mod 4

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

$$\boxed{A \rightarrow a\alpha}$$

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 variables

Simplification of grammer

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

Step1: Removal of useless symbol (Non terminal)

Step2: Removal of unit production

Step3: Removal of ϵ production.

Step1: Removal of useless symbol:

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

Step2: Removal of unit production

* A production of the form $A \to B$ 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 $A \to \epsilon$. To remove ϵ production, we compute nullable set.

2

Two algorithms used to simplify CFG.

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

(1) Grammer
$S \rightarrow AB \mid AC$
$A \rightarrow aAb \mid \epsilon$
$B \rightarrow aA$
$C \rightarrow bca$
$D \rightarrow AB$

→ remove unproductive
→ remove unreachable

$X \rightarrow \alpha$

all symbol in RHS is productive LHS also as productive

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

→ $P = \{a, b, A\}$ | $A \rightarrow aAb \mid \epsilon$
→ $P = \{a, b, A, B\}$ | $B \rightarrow aA$
$P = \{a, b, S, A, B\}$ | $S \rightarrow AB$
$P = \{a, b, S, A, B, D\}$ | $D \rightarrow AB$

C as unproductive
∴ drop RHS or LHS with unproductive

∴ $S \rightarrow AB$
$A \rightarrow aAb \mid \epsilon$
$B \rightarrow aA$
$D \rightarrow AB$

Remove unreachable

Initial Reachable = $\{S\}$

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

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

D is unreachable, so Remove D from both from LHS, RHS

$X \to \alpha \underline{A} \beta$

reachable

is yet not reachable

mark 'A' as reachable

After removing useless grammer symbols

$$\begin{cases} S \to AB \\ A \to aAb \mid \epsilon \\ B \to bA \end{cases}$$

$S \to abS \mid abA \mid abB$

$A \to cd$

$B \to aB$

$C \to dc$

→ Remove unproductive

→ Remove unreachable

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

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

Loop — what are gets added as productive

$P = \{a, b, c, d, A\}$ $A \to cd$

$P = \{a, b, c, d, A, C\}$ $C \to dc$

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

B is unproductive we remove from RHS,

$S \to abS \mid abA$

$A \to cd$

$B \to dc$

Remove unreachable symbols.

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

<u>Loop</u>: $R = \{S, A\}$

Left hand reach Right hand

C is unreachable

$\therefore$ $S \to abS \mid abA$

$A \to cd$

<u>ex</u>:

3) $S \to aaB \mid abA \mid aaT$ Remove useless symbols

$A \to aA$ 1. Remove unproductive

$B \to ab \mid b$ 2. Remove unreachable

$C \to ad.$

1. Remove unproductive

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

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

<u>loop</u>: $P = \{a, b, d, C\}$ $C \to ad$

$P = \{a, b, d, C, B\}$ $B \to ab$

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

unproductive are A & T

$\therefore$ $S \rightarrow aaB$

$B \rightarrow ab \mid b$

$C \rightarrow ad$

Remove unreachable

Initial $R = \{S\}$

$R = \{S, B\}$

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

$\therefore$ $S \rightarrow aaB$

$B \rightarrow ab \mid b$

(4) Remove useless symbol | production

$S \rightarrow AB \mid a$

$A \rightarrow BC \mid b$

$B \rightarrow aB \mid C$

$C \rightarrow aC \mid B$

Remove unproductive

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

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

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

$\therefore$ $S \rightarrow \cancel{AB} \mid a \quad a$ $\qquad$ $S \rightarrow a$

$\cancel{A} \rightarrow \cancel{BC} \mid b \quad b$ $\qquad$ $A \rightarrow b$

2. Remove unreachable

$R = \{S\}$ UR :

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

(5) $S \to AB \mid AC$

$A \to aAb \mid bAa \mid a$

$B \to bbA \mid aaB \mid AB$

$C \to abCA \mid aDb$

$D \to bD \mid aC.$

Remove unproductive

$P = \{a, b\}$

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

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

$P = \{a, b, A, B\}$ $B \to bbA$

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

$\therefore S \to AB$

$A \to aAb \mid bAa \mid a$

$B \to bbA \mid aaB \mid AB$

$\therefore$ Remove C & D unproductive remove from grammer

$\Rightarrow$ Remove unreachable

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

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

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 ∈ production

Eliminating ∈ rules

⇒ **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 → (1).
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.

<u>modifiable</u> <u>variable</u>: A rule is modifiable iff it is of the form

$P \to \alpha Q \beta$ for some nullable Q.

(1) $S \to aTa$

$T \to ABC$

$A \to aA \mid C$

$B \to Bb \mid C$

$C \to c \mid \epsilon$

Initial Nullable $N = \{C\}$ $C \to \epsilon$

$N = \{C, B, A\}$ $A \to C$
$B \to C$

$N = \{C, B, A, T\}$ $T \to ABC$

All Nullable NT

$S \to aTa \mid aa$

$T \to ABC \mid BC \mid AC \mid AB \mid A \mid C \mid B$ {one at a time & 2 at a time

$A \to aA \mid a \mid C$

$B \to Bb \mid b \mid C$

$C \to c$ {remove ϵ productions}

② $S \rightarrow aACa$
$A \rightarrow B | a$
$B \rightarrow C | e$
$C \rightarrow eC | \epsilon$

Remove ϵ production
First find nullable

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

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

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

$\therefore$ $S \rightarrow aACa | aCa | aAa | aa$
$A \rightarrow B | a$
$B \rightarrow C | e$
$C \rightarrow eC | e$

Removal of unit production

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

ex : $X \rightarrow Y$ & $S \rightarrow Y$

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

Step 2 : now delete $X \rightarrow Y$ from grammer

Step 3 : Repeat step 1 & 2 untill all production ~~reached~~ removed from grammer.

ex: $S \to 0A \mid 1B \mid C$

$A \to 0S \mid 00$

$B \to 1 \mid A$

$C \to 01$

$S \to 0A$ $A \to 0S$ $B \to 1$ $C \to 01$

$S \to 1B$ $A \to 00$ $B \to A$ — unit production

$S \to C$

~~S →~~

$S \to 0A \mid 1B \mid 01$

$A \to 0S \mid 00$

$B \to 1 \mid 0S \mid 00$

$C \to 01$ //

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: $A \to \epsilon$

⇒ A non terminal generating two non terminal

ex: $S \to AB$

⇒ A non terminal generating termin

eg: $S \to a$

Removal of unit production

A unit production is a production of the form A → B where both A & B are variable

1) Eliminate are unit production from the grammer,

S → AB
A → a
B → C | b
C → D
D → E | bC
E → d | Ab

Non unit production

S → AB
A → a
B → b
D → bC

unit production

B → C
C → D
D → E

It is clear from the depending graph that D → E, so the non unit productions generated from 'E' can also be generated from D.

E → d | Ab

since D → E
D → d | Ab | bc

C → D → E

C → d | Ab | bc

since B → C, C → D, D → E

B → b | d | Ab | bC

The final grammer obtained after eliminating

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

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

$P = \{ S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow b \mid d \mid Ab \mid bC$

$C \rightarrow bC \mid d \mid Ab$

$D \rightarrow bc \mid d \mid Ab$

$E \rightarrow d \mid Ab$

$\}$

S is the Start symbol,,

(2) Eliminate unit productions from the grammer,

$S \rightarrow A0 \mid B$

$B \rightarrow A \mid 11$

$A \rightarrow 0 \mid 12 \mid B$

non unit production

$S \rightarrow A0$

$B \rightarrow 11$

$A \rightarrow 0 \mid 12$

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

$S \rightarrow A0 \mid 11 \mid 0 \mid 12$

$B \rightarrow 0 \mid 12 \mid 11$

$A \rightarrow 0 \mid 12 \mid 11$

The resulting grammer without unit production.

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

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

$P = \{ S \to A0 \mid 11 \mid 0 \mid 12$

$B \to 0 \mid 12 \mid 11$

$A \to 0 \mid 12 \mid 11 \}$

S is start symbol

③ Eliminate unit production

$S \to Aa \mid B \mid Ca$

$B \to aB \mid b$

$C \to Db \mid D$

$D \to E \mid d$

$E \to ab$

unit production

$S \to B$

$C \to D$

$D \to E$

non unit production

$S \to Aa \mid Ca$

$B \to aB \mid b$

$C \to Db$

$D \to d$

$E \to ab$

$S \to B$

$S \to Aa \mid aB \mid b \mid Ca$ &

$D \to E$

$D \to ab \mid d$

$C \to D \quad D \to E.$

$C \to Db \mid ab \mid d$

The resulting grammer is

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

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

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

S is start symbol.

mod 4

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

$$\boxed{A \rightarrow a\alpha}$$

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.

Converting CFG to CNF

A context free grammer $G = (V, \Sigma, R, S)$ is said to be in chomsky Normal form (CNF), iff every rule is of one of the following forms.

$X \rightarrow a$ where $a \in \Sigma$ or

$X \rightarrow BC$ where B & $C \in V - \Sigma$

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

GNF [Greibach Normal Form]

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

$X \rightarrow a\beta$ where $a \in \Sigma$ & ~~$\beta \in V - \Sigma$~~ $\beta \in (V - \Sigma)^*$

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

Conversion takes 4 steps [CFG to CNF]

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

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_c

① convert the given grammer into CNF

$G = S \to aACa \mid aCa \mid aAa \mid aa$

$A \to eC \mid e \mid a$

$B \to eC \mid e$

$C \to e \mid eC$

$S \to aACa$

$A \to B \mid a$

$B \to C \mid e$

$C \to eC \mid \epsilon$

Step1: Remove ϵ production

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

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

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

$G' = \{ S \to aACa \mid aCa \mid aAa \mid aa$

$A \to B \mid a$

$B \to C \mid e$

$C \to eC \mid e$ }

Step2 : Remove unit productions

$G'' = \{ S \to aACa \mid aCa \mid aAa \mid aa$

$A \to eC \mid e \mid a$

$B \to eC \mid e$

$C \to eC \mid e$

$B \to C$

$C \to eC \mid e$

Step3: Remove mixed Symbol

$G''' = \{ S \to T_a A C T_a \mid T_a C T_a \mid T_a A T_a \mid T_a T_a$

$T_a \to a$

$A \to T_e C \mid e \mid a$

$T_e \to e$

$B \to T_e C \mid e$

$C \to T_e C \mid e$

Step4: Remove Long production

$G'''' = \{ S \to T_a S_1 \mid T_a S_2 \mid T_a S_3 \mid T_a T_a$

$S_1 \to A S_2$

$S_2 \to C T_a$

$S_3 \to A T_a$

$T_a \to a$

$A \to T_e C \mid e \mid a$

$T_e \to e$

$B \to T_e C \mid e$

$C \to T_e C \mid e$

$S \to T_a \underline{A C T_a}$

$S \to T_a S_1$

$S_1 \to A \underbrace{C T_a}_{S_2}$

② convert the following grammer to CNF

$G = \{ S \rightarrow aSa \mid B$

$B \rightarrow bbC \mid bb$

$C \rightarrow aC \mid \epsilon \}$

CNF form [chomsky Norm form]

$A \rightarrow a$ NT – terminal

$A \rightarrow BC$ NT → NT NT {NT : nonterminal}

Step1: Remove ϵ production.

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

$S \rightarrow aSa \mid B$

$B \rightarrow bbC \mid bb$

$C \rightarrow cC \mid c$

Step2: Remove unit production. {NT → NT, A → B}

$S \rightarrow aSa \mid bbC \mid bb$

$B \rightarrow bbC \mid bb$

$C \rightarrow cC \mid c$

Step3: Remove mixed production {NT → T NT}

$S \rightarrow T_a S T_a \mid T_b T_b C \mid T_b T_b$

$B \rightarrow T_b T_b C \mid T_b T_b$

$C \rightarrow T_c C \mid c$

{NT→T, NT→NT NT}

Step4 : Remove long production

$S \rightarrow T_a S_1 \mid T_b S_2 \mid T_b T_b$

$S_1 \rightarrow S T_a$

$S_2 \rightarrow T_b C$

$B \rightarrow T_b S_2 \mid T_b T_b$

$C \rightarrow T_c C \mid c$

$T_a \rightarrow a$

$T_b \rightarrow b$

$T_c \rightarrow c$

---

(3) convert given CFG to CNF

$S \rightarrow ABC$

$A \rightarrow aC \mid D$

$B \rightarrow bB \mid A \mid \epsilon$

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

$D \rightarrow aa$

Step1: Remove ϵ production

Nullable {C} $C \rightarrow \epsilon$

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

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

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

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

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

$D \rightarrow aa$

Step2 Remove unit production

$$\left\{\begin{array}{l} S \to A \\ B \to A \\ A \to D \end{array}\right.$$

$S \to ABC \mid AC \mid AB \mid aG \mid a \mid aa$

$A \to aG \mid a \mid aa$

$B \to bB \mid b \mid aG \mid a \mid aa$

$G \to Ac \mid Cc \mid c$

$D \to aa$

Step3: Remove Mixed production

$S \to ABC \mid AC \mid AB \mid T_a G \mid a \mid T_a T_a$

$T_a \to a$

$A \to T_a G \mid a \mid T_a T_a$

$B \to T_b B \mid b \mid T_a C \mid a \mid T_a T_a$

$T_b \to b$

$G \to A T_c \mid C T_c \mid c$

$T_c \to c$

$D \to aa$

Step4: Remove Long production

$S \to A S_1 \mid AC \mid AB \mid T_a G \mid a \mid T_a T_a$

$S_1 \to BC$

$T_a \to a$

$A \to T_a G \mid a \mid T_a T_a$

$B \to T_b B \mid b \mid T_a G \mid a \mid T_a T_a$

$T_b \to b$

$G \to A T_c \mid C T_c \mid c$

$T_c \to c$

$D \to aa$

problems on GNF

→ convert the given grammer CNF
→ Eliminate left recursion
→ convert into GNF.

Conversion Rules

1. obtain grammer in CNF
2. Rename variables
3. use substitution methods to obtain production to the form

$A_i \to A_j$ 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 → aBa~~ $S \to aBa$ $B \to b$

Introduce new symbol Nonterminal G

$S \to aBG$
$B \to b$
$G \to a$

2) $S \to Bc$ — {NT, t} it is not in GNF
$B \to b$

$S \to bc$ → $S \to bC$, $C \to c$ } removed $B \to b$

(3) convert the following grammar ... form.

$S \rightarrow AA \mid 0$

$A \rightarrow SS \mid 1$

{ $A \rightarrow BC$
$A \rightarrow a$
CNF form }

(1) This grammar is in CNF

(2) Renaming $S = A_1$ $A = A_2$

$A_1 \rightarrow A_2A_2 \mid 0$ ——— (a)

$A_2 \rightarrow A_1A_1 \mid 1$

(3) $A_i \rightarrow A_j$ $i < j$

so use substitution rule $\underline{\underline{A_2}} \rightarrow \underline{A_2}\underbrace{A_2A_1}_{\alpha_1} \mid \underbrace{0A_1}_{\beta_1} \mid \underbrace{1}_{\beta_2}$

left recursion Remove left recursion

$A \rightarrow A\alpha_1 \mid A\alpha_2 \cdots \mid A\alpha_n \mid \beta_1 \mid \beta_2 \cdots \beta_m$

left recursion A's

i.e $A \rightarrow \beta_1 X \mid \beta_2 X \cdots \mid \beta_m X$ {X new symbol}

$X \rightarrow \alpha_1 X \mid \alpha_2 X \cdots \mid \alpha_n X \mid \epsilon$

$A_2 \rightarrow 0A_1X \mid 1X$ {instead of A' replace with X}

$X \rightarrow A_2A_1X \mid \epsilon$

Rewrite remove ϵ $\Rightarrow A_2 \rightarrow 0A_1 \mid 1 \mid 0A_1X \mid 1X$ $\rightarrow$ GNF ✓ ——— (b)

$X \rightarrow A_2A_1X \mid A_2A_1$ ——— (c)

<u>production</u> (a) $A_1 \rightarrow A_2A_2 \mid 0$ Not in GNF

(b) $A_2 \rightarrow 0A_1 \mid 1 \mid 0A_1X \mid 1X$ — in GNF

(c) $X \rightarrow A_2A_1X \mid A_2A_1$ — Not in GNF

(b) i.e. in GNF

(a) Substitute A_2 in RHS

$A_1 = 0A_1A_2 | 1A_2 | 0A_1XA_2 | 1XA_2 | 0$ — in GNF

(c) Substitute A_2

$X \rightarrow 0A_1A_1X | 1A_1X | 0A_1XA_1X | 1XA_1X |$
$0A_1A_1 | 1A_1 | 0A_1XA_1 | 1XA_1$ — in GNF

(2) convert the following grammar to GNF

$A \rightarrow BC$

$B \rightarrow CA | b$

$C \rightarrow AB | a$

(1) All production are in CNF

(2) $A = A_1$ $B = A_2$ $C = A_3$

$A_1 \rightarrow A_2A_3$ $A_2 \rightarrow A_3A_1 | b$ $A_3 \rightarrow A_1A_2 | a$

not in format $\because 3 > 1$

(3) $A_i \rightarrow A_j\alpha$ $i < j$

Substitution

$A_3 \rightarrow A_1A_2 | a$ — substitute for A_1

$A_3 \rightarrow A_2A_3A_2 | a$ $i \nless j$ still $3 > 2$

substitute for A_2

Rewrite

$A_3 \rightarrow \underbrace{A_3A_1A_3A_2}_{\alpha} | \underbrace{bA_3A_2}_{\beta_1} | \underbrace{a}_{\beta_2}$ it has left recursion

Formula for remove left recursion

$$\left\{ \begin{array}{l} A \to A\alpha_1 \mid \beta_1 \mid \beta_2 \\ \quad A \to \beta_1 X \mid \beta_2 X \\ \quad X \to \alpha_1 X \mid \epsilon \end{array} \right.$$

$A_3 \to bA_3A_2X \mid aX$
$X \to A_1A_3A_2X \mid \epsilon$ } Remove ϵ

① $A_3 \to bA_3A_2X \mid aX \mid bA_3A_2 \mid a$
② $X \to A_1A_3A_2X \mid A_1A_3A_2$

① $A_1 \to A_2A_3$ × not in GNF

② $A_2 \to A_3A_1 \mid b$ × not in GNF

③ $A_3 \to bA_3A_2X \mid aX \mid bA_3A_2 \mid a$ - GNF

④ $X \to A_1A_3A_2X \mid A_1A_3A_2$ not in GNF

Rewrite 2 by substituting for A_3

$A_2 \to bA_3A_2XA_1 \mid aXA_1 \mid bA_3A_2A_1 \mid aA_1 \mid b$ → in GNF

Rewrite 1 by substituting for A_2

$A_1 \to bA_3A_2XA_1A_3 \mid aXA_1A_3 \mid bA_3A_2A_1A_3 \mid$
$aA_1A_3 \mid bA_3$ → in GNF

(4) $X \rightarrow bA_3A_2XA_1A_3 - A_3A_2X \mid$ { substitute A_1

$aXA_1A_3 - A_3A_2X \mid$

$bA_3A_2A_1A_3 - A_3A_2X \mid$

$aA_1A_3 - A_3A_2X \mid$

$bA_3 - A_3A_2X \mid$

$bA_3A_2XA_1A_3 \; A_3A_2 \mid$

$aXA_1A_3 \; A_3A_2 \mid$

$bA_3A_2A_1A_3 \; A_3A_2 \mid$

$aA_1A_3 \; A_3A_2 \mid$

$bA_3 \; A_3A_2 \mid$

$\longrightarrow$ in GNF

pumping lemma for context free language

(3) convert into GNF

$S \rightarrow CA \mid BB$

$B \rightarrow b \mid SB$

$C \rightarrow b$

$A \rightarrow a$

(1) This grammer in CNF

(2) Renaming $S = A_1$, $C = A_2$, $A = A_3$, $B = A_4$

$A_1 \to A_2 A_3 \mid A_4 A_4$

$A_4 \to b \mid A_1 A_4$

$A_2 \to b$

$A_3 \to a$

(3) $A_i \to A_j K$ $\underline{\underline{i < j}}$ & should never be $i \geq j$

$A_1 \to A_2 A_3 \mid A_4 A_4$

$A_4 \to b \mid A_1 A_4$

$A_2 \to b$

$A_3 \to a$

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

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

$A_4 \to b \mid b A_3 A_4 \mid \underline{A_4 A_4 A_4}$
$\Downarrow$
left recursion

Remove left recursion Introduce new variable.

$Z \to A_4 A_4 Z \mid A_4 A_4$

$A_4 \to b \mid b A_3 A_4 \mid bZ \mid b A_3 A_4 Z$

$Z \to A_4 A_4 Z \mid A_4 A_4$

$A A A$
$A A C \mid A A$

$A_4 = b \mid b A_3 A_4 \mid A_4 \ldots$

$A_4 \to b \mid A_3 A_4$

$A_4 \to A_4 A_4 A_4$

new grammer

violate condition

$A_1 \to A_2A_3 \mid A_4A_4$

$A_4 \to b \mid bA_3A_4 \mid bZ \mid bA_3A_4Z$

$Z \to A_4A_4 \mid A_4A_4Z$

$A_2 \to b$

$A_3 \to a$

Replace A_2 & A_4

$A_1 \to bA_3 \mid bA_4 \mid bA_3A_4A_4 \mid bZA_4 \mid bA_3A_4ZA_4$

$A_4 \to b \mid bA_3A_4 \mid bZ \mid bA_3A_4Z$

$Z \to bA_4 \mid bA_3A_4 \mid bZA_4 \mid$
$bA_3A_4ZA_4 \mid$
$bA_4Z \mid bA_3A_4A_4Z \mid bZA_4Z \mid bA_3A_4ZA_4Z$

$A_2 \to b$

$A_3 \to a$

∴ It is in GNF//

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

$|w| \geq n$

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

$|vxy| \leq n$

$|vy| \neq \epsilon$

$\forall \; k \geq 0$ the string $uv^k x y^k z \in L$

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 $|w| \geq n$
4. Now find a string 'w' in L such that $|w| \geq n$
5. Divide w into ~~uvxy~~ uvxyz
6. show that $uv^k x y^k z \notin L$ 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.

$$\underbrace{x}_{u}\,\underbrace{xx}_{v}\,\underbrace{xyyyyz}_{x}\,\underbrace{z}_{y}\,\underbrace{zz}_{z}$$

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

$u v^k x y^k z$ (Let $k = 2$)

$\Rightarrow u v^2 x y^2 z$

$\Rightarrow xxxxxxyyyyzzzzz$

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

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

$u = x$
$v = xx$
$x = xyyyyz$
$y = z$
$z = zz$

$\therefore$ The given language is not context free.

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

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

$n=4$ so $x^4 y^4 z^4$

$\Rightarrow \underbrace{xx}_{u}\underbrace{xxyy}_{v}\underbrace{y}_{x}\underbrace{y}_{y}\underbrace{zzzz}_{z}$

$\Rightarrow$ 5 parts

$uvxyz$

$u = xx$

$v = xxyy$

$x = y$

$y = y$

$z = zzzz$

$\Rightarrow$ Let assume $k=2$

$\Rightarrow u v^k x y^k z$

$\Rightarrow u v^2 x y^2 z$

$\Rightarrow xxxxyyxxyyyyyzzzz$

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

② $L = \{a^p \mid p$ 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 $p > n$.

Let $|vy| = k \leq n_0$

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

then a^p can be written as

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

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

$$a^{p-k} a^{k(p-k)} = a^{(p-k)(k+1)}$$

↳ see solution in nextpage

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
L must have a pumping length (say p)
Now take string s such that

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

devide string into 5 parts $uvxyz$

such that $|vxy| \leq n$, $|vy| \neq \epsilon$

eg take $p = 5$

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

$$\underbrace{000001}_{u}\ \underbrace{1\,1\,1}_{v\,x\,y}\ \underbrace{1000001111}_{z}$$

uv^2xy^2z Let $k=2$

0000011111110000011111

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

The given language is not context free.

$a^{p-k} \; a^{k(p-k)}$

$a^{p-k} \; a^{kp-k^2}$

$a^{p-k+kp-k^2}$

$a^{p-k+kp-k^2}$

$a^{(p+kp)-(k+k^2)}$

$a^{(p+kp)-(k+k^2)}$

$a^{p(1+k)-k(1+k)}$

$a^{(1+k)(p-k)}$

---

(4) $L = \{a^n b^n c^n : n \geq 0\}$ is not CFL

Assume L is ~~context~~ free

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

* $|w| \geq k$ we divid string uvxyz

$|vxy| \leq k$

$|vy| \neq 0$ At $q \geq 0$, $uv^q x y^q z$

Let $k=4$

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

$\underset{u}{a}\underset{v}{a}\underset{x}{a}\underset{y}{a}\underbrace{bbbbcccc}_{z}$

$|vxy| \leq 4$

$aaa \leq 4$

$3 \leq 4$

$|vy| \neq 0$

$2 \neq 0$

Let $q = 2$

uv^2xy^2z

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

aaaaaabbbbcccc

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

$\therefore$ not CFL

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

Assume L is context free

Let $w = a^k$

* $|w| \geq k$ ~~uvxyz~~ we divide string into uvxyz

$|vxy| \leq k$

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

Let $k = 2$

$w = a^{2^2}$

$= a a \epsilon a a$

$\ \ u\ v\ x\ y\ z$

$|vxy| \leq 2$

$a \epsilon a \leq 2$

$2 \leq 4$

Let $q = 2$

$uv^2 xy^2 z$

$aaaaaaa \notin L$

$\therefore$ not CFL

(6) $L = \{wcw : w \in \{a,b\}^*\}$ is not CFL

Assume L is context free

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

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

$|vxy| \leq k$

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

Let $k = 4$

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

$w = aaaa\underbrace{bbbb\,c\,aaaabbbb}_{z}$

$\ \ u\ v\ x\ y$

$|vxy| \leq 4$

$aaa \leq 4$

$3 \leq 4$

$|vy| \neq 0$

$2 \neq 0$

Let $q = 2$

$uv^2 xy^2 z$

$a(a)^2 a(a^2) c\,aaaa\,bbbb$

$aaaabbbb$

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

$\notin L$

$\therefore$ not CFL

Closure properties of CFL.

1) union, concatenation, Kleen star

If L_1 & L_2 are context free language & 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 G_3 and $S_3 \in (V_1 \cup V_2)$

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

$L_3 = L_1 \cup L_2$

(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 grammer G_4 and $S_4 \in (V_1 \cup V_2)$

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

$\therefore L_3 = L_1 \cdot L_2$

(iii) $G_5 = (V_1 \cup S_5, T_1, P_5\ S_5$

S_5 is a the start symbol of grammer G_5

$P_5 = P_1 \cup \{S_5 \to S_1 S_5 | \epsilon\}$

$T L_5 = L_s^*$