A_1 A_1 / b.$$

$$A_2 \rightarrow A_2 A_2 A_1 / a A_1 / b \quad (\text{Replace } A_1 \rightarrow A_2 A_2 / a)$$

4) Eliminate the left recursion.

$$A_2 \rightarrow a A_1 B / b B$$

$$B \rightarrow A_2 A_1 B / \epsilon$$

eliminate ϵ -production $B \rightarrow \epsilon$

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

$$B \rightarrow A_2A_1B \mid A_2A_1$$

The grammar is,

$$A_1 \rightarrow A_2A_2 \mid a$$

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

$$B \rightarrow A_2A_1B \mid A_2A_1$$

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

Substitute A_2 in A_1 & B .

~~$$A_1 \rightarrow A_2A_2 \mid a$$~~

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

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

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

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 $w = uvwx^i y$ such that

(i) $uv \neq \epsilon$

(ii) $|vwx| \leq n$

(iii) for all $i \geq 0$, the string $uv^iwx^i y$ 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.

v) 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) = $aabbbccc$

Divide the string(w) into $uvwx^2y$.

Case (i):-

$$w = \frac{a}{u} \frac{a}{v} \frac{a}{w} \frac{b}{x} \frac{b}{y} \frac{c}{c} \frac{c}{c}$$

Let $x^2 = b^2$

$$uv^2wx^2y = \frac{a}{u} \frac{a}{v^2} \frac{a}{w} \frac{b}{x^2} \frac{b}{y} \frac{c}{c} \frac{c}{c}$$

$$= aababbbccc$$

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

So, the string $a^5 b^3 c^5$ 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):-

$$w = \frac{aa}{u} \frac{ab}{v} \frac{bb}{w} \frac{ccc}{x} \frac{c}{y}$$

$$uv^2wx^2y = aaababbbcccc$$

So, the string $aaababbbcccc$ 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

Soln (iii): $w = \underline{a a a b b b c c c}$
 $u v w x y$

$$u v^2 w x^2 y = a a a a b b b c c c$$

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

So, the string $a^4 b^4 c^3$ 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.

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

Solution: string = $\underline{a b a b a b a b}$

divide the string into two

$$= \begin{array}{cccccc} a & b & a & b & a & b \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ u & v & w & x & y & \end{array}$$

$$u v^2 w x^2 y = a b b a b a b a b$$

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

Solution:-

Let $n = 7$.

String(w) = $aaaaaaa$

Divide the string into $uvwx^2y$.

$$w = \frac{aaaaaa}{uvwx^2y}$$

$$uv^2wx^2y = aaaaaaaaa$$

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

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

Closure properties of context free languages:-

CFL are closed under

- 1) union
- 2) concatenation
- 3) Kleene closure.

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

Example:- $L_1 = \{a^n b^n \mid n \geq 1\}$

CFG for L_1 is

$$S_1 \rightarrow aAb \mid ab$$

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

CFG for L_2 is

$$S_2 \rightarrow cBd \mid \epsilon$$

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

$$= \{a^n b^n \mid n \geq 1\} \cup \{c^m d^m \mid m \geq 0\}$$

CFG for $L_1 \cup L_2$

$S \rightarrow S_1 \mid S_2$
$S_1 \rightarrow aAb \mid ab$
$S_2 \rightarrow cBd \mid \epsilon$

2) Concatenation: If L_1 and L_2 are two context free languages, then their concatenation $L_1 \cdot L_2$ will also be context free.

Example: $L_1 = \{a^n b^n \mid n \geq 1\}$

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

$L = L_1 \cdot L_2$

$= \{a^n b^n c^m d^m \mid n \geq 1, m \geq 1\}$ is also context free.

CFM for L is

$S \rightarrow S_1 S_2$

$S_1 \rightarrow a S_1 b \mid ab$

$S_2 \rightarrow c S_2 d \mid cd$

3) Kleene closure: If L is a context free language,

then L^+ is also a context free.

Example: $L = \{a^n b^n \mid n \geq 0\}$

CFM for L is $S \rightarrow a S b \mid \epsilon$

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

CFM for L^+ is $S_1 \rightarrow S S_1 \mid \epsilon$