Content Free Grammar

It describes content free languages. it is formally defined by 4 tuples:

$G = (V, T, P, S)$ where

V → Non empty set of variables / non terminals.

T → Non empty set of Terminals.

P → Non empty set of productions of the form

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

S → Start symbol / Variable.

Example :- $G = (V, T, P, S)$ with $V = \{S, A\}$ $T = \{a, b\}$

$P = \{ S \rightarrow aSAb \mid a,\ A \rightarrow bA \mid b \}$

Derivation :- Derivation starts with start symbol.

$S \Rightarrow aSAb$	$S \rightarrow aSAb$
$\Rightarrow aaSAbAb$	$S \rightarrow aSAb$
$\Rightarrow aaaAbAb$	$S \rightarrow a$
$\Rightarrow aaabbAb$	$A \rightarrow b$
$\Rightarrow aaabbbb$	$A \rightarrow b$

Sentential form

The process of deriving strings containing terminal

(5) $L(G) = \{ a^n b^m \mid n \geq 1, m \geq 0 \}$

$V = \{S, A, B\}$ $T = \{a, b, \epsilon\}$ $\{ S \rightarrow AB,\ A \rightarrow aA \mid a,\ B \rightarrow bB \mid \epsilon \}$

symbols starting from the start symbol by applying some of its productions in G.

Derivation are of two types, namely

i) Leftmost derivation (LMD)

ii) Rightmost derivation (RMD)

eg:- LMD

$S \rightarrow aSAb$
$\rightarrow aaSAbAb$
$\rightarrow aaaAbAb$
$\rightarrow aaabbAb$
$\rightarrow aaabbbb$

RMD

$S \rightarrow aSAb$
$\rightarrow aSbb$
$\rightarrow aaSAbbb$
$\rightarrow aaSbbbb$
$\rightarrow aaabbbb$

$S \rightarrow aSAb$
$A \rightarrow b$
$S \rightarrow aSAb$
$A \rightarrow b$
$S \rightarrow a$

Language defined by CFG :-

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

1) For generating a language that generates equal number of a's and b's in the form $a^n b^n$, the content free grammar will be defined as

$G = \{ (S, A) (a, b), (S \rightarrow aAb, A \rightarrow aAb \mid \epsilon) \}$

$S \rightarrow aAb$
$\rightarrow aaAbb$ (by $A \rightarrow aAb$)
$\rightarrow aaaAbbb$
$\rightarrow aaabbb$ (by $A \rightarrow \epsilon$)
$\rightarrow a^3 b^3 \Rightarrow a^n$

1) Construct CFG for the language having any number of a's over the set $\Sigma = \{a\}$ over production rule $S \to as$, $S \to \epsilon$ & derive i/p : aaaaaaa

$\Sigma = \{a\}$

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

$R.E = a^*$

$S \Rightarrow aS$	$S \to as$
$\Rightarrow aas$	
$\Rightarrow aaas$	$S \to as$
$\Rightarrow aaaas$	$S \to as$
$\Rightarrow aaaaas$	
$\Rightarrow aaaaaaas$	
$\Rightarrow aaaaaaaa$	

2) Construct a CFG for language

$L = \{ WC\,WR \mid \text{where } W \in (a,b)^* \}$

$L = \{ aacaa,\ bcb,\ abcba,\ abbcbba \ldots \}$

The grammar Could be $S \to aSa$, $S \to bSb$, $S \to c$

$S \Rightarrow aSa$

$\Rightarrow abSba$

$\Rightarrow abbSbba$

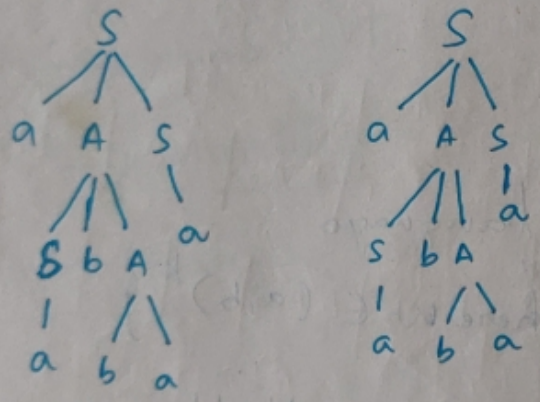
$\Rightarrow abbcbba$

1) Consider the following G: $S \to aAS$ (1), $S \to a$ (2), $A \to SS$ (4), $A \to ba$ (5) for the input string "aabbaa". find i) Leftmost derivation. ii) Rightmost derivation iii) Derivation tree.

i) $S \overset{LM}{\Rightarrow} aAS$ by (1)
$\Rightarrow aSbAS$ by (3)
$\Rightarrow aabAS$ by (2)
$\Rightarrow aabbas$ by (5)
$\Rightarrow$ aabbaa by (2)

ii) $S \overset{RM}{\Rightarrow} aAS$ by (1)
$\overset{RM}{\Rightarrow} aAa$ by (2)
$\overset{RM}{\Rightarrow} aSbAa$ by (3)
$\overset{RM}{\Rightarrow} aSbbaa$ by (5)
$\overset{RM}{\Rightarrow}$ aabbaa by (2)

iii)



2) Derive the string "abb" for left most derivation & Right most derivation using CFG given by

$S \to AB \mid \epsilon$ $A \to aB$ $B \to Sb$

LM
$S \to AB$
$\to aBB$
$\to asbB$
$\to abSb$
$\to abb$
$\to a^3b^3 \Rightarrow a^n$

RM
$S \to AB$
$\to ASb$
$\to AEb$
$\to aBb$
$\to asbb$
$\to aEbb$
$\to abb$

Given the language, deriving the grammar

$L(G) = \{ a^n, n \geq 0 \}$

Step1 :- Simplifying the language

$L(G) = \{ a^0, a^1, a^2, a^3 \text{---} \}$

$= \{ \epsilon, a, aa, aaa \text{---} \}$

Step2 :- To derive the grammar, always starts with first case

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

$V = \{ A \}$ $T = \{ \epsilon, a \}$ $\{ A \to \epsilon \mid aA \}$ OR $\{ A \to \epsilon, A \to aA \}$

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

$L(G) = \{ a, aa, aaa \text{---} \}$

$V = \{ B \}$ $T = \{ a \}$ $\{ B \to a, B \to aB \}$

3) $L(G) = \{ a^{4n+1} \mid n \geq 0 \}$

$L(G) = \{ a, a^5, a^9, a^{13} \text{---} \}$

$V = \{ S \}$ $T = \{ a, \epsilon \}$ $\{ S \to a, S \to a^4 S \}$

4) $L(G) = \{ w \in a^* \mid |w| \bmod 5 = 2 \}$

$L(G) = \{ a^2, a^7, a^{12} \text{---} \}$

$V = \{ S \}$ $T = \{ a \}$ $\{ S \to a^2, a^5 S \}$

(5) $L(G) = \{ \text{[illegible]} \mid n \geq 1, m \geq 0 \}$

$V = \{ S, A, B \}$ $T = \{ a, b, \epsilon \}$ $\{ S \to AB, A \to a\text{[illegible]} \mid a, B \to bB \mid \epsilon \}$

6) $L(G) = \{ a^n b^m \mid (n+m) \text{ is even} \}$

$V = \{S, A, B\}$ $T = \{a, b, \epsilon\}$ $\{ S \to AB \mid aAbB,\ A \to aa$

$B \to bbB \mid \epsilon \}$

7) $L(G) = \{ a^n b^m \mid (n+m) \text{ is odd} \}$

$V = \{S, A, B\}$ $T = \{a, b, \epsilon\}$ $\{ S \to AbB \mid aAB,\ A \to aaA \mid \epsilon,$

$B \to bbB \mid \epsilon \}$

8) $L(G) = \{ a^n b^m \mid n \geq 3,\ m \geq 2 \}$

$V = \{S, A, B\}$ $T = \{a, b, \epsilon\}$ $\{ S \to aaaAbbB,$

$A \to aA \mid \epsilon,\ B \to bB \mid \epsilon \}$

9) $L(G) = \{ a^n b^m \mid n \geq 0 \}$

$V = \{S, A, B\}$ $T = \{a, b, \epsilon\}$ $\{ S \to aSb \mid \epsilon \}$

10) $L(G) = \{ a^n b^n c^m \mid n \geq 1,\ m \geq 0 \}$

$V = \{S, A, B\}$ $T = \{a, b, \epsilon\}$ $\{ S \to AB,\ A \to aAab \mid ab$

$B \to CB \mid \epsilon \}$

11) $L(G) = \{ a^n b^n c^m d^m \mid n, m \geq 0 \}$

$\{ S \to AB,\ A \to aAb \mid \epsilon,\ B \to cBd \mid \epsilon \}$

12) $L(G) = \{ a^n b^m \mid n \neq m \text{ and } n, m \geq 1 \}$

$\{ S \to aSb \mid aAb \mid aBb,\ A \to aA \mid a\ \ B \to bB \mid b \}$

$L = \{ a^n b^m \mid n \geq m \}$ (4)

$L(G) = \{ ab, aab, aabb, aaabb \ldots \}$

$\{ S \rightarrow aAb \mid aSb, \; A \rightarrow aA \mid \epsilon \}$

14) Find the grammar that generates all strings of a's & b's where each string is starting and ending with same symbol.

$R.E = a(a+b)^*a + b(a+b)^*b \mid a \mid b$

$S \rightarrow aAa \mid bAb \mid a \mid b$

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

15) Starting and ending with different symbol.

$R.E = a(a+b)^*b \mid b(a+b)^*a \mid a \mid b$

$S \rightarrow aAb \mid bAa \mid a \mid b$

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

16) $L = \{ a^n b^m \mid m > n \}$

$S \rightarrow aSb \mid bA$

$A \rightarrow bA \mid \epsilon$

17) $L = \{ a^n b^n \mid n \geq 0 \}$

$S \rightarrow aSb \mid \epsilon$

$V = \{ S \} \quad T = \{ a, b, \epsilon \} \quad \{ S \rightarrow aSb \}$

Generating CFG from regular expression

i) $R = (a+b)^* aa (a+b)^*$

$L = \{ aa, baab, aaa\ bbbaa, aabab \ldots \}$

$S \rightarrow AaaA \mid aa$

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

ii) $R = 0^*1\ (0+1)^*$

$L = \{ 1, 01, 10101, 001, 100001. \ldots \}$

$S \rightarrow A1B$

$A \rightarrow 0A \mid \epsilon$

$B \rightarrow 0B \mid 1B \mid \epsilon$

iii) Write a CFG to generate string having any combi-nations of a's and b's except null.

$L = \{ a, b, ab, aaab, abab, baa, bab, bbb \ldots \}$

$S \rightarrow aS \mid bS \mid a \mid b$

iv) Write a CFG to generate string that belongs to the language of palindromes with alphabets.

$L = \{ aba, aabaa, bab, a, b, \epsilon, aa, bbb \ldots \}$

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

Generate a CFG to generate string given the R.Ex a*

(5)

$S \to \epsilon$

$S \to aS$

vi) $RE = (a+b)^*$

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

vii) $R.E = (a+b)^* a (a+b)^*$

$L = \{ a, bab, aab, \text{---} \}$

$S \to AaA \mid a$

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

viii) $R.E = a^* + a(a+b)^*$

$S \to X \mid Y$

$X \to \epsilon \mid aX$

$Y \to aZ$

ix) $R.E = (a+b)^*(a+b) + (ab+ba)^*(a+b)^* B(a+b)+a$

$S \to XYZ$

$Z \to a$

$X \to PQ$

$P \to \epsilon \mid aP \mid bP$

$Q \to a \mid b$

$Y \to LPNQ$

Define a Context free grammar.

A CFG is a way of describing languages by recursive rules called productions. A CFG Consists of a set of variables, a set of terminal symbols, and a start variable, as well as the productions. Each production Consists of a head variable and a body Consisting of a string of zero or more variables and/or terminals.

A CFG is denoted by $G = (V, T, P, S)$ where

V → Finite set of variables/ non terminals.

T → Finite set of terminals.

P → Finite set of productions.

S → start symbol.

What are the applications of Context free languages?

Context free languages are used in the following:

i) programming languages.

ii) Formalizing the ~~notation~~ notation of parsing.

iii) Translation of programming languages.

iv) String processing applications.

3) What are the important applications of context free grammars.

* Context free grammars are used as basis for compiler design and implementation.

* Designers of Compilers use such grammars to implement Compiler's Components, such as Scanners, parsers, code generators and code Synthesizers.

* Context-free grammars are used as specification mechanisms for programming languages.

* The implementation of almost any programming languages is preceded by a context-free grammar that specifies it.

Example

Consider the grammar $G_4 = (\{E, T, F\}, \{a, +, *, (,)\}, R, E)$ in which R is

$$E \to E+T \mid T$$

$$T \to T*F \mid F$$

$$F \to (E) \mid a$$

Then $L(G_4)$ is the language of arithmetic expressions.

* program synthesis is done by CFG.

What is the language generated by CFG or G?

The language generated by grammar G is $L(G) = \{w \mid w$ in T^* and $S \overset{*}{\Rightarrow} w\}$ in which

* the string contains only terminals.
* The string can be derived from start symbol S.

What is CFL ?

The language of the CFG is the set of terminal strings we can so derive; it is called a Content-free language. The language generated by grammar G is

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

What is sentential form ?

Any step in a derivation is a string of terminals and/or variables. We call such a string a sentential form. If the derivation is leftmost, then the string is a left sentential form.

What is a Formal language ?

Language is a set of valid strings from some alphabet. The set may be empty, finite or infinite. $L(M)$ is the language defined by machine M and $L(G)$ is the language defined by Content free grammars.

The two notations for specifying formal languages are:
i) Grammar or regular Expression (Generative approach)
ii) Automaton (Recognition approach) (7)

What is a parser ?

A parser for grammar G is a program that takes as input a string w and produces as output either a parse tree for w, if w is a sentence of G or an error message indicating that w is not a sentence of G.

Define left most derivation and right most derivation ?

Left most derivation :- In this method, we replace the left most non-terminal by one of its production in the grammar. Such a derivation is known as left most derivation and it is represented by using the relation $\Rightarrow$ $*\Rightarrow$ $\underset{lm}{*\Rightarrow}$ for one or more steps respectively

Right most derivation :- In this method, we replace the right most non terminal by one of its production in the grammar. Such a derivation is known as right most derivation and its represented by using the relation $\Rightarrow$ and $\underset{rm}{*\Rightarrow}$ for one or more steps respectively.

Define parse tree or Derivation tree.

The strings that are derived from the CFG can be represented in a tree format known as parse tree or derivation tree. This parse tree clearly shows how the symbols of a terminal string

are grouped into substrings, each of which belongs to the language of one of the variables of the grammar.

What are the two major normal forms of Context-free grammar?

The two Normal forms are:

i) Chomsky Normal Form (CNF)

ii) Greibach Normal Form (GNF)

How do you simplify the Context-free grammar?

i) First eliminate useless symbols, where the variables or terminals that do not appear in any derivation of a terminal string from the start symbol.

ii) Next eliminate ϵ-production which is of the form $A \to \epsilon$ for some variable A.

iii) Eliminate unit production, which are of the form $A \to B$ for variables A, B.

iv) Finally use any of the normal forms to get the simplified CFG.

1) Convert the given CFG to CNF, Consider the given grammar G as

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

$A \to aBB \mid \epsilon$

$B \to Aa \mid b$

To Convert CFG to CNF

$S \to \epsilon$

$A \to AB$

$A \to a$

$S_0 \rightarrow a \mid XA \mid Ax \mid b$

$S \rightarrow a \mid XA \mid Ax \mid b$

$A \rightarrow RB$

$R \rightarrow XB$

$B \rightarrow Ax \mid b \mid a$

$X \rightarrow a$

∴ The given grammar in CNF

Steps to be followed to Convert CFG to CNF

i) If the start symbol S occurs on right side, then create a new start symbol s' and a new production $s' \rightarrow S$

ii) Remove Null production.

iii) Remove unit production.

iv) Replace each production $A_1 \rightarrow B_1 \cdots B_n$ where $n > 2$ with $A_1 \rightarrow B_1 C$

v) If the right side of any production is in the form $A \rightarrow aB$ where 'a' is a terminal and A & B are non-terminals Then the production is replaced by $A \rightarrow XB$ and $X \rightarrow a$.

1) Convert the following CFG to CNF

$S \rightarrow ASA \mid aB$

$A \rightarrow B \mid S$

$B \rightarrow b \mid \in$

Step 1 :- New production has to be created.

$S_1 \rightarrow S$ ✓ (New prod)

$S \rightarrow a \mid aA \mid B$

$A \rightarrow aBB \mid \epsilon$

$B \rightarrow Aa \mid b$

Step 2 :- $A \rightarrow \epsilon$ null production, we have to remove ϵ

$S_1 \rightarrow S$

$S \rightarrow a \mid aA \mid B$

$A \rightarrow aBB$

$B \rightarrow Aa \mid b \mid a$

Step 3 :- Remove unit production, replace B with B production rule.

$S_0 \rightarrow a \mid aA \mid Aa \mid b$

$A \rightarrow aBB$

$B \rightarrow Aa \mid b \mid a$

$S \rightarrow a \mid aA \mid Aa \mid b$

Step 4 :- In the production rule, which are violating CNF

$S_0 \rightarrow aA \mid Aa \mid a \mid b$

$S \rightarrow aA \mid Aa \mid$

$A \rightarrow aBB$

$B \rightarrow Aa$

$X \rightarrow a \Rightarrow S_0 \rightarrow a \mid XA \mid Ax \mid b$ $\quad B \rightarrow Ax \mid b \mid a$

$S \rightarrow a \mid XA \mid Ax \mid b$ $\quad X \rightarrow a$

$A \rightarrow xBB$

Step1 :— Since S appears in RHS, we add a new state S' and $S' \to S$ is added to the production

$S' \to S$

$S \to ASA \mid aB$

$A \to B \mid S$

$B \to b \mid \epsilon$

Step2 :- Remove the null production : $B \to \epsilon$ and $A \to \epsilon$

After removing $B \to \epsilon$: $S' \to S$, $S \to ASA \mid aB \mid a$,

$A \to B \mid S \mid \epsilon$

$B \to b$

After removing $A \to \epsilon$: $S' \to S$, $S \to ASA \mid aB \mid a \mid AS \mid SA \mid S$

$A \to B \mid S$

$B \to b$

Step3 :- Remove unit production : $S' \to S$, $S \to S$, $A \to B$,

$A \to S$ | After removing $S' \to S$: $S' \to ASA \mid aB \mid a \mid AS \mid SA$

$A \to B \mid S$

$B \to b$

After removing $S \to S$: $S \to ASA \mid aB \mid a \mid AS \mid SA$

$A \to B \mid S$

$B \to b$

$S \to ASA \mid aB \mid a \mid AS \mid AS$

After Removing $A \to B$: $S' \to ASA \mid aB \mid a \mid AS \mid SA$

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

$A \to b \mid S$

$B \to b$

After removing $A \to S$

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

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

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

$B \to b$

step4 :- Now find out the production of the form (should not have more than two variable)

$S' \to ASA$ $S \to ASA$ and $A \to ASA$

$S' \to Ax \mid aB \mid a \mid AS \mid SA$

$S \to Ax \mid aB \mid a \mid AS \mid SA$

$A \to b \mid Ax \mid aB \mid a \mid AS \mid SA$

$B \to b$

$x \to SA$

step5 :- Now change the productions $S' \to aB$, $S \to aB$ and $A \to aB$

$S' \to Ax \mid YB \mid a \mid AS \mid SA$

$S \to Ax \mid YB \mid a \mid AS \mid SA$

$A \to b \mid Ax \mid YB \mid a \mid AS \mid SA$

$B \to b$

$X \to SA$

$Y \to a$

Converting the grammar from CFG to CNF 1.1

(10)

1) $S \rightarrow aA \mid aBB$

$A \rightarrow aaA \mid \epsilon$

$B \rightarrow bB \mid bbC$

$C \rightarrow B$

Step 1 :- Remove the ϵ-production in the grammar

$S \rightarrow aA \mid aBB \mid a$

$A \rightarrow aaA \mid aa$

$B \rightarrow bG \mid bbC$

$G \rightarrow B$

Step 2 :- Remove the unit production in the grammar

$S \rightarrow aA \mid aBB \mid a$

$A \rightarrow aaA \mid aa$

$B \rightarrow bC \mid bbC$

$C \rightarrow bC \mid bbC$

Step 3 :- Remove useless variables in the CFG grammar.

The variables B and C does not terminates the string, hence it is useless production, is to be removed.

$$S \rightarrow aA \mid a$$
$$A \rightarrow aaA \mid aa$$

Step4 :- The production should not end with the variables.

$$S \rightarrow X_aA \mid a$$
$$A \rightarrow X_aX_aA \mid X_aX_a$$
$$X_a \rightarrow a$$
$$Y \rightarrow X_aX_a$$

Then
$$S \rightarrow X_aA \mid a$$
$$A \rightarrow YA \mid X_aX_a$$
$$X_a \rightarrow a$$
$$Y \rightarrow X_aX_a$$

General steps to be followed in Converting CFG to CNF.

1) If the starting symbol occurs in the right side of the production, then replace the production with a new production.

(2) Remove the ϵ-production in the grammar.

(3) Remove the unit production in the grammar.
(4) Remove the useless production in the grammar. (11)
(5) If the production ends with the variable, then the production to be replaced.
(6) The production should have atmost of two variables.

The grammar of the two forms is called as Chomsky normal form when it is of the form:

i) $A \rightarrow a$

ii) $S \rightarrow AB$

the variable should ~~not~~ end with the terminal & the variable produces atmost of two variables in the production.

2) Consider the following CFG

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

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

$B \rightarrow 0BB \mid 1$

where S is the start symbol. Convert this CFG to CNF.

Step 1 :- Consider the production, $S \rightarrow 1A \mid 0B$, This is not in CNF. So replace 1 by P and 0 by q, by adding the production,

14

$P \to 1$, $q \to 0$. we get

$S \to PA \mid qB$

$P \to 1$,

$q \to 0$.

Now the CFG is,

$S \to PA \mid qB$

$P \to 1$

$q \to 0$

$A \to PAA \mid qS \mid 0$

$B \to qBB \mid 1$.

In the above CFG, the productions.

$A \to PAA$

$B \to qBB$

are not in CNF.

Step2: Replace AA by R and BB by T. Then we get,

$A \to PR$

$B \to QT$

$R \to AA$

$T \to BB$

Now the CFG is

$S \to PA \mid qB$

$P \to 1$

$q \to 0$

$A \to PR \mid qS \mid 0$,

$B \to qT \mid 1$, $R \to AA$, $T \to BB$.

Where S is the start symbol. Above CFG is in CNF.

3) $S \rightarrow ABA$

$A \rightarrow aA \mid \epsilon$

$B \rightarrow bB \mid \epsilon$

Step1

Remove ϵ-production.

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

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid \epsilon$

Remove ϵ-production

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

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

Step2 :- Remove unit production.

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

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

Step3 :- put $X \rightarrow AB$

$A_1 \rightarrow a$

$B_1 \rightarrow b$

$S \rightarrow XA$

$S \rightarrow A_1A$

$S \rightarrow B_1B$

$A \rightarrow A_1A \mid a$

$B \rightarrow B_1B \mid b$

Step4 :- $S \rightarrow XA \mid AA \mid BA \mid AB \mid A_1A \mid a \mid B_1B \mid b$

$A \rightarrow A_1A \mid a$ $B \rightarrow B_1B \mid b$.

28

2) $S \rightarrow ab|AB$ $abAB$

$A \rightarrow bAB \mid \epsilon$

$B \rightarrow BAa \mid A \mid \epsilon$

Step1 :- Remove ϵ-production.

$S \rightarrow abAB \mid abB \mid abA \mid ab$

$A \rightarrow bAB \mid bB \mid bA \mid b$

$B \rightarrow BAa \mid Ba \mid A \mid Aa \mid a$

Step2 :- Remove unit production

$S \rightarrow abAB \mid abB \mid abA \mid ab$

$A \rightarrow bAB \mid bB \mid bA \mid b$

$B \rightarrow BAa \mid Ba \mid Aa \mid a \mid bAB \mid bB \mid bA \mid b$

Step3 :- $A_1 \rightarrow a$ $B_1 \rightarrow b$ $C_1 \rightarrow A_1B_1$ $C_2 \rightarrow AB$

$S \rightarrow A_1B_1AB$

$\rightarrow C_1C_2$

$abB \rightarrow A_1B_1B$

$\rightarrow C_1B$

$abA \rightarrow A_1B_1A$

$\rightarrow C_1A$

$S \to C_1C_2 \mid C_1B \mid C_1A \mid A_1B_1$

$A \to bAB \mid bB_1 \mid bA$
$\to B_1AB \mid B_1B \mid B_1A$
$\to B_1C_2$

$A \to B_1C_2 \mid B_1B \mid B_1A \mid b$

$B \to BAA_1 \mid C_3A_1 \mid BA_1 \mid AA_1 \mid a \mid b \mid B_1C_2$

$C_3 \to BA$

$B \to C_3A_1 \mid BA_1 \mid AA_1 \mid a \mid b \mid B_1B \mid B_1A \mid B_1C_2$

$B \to B_1C_2$

1). Consider the grammar $G = (N, T, P, S)$ where $N = \{S, A\}$, $T = \{a, b, c\}$, the production rules in p are: $S \to aSc$, $S \to aAc$, $A \to b$. Find the language generated by the grammar.

A typical derivation in the grammar is:

$S \Rightarrow aSc$
$\Rightarrow aaScc$
$\Rightarrow aaaAccc$
$\Rightarrow aaabccc$

The language generated as

$$L(G) = \{a^n b c^n \mid n \geq 1\}$$

2) Let $G = (N, T, P, S)$ where $N = \{S, B\}$ $T = \{a, b, c\}$; P has the following rules:

$S \to aSBC$, $S \to abc$, $cB \to Bc$, $bB \to bb$. Find the language generated by the grammar.

$S \Rightarrow abc$ hence $abc \in L(G)$

$S \Rightarrow aSBC$
$\Rightarrow aabcBc$
$\Rightarrow aabBcc$
$\Rightarrow aabbcc$ $a^2b^2c^2 \in L(G)$

$S \Rightarrow aSBC$
$\Rightarrow aaSBcBc$
$\Rightarrow aaabcBcBc$
$\Rightarrow aaabBccBc$
$\Rightarrow aaabBcBcc$
$\Rightarrow aaabBBccc$
$\Rightarrow aaabbbccc$ $a^3b^3c^3 \in L(G)$

Hence $L(G) = \{a^n b^n c^n \mid n \geq 1\}$

3) Let $G = (N, T, P, S)$, where $N = \{S\}$, $T = \{a, b\}$, P consists of the following rules:

$S \to aS$, $S \to bS$, $S \to \epsilon$. Find the string generated by the grammar.

$S \Rightarrow aS$
$\Rightarrow abS$
$\Rightarrow abbS$
$\Rightarrow abbaS$ → abbaaS, abbaabS, abbaab

3) Consider the following CFG,

$S \rightarrow a|b|CSS$,

where S is the start symbol. Convert this CFG to CNF.

Step1 :- The production, $S \rightarrow CSS$ is not in CNF. So replace SS by P, we get

$S \rightarrow CP$

$P \rightarrow SS$

Now, the CFG becomes,

$S \rightarrow a|b|CP$,

$P \rightarrow SS$, where S is the start symbol.

above CFG is in CNF.

4) Consider the following CFG

$S \rightarrow abSb|a|aAb$,

$A \rightarrow bS|aAAb$, where S is the start symbol.

Convert this CFG to CNF.

Step1 :- Consider the production $S \rightarrow abSb|aAb$ is not in CNF. So replace Ab by P, we get

$S \rightarrow abSb|aP$

$P \rightarrow Ab$

Now the grammar becomes,

$S \rightarrow abSb|a|aP$

$P \rightarrow Ab$

$A \rightarrow bS|aAp$

16 Now replace Sb by R, we get

$S \to abR \mid a \mid aP$

$P \to Ab$

$A \to bS \mid aAP$

$R \to Sb$

Now replace bR with T, we get

$S \to aT \mid a \mid aP$

$P \to Ab$

$A \to bS \mid aAP$

$R \to Sb$

$T \to bR$

Now replace aA with U, we get

$S \to aT \mid a \mid aP$

$P \to Ab$

$A \to bS \mid UP$

$R \to Sb$

$T \to bR$

$U \to aA$

Now replace a with V and b with W, we get

$S \to VT \mid a \mid VP$

$P \to AW$

$A \to WS \mid UP$

$R \to SW$

$T \to WR$

$U \to VA$, $V \to a$, $W \to b$ where S is the start symbol. Hence it is CNF.

4) Find the grammar in CNF Equivalent to (15)

$S \rightarrow aAbB$

$A \rightarrow aA \mid b$

$B \rightarrow bB \mid b$

Step1 :- i) $S \rightarrow C_aAC_bB$

Add new productions $C_a \rightarrow a$ and $C_b \rightarrow b$

ii) $A \rightarrow aA$ becomes

$A \rightarrow C_aA$

iii) $B \rightarrow bB$ becomes

$B \rightarrow C_bB$

Step2 :- i) $S \rightarrow C_aAC_bB$ is Converted as

$S \rightarrow C_aD_1$ where

$D_1 \rightarrow AC_bB$

ii) $A \rightarrow C_aA$ (already in CNF)

iii) $B \rightarrow C_bB$ (already in CNF)

Step3 :-

$S \rightarrow C_aD_1$

$D_1 \rightarrow AD_2$

$D_2 \rightarrow C_bB$

$A \rightarrow C_aA$

$B \rightarrow C_bB$

$A \rightarrow a$

$B \rightarrow b$

5) 18 Convert the grammar with productions:

$S \rightarrow ABa$

$A \rightarrow aab$

$B \rightarrow Ac$ to Chomsky normal form.

Step1 :- Introduce new variables B_a B_b B_c & substitute to a, b, c, Then $S \rightarrow ABB_a$

$A \rightarrow B_aB_aB_b$

$B \rightarrow AB_c$

$B_a \rightarrow a$

$B_b \rightarrow b$

$B_c \rightarrow c$

Step2 :- In the second step, introduce additional variables to get the first two productions into normal form & we get the final result:

$S \rightarrow AD_1$

$D_1 \rightarrow BB_a$

$A \rightarrow B_aD_2$

$D_2 \rightarrow B_aB_b$

$B \rightarrow AB_c$

$B_a \rightarrow a$

$B_b \rightarrow b$

$B_c \rightarrow c$

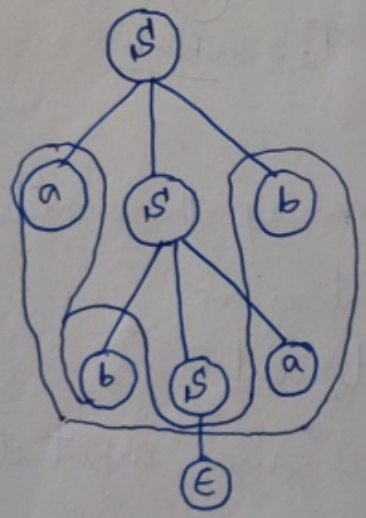
Ambiguous Grammar :- A Grammar is said to be Ambiguous, if there exists two or more derivation tree for a string w (That means two or more left most derivation trees) (16)

Example :- $G = (\{S\}, \{a+b, +, *\}, P, S)$ where P consists of $S \to S+S \mid S*S \mid a \mid b$. The string $a+a*b$ can be generated.

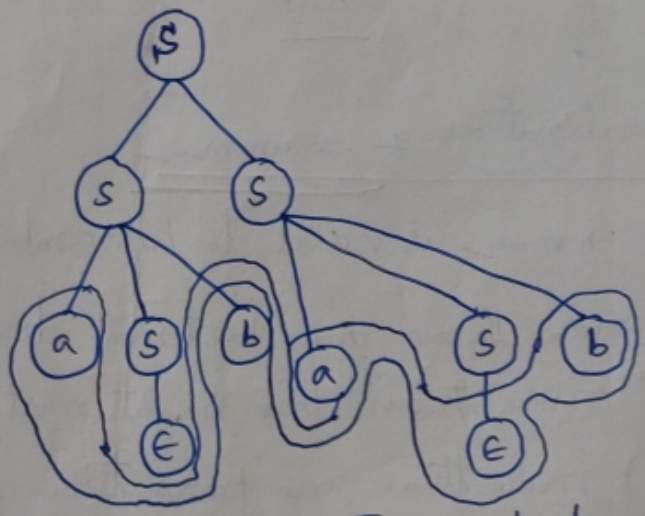
$S \to S+S$
$\to a+S$
$\to a+S*S$
$\to a+a*S$
$\to a+a*b$

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

2) $G = (\{S\}, \{a,b\}, P, S \to aSb \mid bSa \mid SS \mid \epsilon)$ The string abab Can be generated. Show the grammar is ambiguous by Constructing parse tree



$T_1 = abab$



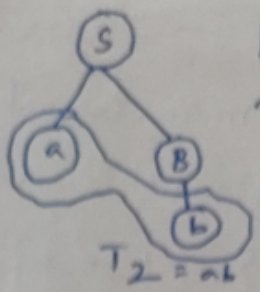
$T_2 = abab$

$\therefore$ it is ambiguous

3) prove that the grammar $S \to aB \mid ab$ $A \to aAB \mid a$, $B \to Abb \mid b$ is ~~ots~~ ambiguous grammar.

The string to be Considered in the grammar ab

T_1 : ab

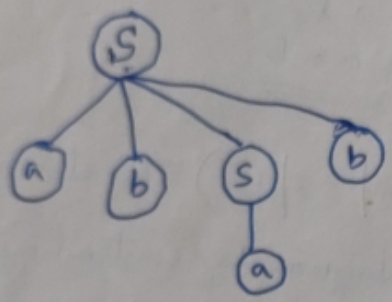


T_2 : ab

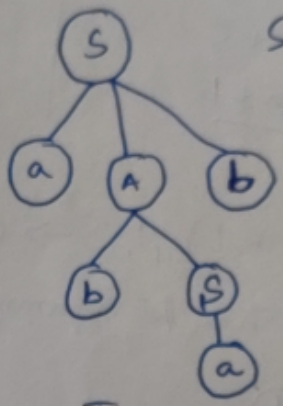
For string 'ab' There are more than one distinct derivation trees. Therefore it is ambiguous.

4) prove that the grammar $S \rightarrow a \mid aAb \mid abSb$, $A \rightarrow aAAb \mid bS$ is ambiguous.

The string to be considered is "abab"



T_1 : abab



T_2 : abab

Since two parse trees are obtained from the grammar. Hence the grammar is ambiguous.

Generalization of grammar

A grammar is said to be ambiguous if there exists

i) more than one leftmost derivation or

ii) more than one rightmost derivation or

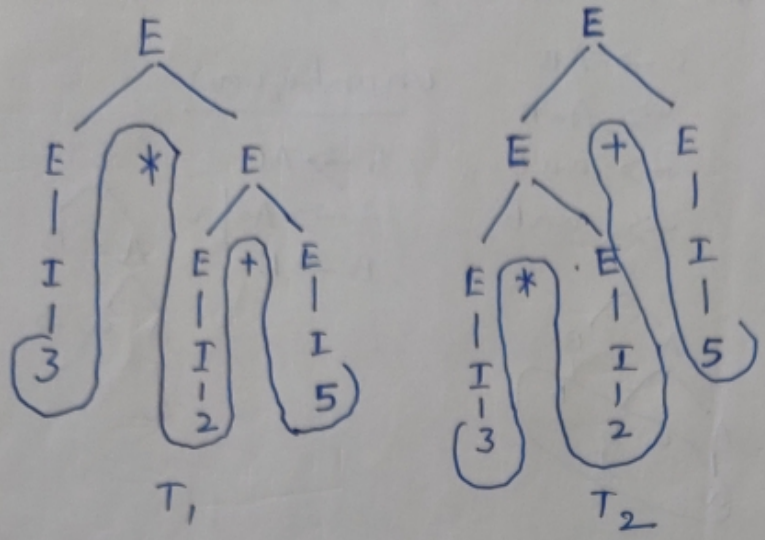
iii) more than one parse tree for given input string.

* If the grammar is not ambiguous, then we call unambiguous.

* If the grammar has ambiguity, then it is not good for Compiler Construction.

* No method can automatically detect & remove the ambiguity. But we can remove ambiguity by re-writing the grammar

1) Let us consider a grammar with production rule
$E \to I$, $E \to E+E$, $E \to E*E$, $E \to (E)$, $I \to \epsilon|0|1|2 \cdots 9$.
The string 3*2+5 can be generated, by constructing the parse tree. (17)



2) Consider the following grammar & find the left most derivation, rightmost derivation and parse tree for the string $+*-xyxy$.

$E \to +EE \mid *EE \mid -EE \mid x \mid y$

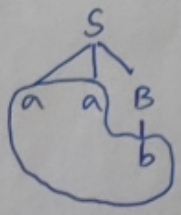
LMD

$E \to +EE$
$\to +*EEE$
$\to +*-EEEE$
$\to +*-xEEE$
$\to +*-xyEE$
$\to +*-xyxE$
$\to +*-xyxy$

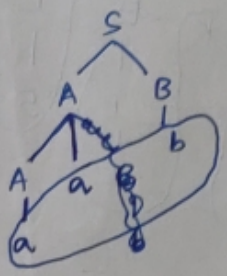
$E \to +EE$
$\to +Ey$
$\to +*EEy$
$\to +*Exy$
$\to +*-EExy$
$\to +*-Eyxy$
$\to +*-xyxy$

3) Consider the grammar $S \to AB \mid aaB$, $A \to a \mid Aa$, $B \to b$. Generate the string aab from the grammar. Check the grammar is ambiguous? Construct an unambiguous grammar equivalent to G.

$S \to aaB$
$\to aab$

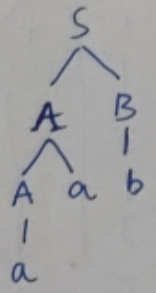


$S \to AB$
$\to AaB$
$\to aaB$
$\to aab.$



Unambiguous

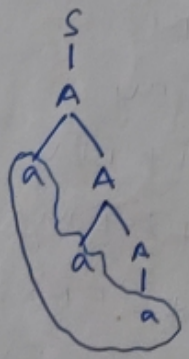
$S \to AB$
$A \to Aa \mid a$
$B \to b$

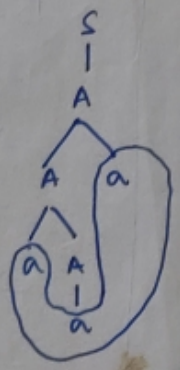


4) Consider the grammar $A \to aA \mid Aa \mid a$, $S \to A$. Generate the string aaa from the grammar. Check the grammar is ambiguous?

$S \to A$
$\to aA$
$\to aaA$
$\to aaa$

$S \to A$
$\to Aa$
$\to aAa$
$\to aaa$

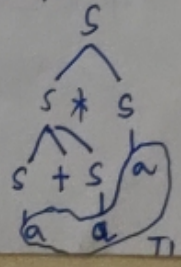


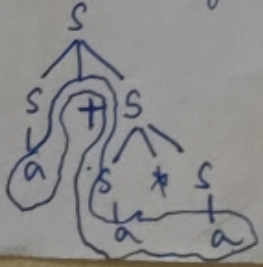


5) Consider the grammar $S \to S+S \mid S*S \mid a$. Generate the string a+a*a. & Convert the grammar into unambiguous.

$S \to S*S$
$\to S+S*S$
$\to a+S*S$
$\to a+a*a$

$S \to S+S$
$\to a+S$
$\to a+S*S$
$\to a+a*S$
$\to a+a*a$



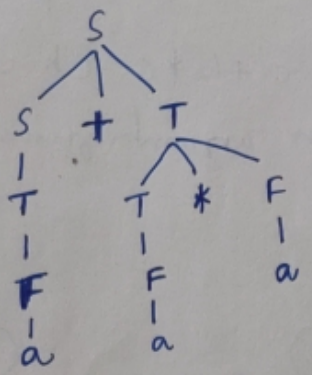


Note !: Associativity & precedence problem (level). Use left-associativity, so that the tree grows in one direction. or use right associativity, so that the tree grows in one direction.

$S \rightarrow S + T \mid T$

$T \rightarrow T * F \mid F$

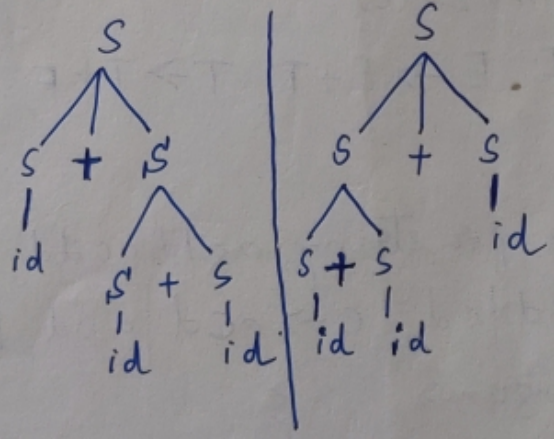
$F \rightarrow a$



Since T_1 parse tree is logically incorrect, Since + operator has more precedence than * operator.

6) Consider the grammar $S \rightarrow S+S \mid id$. Determine whether the grammar is ambiguous, if it is ambiguous, Construct an unambiguous grammar Equivalent to G

consider the string id+id+id.

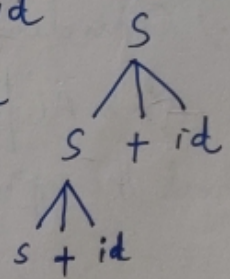


the grammar is ambiguous.

Unambiguous grammar

$S \rightarrow S + id$

$S \rightarrow id$



7) Consider the grammar $S \rightarrow aSbS \mid bSaS \mid \epsilon$ ~~$S \rightarrow AB \mid aaB$~~, ~~$A \rightarrow a$~~ Check the grammar is ambiguous.

$S \rightarrow aSbS$

$\rightarrow a\epsilon bS$

$\rightarrow a\epsilon b\epsilon$

$\rightarrow ab$

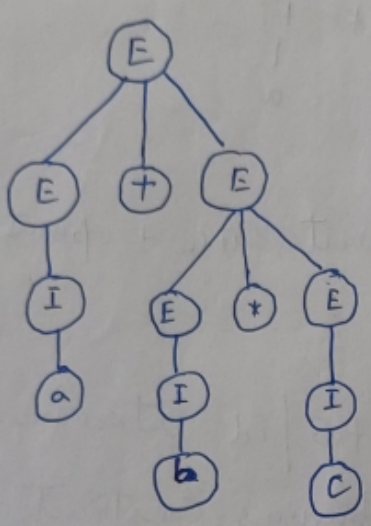
$S \rightarrow bSaS$

$\rightarrow b\epsilon aS$

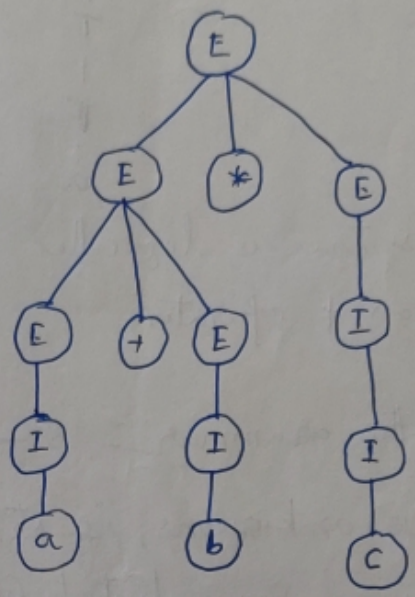
$\rightarrow b a\epsilon$

$\rightarrow ba$ //

7) Consider the grammar $G=(V, T, E, P)$ with $V=\{E, I\}$, $T=\{a, b, c, +, *, (,)\}$ and productions $E \to I$, $E \to E+E$, $E \to E*E$, $E \to (E)$ $I \to a|b|c$. Construct derivation tree for the string $a+b*c$. Check the grammar is ambiguous. Construct an unambiguous grammar equivalent to G



$T_1 = a+b*c$



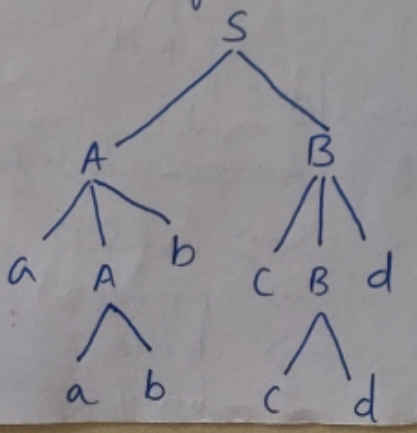
$T_2 = a+b*c$

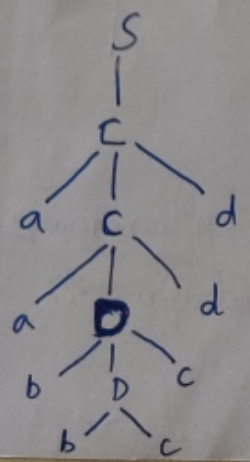
Now unambiguous grammar is

$E \to T$ $T \to F$, $F \to I$ $E \to E+T$, $T \to T*F$, $F \to (E)$

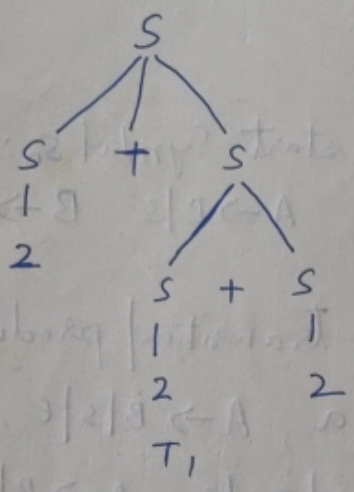
$I \to a|b|c$

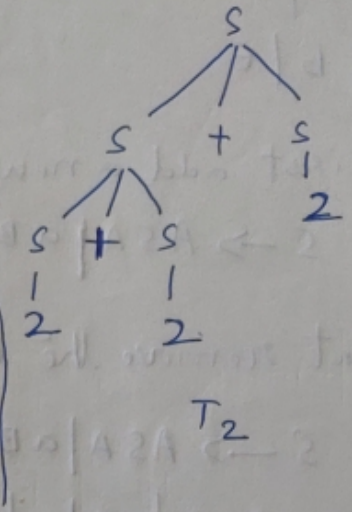
8) Consider the grammar for string aabbccdd $S \to AB|C$ $A \to aAb|ab$, $B \to cBd|cd$, $C \to acd|aDd$ $D \to bDc|bc$ Check the grammar is ambiguous.





9) The grammar $S \to S+S \mid 2$. Check the grammar has ambiguity? if it is ambiguous, construct an unambiguous grammar Equivalent to G.



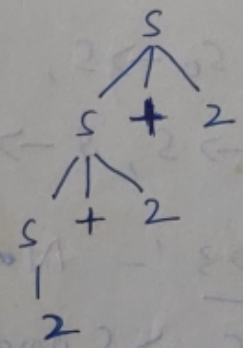


The grammar is ambiguous.

The unambiguous grammar will be

$S \to S+2$

$S \to 2$



Removing ambiguity

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

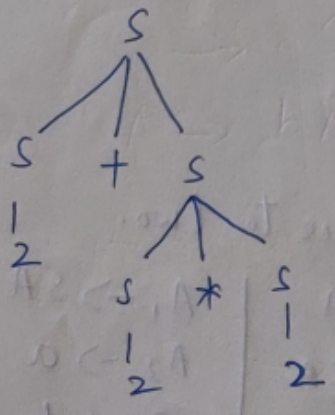
If the grammar has right associative operator $(\uparrow)$, Then induce the right recursion.

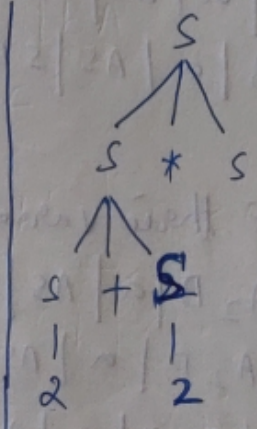
10) $S \to S+S \mid S*S \mid 2$

$S \to S+T \mid T$

$T \to T*F \mid F$

$F \to 2$





1) Convert the following grammar into CNF.

$S \to ASA \mid aB$

$A \to B \mid S$

$B \to b \mid \epsilon$

Step 1 :- First add a new start Symbol S_0:

$S_0 \to S,\ S \to ASA \mid aB,\ A \to B \mid S,\ B \to b \mid \epsilon$

Step 2 :- Next remove the ϵ-transition production:

$S_0 \to S,\ S \to ASA \mid aB \mid a,\ A \to B \mid S \mid \epsilon,\ B \to b$

$S_0 \to S,\ S \to ASA \mid aB \mid a \mid AS \mid SA \mid S,\ A \to B \mid S,\ B \to b$

Step 3 :- Next remove unit production, starting with $S_0 \to S$ and $S \to S$: $A \to B$, $A \to S$

$S_0 \to ASA \mid aB \mid a \mid AS \mid SA$

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

$A \to b \mid S$

$B \to b$

$S_0 \to ASA \mid aB \mid a \mid AS \mid SA$

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

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

$B \to b$

Step 4 :- Converting the three variables to two.

$S_0 \to AA_1 \mid A_2B \mid a \mid AS \mid SA$

$S \to AA_1 \mid A_2B \mid a \mid AS \mid SA$

$A \to b \mid AA_1 \mid A_2B \mid a \mid AS \mid SA$

$A_1 \to SA$

$A_2 \to a$

$B \to b.$