UNIT = 3

content free Grammar (CFG:):

CFG is a 4 tuple $G = (V, \Sigma, S, P)$, where

V is ~~disjoint~~ a Finite set of Variables or nonterminals

Σ is a Finite set of Terminals.

S is a starting Variable.

P is a Finite set of formula or Production rules of the form $A \to \alpha$ where $A \in V$, $\alpha \in (V \cup T)^*$

ex. construct a CFG. generating $L = a^n b^n \; \forall n \geq$

Let $G = \{V, \Sigma, P, S\}$

where Production rules P as

$$S \to ASB / \varepsilon$$
$$A \to a$$
$$B \to b$$

$V = \{S, A, B\}$

$\Sigma = \{a, b\}$

$S = \{S\}$

Language of CFG: set of all string which will be generating by Production rule of CFG is called Language of CFG.

$$L(G) = \{w : \text{where } S \xrightarrow{*} w, \; w \in T^*\}$$

Conventions used in grammar:

(i) Lower case letter (a, b, ... z), digits, operator, Parenthisis are used as terminals of Grammar

(ii) Upper case letter (A, B, ... Z) are use as Variable of Grammar.

(iii) S is used as start Variable.

(IV) u, v, w, x, y, z are used as string of Terminals.

(V) α, β, γ are used as string of Variables & Terminals.

Prob: write a CFG for $L = \{wcw^R : w \in (a,b)$

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

$P =$:

$S \to aSa \,/\, bSb$

$S \to c$

$V = \{ S \}$

$T = \{ a, b, c, \}$

$S = \{ S \}$

check that abbcbba can be drived from the given CFG.

$S \to aSa$

$\to abSba \to abbSbba$

$\to abbcbba$

a write a CFG for the R.E. $r = 0^* 1 (0+1)^*$

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

where P:

$S \rightarrow A1B$

$A \rightarrow 0A / \varepsilon$.

$B \rightarrow 0B / 1B / \varepsilon$

$V = \{S, A, B\}$

$T = \{0, 1\}$

$S = \{S\}$

Let us see the derivation of the string 00101

$S \rightarrow A1B$

$\rightarrow 0A10B$

$\rightarrow 00A101B$

$\rightarrow 00101$

~~Q~~

<u>Parse tree / Derivation tree</u>:

it is a rooted tree used to represent string of the grammar with following properties.

(i) Root of the tree will always the start variable of the grammar.

(ii) All internal nodes are variable of the grammar.

(iii) All leaf nodes are terminals of the grammar.

(IV) ε may also be leaf node but it should not have any sibling.

(V) string represented by parse tree can be obtained from reading leaf node left to right.

Left most Derivations & Right most Derivation:

in order to restrict to no. of choices of Replacement of variables, if at each step we replace the left most variable by one of its Production such a derivation is called a LMD.

Similarly it is possible that at each step the right most variable is replaced by one of its Production such a derivation is called a RMD.

Q write a CFG for $L = \{a^n b^m \ \forall n, m \geq 0\}$

$S \to AB$
$A \to aA / \varepsilon$
$B \to bB / \varepsilon$

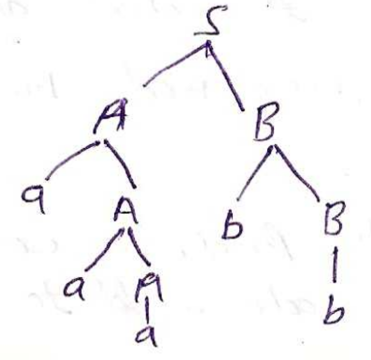
LMD for aaabb

$S \to AB$
$\to aAB$
$\to aaAB$
$\to aaaB$
$\to aaabB$
$\to aaabb$

RMD for aaabb

$S \to AB$
$A \to AbB$
$A \to Abb$
$A \to aAbb$
$A \to aaAbb$
$A \to aaabb$

Parse Tree:



Write the CFG for language $L = \{a^n b^m c^{2m} : n, m >=$

$S \rightarrow AB$

$A \rightarrow aA / \varepsilon$

$B \rightarrow bBcc / \varepsilon$

simplification of CFG:

CFG may contain different types of useless symbols, unit Productions & null Productions, These type of symbols & Productions increase the number of steps in generating a language from a CFG. Reduced grammar contains less no. of non terminals & Productions, so the time complexity for ~~CFG~~ the language generating Process becomes less from the reduced grammar.

There are 3 Processes.

(i) Removal of useless symbols.
(ii) Removal of unit Productions.
(iii) Removal of null Productions.

(i) Removal of useless symbols

two types of useless symbols.

(a) Non generating symbols: which do not Produce any terminal string

(b) Non Reachable symbols: which can not be Reached from starting symbols.

ex. Remove useless symbols from the given CFG.

$S \rightarrow AC/BA$
$C \rightarrow CB/AC$
$A \rightarrow a$
$B \rightarrow aC/b$

here c is nongenerating symbols as it doesnot Produce any terminal string.

So we have to remove C. To remove C, all the Productions containing C as a symbol. (LHS & RHS)

$S \to BA$
$A \to a$
$B \to b$

in This grammar no nonreachable symbols.
So the grammar is without useless symbols.

Q Remove the useless symbols from the given CFG.

$S \to aB / bX$
$A \to BAd / bSX / a$
$B \to aSB / bBX$
$X \to SBd / aBx / ad$

B is non generating symbols. so remove it

~~S → aB~~ $S \to bX$
$A \to bSX / a$
$X \to ad$

here A is non reachable so remove it from LHS ~~from (LHS & RHS~~

$S \to bX$
$X \to ad$

(II) Removal of unit Productions:

if in R.H.S. is a only single symbol that is variable (non terminal) is called unit Production.

For remove unit Production, Replace RHS variable by its Productions.

ex. Remove unit Production From CFG

$S \rightarrow ABC / AB / B$
$A \rightarrow 0A / 0$
$B \rightarrow 1B / 1$
$C \rightarrow 01C / 01$

Sol.. Here $S \rightarrow B$ is unit Production so in this Production B replaced by its Production $\{B \rightarrow 1B/1\}$

so grammar without unit Production is

$S \rightarrow ABC / AB / 1B / 1$
$A \rightarrow 0A / 0$
$B \rightarrow 1B / 1$
$C \rightarrow 01C / 01$

Q. Remove unit Production

$S \rightarrow A / B$
$A \rightarrow 0A / 0 / C$
$B \rightarrow 1B / 1 / D$
$C \rightarrow 0$
$D \rightarrow 1$

Sol. here

$S \to A,\ S \to B,\ A \to C,\ B \to D$ are unit Production

so remove its one by one

remove $S \to A,\ S \to B$

$S \to 0A/0/C/1B/1/D$

Now $S \to C,\ S \to D$ is also unit Production

so

$S \to 0A/0/0/1B/1/1$

here $S \to 0,\ S \to 1$ repeat two times so remove one time.

$S \to 0A/0/1B/1$

Now remove $A \to C,\ B \to D$

$S \to 0A/0/1B/1$

$A \to 0A/0/0$

$B \to 1B/1/1$

$C \to 0$

$D \to 1$

here $A \to 0,\ B \to 1$ repeat two times so Remove one time.

So Grammar without unit Production is

$S \to 0A/0/1B/1$

$A \to 0A/0$

$B \to 1B/1$

$C \to 0$

$D \to 1$

Q. $S \to A$
$A \to S / 0$ } remove unit Production

$S \to S / 0$ here $S \to S$, $A \to A$ is meaningless
$A \to A / 0$

so $S \to 0$
$A \to 0$ } CFG without unit Production.

(iii) Removal of null Production:

for remove $A \to \varepsilon$, Replace A by ε in all Productions.

Note: Null Production Permanently remove only when CFG Language not contain ε.

ex. $S \to AB$
$A \to 0A / \varepsilon$
$B \to 1B / \varepsilon$ } ε cant remove b/c L contain ε.

Q $S \to 0A1B$
$A \to 0A / \varepsilon$
$B \to 1B / \varepsilon$

remove $A \to \varepsilon$

$S \to 0A1B / 01B$
$A \to 0A$
$B \to 1B / \varepsilon$

remove $B \to \varepsilon$

$S \to 0A1B / 01B / 0A1 / 01$
$A \to 0A$
$B \to 1B$

Note: For simplification of any CFG apply these Rule in sequence of

(i) Remove null Productions.

(ii) Remove unit Productions.

(iii) Remove useless symbols.

Q. Simplified the CFG

$S \to AADE / ACD$

$A \to aAb / \varepsilon$

$C \to aC$

$D \to aDa / bDb / \varepsilon$

$E \to F / ab$

$F \to b$

Firstly remove Null Production here $A \to \varepsilon$, $D \to \varepsilon$ remo

so $S \to AADE / ACD / ADE / DE / CD / AAE / AC / AE / E$

$A \to aAb / ab$

$C \to aC$

$D \to aDa / bDb / aa / bb$

$E \to F / ab$

$F \to b$

Now remove unit Production, remove $S \to E$, $E \to F$

$S \to AADE / ACD / ADE / DE / CD / AAE / AC / AE / F / ab$

$A \to aAb / ab$

$C \to aC$

$D \to aDa / bDb / aa / bb$

$E \to b / ab$

$F \to b$

$S \rightarrow F$ is also unit Production

so $S \rightarrow AADE / ACD / ADE / DE / CD / AAE / AC / AE / b / ab$

$A \rightarrow aAb / ab$

$C \rightarrow aC$

$D \rightarrow aDa / bDb / aa / bb$

$E \rightarrow b / ab$

$F \rightarrow b$

Now Remove useless symbols.

here C is Non generating symbols and F is unreachable symbols. so remove them.

$S \rightarrow AADE / ADE / DE / AAE / AE / b / ab$

$A \rightarrow aAb / ab$

$\cancel{C \rightarrow aC}$

$D \rightarrow aDa / bDb / aa / bb$

$E \rightarrow b / ab$

This grammar is simplified grammar.

Normal form of the grammar:

(i) Chomsky Normal Form (CNF):

Any grammar that doesnot generate ϵ as a string can be written in the following formate.

$$A \rightarrow BC / a$$

it means in all the Production in the grammar R.H.S. will be either two variab[les] or single terminal.

Convert any CFG to CNF Form:

(i) if the given grammar is not simplified then firstly simplify the grammar.

(ii) Arrange that all RHS string of length 2 or more consist only of variables.

(iii) Break RHS string of length 3 or more into a cascade of Production, each with RHS string consisting of two Variables.

ex

$S \rightarrow AB$

$A \rightarrow 0A/0$

* $B \rightarrow 1B/1$

The given grammar is simplified grammar so CNF form of this grammar

$S \to CA / 0 / DB / 1$

$A \to CA / 0$

$B \to DB / 1$

$C \to 0$

$D \to 1$

* Prob. convert in CNF

$S \to ABCD$

$A \to aA / \varepsilon$

$B \to bB / \varepsilon$

$C \to cC / \varepsilon$

$D \to d$

Sol.

firstly simplify the grammar

remove null production

$S \to ABCD / BCD / ABD / BD / ACD / CD / AD / D$

$A \to aA / a$

$B \to bB / b$

$C \to cC / c$

$D \to d$

remove unit production

$S \to ABCD / BCD / ABD / BD / ACD / CD / AD / d$

$A \to aA / a$

$B \to bB / b$

$C \to cC / c$

$D \to d$

No one here useless symbal so given grammar is simplified form.
Now convert in CNF

$S \to FE / BE / AE / FD / CD / AD / BD / d$

$F \to AB$

$E \to CD$

$A \to A'A / a$

$B \to B'B / b$

$A' \to a$

$B' \to b$

$C \to C'C / c$

$C' \to c$

$D \to d$

* Prob: ~~chag~~ change the fallowing grammar in CNF

$S \to 1A / 0B$

$A \to 1AA / 0S / 0$

$B \to 0BB / 1$

given grammar is simplified so CNF is

$S \to CA / DB$

$A \to CE / DS / 0$

$B \to DF / 1$

$C \to 1$

$D \to 0$

$E \to AA$

$F \to BB$

Theorem: Let $A \to \alpha_1 B \alpha_2$ and

$B \to \beta_1 / \beta_2 / \beta_3 \cdots / \beta_n$ are the production of the grammar then it can be written as

$$A \to \alpha_1 \beta_1 \alpha_2 / \alpha_1 \beta_2 \alpha_2 / \cdots \alpha_1 \beta_n \alpha_2$$

ex.

$A \to aB / aBb$

$B \to c/d$

$\Rightarrow A \to ac / ad / acb / adb$

Theorem: Let $A \to \overbrace{A\alpha_1 / A\alpha_2 / \cdots A\alpha_r}^{\text{Left recursive}} / \beta_1 / \beta_2 \cdots \beta_p$

{ β_i's do not start with A

then it can be witten as

Remove left recursion $\to$ { $A \to \beta_1 / \beta_2 / \cdots / \beta_p / \beta_1 B / \beta_2 B / \cdots \beta_p B$ and

$B \to \alpha_1 / \alpha_2 / \cdots / \alpha_r / \alpha_1 B / \alpha_2 B / \cdots \alpha_r B$

where B is a new variable.

ex. $A \to A\underset{\alpha_1}{\underline{b}} / A\underset{\alpha_2}{\underline{ab}} / \underset{\beta_1}{\underline{cd}} / \underset{\beta_2}{\underline{de}} / \underset{\beta_3}{\underline{f}}$

$\Rightarrow$

$A \to cd / de / f / cdB / deB / fB$

$B \to b / ab / bB / abB$

Greibach Normal Form (GNF)

In GNF, we put restrictions not on the length of the right side of a Production but on the positions in which terminals & variables can appear.

Any grammar that doesnot generate ε as a string can be written in the following formate.

$$A \rightarrow a\alpha \,/\, a$$

it means, in the RHS of any Production either single terminal followed by string of Variable or only single terminal.

ex

$A \rightarrow a \,/\, aB$

$B \rightarrow bD \,/\, bDA \,/\, b$

$D \rightarrow a \,/\, bAB$

Q. convert is in GNF

$A \rightarrow AAb / Ab / a$

The given grammar is simplified so

by using Leema(2)

$A \rightarrow a$
$A \rightarrow aB$
$B \rightarrow Ab / b$
$B \rightarrow AbB / bB$

by using Leema (i)

$A \rightarrow a / aB$
$B \rightarrow ab / aBb / b / abB / aBbB / bB$

Now equivelent GNF grammar is

$A \rightarrow a / aB$
$B \rightarrow aB' / aBB' / b / aB'B / aBB'B / bB$
$B' \rightarrow b$

Q convert into GNF form.

$S \rightarrow AB$
$A \rightarrow aA / bB / b$
$B \rightarrow b$

Sol. Given CFG is in simplified form so GNF is

$S \rightarrow bBB / aAB / bB$ by using Leema (1)
$A \rightarrow aA / bB / b$
$B \rightarrow b$

Now the Grammar is in GNF.

Q. $S \rightarrow absb / aa$

Sol. the given CFG is in simplified form so GNF is

$S \rightarrow aBSB / aA$
$B \rightarrow b$
$A \rightarrow a$

Now the CFG is in GNF.

Q $A \rightarrow ABb / AB / a$
$B \rightarrow Bc / b$

The given grammar is in simplified form so by using leema 2

$A \rightarrow a$
$A \rightarrow aA'$
$A' \rightarrow Bb / B$
$A' \rightarrow BbA' / BA'$
$B \rightarrow b / bB'$
$B' \rightarrow c / cB'$

by using Leema (1)

$A \rightarrow a / aA'$
$A' \rightarrow bb / bB'b / b / bB'$
$A' \rightarrow bbA' / bA' / bB'bA' / bB'A'$
$B \rightarrow b / bB'$
$B' \rightarrow c / cB'$

Now GNF is

$A \rightarrow a / aA'$
$A' \rightarrow bD / bB'D / b / bB'$
$A' \rightarrow bDA' / bA' / bB'DA' / bB'A'$
$D \rightarrow b$
$B \rightarrow b / bB'$
$B' \rightarrow c / cB'$

Q $A \rightarrow Ab / Aab / cd / de$

Sol. given CFG is in simplified farm so

by using Leema (2)

$A \rightarrow cd / de / cdA' / deA'$
$A' \rightarrow b / ab / bA' / abA'$

Now GNF is

$A \rightarrow cD / dE / cDA' / dEA'$
$A' \rightarrow b / aB / bA' / aBA'$
$D \rightarrow d$
$E \rightarrow e$
$B \rightarrow b$