2/3/20

UNIT-IV
NORMAL FORMS

Context Free Grammar can be written in some Standard forms is known as Normal Forms. In CFG left side only non-terminals, right side combination of terminals & Non-terminals. then We need to normalize such grammar. These Normal Forms certain reductions on the production of CFG. then CFG contains fixed no.of terminals & non-terminals.

→ Normal Forms are categorized into 2 types

1) chomsky's Normal Form
2) Greibach Normal Form

1. chomsky's Normal Form (CNF):

The chomsky's Normal Form can be defined as

Non-terminal → Non-terminal, Non-terminal

Non-terminal → Terminal

Ex:- $A \rightarrow BC$

$B \rightarrow b$

Here, A, B, C are non-terminals, b is terminal

→ Before converting the grammar into CNF it should be in reduced form that means remove all the useless symbols, ε-productions & unit productions from it.

→ After that the given reduced grammar can be converted into CNF.

→ If the language has an empty string i.e, ε then the following ε-production is allowed in CNF.

$S \to \epsilon$

Here, S is the start symbol.

Conversion from CFG to CNF:-

Step 1:- eliminate useless symbols, ϵ-production & unit-productions

Step-2: eliminate terminal symbols in right hand side.

Step-3: Restrict number of non-terminals on the right hand side.

problems:-

1) convert the following grammar into CNF.

$P: S \to aAD$
$A \to aB \mid bAB$
$B \to b$
$D \to d$

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

$P': B \to b$
$D \to d$

$S \to aAD$
$S \to C_a AD \quad [C_a = a]$
$\boxed{S \to C_a C_c} \quad [AD = C_c]$

$A \to aB$
$\boxed{A \to C_a B} \quad [\because C_a = a]$

$A \to bAB \quad [b = C_b]$
$A \to C_b AB$
$\boxed{A \to C_b C_d} \quad [\because AB = C_d]$

$P': B \to b \quad S \to C_a C_c$
$D \to d \quad A \to C_a B \mid C_b C_d$

2) $S \to aAbB$

$A \to aA \mid a$

$B \to bB \mid b$

P^{1}: $A \to a$

$B \to b$

$S \to aAbB$

$S \to C_a A C_b B \quad \left[\because \begin{array}{l} C_a = a \\ C_b = b \end{array}\right]$

$S \to C_a A C_1 \quad (\because C_b B = C_1)$

$\boxed{S \to C_a C_2} \quad (\because A C_1 = C_2)$

$A \to aA$ $\quad$ $B \to bB$

$A \to C_a A$ $\quad$ $B \to C_b B$

P^{1}: $S \to C_a C_2$

$A \to C_a A \mid a$

$B \to C_b B \mid b$

3) $S \to 1A \mid 0B$

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

$B \to 0BB \mid 1$

$P^{1} \Rightarrow B \to 1$

$A \to 0$

$S \to 1A$

$S \to C_a A \quad [\because C_a = 1]$

$S \to 0B$

$S \to C_b B \quad [\because C_b = 0]$

$A \to 1AA$

$A \to C_a AA$

$A \to C_a C_c \quad (AA = C_c)$

$A \to 0S$

$A \to C_b S$

$B \to 0BB$

$B \to C_b BB$

$B \to C_b C_d$

P^{1}:- $S \to C_a A \mid C_b B$

$A \to C_a C \mid C_b S \mid 0$

$B \to C_b C_d \mid 1$

4) $S \to bA \mid aB$

$A \to bAA \mid aS \mid a$

$B \to aBB \mid bS \mid b$

P':- $A \to a$

$B \to b$

$S \to bA$

$S \to C_b A$ ($\because b = C_b$)

$S \to aB$

$S \to C_a B$ ($\because C_a = a$)

$A \to bAA$

$A \to C_b A A$

$A \to C_b C_c$ ($\because AA = C_c$)

$A \to aS$

$A \to C_a S$

$B \to aBB$

$B \to C_a BB$

$B \to C_a C_d$ ($\because BB = C_d$)

$B \to bS$

$B \to C_b S$

P':- $S \to C_b A \mid C_a B$

$A \to C_b C_c \mid C_a S \mid a$

$B \to C_a C_d \mid C_b S \mid b$

5. Design a CFG for language

$L = \{a^{4n} \mid n \geq 1\}$ and convert CFG $\to$ CNF

$n=1$ $L = \{a^4\} = \{a,a,a,a\}$

$n=2$ $L = \{a^8\} = \{a,a,a,a,a,a,a,a\}$

CFG:- $S \to aaaaS \mid \epsilon$

$S \to aaaa$

P':- $S \to \epsilon$

$S \to aaaaS$

$S \to C_a C_a S \quad (\because aa = C_a)$

$S \to C_b S \quad (\because C_a C_a = C_b)$

$S \to \epsilon \mid C_b S \mid C_a C_a$

6) $S \to ASA \mid aB$

$A \to B \mid S$

$B \to b \mid \epsilon$

Remove ϵ- production.

$S \to ASA \mid a$

$A \to S$

$B \to b$

Remove useless symbols

$S \to ASA \mid a$

$A \to S$

Remove unit production

$S \to a \mid ASA$, $A \to ASA \mid a$

$S \to ASA$

$S \to C_a A \quad (\because AS = C_a)$

P':- $S \to C_a A \mid a$

$A \to C_a A \mid a$

7) $S \to aSa \mid bSb \mid a \mid b$

$P' :- S \to a$

$S \to b$

$S \to aSa$

$S \to C_a S C_a$ ($\because C_a = a$)

$S \to C_b C_a$ ($\because C_b = C_a S$)

$S \to bSb$

$S \to C_c S C_c$ ($\because C_c = b$)

$S \to C_d C_c$ ($\because C_d = C_c S$)

$P' :- S \to C_b C_a \mid C_d C_c \mid a \mid b$

8) $S \to aAS \mid a$

$A \to SbA \mid SS \mid ba$

$P' :- S \to a$

$A \to SS$

$S \to aAS$

$S \to C_a AS$ ($\because a = C_a$)

$S \to C_a C_b$ ($\because AS = C_b$)

$A \to SbA$

$A \to SC_c A$ ($\because C_c = b$)

$A \to C_d A$ ($\because SC_c = C_d$)

$A \to ba$

$A \to C_a C_c$

$P^!$:- $S \to C_a C_b$

$A \to C_d A \mid SS \mid C_a C_c$

9) $S \to aB$

$S \to bA$

$A \to a$

$A \to aS$

$A \to bAA$

$B \to b$

$B \to aS$

$B \to aBB$

CNF

$N \to NN$

$N \to T$

$P^!$:- $A \to a$

$B \to b$

$S \to aB$

$S \to C_a B$ $(\because C_a = a)$

$S \to bA$

$S \to C_b A$ $(\because C_b = b)$

$A \to aS$

$A \to C_a S$

$A \to bAA$

$A \to C_b AA$ $(\because C_c = AA)$

$A \to C_b C_c$

$B \to aS$

$B \to C_aS$

$B \to aBB$

$B \to C_aBB \quad (\because C_d = BB)$

$B \to C_aC_d$

P^1 :- $S \to C_aB$

$S \to C_bA$

$A \to a$

$A \to C_aS$

$A \to C_bC_c$

$B \to C_aS$

$B \to b$

$B \to C_aC_d$

10) $S \to ABA$

$A \to aA \mid \epsilon$

$B \to bB \mid \epsilon$

Remove ϵ-production

$S \to ABA \mid AB \mid BA \mid AA \mid A \mid B$

$A \to aA \mid a$

$B \to bB \mid a$

$A \to C_aA \quad (\because C_a = a)$

$B \to C_bB \quad (\because C_b = b)$

$S \to ABA$

$S \to C_cA \quad (\because C_c = AB)$

P^1 :- $S \to C_cA \mid AB \mid BA \mid AA \mid a \mid b$

$A \to C_aA \mid a$

$B \to C_bB \mid b$

Scanned with CamScanner

3/3/20

GNF: Greiback Normal Form.

A CFG is in GNF if all the production rules satisfies these conditions.

i) Start symbol generating epselon 'ϵ'

$S \rightarrow \epsilon$

ii) A Non-terminal generating a terminal

$A \rightarrow a$

iii) A Non-terminal generating a terminal which is followed by any no-of non-terminals

$A \rightarrow aBCDE \cdots$

Ex:- G1: $S \rightarrow aAB \mid aB$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

It is in form of GNF. since it following all conditions of GNF

Ex:- G2: $S \rightarrow aAB \mid aB$

$A \rightarrow aA \mid \epsilon$

$B \rightarrow bB \mid \epsilon$

It is not in the form of GNF since Non-terminal $A \rightarrow \epsilon$

Steps to convert CFG to GNF

1. check if the given grammar has any useless productions, null productions and unit productions and remove if they exist.

2. check whether the grammar is in the form of CNF or not. If it is not then convert into CNF.

3. change the names of the non-terminal symbols into some A_i in ascending order of i.

4. After Alter the rules with the new non-terminal symbols such that if the production is in the form of $A_i \rightarrow A_j X$ then $i < j$ and should never be $i \geq j$

→ If $i \geq j$ then replace the non-terminals with other productions.

→ If $i = j$ then there is a possibility of left recursion.

5. Remove left recursion. to remove it

6. To remove the left recursion from grammar we need to follow some rules

→ $A \rightarrow A\alpha_1 | A\alpha_2 -- | \beta_1 | \beta_2 ----$

Here, now new terminal introduced that is "Z"

A - production:

$A \rightarrow \beta_1 | \beta_2 ---$

$A \rightarrow \beta_1 Z | \beta_2 Z ---$

Z - production:

$Z \rightarrow \alpha_1 | \alpha_2 ---$

$Z \rightarrow \alpha_1 Z | \alpha_2 Z ---$

7. After removing left recursion. convert remaining productions into GNF format.

1) convert the following grammar into GNF:

P: $S \rightarrow AA | a$

$A \rightarrow SS | b$

Step 1: In the given grammar no useless symbols, no unit productions, no ϵ productions

Step 2: The given grammar is already in form of CNF

3: Replace the Non-terminals with new non-terminals i.e

$S \rightarrow A_1$

$A \rightarrow A_2$

4: Replace these new non-terminals in the given productions

i.e, $A \rightarrow A_2 A_2 | a$

$A_2 \rightarrow A_1 A_1 | b$

Scanned with CamScanner

4. The given production is the form of

$A_i \rightarrow A_j X$

→ for first production:

$A_1 \rightarrow A_2 A_2$ here $i=1$, $j=2$, $i<j \Rightarrow 1<2$

it satisfies the given condition then there is no problem with the production

→ 2^{nd} production:- $A_2 \rightarrow A_1 A_1$ here $i=2$, $j=1$

$i>j$ then replace the A_1 with the A_1 production (1st production) i.e,

$A_1 \rightarrow A_2 A_2 / a$

→ After replacing A_1 in the A_2 we will get this production $A_2 \rightarrow A_2 A_2 A_1 | aA_1 | b$

but in the above production $i=j$ then there is a posibility of left-recursion.

5. eliminate the left-recursion from the A_2 production i.e $\underset{A}{A_2} \rightarrow \underset{A}{A_2}\underset{\alpha}{A_2 A_1} | \underset{\beta_1}{aA_1} | \underset{\beta_2}{b}$

here A is A_2, $\alpha = A_2 A_1$, $\beta_1 = aA_1$, $\beta_2 = b$

A-production :- $\left\{ \begin{array}{l} A \rightarrow \beta_1 | \beta_2 \\ A \rightarrow \beta_1 Z | \beta_2 Z \end{array} \right\}$

i.e A_2-production:- $A_2 \rightarrow aA_1 | b$

$A_2 \rightarrow aA_1 Z | bZ$

Z-production :- $\left\{ \begin{array}{l} Z \to \alpha_1 \\ Z \to \alpha_1 Z \end{array} \right\}$

i.e, $Z \to A_2 A_1$

$Z \to A_2 A_1 Z$

→ finally after removing left recursion A_2 productions are in the form of GNF but A_1 production, Z productions are not in the form of GNF.

→ then Replace A_2 productions in the A_1 & Z productions

→ after replacing :- $A_1 \to A_2 A_2 \mid a$

$\left\{ \begin{array}{l} A_2 \to aA_1 \mid b \\ A_2 \to aA_1 Z \mid bZ \end{array} \right\}$

* $A_1 \to aA_1A_2 \mid bA_2 \mid aA_1ZA_2 \mid bZA_2 \mid a$

replace A_2 in Z production ⇒ $\left\{ \begin{array}{l} Z \to A_2A_1 \\ Z \to A_2A_1Z \end{array} \right\}$

* $Z \to aA_1A_1 \mid bA_1 \mid aA_1ZA_1 \mid bZA_1$

* $Z \to aA_1A_1Z \mid bA_1Z \mid aA_1ZA_1Z \mid bZA_1Z$

finally the grammar $G' = (Q', T, P', A_1)$

$Q' = \{A_1, A_2, Z\}$ $T = \{a, b\}$

$P^1 :\rightarrow A_1 \rightarrow aA_1A_2 \mid bA_2 \mid aA_1ZA_2 \mid$
$bZA_2 \mid a$

$A_2 \rightarrow aA_1 \mid b$

$A_2 \rightarrow aA_1Z \mid bZ$

$Z \rightarrow aA_1A_1 \mid bA_1 \mid aA_1ZA_1 \mid bZA_1$

$Z \rightarrow aA_1A_1Z \mid bA_1Z \mid aA_1ZA_1Z \mid$
bZA_1Z

2) consider the grammar and convert it into equivalent GNF

$S \rightarrow CA \mid BB$

$B \rightarrow b \mid SB$

$C \rightarrow b$

$A \rightarrow a$

Step:-1 : no useless symbols, no ϵ- productions, no unit productions.

2 : It is in CNF.

3 : Replacing non-terminals with A_i i.e,

$S \rightarrow A1$

$A \rightarrow A2$

$B \rightarrow A3$

$C \rightarrow A4$

$\rightarrow$ after replacing the grammer is

$A_1 \to A_4 A_2 \mid A_3 A_3$

$A_3 \to b \mid A_1 A_3$

$A_4 \to b$

$A_2 \to a$.

2) $S \to A$

$A \to aBa \mid a$

$B \to bAb \mid b$

3) $S \to AB$

$A \to BSB$

$A \to a$

$B \to b$

4) $A_1 \to A_2 A_3$

$A_2 \to A_3 A_1$

b

$A_3 \to A_1 A_2$

a

A-> The given production is in form of $A_i \to A_j X$

In 1^{st} production i.e $A_1 \to A_4 A_2$, $i=1, j=4$

$i<j$ it satisfies the given condition

$A_1 \to A_3 A_3$, $i=1, j=3$ $i<j$.

2^{nd} production:=> $A_3 \to A_1 A_3$, $i=3, j=1$

$i>j$.

replacing A_1 with A_1-production i.e

$A_1 \to A_4 A_2 / A_3 A_3$ in A_3 then

$A_3 \to A_4 A_2 A_3 /$ $i=3, j=4$ $i<j$

$A_3 A_3 A_3 / b$

$A_3 \to A_4 A_2 A_3$

$i=3$ $j=4$ $i<j$

replacing A_4 with A_4-production

$A_3 \to b A_2 A_3$

it is in GNF

$\underbrace{A_3}_{A} \to \underbrace{A_3}_{A} \underbrace{A_3 A_3}_{\alpha} \mid \underbrace{b}_{\beta}$

$\cancel{A_3 \to A}$

$\underbrace{A_3}_{A} \to \underbrace{A_3}_{A} \underbrace{A_3 A_3}_{\alpha} \mid \underbrace{b A_2 A_3}_{\beta_1} \mid \underbrace{b}_{\beta_2}$

Scanned with CamScanner

A-production : Z: production

$A \to \beta$ $Z \to \alpha$

$A \to \beta Z$ $Z \to \alpha Z$

$Z \to A_3 A_3$ $Z \to bA_2A_3A_3$

$A_3 \to bA_2A_3 | b$ $Z \to A_3A_3Z$ $Z \to bA_2A_3A_3Z$

$A_3 \to bA_2A_3Z | bZ$

$A_1 \to A_4A_2 | A_3A_3$

$A_1 \to bA_2 | bA_3 | bA_2A_3A_3$

$P' : \to A_1 \to bA_2 | bA_3 | bA_2A_3A_3 | bA_2A_3ZA_3 | bZA_3$

$A_3 \to bA_2A_3 | b$

$A_3 \to bA_2A_3Z | bZ$

$Z \to bA_2A_3A_3 | bA_2A_3A_3Z | bA_3 | bZA_3 | bA_2A_3ZA_3 | bA_2A_3A_3Z | bA_3Z | bZA_3Z$

$A_2 \to a$

$A_4 \to b$

another

$\to$ $S \to A_1$ $C \to A_2$ $A \to A_3$ $B \to A_4$

after replacing in grammar

$A_1 \to A_2A_3 | A_4A_4$

$A_4 \to b | A_1A_4$

$A_2 \to b$

$A_3 \to a$

$A_1 \to A_2 A_3$ $\quad$ $A_1 \to A_4 A_4$

$i < j$ (no problem) $\quad$ $i < j$ (no prob)

$A_4 \to A_1 A_4$

$i > j$

Replacing A_1 production

$A_4 \to A_2 A_3 A_4 \mid A_4 A_4 A_4 \mid b$

$A_4 \to A_2 A_3 A_4$

$i > j$

$$\underbrace{A_4}_{A} \to \underbrace{A_4}_{A} \underbrace{A_4 A_4}_{\alpha} \mid \underbrace{b A_3 A_4}_{\beta_1} \mid \underbrace{b}_{\beta_2}$$

$A_4 \to b A_3 A_4$

GNF

A- production

$A \to \beta \quad A \to \beta Z$

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

Z: production

$Z \to \alpha \quad Z \to \alpha Z$

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

$P' \to \quad A_1 \to b A_3 \mid b A_4 \mid b A_3 A_4 A_4 \mid b A_3 A_4 Z A_4$

$A_2 \to b$

$A_3 \to a$

$A_4 \to b A_3 A_4 \mid b A_3 A_4 \mid b A_4 A_4 \mid$

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

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

2) $S \to A$

$A \to aBa | a$

$B \to bAb | b$

1) unit production

$S \to A$

$S \to aBa | a$

$B \to bAb | b$

$A \to aBa | a$

2) converting to CNF

$S \to aBa$

$S \to C_aBa$ {∵ $a = C_a$}

$\boxed{S \to C_aC_b} | a$ [$BC_a = C_b$]

$B \to bAb | b$

$B \to C_cAb$ (∵ $b = C_c$)

$\boxed{B \to C_cC_d | b}$ ($AC_c = C_d$)

$A \to aBa | a$

$\boxed{A \to C_aC_b | a}$

It is in CNF

$S \to C_aC_b | a$

$A \to C_aC_b | a$

$B \to C_cC_d | b$

3)

$S \to A_1$ $C_a \to A_4$

$A \to A_2$ $C_b \to A_5$

$B \to A_3$ $C_c \to A_6$

$C_d \to A_7$

3

$A_1 \to A_4A_5 | a$

$A_2 \to A_4A_5 | a$

$A_3 \to A_6A_7 | b$

$A_5 \to A_3A_4$ ($C_b = BC_a$)

$A_7 \to A_2A_6$ ($C_d = AC_c$)

$A_6 \to b$; $A_4 \to a$

4) $A_1 \to A_4A_5$ ($i<j$)

$A_2 \to A_4A_5$ ($i<j$)

$A_3 \to A_6A_7$ ($i<j$)

$A_5 \to A_3A_4$ ($i>j$)

$A_5 \to A_6A_7A_4 | bA_4$

$\boxed{A_5 \to bA_7A_4 | bA_4}$

$A_7 \to A_2A_6$

$A_7 \to A_4A_5A_6 | aA_6$

$\boxed{A_7 \to aA_5A_6 | aA_6}$

3) $S \to AB$

$A \to BSB$

$A \to a$

$B \to b$

i) converting to CNF

$\boxed{S \to AB}$

$A \to BSB$

$\boxed{A \to CB}$ $(BS = C)$

$\boxed{A \to a}$

$\boxed{B \to b}$

ii) Replacing with Non-terminals

$S \to A1$ $B \to A3$

$A \to A2$ $C \to A4$

$A1 \to A2A3$

$A2 \to A4A3$

$A2 \to a$

$A3 \to b$

$A4 \to A3A1$

iv) $A_1 \to A_2A_3 (i<j)$

$A_2 \to A_4A_3 (i<j)$

$A_4 \to A_3A_1 (i>j)$

$A_4 \to bA_1$

$A_2 \to a$

$A_3 \to b$

$A_1 \to aA_3$

$A_2 \to A_3A_1A_3$

$A_2 \to bA_1A_3$

4) $A_1 \to A_2A_3$

$A_2 \to A_3A_1 | b$

$A_3 \to A_1A_2 | a$

$A_1 \to A_2A_3 (i<j)$

$A_2 \to A_3A_1 | b (i<j)$

$A_3 \to A_1A_2 (i>j)$

$A_3 \to A_2A_3A_2$

$A_3 \to A_3A_1A_3A_2 | bA_3A_2$

$\underbrace{A_3}_{A} \to \underbrace{A_3}_{A}\underbrace{A_1A_3A_2}_{\alpha} | \underbrace{bA_3A_2}_{\beta}$

A: production:

$A \to \beta$ $A_3 \to bA_3A_2$

$A \to \beta Z$ $A_3 \to bA_3A_2Z$

Z: production:

$Z \to \alpha$ $Z \to A_1A_3A_2$

$Z \to \alpha Z$ $Z \to A_2A_3A_3A_2$

$Z \to bA_3A_3A_2 |$

$A_3A_1A_3A_3A_2 |$

aA_1A_3

6/3/20

pumping Lemma:

pumping Lemma is used to prove the given Language is not context free Language.

→ L is a context Free Language, there is a pumping Length 'n' such that any string $z \in L$ then the Length of string $|z| \geq n$. Here n is the natural number.

→ Here we can break $z = uvwxy$

→ Given conditions i) $|vx| \geq 1$ iii) $uv^k wx^k y \in L$

ii) $|vwx| \leq n$ for all $k \geq 0$

Steps to prove that given language is not CFL [We are proving using contradiction]

1) Assume that L is context Free Language

2) The pumping length is n

3) All strings longer than the n then only we are able to pump the string. where $|z| \geq n$

4) Now find a string in L. where $|z| \geq n$

5) divide z into 5 parts uvwxy

6) show that $uv^k wx^k y$ is not belong to L for some k.

7) Then consider the different ways that z can be divided into uvwxy

8) Then z cannot be pumped.

① Find out whether the language $L = \{x^n y^n z^n / n \geq 1\}$ is CFL or not

$L = \{xyz, xxyyzz, xxxyyyzzz, \ldots\}$

$z = xxxx\,yyyy\,zzzz$ let $n = 10$

$|z| \geq n$ $|12| \geq 10$

Case 1: v, x each contain only one type of symbols

$$\underline{xxxxyyyyzzzz}$$
$u\ v\ w\ x\ y$

$u = xx \quad v = xx$
$w = yyyyz \quad x = z$
$y = zz$

let
$k = 2 \Rightarrow uv^2wx^2y$
$\Rightarrow xx\,xx\,xx\,yyyyz\,zzzz \Rightarrow x^6y^4z^5 \notin L$

case 2: v, x has different kinds of symbols

$$\underline{xxxxyyyyzzzz}$$
$u\ v\ w\ x\ z$

$u = xxx \quad v = xy$
$w = yy \quad x = yz$
$y = zzz$

$k = 2$
$uv^2wx^2y \Rightarrow xxxxyxyyyyzyzzzz$
$x^4y\,xy^4zyz^4 \notin L$

∴ It is not CFL

2) show that $L = \{ww \mid w \in (0,1)^*\}$ is not CFL

$L = \{00, 11, 0101, 001001, 1010, 011011, \ldots\}$

let $z = 001001$

case 1:- $\underbrace{00}_{u}\underbrace{1}_{v}\underbrace{0}_{w}\underbrace{0}_{x}\underbrace{1}_{y}$ $\quad u=00,\ v=1,\ w=0,\ x=0,\ y=1$

$k=2 \Rightarrow uv^2wx^2y \Rightarrow 00110001 \notin L$

case 2:- $\underbrace{0}_{u}\underbrace{01}_{v}\underbrace{0}_{w}\underbrace{0}_{x}\underbrace{1}_{y}$ $\quad u=0,\ v=01,\ w=0,\ x=0,\ y=1$

$k=2 \Rightarrow uv^2wx^2y \Rightarrow 001010001 \notin L$

∴ It is not CFL

3) $L = \{ss^T \mid s \in (a,b)^*\}$

$L = \{abba, aabbaa, abbbba, \ldots\}$

$z = ababbbbaba$

case:- 1 :- $\underset{u}{ab}|\underset{v}{a}|\underset{w}{bb\ bba}|\underset{x}{b}|\underset{y}{a}$

$k=2 \Rightarrow uv^2wx^2y \Rightarrow abaabbbbaaba$

$abaabbbbabba \notin L$

case:- 2 $\underset{u}{a}|\underset{v}{ba}|\underset{w}{bb\ b}|\underset{x}{ba}|\underset{y}{b}$

$k=2$

$\rightarrow uv^2wx^2y \Rightarrow abababbbbababab \notin L$

$\therefore$ It is not CFL

4) $L = \{a^n b^{2n} a^n \mid n \geq 0\}$

$L = \{\epsilon, abba, aabbbbaa, ---\}$

$z = aabbbbaa$

case 1:- $\underset{u}{a}|\underset{v}{a}|\underset{w}{bbb}|\underset{x}{b}|\underset{y}{aa}$

$k=2$ $uv^2wx^2y \Rightarrow aaabbbbbaa \notin L$

case 2:- $\underset{u}{a}|\underset{v}{ab}|\underset{w}{bb}|\underset{x}{ba}|\underset{y}{a}$

$k=2$ $uv^2wx^2y \Rightarrow aababbbbabaa \notin L$

$\therefore$ It is not CFL

5) $L = \{a^i b^j c^k \mid i<j<k\}$

$L = \{abbccc, aabbbcccc, ---\}$

$z = aabbbcccc$

case:- 1 $\underset{u}{aa}|\underset{v}{bb}|\underset{w}{bc\ c}|\underset{x}{c}|\underset{y}{c} \Rightarrow$ $k=2$ uv^2wx^2y

~~abbbcccc~~

$aabbbbbccccc \notin L$

Case:-2 $\underbrace{a}_{u}\underbrace{a}_{v}\ \underbrace{bbb\,cc}_{w}\underbrace{c}_{x}\underbrace{c}_{y}$

$k=2 \Rightarrow uv^2wx^2y \Rightarrow aababbbcccccy \notin L$

$\therefore$ It is not CFL

closure properties of context free languages:

1. closed under union.

→ If L_1 & L_2 are context free languages then $L = L_1 \cup L_2$ is also context free language i.e

CFLs are closed under union.

proof:- We will consider 2 languages L_1 & L_2 which are CFL

⇒ We can give these languages using CFG's G_1 and G_2 such that $G_1 \in L_1$, $G_2 \in L_2$

The G_1 can be given as

$G_1 = \{V, \Sigma, P_1, S_1\}$ where P_1 can be given as

$P_1 = \{ S_1 \rightarrow A_1 S_1 A_1 \mid B_1 S_1 B_1 \mid e$

$A_1 \rightarrow a$

$B_1 \rightarrow b \}$

Here $V_1 = \{S_1, A_1, B_1\}$ and $\Sigma = \{a, b\}$ and

S_1 is start symbol.

Similarly, $G_2 = (V_2, \Sigma, P_2, S_2)$

$V_2 = \{S_2, A_2, B_2\}$ and S_2 is start symbol

$P_2 = \{ S_2 \rightarrow a A_2 A_2 \mid b B_2 B_2$

$A_2 \rightarrow b$

$B_2 \rightarrow a$

$\}$

Now, $L = L_1 \cup L_2$ gives $G \in L$

This G can be written as $G = \{V, \Sigma, P, S\}$

$V = \{S_1, A_1, B_1, S_2, A_2, B_2\}$

$P = \{P_1 \cup P_2\}$ $\Sigma = \{a, b\}$

then S is start symbol

$P = \{S \to S_1 | S_2$

$S_1 \to A_1 S_1 A_1 | B_1 S_1 B_1 | \epsilon$

$A_1 \to a$ $\quad S_2 \to a A_2 A_2 | b B_2 B_2$

$B_1 \to b$ $\quad A_2 \to b$

$B_2 \to a \}$

∴ Thus, G is CFG

2) If L_1 & L_2 are two CFL then $L_1 L_2$ is CFG. That means CFL's are closed under concatenation.

proof:- Let L_1 is CFL which can be represented by a CFG is G_1 such that $G_1 \in L_1$ and $G_1 = \{V_1, \Sigma, P_1, S_1\}$

$V_1 = \{S_1, A_1, B_1\}$, $\Sigma = \{a, b\}$

S_1 is a start symbol

and P_1 is a set of production rules

$P_1 = \{S_1 \to A_1 S_1 A_1 | B_1 S_1 B_1 | \epsilon$

$A_1 \to a$

$B_1 \to b\}$

Similarly, L_2 is CFL which can be represented by a CFG is G_2. Such that $G_2 \in L_2$ and

$G_2 = \{V_2, \Sigma, P_2, S_2\}$

$V_2 = \{S_2, A_2, B_2\}$

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

S_2 is a start symbol.

P_2 is set of production rules

$$P_2 = \{S_2 \rightarrow aA_2A_2 \mid bB_2B_2$$
$$A_2 \rightarrow b$$
$$B_2 \rightarrow a\}$$

Now $L = L_1 \cdot L_2$ can be obtained by G such that $G = G_1 \cdot G_2$. Therefore,

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

$V = \{S, S_1, A_1, B_2, S_2, A_2, B_2\}$

Where S is a start symbol. Then the production rule P. can be given as

$$P = \{S \rightarrow S_1 \mid S_2$$
$$S_1 \rightarrow A_1S_1A_1 \mid B_1S_1B_1 \mid \epsilon$$
$$A_1 \rightarrow a$$
$$B_1 \rightarrow b$$
$$S_2 \rightarrow aA_2A_2 \mid bB_2B_2$$
$$A_2 \rightarrow b$$
$$B_2 \rightarrow a\}$$

As grammar 'G' is CFG the language 'L' produced by 'G' is CFL.

Hence, CFL's are closed under concatenation

If L_1 is CFL

Concatenation:- $L_1 \to G_1$
$L_2 \to G_2$

$L_1 \cdot L_2 \to G = \{V, \Sigma, P, S\}$

$G_1 = \{S_1 \to aA$
$A \to a\}$

$G_2 = \{S_2 \to bB$
$B \to b\}$

$S \to S_1 \cdot S_2$
$S_1 \to aA$
$A \to a$
$S_2 \to bB$
$B \to b$

3) If L_1 is CFL then L_1^* is also CFL.

that means CFL is closed kleen closure.

$L_1 =$
$G_L = \{V_1 \cup (S), T_1 \cup \Sigma, S_1, P_1 \cup S \to S_1 S \mid e\}$

proof:- Let L be CFL represented by G_1

such that $G_1 \Rightarrow \in L_1$

The CFG G_1 can be given as

$G_1 = \{V_1, \Sigma, P, S_1\}$ where S_1 is start symbol

$V_1 = \{S_1, A_1, B_1\}$ $\Sigma = \{a, b\}$

$P_1 = \{S_1 \to A_1 S_1 A_1 / B_1 S_1 B_1 / \epsilon$
$A_1 \to a$
$B_1 \to b\}$

Scanned with CamScanner

Now $L = L_1^*$ can be represented by a Grammar G such that $G = \{V, \Sigma, P, S\}$

$V = \{S_2, S_1, A_1, B_1\}$ and $P = S \rightarrow S_1 S | \epsilon$

$S_1 \rightarrow A_1 S_1 A_1 | B_1 S_1 B_1$

$A_1 \rightarrow a$

$B_1 \rightarrow b \}$

Thus Grammar G is a CFG and language produced by G is also CFG.

Hence CFL are closed under kleen closure.

4) If L_1 & L_2 are two CFL's then $L = L_1 \cap L_2$ may be CFL (or) may not be CFL. That means L is not closed under intersection

let $L_1 = \{a^n b^n c^m \mid n, m \geq 0\}$

$S_1 \rightarrow S_1 C$

$S_1 \rightarrow a S_1 b | \epsilon$

$C \rightarrow cC | \epsilon$

$L_2 = \{a^n b^m c^m \mid n, m \geq 0$

$S \rightarrow S_2 A$

$S_2 \rightarrow S_2 bc | \epsilon$

$A \rightarrow aA | \epsilon$

then $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$ is not CFL

$\therefore$ CFG are not closed under Intersection.

5) If L_1 is CFL then L_1' may (or) may not be CFL. That means CFL is not closed under complement

i.e $L_1 \cap L_2 = \overline{L_1}$

$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$

L_1 & L_2 are CFL. $\overline{L_1}$ & $\overline{L_2}$ may not be CFL

Decision properties of CFL:-

→ Given languages, specified in any one of the four means we can develop algorithms that answer the following questions.

1. Is a given string in the language?
2. Is the language empty?
3. Is the language finite?

(1) Emptiness test

a) Generating test

b) Reachability test

(2) Membership test

PDA acceptance

(3) Finiteness Test

(4) Infiniteness testing

Emptiness: Remove useless symbols and production. If S is useless, then $L(G)$ is Empty.

-) By the proof of the pumping lemma, if a grammar is CNF has p states, the longest string, not subject to the pumping lemma will have length.

$$n = 2^{p+1}$$

Systematically generate all strings with lengths less than n:

Test each one using membership algorithm if all fail and epsilon not equal to L. then L is empty else L is not empty.

Finiteness:

Just as with RL's a language is infinite if there is a string x with length between 'n' and '2n'

with RL's n = no. of states in an FA.

with CFL's $n = 2^{p+1}$ where p is the number of variables in the CFG.

Systematically generate all strings with lengths between n and 2n.

Run through membership alg if one passes, L is infinite, if all fail, L is finite

Sad facts about CFL's

There is no "algorithm" to determine if given a grammar G, G is ambiguous.

Membership:

Unlike FA's we can't just run the string through the machine and see where it goes. Since PDA's are non-deterministic.

It must consider all possible paths. Instead, start with your grammar in CNF.

The proof of the pumping lemma states that the longest derivation path of a string size of 'n' will be '2n-1'.

Systematically generate all derivation with one step, then two steps, ---- then 2n-1 steps. Where the length of the string tested = n. If one of the derivation derive x, return true else return false.

Infiniteness testing:

The idea is like that for regular languages look for 'loop' in CNF grammar

Undecidable problems of CFL:

→ Is a given CFG 'G' is ambigous?

→ Is a given CFL inherently ambigous?

→ Is the intersection of two CFL's empty?

→ Are two CFLs the same!

→ Is a given L(G) equal to Σ^*?

complexity of converting among CFG & PDA:-

→ There is an algorithm to convert a given CFG to an equivalent PDA.

→ The PDA guesses the left most derivation.

→ There is an algorithm to convert PDA to CFG and that the conversion is more complex.

12/3/20

Two stack PDA: It is equivalent to turing Machine

Instead of single stack we are going to use two stacks to accept the given language.

→ Two stack PDA contains 6 tuples

$\{Q, \Sigma, \Gamma, \delta, q_0, F\}$

Here Z_0 is not there.

$\delta : (Q \times (\Sigma \cup \epsilon) \times \Gamma \times \Gamma) \rightarrow Q \times \Gamma^* \times \Gamma^*$

construct a 2 stack PDA for the given language $L = \{a^n b^n c^n \mid n \geq 1\}$