

$$P = \{ S \rightarrow Aa \mid ab \mid b \mid ca \\ B \rightarrow ab \mid b \\ C \rightarrow Db \mid ab \mid d \\ D \rightarrow d \mid ab \\ E \rightarrow ab \}$$

~~mod 11~~ S is start symbol.

Normal Form For Context Free Language

Greibach Normal Form [GNF]

In GNF there is no restriction on the number of symbols on the right hand side but there is restriction on the terminals and variables appear on the right hand side of the production.

Definition: let $G = (V, T, P, S)$ be a CFG, the CFG G is said to be in GNF if all the productions are of the form

$$A \rightarrow a\alpha$$

where $a \in T$ and $\alpha \in V^*$ i.e. the first symbol on the right hand side of the production must be a terminal & it can be followed by zero or more variable

Simplification of grammar

Simplification of context free grammar takes the following major steps are.

Step 1: Removal of useless symbol (non terminal)

Step 2: Removal of unit production

Step 3: Removal of ϵ production.

Step 1: Removal of useless symbol:

A symbol is termed useless if it is not useful i.e. it not generate any string.

Step 2: Removal of unit production

* A production of the form $A \rightarrow B$ is called unit production.

* To remove unit production, for each non terminal A , we compute A^+ , the set of non-terminal derived from A via unit production.

Removal of ϵ production

A symbol A is said to be nullable if $A \rightarrow \epsilon$. To remove ϵ production, we compute nullable set.

TWO algorithms used to simplify CFG

- a) To find and remove unproductive
- b) To find and remove unreachable variable

① Grammer $S \rightarrow AB|AC$

$$A \rightarrow aAb|\epsilon$$

$$B \rightarrow aA$$

$$C \rightarrow bca$$

$$D \rightarrow AB$$

\rightarrow remove unproductive
 \rightarrow remove unreachable

$$\text{Initial UP} = \{S, A, B, C, D\}$$

$$\text{Initial P} = \{a, b\}$$

$$\rightarrow P = \{a, b, A\}$$

$$\rightarrow P = \{a, b, A, B\}$$

$$P = \{a, b, S, A, B\}$$

$$P = \{a, b, S, A, B, D\}$$

$$A \rightarrow aAb|\epsilon$$

$$B \rightarrow aA$$

$$S \rightarrow AB$$

$$D \rightarrow AB$$

$X \rightarrow \alpha$
↑
all symbol
in RHS is
productive

LHS also
as productive

C as unproductive

\therefore drop RHS or LHS with unproductive

$$S \rightarrow AB$$

$$A \rightarrow aAb|\epsilon$$

$$B \rightarrow aA$$

$$D \rightarrow AB$$

Remove unreachable

Initial Reachable = {S}

UR = {A, B, D} [mark every non-terminal symbol as unreachable]

Reachable = {S, A, B} $S \rightarrow AB$

D is unreachable, so

Remove D from both from LHS, RHS

After removing useless grammar symbols

$$\begin{cases} S \rightarrow AB \\ A \rightarrow aAb \\ B \rightarrow bA \end{cases}$$

$X \xrightarrow{\quad} \cancel{A} \beta$
 ↓
 reachable is yet not
 mark 'A' as reachable

$S \rightarrow abS \mid aba \mid abB$

A $\rightarrow cd$

B $\rightarrow aB$

C $\rightarrow dc$

→ Remove unproductive

→ Remove unreachable

Initial UP = {S, A, B, C}

initial P = {a, b, c, d}

loop $\rightarrow P = \{a, b, c, d, A\}$ A $\rightarrow cd$
 what are {
 gets added } P = {a, b, c, d, A, C} C $\rightarrow dc$

as productive P = {a, b, c, d, A, C, S} S $\rightarrow abS$
 S $\rightarrow abA$

B is unproductive we remove from RHS,

$$S \rightarrow abS \mid aba$$

$$A \rightarrow cd$$

$$B \rightarrow dc$$

Remove unreachable symbols.

$$R = \{S\} \cup R\{A, C\}$$

Loop: $R = \{S, A\}$

C is unreachable

lefthand reaches
right hand

$$\therefore S \rightarrow abS \mid aba$$

$$A \rightarrow cd$$

Ex:

$$S \rightarrow aab \mid abA \mid aaT$$

remove useless symbols

1. remove unproductive
2. remove unreachable

$$A \rightarrow AA$$

$$B \rightarrow ab \mid b$$

$$C \rightarrow ad$$

1. Remove unproductive

$$\text{Initial } P = \{a, b, d\}$$

$$\text{initial UP} = \{S, A, B, C, T\}$$

loop:

$$P = \{a, b, d, c\} \quad C \rightarrow ad$$

$$P = \{a, b, d, c, B\} \quad B \rightarrow ab$$

$$P = \{a, b, d, c, B, S\}$$

2 do -> 2

Add S -> 2

unproductive are A & T

$$\therefore S \rightarrow aaB$$

$$B \rightarrow ab/b$$

$$C \rightarrow ad$$

Remove unreachable

$$\text{Initial } R = \{S\}$$

$$R = \{S, B\}$$

$$\therefore UR = \{C\}, C \text{ is unreachable}$$

$$\therefore S \rightarrow aaB$$

$$\underline{B \rightarrow ab/b}$$

(4) Remove useless symbol / production

$$S \rightarrow AB/a$$

$$A \rightarrow BC/b$$

$$B \rightarrow aB/c$$

$$C \rightarrow ac/B$$

Remove unproductive

$$P = \{a, b\} \quad UP = \{S, A, B, C\}$$

$$P = \{a, b, S\} \quad S \rightarrow a$$

$$P = \{a, b, S, A\} \quad A \rightarrow B$$

$$\therefore \begin{array}{ll} S \not\rightarrow \cancel{a} \cancel{b} \not\rightarrow a & S \rightarrow a \\ \cancel{a} \not\rightarrow \cancel{b} \not\rightarrow b & A \rightarrow b \end{array}$$

2. Remove unreachable

$$R = \{S\} \quad UR =$$

$\therefore S \rightarrow a$ [A is unreachable]

5
 $S \rightarrow AB \mid AC$

$$A \rightarrow aAb \mid bAa \mid a$$

$$B \rightarrow bba \mid aab \mid AB$$

$$C \rightarrow abcA \mid adb$$

$$D \rightarrow bD \mid ac.$$

Remove unproductive

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

$$UP = \{S, A, B, C, D\}$$

$$P = \{a, b, A\} \quad A \rightarrow a$$

$$P = \{a, b, A, B\} \quad B \rightarrow bba$$

$$P = \{a, b, A, B, S\} \quad S \rightarrow AB$$

$$\therefore S \rightarrow AB$$

$$A \rightarrow aAb \mid bAa \mid a$$

$$B \rightarrow bba \mid aab \mid AB$$

\therefore Remove C & D unproductive remove from grammar

\Rightarrow Remove unreachable

$$R = \{S\} \quad UR = \{A, B\}$$

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

There are no unreachable symbols
∴ there is no change

$$S \rightarrow AB$$

$$A \rightarrow aAb \mid bAA \mid a$$

$$B \rightarrow bBA \mid aaB \mid AB$$

Remove ϵ production

Eliminating ϵ rules

\Rightarrow Nullable variable:

A variable x is nullable iff either

(i) There is a rule $x \rightarrow \epsilon$ or

(ii) There is a rule $x \rightarrow PQR$ and P, Q, R are all nullable.

So compute N , the set of nullable variable as follows.

1. Set N to the set of variable that satisfy (i).
2. until an entire pass is made without adding anything to N do evaluate all other variables with respect to (2)

If any variable satisfies (2) & is not in N, insert it.

modifiable variable: A rule β is modifiable iff it is of the form

$P \rightarrow Q\beta$ for some nullable Q .

(2)

$$\begin{aligned} S &\rightarrow aTa \\ T &\rightarrow ABG \\ A &\rightarrow aA | G \\ B &\rightarrow Bb | G \\ G &\rightarrow c | \epsilon \end{aligned}$$

Initial nullable $N = \{c\} \cup C \rightarrow \epsilon$

$$N = \{c, B, A\} \quad \begin{aligned} A &\rightarrow C \\ B &\rightarrow C \end{aligned}$$

$$N = \{c, B, A, T\} \quad T \rightarrow ABC$$

All nullable N^T

$$S \rightarrow aTa | aa$$

$$T \rightarrow ABC | BC | AC | AB | A | C | B \quad \begin{cases} \text{one at a time} \\ \& 2 \text{ at a time} \end{cases}$$

$$A \rightarrow aA | a | G$$

$$B \rightarrow Bb | b | G$$

$$C \rightarrow c$$

{ remove ϵ production }

$$\begin{aligned}
 ② \quad S &\rightarrow aA\bar{a} \\
 A &\rightarrow B/a \\
 B &\rightarrow c/c \\
 C &\rightarrow eC/e
 \end{aligned}$$

Remove ϵ production
first find nullable

$$N = \{C\} \quad C \rightarrow \epsilon$$

$$N = \{B, C\} \quad B \rightarrow C$$

$$N = \{A, B, C\} \quad A \rightarrow B, B \rightarrow C, C \rightarrow \epsilon$$

$$\therefore S \rightarrow aA\bar{a} | a\bar{a} | a\bar{a} | a\bar{a}$$

$$A \rightarrow B/a$$

$$B \rightarrow C/c$$

$$C \rightarrow eC/e$$

====

Removal of unit production

The unit productions are the productions in which one non-terminal gives another non-terminal.

$$\text{ex: } X \rightarrow Y \quad \& \quad S \rightarrow Y$$

Step 1: To remove $X \rightarrow Y$ add production $X \rightarrow a$ to grammar rule, $Y \rightarrow a$ occurs in the grammar whenever

Step 2: now delete $X \rightarrow Y$ from grammar,

Step 3: repeat step 1 & 2 until all production ~~reached~~ removed from grammar.

ex: $S \rightarrow 0A|1B|C$

$A \rightarrow 0S|00$

$B \rightarrow 1|A$

$C \rightarrow 01$

$S \rightarrow 0A \quad A \rightarrow 0S \quad B \rightarrow 1 \quad C \rightarrow 01$

$S \rightarrow 1B \quad A \rightarrow 00 \quad B \rightarrow A$ — unit production

$\boxed{S \rightarrow C}$

$\cancel{S \rightarrow A}$

$S \rightarrow 0A|1B|01$

$A \rightarrow 0S|00$

$B \rightarrow 1|0S|00$

$C \rightarrow 01 //$

Chomsky Normal Form [CNF]

CNF stands for Chomsky normal form. A CFG is in CNF if all production rules satisfy one of the following conditions.

⇒ Start symbol generating ϵ

eg: $A \rightarrow \epsilon$

⇒ A non-terminal generating two non-terminal

ex: $S \rightarrow AB$

⇒ A non-terminal generating terminal

eg: $S \rightarrow a$

Removal of unit production

A unit production is a production of the form $A \rightarrow B$ where both A & B are variable.

1) Eliminate any unit production from the grammar,

$$S \rightarrow AB$$

$$A \rightarrow a$$

$$B \rightarrow c/b$$

$$C \rightarrow D$$

$$D \rightarrow E/bc$$

$$E \rightarrow d/Ab$$

Non unit production

$$S \rightarrow AB$$

$$A \rightarrow a$$

$$B \rightarrow b$$

$$D \rightarrow bc$$

Unit production

$$B \rightarrow c$$

$$C \rightarrow D$$

$$D \rightarrow E$$

It is clear from the dependency graph that $D \rightarrow E$, so the non unit productions generated from 'E' can also be generated from D .

$$E \rightarrow d/Ab$$

Since

$$D \rightarrow E$$

$$D \rightarrow d/Ab/bc$$

$$C \rightarrow D \rightarrow E$$

$$G \rightarrow d/Ab/bc$$

Since $B \rightarrow C$, $C \rightarrow D$, $D \rightarrow E$

$$B \rightarrow b/d/Ab/bc$$

The final grammar obtained after eliminating

$$V = \{ S, A, B, C, D, E \}$$

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

$$P = \{ S \rightarrow AB \}$$

$$A \rightarrow a$$

$$B \rightarrow b/d/Ab/bc$$

$$C \rightarrow bc/d/Ab$$

$$D \rightarrow bc/d/Ab$$

$$E \rightarrow d/Ab$$

S is the start symbol,

② Eliminate unit productions from the grammar.

$$S \rightarrow AO/B$$

$$B \rightarrow A/11$$

$$A \rightarrow 0/12/B$$

non unit production

$$S \rightarrow AO$$

$$B \rightarrow 11$$

$$A \rightarrow 0/12$$

unit production

$$S \rightarrow B$$

$$B \rightarrow A$$

$$A \rightarrow B$$

It is clear from the dependence graph that
 $S \rightarrow B$, $B \rightarrow A$, & $A \rightarrow B$, so the new production
for S , A & B are

$$S \rightarrow AO/11/0/12$$

$$B \rightarrow 0/12/11$$

$$A \rightarrow 0/12/11$$

The resulting grammar without unit production.

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

$$T = \{ 0, 1, 2 \}$$

$$P = \{ S \rightarrow A01110112 \}$$

$$B \rightarrow 0112111$$

$$A \rightarrow 0112111 \}$$

S is start symbol

③ Eliminate unit production

$$S \rightarrow Aa | B | Ca$$

$$B \rightarrow ab | b$$

$$C \rightarrow Db | D$$

$$D \rightarrow E | d$$

$$E \rightarrow ab$$

unit production

$$\left\{ \begin{array}{l} S \rightarrow B \\ C \rightarrow D \\ D \rightarrow E \end{array} \right.$$

non unit production

$$S \rightarrow Aa | Ca$$

$$B \rightarrow ab | b$$

$$C \rightarrow Db$$

$$D \rightarrow d$$

$$E \rightarrow ab$$

$$S \rightarrow B$$

$$S \rightarrow Aa | ab | b | Ca$$

$$D \rightarrow E$$

$$D \rightarrow ab | d$$

$$C \rightarrow D \quad D \rightarrow E$$

$$C \rightarrow Db | ab | d$$

The resulting grammar is

$$V = \{ S, A, B, C, D, E \}$$

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

$$P = \{ S \rightarrow aA | aB | b | ca$$

$$B \rightarrow ab | b$$

$$C \rightarrow Db | ab | d$$

$$D \rightarrow d | ab$$

$$E \rightarrow ab \}$$

~~mod 11~~
S is start symbol.

Normal Form for context free language

Normal Form for context free language

Greibach Normal Form [GNF]

In GNF there is no restriction on the number of symbols on the right hand side but there is restriction on the terminals and variables appear on the right hand side of the production.

Definition: Let $G = (V, T, P, S)$ be a CFG, the CFG G is said to be in GNF if all the productions are of the form

$$A \rightarrow \alpha\beta$$

where $\alpha \in T$ and $\beta \in V^*$ i.e. the first symbol on the right hand side of the production must be a terminal & it can be followed by zero or more variable

Converting CFG to CNF

A context free grammar $G = (V, E, R, S)$ is said to be in Chomsky Normal Form (CNF), iff every rule is of one of the following forms.

$x \rightarrow a$ where $a \in E$ or

$x \rightarrow BC$ where $B, C \in V - E$

ex: $S \rightarrow AB, A \rightarrow a, B \rightarrow b$

GNF [Greibach Normal Form]

GNF is a context free grammar $G = (V, E, R, S)$ where all rules have one of the following forms:

$x \rightarrow a\beta$ where $a \in E$ & ~~both~~
 $\beta \in (V - E)^*$

ex: $S \rightarrow aA | aAB, A \rightarrow, B \rightarrow b$

Conversion takes 4 steps [CFG to CNF]

There exist 4-steps algorithm to convert a CFG g into a new grammar g' such that: $L(g) = L(g') - \{\epsilon\}$

1. $g'' =$ remove EPS $S \rightarrow \epsilon$

2. $g''' =$ remove unit $A \rightarrow B$

3. $g''' =$ remove mixed $A \rightarrow AB$

4. $g^IV =$ remove long $S \rightarrow ABCD$

return g'

① convert the given grammar into CNF

$$G = \{ S \rightarrow aACa | aCa | aAa | aa \\ A \rightarrow eC | e | a \\ B \rightarrow eC | e \\ C \rightarrow e | ec \}$$

$$S \rightarrow aACa$$

$$A \rightarrow B | a$$

$$B \rightarrow C | e$$

$$C \rightarrow eC | \epsilon$$

Step 1: Remove ϵ production

$$N = \{ C \} \quad C \rightarrow \epsilon$$

$$N = \{ B, C \} \quad B \rightarrow C \quad C \rightarrow \epsilon$$

$$N = \{ A, B, C \} \quad A \rightarrow B \rightarrow C \quad C \rightarrow \epsilon$$

$$G' = \{ S \rightarrow aACa | aCa | aAa | aa \\ A \rightarrow B | a \\ B \rightarrow C | e \\ C \rightarrow eC | e \}$$

Step 2: Remove unit production

$$G'' = \{ S \rightarrow aACa | aCa | aAa | aa \\ A \rightarrow eC | e | a \\ B \rightarrow eC | e \\ C \rightarrow eC | e \}$$

$$\begin{array}{l} B \rightarrow C \\ C \rightarrow eC | e \end{array}$$

Step3: Remove mixed symbol

$$G''' = \left\{ \begin{array}{l} S \rightarrow T_a A C T_a \mid T_a C T_a \mid T_a A T_a \mid T_a T_a \\ T_a \rightarrow a \\ A \rightarrow T_e C \mid e \mid a \\ T_e \rightarrow e \\ B \rightarrow T_e C \mid e \\ C \rightarrow T_e C \mid e \end{array} \right.$$

Step4: Remove long production

$$G'''' = \left\{ \begin{array}{l} S \rightarrow T_a S_1 \mid T_a S_2 \mid T_a S_3 \mid T_a T_a \\ S_1 \rightarrow A S_2 \\ S_2 \rightarrow C T_a \\ S_3 \rightarrow A T_a \\ T_a \rightarrow a \\ A \rightarrow T_e C \mid e \mid a \\ T_e \rightarrow e \\ B \rightarrow T_e C \mid e \\ C \rightarrow T_e C \mid e \end{array} \right. \right\} \left\{ \begin{array}{l} S \rightarrow T_a \frac{A C T_a}{S_1} \\ S \rightarrow T_a \frac{S_1}{S_2} \\ S_1 \rightarrow A \frac{C T_a}{S_2} \end{array} \right.$$

Q) Convert the following grammar to CNF

$$G = \begin{array}{l} S \rightarrow aSa | B \\ B \rightarrow bbG | bb \\ G \rightarrow cG | \epsilon \end{array}$$

CNF form [Chomsky normal form]
NT - terminal
 $A \rightarrow a$ NT \rightarrow NT NT {NT: nonterminal}
 $A \rightarrow Bc$

Step 1: Remove ϵ production.

$$\text{Nullable} = \{C\} \quad C \rightarrow \epsilon$$

$$\begin{array}{l} S \rightarrow aSa | B \\ B \rightarrow bbG | bb \\ G \rightarrow cG | c \end{array}$$

Step 2: remove unit production. $\left\{ \begin{array}{l} NT \rightarrow NT \\ A \rightarrow B \end{array} \right\}$

$$\begin{array}{l} S \rightarrow aSa | bbG | bb \\ B \rightarrow bbG | bb \\ C \rightarrow cG | c \end{array}$$

Step 3: Remove mixed production $\{NT \rightarrow T NT\}$

$$\begin{array}{l} S \rightarrow \tau_a S \tau_a | \tau_b \tau_b G | \tau_b \tau_b \\ B \rightarrow \tau_b \tau_b C | \tau_b \tau_b \\ G \rightarrow \tau_c G | c \end{array}$$

Step 4: Remove long production

$$\left\{ \begin{array}{l} NT \rightarrow T \\ NT - NTNT \end{array} \right\}$$

$$S \rightarrow TaS_1 \mid TbS_2 \mid TbTb$$

$$S_1 \rightarrow STa$$

$$S_2 \rightarrow TbC$$

$$B \rightarrow TbS_2 \mid TbTb$$

$$C \rightarrow TcG \mid c$$

$$Ta \rightarrow a$$

$$Tb \rightarrow b$$

$$Tc \rightarrow c$$

③ convert given CFG to CNF

$$S \rightarrow ABC$$

$$A \rightarrow BCG \mid D$$

$$B \rightarrow bB \mid A \mid E$$

$$C \rightarrow Ac \mid Cc \mid E$$

$$D \rightarrow aa$$

Step 1: remove ϵ production

$$\text{Nullable } \{C\} \quad C \rightarrow \epsilon$$

$$\text{Nullable } \{C, B\} \quad B \rightarrow \epsilon$$

$$S \rightarrow ABC \mid AC \mid AB \mid A$$

$$A \rightarrow aC \mid a \mid D$$

$$B \rightarrow bB \mid b \mid A$$

$$C \rightarrow Ac \mid Cc \mid C$$

$$D \rightarrow aa$$

Step 2 Remove unit production

$$\begin{cases} S \rightarrow A \\ B \rightarrow A \\ A \rightarrow D \end{cases}$$

$$S \rightarrow ABC | AC | AB | aG | a | aa$$

$$A \rightarrow aG | a | aa$$

$$B \rightarrow bB | b | aG | a | aa$$

$$C \rightarrow Ac | Cc | c$$

$$D \rightarrow aa$$

Step 3: Remove mixed production

$$S \rightarrow ABC | AC | AB | TaG | \alpha | TaTa$$

$$Ta \rightarrow a$$

$$A \rightarrow TaG | a | TaTa$$

$$B \rightarrow TbB | b | TaC | a | TaTa$$

$$Tb \rightarrow b$$

$$C \rightarrow ATc | CTc | c$$

$$Tc \rightarrow c$$

$$D \rightarrow aa$$

Step 4: Remove long production

$$S \rightarrow AS_1 | AC | AB | TaG | a | TaTa$$

$$S_1 \rightarrow BC$$

$$Ta \rightarrow a$$

$$A \rightarrow TaG | a | TaTa$$

$$B \rightarrow TbB | b | TaC | a | TaTa$$

$$Tb \rightarrow b$$

$$C \rightarrow ATc | CTc | c$$

$$Tc \rightarrow c$$

$$D \rightarrow aa$$

problems on GNF

7

- convert the given grammar CNF
- Eliminate left recursion
- convert into GNF.

Conversion Rules

1. obtain grammar in CNF

2. Rename variables

3. use substitution method to obtain production
to the form

$A_i \rightarrow A_j$ where $i < j$

4. Remove left Recursion

Apply 3 and 4 until grammar is in GNF.

①

convert the following grammar to GNF
form.

~~$S \rightarrow ABc$~~ $S \rightarrow aBc$ $B \rightarrow b$

$S \rightarrow aBG$

$B \rightarrow b$

$G \rightarrow c$

②

$S \rightarrow Bc$ — {NT, t} it is not in GNF

$B \rightarrow b$

$S \rightarrow bc$ $\rightarrow S \rightarrow bG$ } removed $B \rightarrow b$
 $c \rightarrow c$

(3) convert the following grammar to CNF form.

$$S \rightarrow AA|0$$

$$A \rightarrow SS|1$$

$$\left\{ \begin{array}{l} A \rightarrow BC \\ A \rightarrow 9 \\ \text{CNF form} \end{array} \right.$$

① This grammar is in CNF

② Renaming $S = A_1$ $A = A_2$

$$A_1 \rightarrow A_2 A_2 | 0 \quad @$$

$$A_2 \rightarrow A_1 A_1 | 1$$

③ $A_i^0 \rightarrow A_j \quad i < j$
so use substitution rule $A_2 \rightarrow A_2 A_2 A_1 | 0 A_2 | 1$
left recursion remove left recursion

$$A \rightarrow A\alpha_1 | A\alpha_2 \dots | A\alpha_n | \beta_1 | \beta_2 \dots | \beta_m$$

left recursion A's

$$\text{i.e } A \rightarrow \beta_1 \cancel{\alpha} | \beta_2 \cancel{\alpha} \dots | \beta_m \cancel{\alpha}$$

$$\cancel{\alpha} \rightarrow \alpha_1 \cancel{\alpha} | \alpha_2 \cancel{\alpha} \dots | \alpha_n \cancel{\alpha} | \epsilon$$

$$A_2 \rightarrow 0A_1 \cancel{\alpha} | \cancel{\alpha} A_1$$

$$\cancel{\alpha} \rightarrow A_2 A_1 \cancel{\alpha} | \epsilon$$

rewrite $\Rightarrow A_2 \rightarrow 0A_1 | 1 | 0A_1 \cancel{\alpha} | \cancel{\alpha} A_1 \rightarrow \text{GNF } \checkmark \quad b$
remove $\cancel{\alpha} \rightarrow A_2 A_1 \cancel{\alpha} | A_2 A_1 \quad c$

production $a) A_1 \rightarrow A_2 A_2 | 0 \quad \text{not in GNF}$

$b) A_2 \rightarrow 0A_1 | 1 | 0A_1 \cancel{\alpha} | \cancel{\alpha} A_1 \rightarrow \text{in GNF}$

$c) \cancel{\alpha} \rightarrow A_2 A_1 \cancel{\alpha} | A_2 A_1 \rightarrow \text{not in GNF}$

(b) \Leftrightarrow in GNF

(c) substitute A_2 in RHS

$$A_1 = OA_1A_2 \mid 1A_2 \mid OA_1 \times A_2 \mid 1 \times A_2 \mid O - \text{in GNF}$$

(d) substitute A_2

$$x \rightarrow OA_1A_2 \times \mid 1A_2 \times \mid OA_1 \times A_2 \times \mid 1 \times A_2 \times \mid$$

$$OA_1A_2 \mid 1A_2 \mid OA_1 \times A_2 \mid 1 \times A_2 - \text{in GNF}$$

Q2 convert the following grammar to GNF

$$A \rightarrow BC$$

$$B \rightarrow CA \mid b$$

$$C \rightarrow AB \mid a$$

① All production are in CNF

$$A = A_1, B = A_2, C = A_3$$

$$\underline{A_1 \rightarrow A_2 A_3}, \underline{A_2 \rightarrow A_3 A_1} \mid b, \underline{A_3 \rightarrow A_1 A_2} \mid a$$

$$③ A_i \rightarrow A_j x \quad i < j \quad \begin{matrix} \uparrow \text{not in format} \\ \therefore 3 \geq 1 \end{matrix}$$

Substitution

$$A_3 \rightarrow A_1 A_2 \mid a - \text{substitute for } A_1$$

$$A_3 \rightarrow A_2 A_3 A_2 \mid a \quad i \notin j \quad \text{still } 3 > 2$$

\rightarrow substitute for A_2

Rewrite

$$A_3 \rightarrow \underbrace{A_3 A_1}_{\alpha} \underbrace{A_3 A_2}_{\beta_1} \mid b \underbrace{A_3 A_2}_{\beta_1} \mid a \quad \text{it has left recursion}$$

formula for remove left recursion

$$\left. \begin{array}{l} A \rightarrow A\alpha_1 | \beta_1 | \beta_2 \\ A \rightarrow \beta_1 X | \beta_2 X \\ X \rightarrow \alpha_1 X | \epsilon \end{array} \right\}$$

$$\left. \begin{array}{l} A_3 \rightarrow bA_3A_2X | \alpha X | \epsilon \\ X \rightarrow A_1A_3A_2X | \epsilon \end{array} \right\} \text{Remove } \epsilon$$

① $A_3 \rightarrow bA_3A_2X | \alpha X | bA_3A_2 | a$

② $X \rightarrow A_1A_3A_2X | A_1A_3A_2$

③ $A_1 \rightarrow A_2A_3 *$ not in GNF

④ $A_2 \rightarrow A_3A_1 | b *$ not in GNF

⑤ $A_3 \rightarrow bA_3A_2X | \alpha X | bAA_3A_2 | a$ — GNF

⑥ $X \rightarrow A_1A_3A_2X | A_1A_3A_2$ not in GNF

Rewrite ② by substituting for ⑤ A_3

$$A_2 \rightarrow bA_3A_2X A_1 | \alpha X A_1 | bA_3A_2 A_1 | a A_1 | b$$

↑ in GNF

Rewrite ① by substituting for A_2

$$A_1 \rightarrow bA_3A_2X A_1 A_3 | \alpha X A_1 A_3 | bA_3A_2 A_1 A_3 |$$

$$a A_1 A_3 | b A_3$$

\longleftrightarrow in GNF

(4)

$$\begin{aligned}
 X &\rightarrow bA_3A_2XA_1A_3 - A_3A_2X \\
 &\quad aXA_1A_3 - A_3A_2X \\
 &\quad bA_3A_2A_1A_3 - A_3A_2X \\
 &\quad aA_1A_3 - A_3A_2X \\
 &\quad bA_3 - A_3A_2X \\
 \\
 &\quad bA_3A_2XA_1A_3 \quad A_3A_2X \\
 &\quad aXA_1A_3 \quad A_3A_2X \\
 &\quad bA_3A_2A_1A_3 \quad A_3A_2X \\
 &\quad aA_1A_3 \quad A_3A_2X \\
 &\quad bA_3 \quad A_3A_2X
 \end{aligned}$$

{ Substitute
A₁

\rightarrow in GNF

proper grammar for context free language

(3)

convert into GNF

$$S \rightarrow CA | BB$$

$$B \rightarrow b | SB$$

$$C \rightarrow b$$

$$A \rightarrow a$$

(1)

This grammar in CNF

(2)

Renaming S = A₁, C = A₂, A = A₃, B = A₄

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

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

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

③ $A_i^o \rightarrow A_j k$ $i \leq j$ & should never be $i >$

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

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

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

Remove $A_4 \rightarrow \underline{A_1} A_4$, replace A_1

$A_4 \rightarrow b \mid \underline{A_2 A_3 A_4} \mid A_4 A_4 A_4$ replace A_2

$$A_4 \rightarrow b \mid b A_3 A_4 \mid \underline{A_4 A_4 A_4}$$

\downarrow
left recursion

Remove left recursion introduce new variable.

$$Z \rightarrow A_4 A_4 Z \mid A_4 A_4$$

$$A_4 \rightarrow b \mid b A_3 A_4 \mid b Z \mid b A_3 A_4 Z$$

$$Z \rightarrow A_4 A_4 Z \mid A_4 A_4$$

$$\left. \begin{array}{c} AAA \\ AAC \mid AA \\ A_4 = b \mid b A_3 A_4 / A_4 \\ A_4 \rightarrow b \mid n_3 A_4 \\ A_4 \rightarrow A_4 A_4 A_4 \end{array} \right\}$$

New grammar

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

$$A_4 \rightarrow b | b A_3 A_4 | b Z | b A_3 A_4 Z$$

$$Z \rightarrow A_4 A_4 | A_4 A_4 Z$$

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

Replace A_2 & A_4

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

$$A_4 \rightarrow b | b A_3 A_4 | b Z | b A_3 A_4 Z$$

$$Z \rightarrow b A_4 | b A_3 A_4 | b Z A_4 |$$

$$b A_3 A_4 Z A_4 |$$

$$b A_4 Z | b A_3 A_4 A_4 Z | b Z A_4 Z | b A_3 A_4 Z A_4 Z$$

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

∴ It is in GNR //

Pumping Lemma for context free language

Pumping lemma for CFL is used to prove that a language is not a context free

Assume L is a context free language, there is a pumping length n such that any string $w \in L$ of length $\geq n$ can be written as

$$uvw^n$$

we can break w into 5 string
 $w = uvxyz$ such that

$$|vxy| \leq n$$

$$|vy| \neq \emptyset$$

$\nexists k \geq 0$ such that the string $uv^kxy^kz \in L$

NOTE: select k such that the resulting string is not in L .

prove that a language is not CF using pumping lemma (CFL follows the steps (we prove using contradiction))

- ① Assume that L is context free
- ② the pumping length say n
- ③ All strings longer than n can be pumped
 $|w| > n$
- ④ Now find a string ' w ' in L such that
 $|w| > n$
- ⑤ Divide w into ~~xyz~~: $uvxyz$
- ⑥ show that $uv^kxy^kz \notin L$ for some k
- ⑦ then consider the ways that w can be divided into $uvxyz$
- ⑧ show that none of these can satisfy all the 3 pumping conditions at same time.
- ⑨ w cannot be pumped (contradiction)

① find out whether $L = \{x^n y^n z^n \mid n \geq 1\}$
is context free or not

Let L is context free (contradiction)
 \Rightarrow Now we can take a string such that
 $s = x^n y^n z^n$

\Rightarrow we divide s into 5 parts $u \underline{v} x \underline{y} z$

case(i): $n=4$ so, $s \rightarrow x^4 y^4 z^4$

v & y each contain only one type of symbols.

$x \underline{x} x x \underline{y} y y y \underline{z} z z z z$ \Rightarrow divide into 5 parts
 $u \underline{v} x \underline{y} z$

uv^kxy^kz (Let $k=2$)

$\Rightarrow uv^2xy^2z$

$\Rightarrow x \underline{x} x x x \underline{y} y y y z z z z z$

$\Rightarrow x^6 y^4 z^5$

$x^6 y^4 z^5 \notin L$

$2l = x$
$v = xx$
$x = xy y y y z$
$y = z$
$z = zz$

\therefore the given language is not context free.

case(ii): either v or y has more than one kind of symbol

$$S = x^n y^n z^n$$

$$n=4 \text{ so } x^4 y^4 z^4$$

$$\Rightarrow \underset{u}{\underbrace{xx}} \underset{v}{\underbrace{xyyy}} \underset{x}{\underbrace{yy}} \underset{z}{\underbrace{zzzz}}$$

⇒ 5 parts
 \downarrow
~~uvwxyz~~

$$u=xx$$

$$v=xyyy$$

$$x=y$$

$$y=y$$

$$z=zzzz$$

→ let assume $k=2$

$$\Rightarrow uv^2 xy^5 z$$

$$\Rightarrow uv^2 xy^2 z$$

$$\Rightarrow xxzxyyyxxxyy44444zzzz$$

$$\Rightarrow x^4 y^2 x^2 y^5 z^4 \notin L$$

② $L = \{a^p \mid p \text{ is a prime}\}$ is not context free

If L is context free, we can apply pumping lemma

Let n be the constant of the lemma
 consider sentence a^p , where $p \geq n$.

$$\text{let } |vy| = k \leq n$$

$$\text{then } v\cancel{y} = vy = a^k.$$

then a^p can be written as

$$a^{p-k} a^k.$$

If vy is pumped $(p-k)$ times, we get a sentence

$$a^{p-k} a^{k(p-k)} = a^{(p-k)(k+1)} \rightarrow \text{see solution in next page}$$

It is not a prime,

$\therefore \{a^p \mid p \text{ is a prime}\}$ is not context free.

③

Show that $L = \{ww \mid w \in \{0, 1\}^*\}$ is not context free.

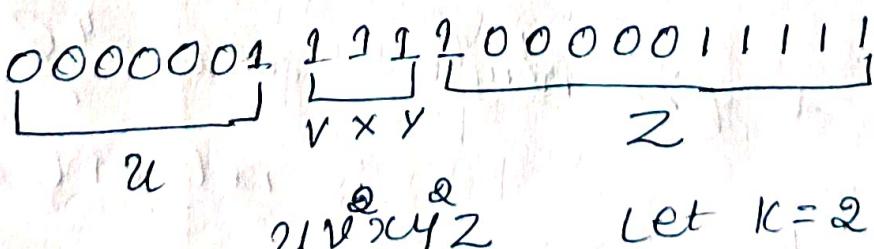
Assume L is context free (say P)
 L must have a pumping length (say p)
Now take string s such that

$$s = 0^p 1^p 0^p 1^p$$

divide string into 5 parts $uvxyz$
such that $|vxy| \leq n$, $|vy| \neq 0$

e.g. take $p = 5$

$$\therefore 0^5 1^5 0^5 1^5$$


 00000011110000011111
 $u \quad vxy \quad z$

uv^2xz let $k=2$

00000111110000011111

$$\therefore 0^5 1^7 0^5 1^5 \in L$$

The given language is not context free.

$$\begin{aligned}
 & a^{P-K} a^{K(P-K)} \\
 & a^{P+K} a^{KP-K^2} \\
 & a^{P-K+KP-K^2} \\
 & a^{P-K+KP-K^2} \\
 & a^{(P+KP)-(K+K^2)} \\
 & a^{(P+KP)-(K+K^2)} \\
 & a^{P(1+K)-K(1+K)} \\
 & a^{(1+K)(P-K)}
 \end{aligned}$$

(4) $L = \{a^n b^n c^n : n \geq 3\}$ is not CFL

Assume L is context free

Let $w = a^k b^k c^k$

* If $|w| > k$ we divide

String $uvwxyz$

$$|vxy| \leq k$$

$$|vy| \neq 0 \quad \text{At } q > 0, uv^q y^q z$$

$$\text{Let } k=4$$

$$w = a^4 b^4 c^4$$

$$\begin{array}{ccccccccc}
 & a & a & a & a & b & b & b & b & c & c & c & c \\
 \hline
 u & | & | & | & | & | & | & | & | & | & | & | & | \\
 x & | & | & | & | & | & | & | & | & | & | & | & | \\
 y & | & | & | & | & | & | & | & | & | & | & | & | \\
 z & | & | & | & | & | & | & | & | & | & | & | & |
 \end{array}$$

$$|vxy| \leq 4$$

$$aaa \leq 4$$

$$3 \leq 4$$

$$\begin{aligned}
 |vy| &\neq 0 \\
 q &\neq 0
 \end{aligned}$$

\Rightarrow Let $q = 2$

$$uv^2xy^2z$$

$$a(a)^2a(a)^2z$$

$$aaaaaaabbb6bccccc$$

$$a^6 b^4 c^4 \notin L$$

∴ not-CFL

⑤ $L = \{a^{n^2} : n \geq 0\}$ is not CFL

Assume L is context free

Let $w = a^K$

* $|w| \geq K$ always we divide string into $uvxyz$

$$|vxy| \leq K$$

$$|vy| \neq 0 \quad \forall q \geq 0 \quad uv^q xy^q z$$

$$\text{Let } K=2$$

$$w = a^{2^2}$$

$$= aaaa
u \underset{|}{v} \underset{|}{x} \underset{|}{y} \underset{|}{z}$$

$$|vxy| \leq 2$$

$$aaa \leq 2$$

$$2 \leq 4 \checkmark$$

$$\text{Let } q=2$$

$$u \underset{aa}{v} \underset{aa}{x} \underset{aa}{y} \underset{aa}{z}$$

\therefore not CFL

⑥ $L = \{w\bar{c}w : w \in \{a, b\}^*\}$ is not CFL

Assume L is context free

let $w = a^K b^K c a^K b^K$

* $|w| \geq K$ we divide $w = uvxyz$

$$|vxy| \leq K$$

$$|vy| \neq 0 \quad \forall q \geq 0 \quad \text{del} \cdot uv^q xy^q z$$

$$\text{Let } K=4$$

$$a^4 b^4 c a^4 b^4$$

$$w = \underbrace{aaaa}_{u} \underbrace{bbbb}_{v} \underbrace{cccc}_{x} \underbrace{aaaa}_{y} \underbrace{bbbb}_{z} \underbrace{cccc}$$

$$|vxy| \leq 4$$

$$aaa \leq 4$$

$$3 \leq 4 \checkmark$$

$$\Rightarrow |vy| \neq 0$$

$$2 \neq 0$$

$$\text{Let } q=2$$

$$u \underset{aa}{v} \underset{aa}{x} \underset{aa}{y} \underset{aa}{z}$$

$$a(a)^2 a(a^2) \underset{aaaa}{c} \underset{aaaa}{a} \underset{aaaa}{b} \underset{aaaa}{b} \underset{aaaa}{b} \underset{aaaa}{b}$$

$$a^6 b^4 c a^4 b^4$$

$$\cancel{a^2}$$

$$\therefore \text{not CFL}$$

Closure properties of CFL

> union, concatenation, Kleen star

If L_1 & L_2 are context free languages then prove $L_1 \cup L_2$, $L_1 L_2$ and L_1^*

(i) $G_1 = (V_1, T_1, P_1, S_1)$

$$G_2 = (V_2, T_2, P_2, S_2)$$

$$G_3 = (V_1 \cup V_2 \cup S_3, T_1 \cup T_2, P_3, S_3)$$

S_3 is a start state q_3 and $S_3 \in (V_1 \cup V_2)$

$$P_3 = P_1 \cup P_2 \cup \{S_3 \rightarrow S_1 | S_2\}$$

$$L_3 = L_1 \cup L_2$$

(ii) $G_4 = (V_1 \cup V_2 \cup S_4, T_1 \cup T_2, P_4, S_4)$

S_4 is a start symbol for the grammar and $S_4 \in (V_1 \cup V_2)$

$$P_4 = P_1 \cup P_2 \cup \{S_4 \rightarrow S_1 | S_2\}$$

$$\therefore L_3 = L_1 \cdot L_2$$

(iii) $G_5 = (V, US_5, T_1, P_5, S_5)$

S_5 is a the start symbol of grammar G_5

$$P_5 = P_1 \cup \{S_5 \rightarrow S_1 | S_5\} \in \mathcal{G}$$

$$T^{L_5} = L_s^*$$