Chomsky Normal Form (CNF) :-

A Context Free Grammar $G = (V, \Sigma, R, S)$ is said to be in CNF if all the Productions are of the form $A \rightarrow BC$ or $A \rightarrow a$ where A, B, C are Variables (Non-Terminal Symbols) and 'a' is a Terminal Symbol.

Example:- The Grammar

$$A \rightarrow BC \mid AC$$
$$B \rightarrow AB \mid a$$
$$C \rightarrow CA \mid CB$$

is in CNF.

If the given grammar is in Chomsky Normal Form (CNF), We Say the grammer is in Simplified form.

Problem: Convert the following Grammar into CNF (Chomsky Normal Form)

$$S \rightarrow ABa$$
$$A \rightarrow aab$$
$$B \rightarrow Ac$$

Solution: terminal Symbols are : a, b, c

Non-Terminal symbols are : S, A, B
(Variables)

step(1):- Convert terminal symbols present in production body (RHS) into Variables.

Production (Rule) $S \rightarrow ABa$ is replaced by $S \rightarrow ABC_1$ and $C_1 \rightarrow a$.

Production $A \rightarrow aab$ is replaced by $A \rightarrow C_1 C_1 C_2$ and $C_2 \rightarrow b$

Production $B \rightarrow Ac$ is replaced by $B \rightarrow AC_3$ and $C_3 \rightarrow c$

Step(ii):- Break those Productions whose production body Contains 3 or more Variables

Production $S \rightarrow A\overbrace{BC_1}^{C_4}$ is divided into $S \rightarrow AC_4$ and $C_4 \rightarrow BC_1$

Production $A \rightarrow C_1 \overbrace{C_1 C_2}^{C_5}$ is divided into $A \rightarrow C_1 C_5$ and $C_5 \rightarrow C_1 C_2$

Resultant Grammar in Chomsky Normal Form (CNF) is:

$$\boxed{\begin{array}{l} S \rightarrow AC_4 \\ A \rightarrow C_1 C_5 \\ B \rightarrow AC_3 \\ C_1 \rightarrow a \\ C_2 \rightarrow b \\ C_3 \rightarrow c \\ C_4 \rightarrow BC_1 \\ C_5 \rightarrow C_1 C_2 \end{array}}$$

(i) Epsilon Productions (ϵ-Productions) :-

In a Content Free Grammar (CFG), Productions of the form $A \to \epsilon$ where A is a Variable is called ϵ-Productions

If Right Hand Side of any Production Contains only epsilon (ϵ) and nothing else, such productions are called epsilon (ϵ)- productions

(ii) Unit Productions :- In a CFG, Productions of the form $A \to B$ where A and B are Non-Terminal Symbols (variables) are Called Unit Productions.

If Right Hand Side of any Production Contains Single Variable, Such Productions are Called Unit Productions

(iii) Useless Productions :- In a CFG, the productions which are not used in the derivation of any string w are Called Useless Productions.

Note: Useless Productions Contain Useless Symbols.

Procedure of Simplifying the given Grammar

(i). Eliminate epsilon (ϵ- Productions)
(ii) Eliminate Unit Productions
(iii) Eliminate Useless Symbols & Productions
(iv) Convert the grammar obtained in step No. 3 into Chomsky Normal Form.

Context Free Languages (CFL)

A Language L is said to be Context Free if there exist either Context Free Grammar (CFG) G or Push Down Automata (PDA) P such that $L(G) = L$ or $L(P) = L$

examples! Following Languages are Context Free Languages

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

i.e. $L = \{ab, aabb, aaabbb, \ldots\ldots\}$

(ii) $L = \{WW^R \mid w \in \{a, b\}^*\}$

i.e $L = \{\underbrace{abba}_{w}\underbrace{abba}_{W^R}, \underbrace{bbab}_{w}\underbrace{babb}_{W^R}, \ldots\ldots\}$

Every Context Free Language (CFL) has either CFG or PDA. For example, the Language $L = \{a^n b^n \mid n \geq 1\}$ is a CFL. Since there exist a CFG (Grammar) $S \to ab \mid aSb$ such that $L(G) = L$.

* Pumping theorem for Context Free Languages (CFL) can be used to show that certain Languages are not Context Free Languages.
* Pumping Theorem (Lemma) for CFL specifies certain properties that any CFL must have. If the given Language fails to satisfy any one property / Condition, then we say the given language is not Context Free.

Pumping theorem says that every CFL have following properties

(i) Language L must contain long string z such that $|z| \geq n$ where n is some constant that is length of string z must be atleast n.

(ii) string z must be breakable into five substrings u, v, w, x & y satisfying the constraints: (i) $|vwx| \leq n$ that is length of middle string vwx must be $\leq n$.

(ii) $vx \neq \epsilon$ (epsilon)
Since substrings v and x are to be pumped (repeated or deleted) they must not be empty.
(iii) for all $i \geq 0$, uv^iwx^iy must also be present in the given Language.
That is two strings v and x are "pumped" any number of times and the resulting string is also present in the Language.

Example: show that the Language $L = \{0^n 1^n 2^n \mid n \geq 1\}$ is not Context Free Language.

Solution: $L = \{\overset{n=1}{012}, \overset{n=2}{001122}, \overset{n=3}{000111222}, \ldots\}$
Put the above language into Pumping theorem test. that is Verify whether above Language satisfies all the Conditions of Pumping theorem.

* Choose long string z such that $|z| \geq n$.
  Let $z = 0^n 1^n 2^n$
* Break string z into five substrings uvwxy satisfying the Constraints
  (i) $|vwx| \leq n$ (ii) $vx \neq \epsilon$.

Break string $z = 0^n 1^n 2^n$ into

$$z = \underbrace{0^i}_{u} \underbrace{0^j}_{v} \underbrace{0^k}_{w} \underbrace{0^l}_{x} \underbrace{1^n 2^n}_{y}$$

substrings:

$u = 0^i$
$v = 0^j$
$w = 0^k$
$x = 0^l$
$y = 1^n 2^n$

Condition (i): $|vwx| \le n$

$|0^j 0^k 0^l| < n$ since

$|0^i 0^j 0^k 0^l| = n$

Condition (ii): both string v and x must not be empty

i.e $vx \neq \epsilon$

substring $v = 0^j \neq \epsilon$ (not empty)
substring $x = 0^l \neq \epsilon$ (not empty)

Condition (iii) For all $i \ge 0$, string uv^iwx^iy must also be present in Language (L)

$i=0$ $uv^iwx^iy = uv^0wx^0y$

$= uwy$

$= 0^i 0^k 1^n 2^n = 0^{i+k} 1^n 2^n$

$i+j+k+l = n$
$i+k = n-(j+l)$

$= 0^{n-(j+l)} 1^n 2^n$

This resulting string (after pumping) is not present in L since (j+l) number of 0's are removed from 0^n as a result of Pumping.

Therefore for $i=0$, uv^iwx^iy is not in L.

$\therefore$ Language L is not Context Free

Pumping theorem for Context Free Languages (CFL) :-

Statement: Let L be a CFL. Let z be a long string in L such that $|z| \geq n$, where n is some constant. String z can be partitioned into $z = uvwxy$ such that $|vwx| \leq n$ and $|vx| \geq n$. For all $i \geq 0$, uv^iwx^iy must also present in Language.

Proof: Since L is a Context Free Language, there exist a Grammar (CFG) $G = (V, \Sigma, R, S)$ which can be simplified into Chomsky Normal Form (CNF). Let z be the long string present in Language L such that $|z| \geq n$ (length of string $z \geq n$).

Since string z is present in the Language, the grammar for L produces parse tree whose yield is string z shown in Fig (1).

Parse Tree:

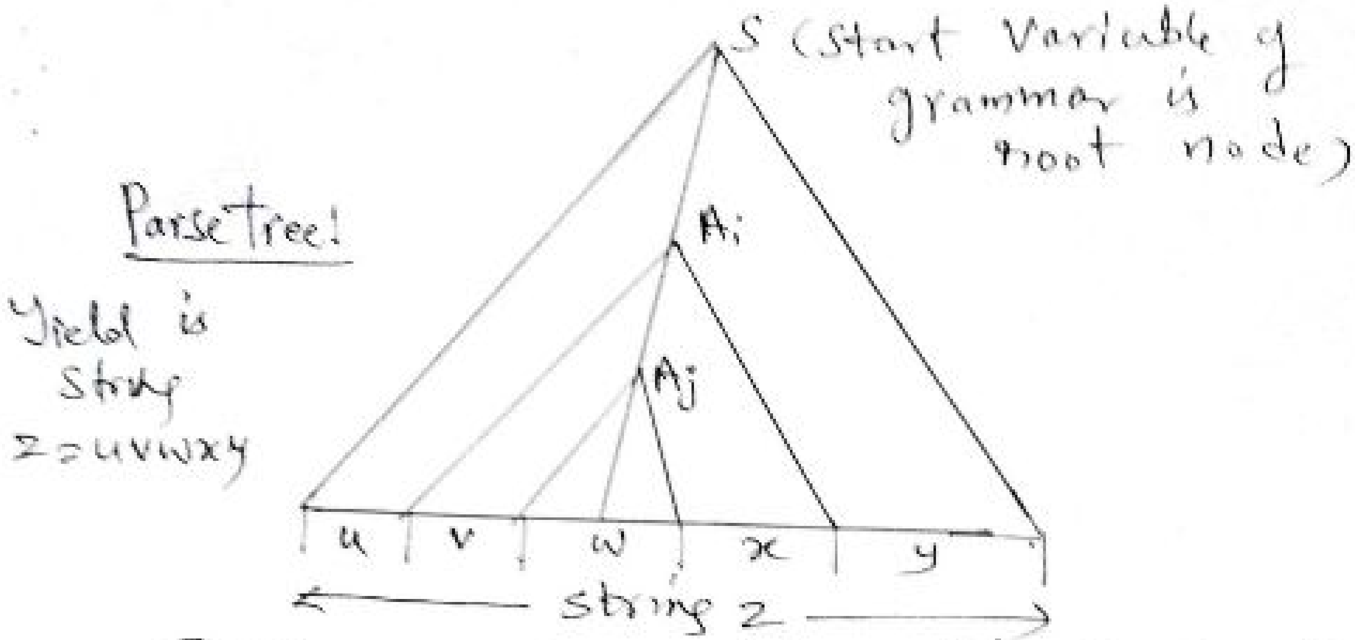


Fig (1) :- Dividing string into z = uvwxy

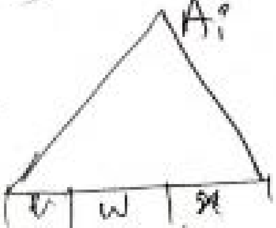
Parse tree show that string z is breakable into z = uvwxy (5 substrings)

We need to show that for all $i \geq 0$ strings uv^iwx^iy are also present in the Language, that is Prove that grammar produces parse trees whose yields are uv^iwx^iy for all $i \geq 0$.

$\underline{i=0}$ $\quad uv^iwx^iy = uv^0wx^0y$

$= uwy$ must be in L.

that is show that there is a parse tree whose yield is uwy, Substitute (replace) subtree 

in place of subtree 

as in fig(1) produce following subtree

Parse tree:

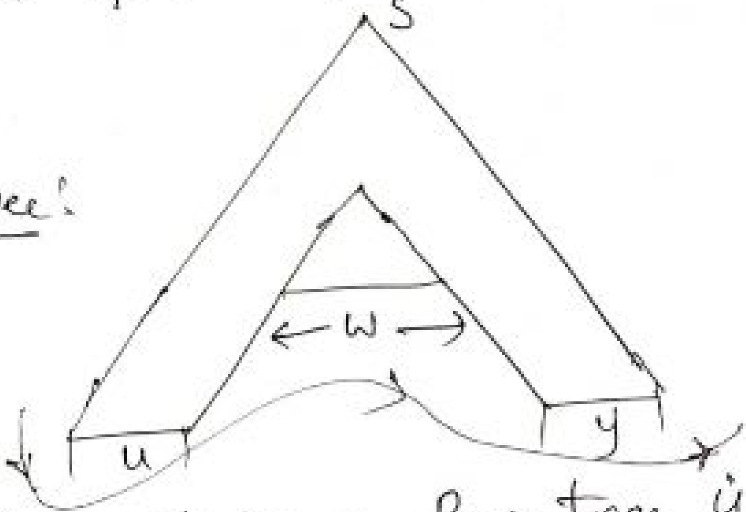
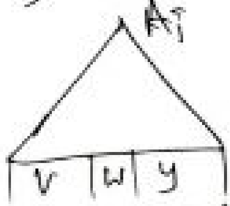
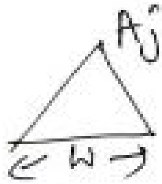


Fig (ii) Yield of Parse tree is string uwy.

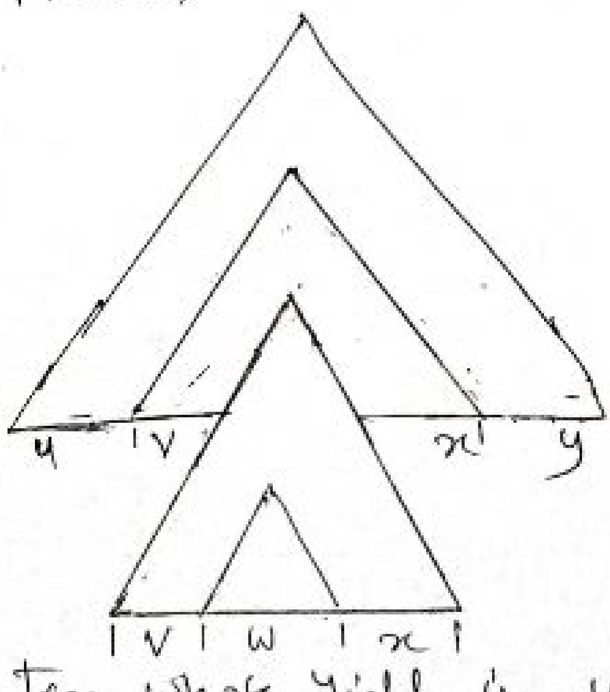
Since Parse tree exist whose Yield is z,

for $i=0$ $uv^0wx^0y = uwy$ is also in L.

Now Substitute subtree 

in place of subtree 

Produces the following Subtree.



Fig(ii): Parse tree whose Yield is $uvvwxxy = uv^2wx^2y$

Since Parse tree is generated whose yield is uv^2wx^2y, for $i=2$ string uv^2wx^2y is also present in Language L.

Similary we can generate Parse Trees for cases $i=3, 4, 5, \ldots$

Since grammar produces Parse Trees for all $i \geq 0$, the given Language is Context Free

∴ thus Pumping theorem is proved.

Closure Properties of Context Free Languages:-

(i) If L_1 and L_2 are Context Free Languages, then $L_1 \cup L_2$ is also a Context Free Language.

Here $L_1 \cup L_2$ is a new Language formed by taking Union of Languages L_1 and L_2

(ii) If L_1 and L_2 are Context Free Languages then $L_1 \cdot L_2$ is also a Context Free Language.

Here $L_1 \cdot L_2$ is a new Language formed by Concatinating Language L_1 with L_2

(iii) If L_1 is a Context Free Language, then L_1^* is also a Context Free Language

Here L_1^* is a new Language obtained after performing closure operation (*) on Language L_1.