Rohit Bale 181 P010 55
TE COMPS.

Q.3) b) A context free grammer $G = [V, T, P, S]$ is said to be in CNF is if every production is of the form:-

$$A \rightarrow a\alpha,$$

where, $a \in T$ is a terminal and α is a string of zero or more variables.

The language $L(G)$ should be without ϵ.

A) Removing left Recursion

Elimination of left recursion is an important step in algorithm used in conversion of a CFG into CNF form.

Left recursive grammar:-

A production of $A \rightarrow A\alpha$ is called left recursive as the left hand side variable appears as the first symbol on the right hand side.

*Language generated by left recursive grammer.

$A \rightarrow A\alpha$ ... left recursive
$A \rightarrow \beta$ ... for termination of recursion.

The language generated by above production is:-

$A \rightarrow A\alpha$ ... (from production $A \rightarrow A\alpha$)
$\rightarrow A\alpha\alpha$ ... (from production $A \rightarrow A\alpha$)
$\rightarrow A\alpha\alpha\alpha$ "
⋮
$\rightarrow A\alpha^n$ "
$\rightarrow \beta\alpha^n$ (from production $A \rightarrow \beta$)

Right recursive grammer for $\beta\alpha^n$:-

$$A \rightarrow \beta B \mid \beta$$ --(where, B generates a string α^n, production $A \rightarrow \beta$ is for termination of recursion)

$$B \rightarrow \alpha B \mid \alpha$$

Thus a left recursive grammer

$$A \rightarrow A\alpha \mid \beta$$

can be written as.

$$A \rightarrow \beta B \mid \beta$$
$$B \rightarrow \alpha B \mid \alpha$$

~~Algorithm for conversion from CFG. to CNF.~~