20/12/2024

MODULE - 5

* Context Free language
  → properties of CFL
    ↳ Normal Forms (CNF, GNF)
  → Pumping lemma for CFL's
  → closure properties of CFL

* Turing Machines
  → Programming Techniques for Turing Machines
  → Multitape Turing Machines

Context Free language

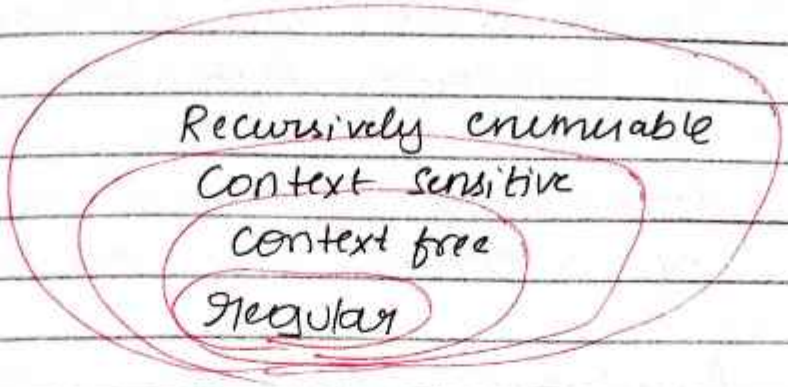
CFL are generated by CFG. The set of all context free lang is identical to the set of languages accepted by pushdown automata, and the set of Regular languages is a subset of CFL.

An inputed lang is accepted by a computational model if it runs through the model and ends in an accepting final state.

All regular languages are CFL but not all CFL are regular. Most arithmetic expressions are generated by CFG, and are therefore CFL.

DATE / /
PAGE NO. | Usha Gold

CFL and CFG have applications in computer science and linguistics such as natural language processing and computer language design



In formal language, a language is defined as a set of strings of symbols that may be constrained by specific rules.

Similarly, the written english lang must follow particular rules, the grammar

Ex: An example of a language that is not regular (proof here) but is context free.

$\{a^n b^n \mid n > 0\}$.

This is the language of all strings that have an equal number of a's & b's

In this notation $a^4 b^4$ can be expanded out to aaaabbbb, where there are four a's and then four b's (so this isn't exponentiation, though the notation is similar)

Properties of CFL

CFL is a language which is generated by a CFG or Type 2 grammar (according to chomsky classification) and get accepted by a pushdown Automata

Some very much important properties of a context free language is

Regularity - CFL are non-Regular PDA language

Closure properties

A set is closed under an operation if doing the operation on a given set always produces a member of the same set.

This means that if one of these closed operations is applied to a context-free language the result will also be a CFL.

Union :- CFL are closed under the union operation. This means that if L and P are both CFL then LUP is also a CFL

Here is a proof that CL

DATE / /
PAGE NO: Usha Gold

Concatenation:- If L and P are both CFL, then LP is also CFL.
The concatenation of a string is defined as follows
$S_1S_2 = vw : v \in S_1$ and $w \in S_2$.

Kleene Star:- If L is a CFL, then L^* is also CFL. The Kleene star can repeat the string or symbol it is attached to any number of times. (including zero times).
The Kleene star basically performs a recursive concatenation of a string with itself.
For example: $\{a, b\}^* = \{\epsilon, a, b, ab, aab, aaab, abb, abbb \cdots\}$ and so on. We've

CFL are not closed under some specific operation, not closed means after doing that operation on a context free language the resultant language not remains be a CFL anymore
Some such operations are
1) Intersection
2) complement
3) subset
4) superset
5) Infinite union
6) Difference, Symmetric Difference.

Pumping lemma for CFL

The pumping lemma for CFL is useful for

a) Generating an infinite number of strings from a given sufficiently long string

b) To prove that certain languages are not context free.

To prove

⇒ Let L be any CFL, then there exist an integer m such that, for any string $w \in L$, with $|w| \geq m$, the string w can be split into five parts as $w = uvxyz$ such that

a) $|vy| \geq 1$

b) $|vxy| \leq m$ → pumping length

c) For all $i \geq 0$ $uv^ixy^i \in L$

Problems

1) Show that $L = \{a^n b^n c^n \mid n \geq 1\}$ is not a CFL

Soln

Let L be a CFL

Let n be a constant

Let $z = a^n b^n c^n$

Split $z = uvxyz$

$u = a^n$

$vxy = b^n$ $\quad |vxy| \leq m$

$z = c^n$

DATE / /
PAGE NO. Usha Gold

$vy = b^{m+n}$, $|vx| \geq 1$

$y = c^n$

$uv^i xy^i z = uvv^{i-1} xyy^{i-1} z$

$= uv x (vy)^{i-1} z$

$= a^n b^n (b^{n-m})^{i-1} c^n$

$= a^n b^n b^{ni-n-mi+m} c^n$

$= a^n b^{n+ni-n-mi+m} c^n$

$=$

$i = 0 \rightarrow uv^i xy^i z = a^n b^m c^n$ subt $i = 0$

Contradiction

$uv^i xy^i z = a^n b^m c^n \notin L$

Hence the given L is not a CFL

[OR]

$p = 4$ so $a^4 b^4 c^4$

case 1: v and y each contain only one type of symbol

a a a a b b b b c c c c

u v x y z

$uv^i xy^i z$

$i = 2$

$uv^2 xy^2 z$

aaaaa abbbbc cc cc

$a^6 b^4 c^5 \notin L$

2) Show that $L = \{ww \mid w \in (0,1)^*\}$ is NOT context Free

Soln Assume that L is CF
L must have a pumping length P
Now we take a string S such that $S = 0^P 1^P 0^P 1^S$
we divide S into parts
$uvxyz$

Case 1: vxy does not straddle a boundary (lies in middle)

$P = 5$

$\underbrace{00000}_{x}\underbrace{1111}_{vxy}\underbrace{100000\,11111}_{z}$

$i = 2$

000001111111000001111

$0^5 1^6 0^5 1^5$

$\neq$

Case 2: vxy straddles the first boundary

$\underbrace{00000}_{u}\underbrace{11}_{v}\underbrace{111}_{x}\underbrace{00000\,11111}_{y\quad z}$

$uv^ixy^iz \qquad i = 2$

uv^2xy^2z

$\underbrace{000}_{u}\underbrace{0001}_{v}\underbrace{111}_{x}\underbrace{1100000\,11111}_{y\quad z}$

$0^7 1^7 0^5 1^5$ contradiction hence the given L is not CFL

$\neq$

case 2b: vxy straddles third boundary

00000 11111 00 0 00 1111 1
v x y z

$uv^i xy^i z$

$i=2 \quad uv^2xy^2z$

00000 11111 000 0000 1 1111 11
u v x y z

$0^5 1^5 0^7 0^7$

$\notin$

3) Show that the $L = \{wwR w \mid w \in \{0,1\}^*\}$ is not context free

$S = \underset{w}{0}\underset{w^R}{11}\underset{w}{00}1 \Rightarrow 0^k 1^{2k} 0^{2k} 1^k$

Soln Case 1: vxy v & y in first 2k 1's

0 1 1 0 0 1
u v x y z

uv²xy²z

0 111 000 1 $\quad 0^1 1^4 0^3 1^1 \notin L$
u v y z

Case 2: v & y in second 2k 0's

0 1 1 0 0 1
u v x y z

01 11 000 1 $\quad 0^1 1^3 0^3 1^1 \notin L$

Case 3: v & y in both 2k 0's & 2k 1's

0 1 100 1
u v x y z

01110001
$0^1 1^3 0^3 1^1 \notin L$

Turning Machine (TM)

Alan Turing introduced a new mathematical model called Turning Machine which is a modified version of the PDA and it's much more powerful than PDA

A PDA is not powerful enough to recognize context sensitive language like $\{a^n, b^n, c^n : n \in N\}$, but TM is a generalized machine which can recognize all type of languages
Regular languages
context free languages
Context sensitive language
unrestricted grammar

It is a hypothetical machine, which can simulate any computer algorithm, no matter how complicated it is.

A TM is a finite state machine with an infinite tape and a tape head that can read or write

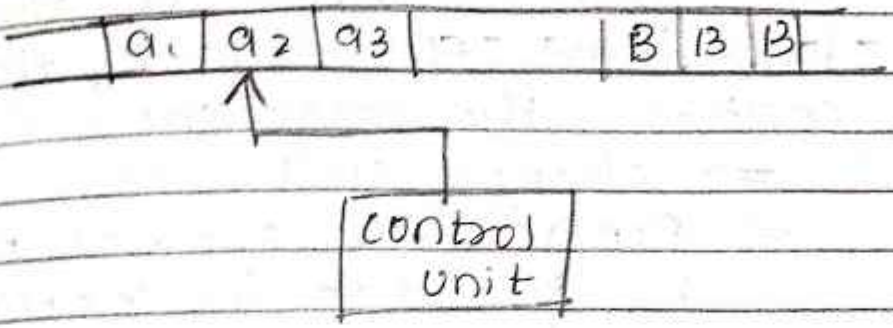
Model of TM

It is a finite automation with the following component.

⇒ Tape

⇒ Read write head

→ control unit.



Tape :- It is used to store info & is divided into cells. Each cell can store only one symbol. The string to be scanned will be stored from the left position of the tape. The remaining infinity of cells each hold the blank (B), which is a special tape symbol that is not an ilp symbol.

The tape is assumed to be infinite both on left and right side of the string.

Read write head : It can read/write a symbol from where it points. It scans one cell of the tape at a time. Initially it is at the leftmost ~~both~~ cell that holds the input.

Control unit: The CU is a finite set of states. It carries out the tasks based on transition table.

In one move the depending upon the symbol scanned by the tape head and the state of the finite control, the machine
=> changes state
=> Replaces the symbol pointed by the tape by another symbol and
=> Moves its head left or right one tape cell.

A TM is a seven tuple
$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$
where
Q → set of finite states
Σ → set of i/p symbols.
Σ is a subset of Γ excluding B
Γ → set of tape symbols
δ → Transitions from $Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$
$q_0 \in Q$ is a the start state of M
B is a special symbol indicating blank character.
$F \subseteq Q$ is a set of final state

Transitions function accepts two parameters namely a state, and a tape symbol and returns a new state after changing the tape symbol and returns a new state after changing the tape symbol and positions read/write head, to the left or right

δ(state, tape symbol) =
(next state, tape symbol, move towards L/R)

Ex:-
The transition $\delta(q_0, a) = (q_1, X, R)$ means that TM is in current state q_0, after scanning the symbol $a \in \Gamma$, TM enters into new state q_1, by replacing the tape symbol a by X and read/write head moves towards right.

Problems

1) Obtain a Turing Machine to accept the language
$L = \{0^n 1^n / n \geq 1\}$

Soln General procedure
It is given that the language accepted by TM should have n number of 0's followed by n number of 1's.
let us take for example
00001111.

Let q_0 be the start state of TM and the read-write head points to the first symbol of the string to be scanned.

0 0 0 0 1 1 1 1

↑

q_0

Step 1 In state q_0, replace 0 by X, change the state to q_1 and move pointer towards right

$$\delta(q_0, 0) = (q_1, X, R)$$

X 0 0 0 1 1 1 1

↑

q_1

Step 2 In state q_1, find left most 1, replace it by Y and change the state to q_2. If we find any 0's while moving right, do not replace it with any other symbol.

$$\delta(q_1, 0) = (q_1, 0, R)$$
$$\delta(q_1, Y) = (q_1, Y, R)$$
$$\delta(q_1, 1) = (q_2, Y, L)$$

X 0 0 0 Y 1 1 1

↑

q_2

DATE / /
PAGE NO: Usha Gold

Step3 The read/write head has to move towards left to obtain left most zero in order to identify left most zero, identify right most X. During this process we may find Y's and 0's. Don't change its value and remain in the same state q_2

$\delta(q_2, Y) = (q_2, Y, L)$
$\delta(q_2, 0) = (q_2, 0, L)$

The resulting configuration is

X 0 0 0 Y 1 1 1
↑
q_2

Step4 TO get leftmost 0, move the pointer to right without changing the symbol X with other symbol. But change the state to q_0

$\delta(q_2, X) = (q_0, X, R)$

The current configuration will be

X 0 0 0 Y 1 1 1
↑
q_2

Step5 Repeat step 1 to 4 to get the following configuration

X X X X Y Y Y Y
↑
q_0

Step 6 In the state q_0, if the scanned symbol is Y, it means there are no more 0's. To check there are no more 1's move the pointer towards right by changing the state to q_3

$$\delta(q_0, Y) = (q_3, Y, R)$$

Step 7. In state q_3, we should see there are only Y's and no more 1's

$$\delta(q_3, Y) = (q_3, Y, R)$$

Repeatedly applying this transition, the following configuration is obtained.

X X X X Y Y Y Y B
↑
q_3

Step 8 Since the string ends with infinite number of blanks, change the state to q_4 which is a final state

$$\delta(q_3, B) = (q_4, B, R)$$

∴ The turing machine to accept the language $\{a^n b^n \mid n \geq 1\}$ is defined by

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

where

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$\Sigma = \{0, 1\}$

DATE / /
PAGE NO. Usha Gold

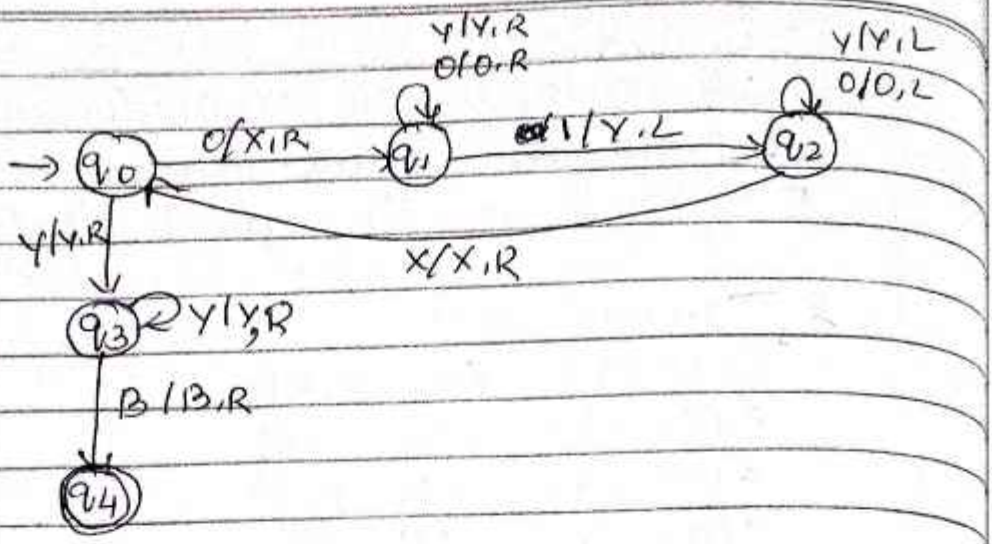
$T = \{0, 1, X, Y, B\}$
$q_0 \in Q$ is the start of machine
$B \in \Gamma$ is the blank symbol
$F = \{q_4\}$ is the final state

δ: shown below

$\delta(q_0, 0) = \{q_1, X, R\}$
$\delta(q_1, 0) = (q_1, 0, R)$
$\delta(q_1, Y) = (q_1, Y, R)$
$\delta(q_1, 1) = (q_2, Y, L)$
$\delta(q_2, Y) = (q_2, Y, L)$
$\delta(q_2, 0) = (q_2, 0, L)$
$\delta(q_2, X) = (q_2, 0, L)$
$\delta(q_2, Y) = (q_0, Y, R)$
$\delta(q_0, Y) = (q_3, Y, R)$
$\delta(q_3, Y) = (q_3, Y, R)$
$\delta(q_3, B) = (q_4, B, R)$

The transitions can also be represented by transition table

δ / Γ	0	1	X	Y	B
$\rightarrow q_0$	(q_1, X, R)	–	–	(q_3, Y, R)	–
q_1	$(q_1, 0, R)$	(q_2, Y, L)	–	(q_1, Y, R)	–
q_2	$(q_2, 0, L)$	–	(q_0, X, R)	(q_2, Y, L)	–
q_3	–	–	–	(q_3, Y, R)	(q_4, B, R)
$*q_4$	–	–	–	–	–



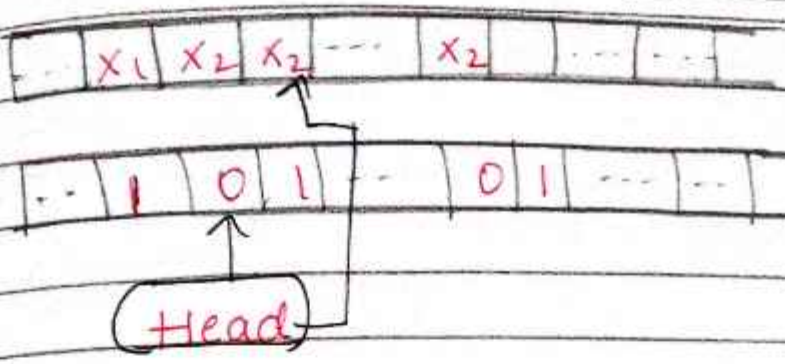
MULTITAPE TURING MACHINE

Multitape TM have many tapes each with its own head, Each of the heads can move independently of the others.

Initially, the first tape is consumed as input, and other tapes are kept blank.

Then the consecutive symbols under its head are read by the machine, the symbols are printed on each tape and its head is moved by the TM.

The expressive power of a Multi-tape turing machine/ computer is similar to that of a K-track turing machine. Single-tape TM can be used to simulate Multi-tape Turing Machine.



There is a single-tape TM for every multi-tape TM.

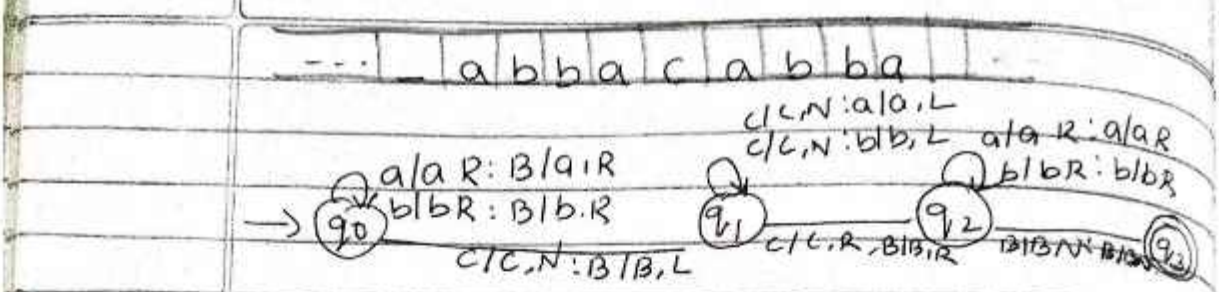
Multitape TM Formal Theorem
We can define a Multi-tape TM formally as a 6-tuple $M = (Q, X, B, \Sigma, q_0, F)$ where

$Q \rightarrow$ is the finite set of states
$X \rightarrow$ is the tape symbol
$\Sigma \rightarrow$ Is the finite set of symbols
$\delta \rightarrow$ Is a transition function
$\delta: Q \times X^k \rightarrow Q \times (X \times \{Left\_shift, Right\_shift\})^k$.
where k is the no. of tape
$q_0 \in Q$ is the initial state
$B \in \Sigma$ is the blank symbol
$F \subseteq Q$ is the set of accepting states

Problem.

1) Show that the given language is accepted by Multi-tape TM
$\{wcw | w \in \{a, b\}^*\}$

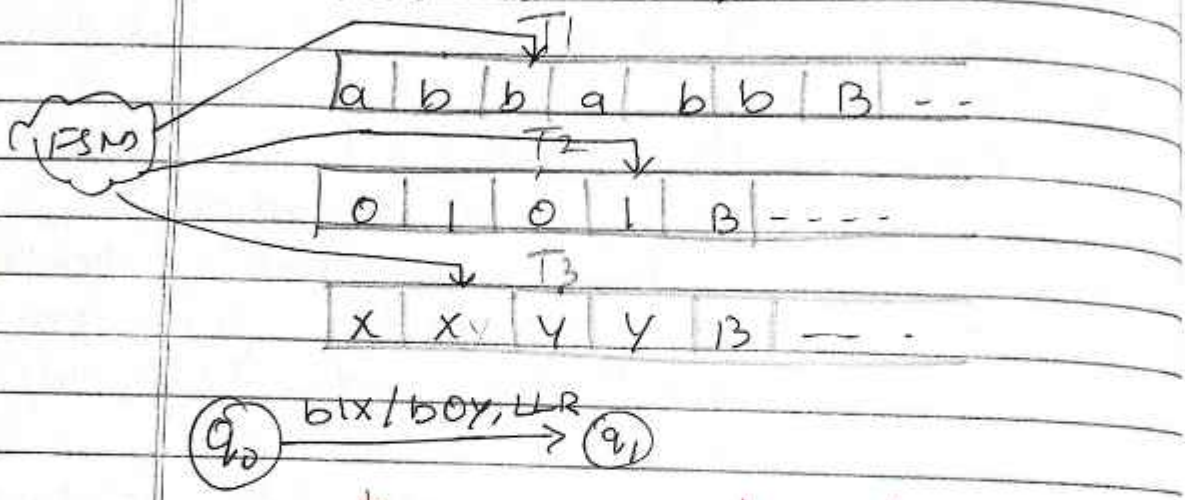
Soln:
consider wcw as abbaCabba



a b b a c a b b a

a b b a

2. Consider the TM

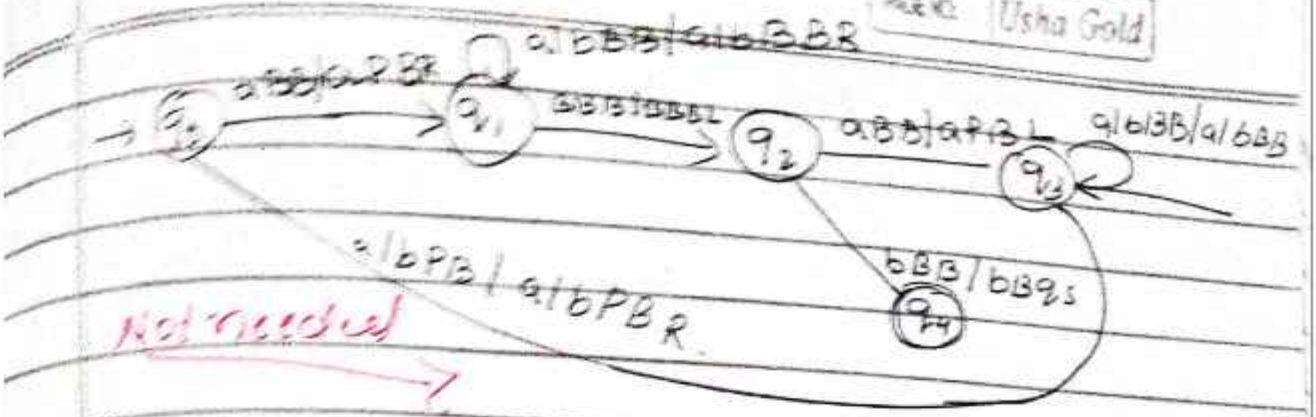


Multitape single Tape equivalence

3 Show that the language $L = \{w \mid w \in \{a, b\}\}$ is accepted by Multitape TM

Soln:

B	a	b	a	B	B
B	B	B	B	B	B
B	B	B	B	B	B



4. Multitrack Turing machine to perform multiplication (M*N)

Ans ⇒ We are using 3 Tracks and Multiplication is performed by repeated Addition

Track 1 → Store M

Track 2 → Store N (used as counter)

Track 3 → 0 (storing the result)

Step 1: Place the given i/p 'M' in track 1

Step 2: Place the given i/p 'N' in track 2

if T1 = 0 then T3 = 0

If T2 = 0 then T3 = 0

Step 3: or else

T3 = T3 + T1

T2 = T2 - 1

until T2 = 0 Repeat

M=5 N=3

5	5	5	5
3	B	B	B
0	B	B	B

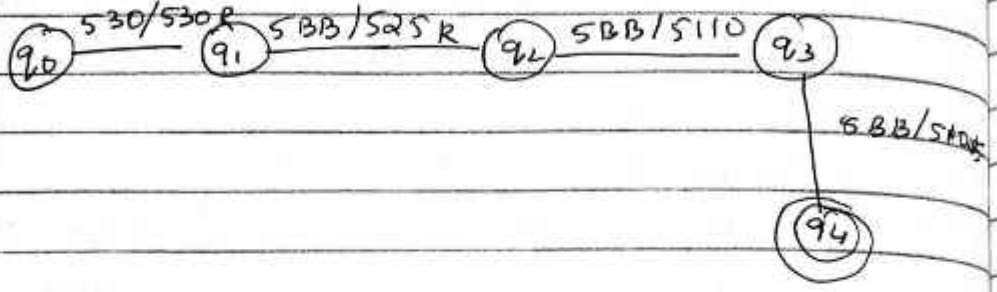
$T_2 = T_2 - 1$

$T_2 = 3-1=2 \quad 2-1=1 \quad 1-1=0$

$T_3 = T_3 + T_1$

$T_3 = 0+5=5 \quad 5+5=10 \quad 5+10=15$

T_1	5	5	5	5
T_2	3	2	1	0
T_3	0	5	10	15



status	5,3,0	5,B,B
→ q_0	$(q_1,(5,3,0),R)$	-
q_1	-	$(q_2(5,2,5),R)$
q_2	-	$(q_3(5,1,10),R)$
q_3	-	$(q_4(5,0,15,R)$
* q_4	-	-

Normal Forms For CFG

When the productions in CFG are made to satisfy certain restrictions, CFG is said to be in NF.

Some of the Normal Forms are

⇒ Chomsky Normal Form

⇒ Greibach Normal Form

Chomsky Normal Form [CNF]

CNF puts restrictions on number of symbols on the RHS of production. That is in CNF the RHS of production cannot more than two symbols. It can be a single terminal or two non-terminals

Definition : CNF

Let $G = (V, T, P, S)$ be a CFG, The grammar. ~~The gram~~ G is said to be in CNF, if all the productions are of the form.

$A \to BC$ or $A \to a$

where A, B and C are non terminals and a is a terminal.

NOTE: The grammar should not have unit productions, ϵ-productions & useless symbols.

Reduction of CFG to CNF

A CFG can be converted to CNF using the following theorem

Theorem

For every CFL where ϵ, generated by a grammar G, there is an equivalent grammar in CNF such that $L(G) = L(G)$

Proof

Let $G = (V, T, P, s)$ be the CFG generating a language, we construct $G' = (V', T', P', s)$ which is in CNF as follows

Step 1: Eliminate ϵ-productions, unit productions and useless symbols from the grammar G.

Step 2: Arrange in such a way that all bodies of length two or more consist only of variables.

For every terminal 'a' that appears in a body of length two or more, create a new variable say A with a production $A \to a$. Now use A in place of a. So that every production has a body i.e either a single terminal or atleast two variable and no terminals

Step 3: Breaks bodies of length three or more into a cascade of productions each with a body consisting of two variables.

Break those productions

$A \to B_1, B_2 \ldots B_k$, for $k \geq 3$

into a group of productions with two variables in each body

i.e introduce $k-2$ new variables

$C_1, C_2 \ldots C_{k-2}$.

The original production is replaced, by the K-1 productions

$A \rightarrow B_1 C_1$

$C_1 \rightarrow B_2 C_2 \cdots C_{K-3}$

$\rightarrow B_{K-2} C_{K-2}$

$C_{K-2} \rightarrow B_{K-1} B_K$

Thus we get G' in CNF Hence proved

Problems

1. Consider the grammar $G = (\{S, A, B\}, \{a, b\}, P, S)$ where $P = \{S \rightarrow bA \mid aB$

$A \rightarrow bAA \mid as \mid a$

$B \rightarrow aBB \mid bS \mid b$

Find an equivalent grammar in CNF

Soln Step 1: No useless, $\in$- unit production

$A \rightarrow a$

$B \rightarrow a$

These productions are already in CNF

Step 2: Replace terminals by variables

$S \rightarrow bA$	is replaced by	$S \rightarrow C_b A, C_b \rightarrow b$
$S \rightarrow aB$	"	$S \rightarrow C_a B, C_a \rightarrow a$
$A \rightarrow bAA$	"	$A \rightarrow C_b AA, C_b \rightarrow b$
$A \rightarrow aS$	"	$A \rightarrow C_a S, C_a \rightarrow a$
$B \rightarrow aBB$	"	$B \rightarrow C_a BB, C_a \rightarrow a$
$B \rightarrow bS$	"	$B \rightarrow C_b S, C_b \rightarrow b$

Step 3: Restrict the number of variables on RHS to two.

$A \to C_b AA$ is replaced by $A \to C_b D_1, D_1 \to AA$

$B \to C_a BB$ " $B \to C_a D_2, D_2 \to BB$

The final grammar in CNF is obtained by combining all the productions in step 1, 2 & 3

The grammar in CNF $G' = (V', T', P', S)$

where $V' = \{S, A, B, C_a, C_b, D_1, D_2\}$

$T = \{a, b\}$

$P = \{ S \to \epsilon \mid C_b A \mid C_a B$

$A \to C_a S \mid C_b D_1 \mid a$

$B \to C_b S \mid C_b D_1 \mid a$

$D_1 \to AA$

$D_2 \to BB$

$C_a \to a$

$C_b \to b$

$\}$

GREIBACH NORMAL FORM (GNF)

In CNF there is restriction on the no of symbols on the RHS of the production (i.e) it should be either two variables or one terminal

In GNF there is no restriction on the length of RHS of a product but restrictions occur on the position in which terminals and variables appear.

Definition: GNF

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

$A \to a\alpha$ where $a \in T$ and $\alpha \in V^*$

i.e. The first symbol on the RHS of the production should be a terminal followed by zero or more number of variables.

Reduction of CFG to GNF

Theorem

Let $G = (V, T, P, S)$ be a CFG, generating the language L without ϵ. An equivalent grammar G_1 generating the same language exists for which each production is of the form

$A \to a\alpha$

Step 1: Obtain the grammar in CNF

Step 2: Rename the non-terminals to

$A_1, A_2 \dots A_n$

Step 3: Using substitution method, obtain the production to the form

$A_i \to A_j \alpha$ for $i < j$

where $\alpha \in V^*$

Step 4: After substitution, if the grammar has left recursion, eliminate left recursion.

Step 5: Repeat step 3 and/or step 4 to get the grammar in GNF

Problem 2

Reduce the following grammar to CNF

$S \rightarrow A B a$

$A \rightarrow a a b$

$B \rightarrow A c$

Solution

Step 1: No unit production, $\in$ - productions and useless symbols.

Step 2: Replace terminals by variables.

$S \rightarrow A B a$, is replaced by $S \rightarrow A B C_a$, $C_a \rightarrow a$

$A \rightarrow a a b$, is replaced by $A \rightarrow C_a C_a C_b$, $C_a \rightarrow a, C_b \rightarrow b$

$B \rightarrow A c$, is replaced by $B \rightarrow A C_c$, $C_c \rightarrow c$

Step 3: Restrict the number of variables on the RHS to two

$S \rightarrow A B C_a$ is replaced by $S \rightarrow A D_1$, $D_1 \rightarrow B C_a$

$A \rightarrow C_a C_a C_b$ is replaced by $A \rightarrow C_a D_2$, $D_2 \rightarrow C_a C_b$

$\therefore$ The grammar in CNF $\boldsymbol{G' = (V', T', P', S)}$

where $V' = \{S, A, B, D_1, D_2, C_a, C_b, C_c\}$ $T' = \{a, b, c\}$ $S = S$

$P' = \{S \rightarrow A D_1$

$A \rightarrow C_a D_2$

$B \rightarrow A C_c$

$D_1 \rightarrow B C_a$

$D_2 \rightarrow C_a C_b$

$C_a \rightarrow a$

$C_b \rightarrow b$

$C_c \rightarrow c\}$

Problem 3

Reduce the grammar into CNF

$S \rightarrow a X X$

$X \rightarrow a S | b S | a$

Solution

Step 1: No useless symbols, $\in$ - productions and unit productions

$X \rightarrow a$, already in CNF.

Step 2: Replace terminals by variables.

$S \rightarrow a X X$, is replaced by $S \rightarrow C_a X X$, $C_a \rightarrow a$

$X \rightarrow a S$, is replaced by $X \rightarrow C_a S$, $C_a \rightarrow a$

$X \to bS$, is replaced by $X \to C_b S$, $C_b \to b$

Step 3: Restrict the number of variables on the RHS to two.

$S \to C_a XX$, is replaced by $S \to C_a D_1$, $D_1 \to XX$

$\therefore$ The grammar in CNF $G' = (V', T', P', S)$

where $V' = \{S, X, D_1, C_a, C_b\}$ $T' = \{a, b\}$ $S = S$

$P' = \{ S \to C_a D_1$

$X \to C_a S \mid C_b S$

$D_1 \to XX$

$C_a \to a$

$C_b \to b \}$

PROBLEM 4

Reduce the following grammar into CNF

$S \to 0A \mid 1B$

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

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

SOLUTION

Step 1: No useless symbols, $\in$ - productions and unit productions

$A \to 1, B \to 0$, already in CNF

Step 2: Replace terminals by variables

$S \to 0A$, is replaced by $S \to C_0 A$, $C_0 \to 0$

$S \to 1B$, is replaced by $S \to C_1 B$, $C_1 \to 1$

$A \to 0AA$ is replaced by $A \to C_0 AA$, $C_0 \to 0$

$A \to 1S$, is replaced by $A \to C_1 S$, $C_1 \to 1$

$B \to 1BB$, is replaced by $B \to C_1 BB$, $C_1 \to 1$

$B \to 0S$, is replaced by $B \to C_0 S$, $C_0 \to 0$

Step 3: Restrict the number of variables on the RHS by two

$A \to C_0 AA$, is replaced by $A \to C_0 D_1$, $D_1 \to AA$

$B \to C_1 BB$, is replaced by $B \to C_1 D_2$, $D_2 \to BB$

$\therefore$ The equivalent grammar in CNF $G' = (V', T, P', S)$

where $V' = \{S, A, B, D_1, D_2, C_0, C_1\}$ $T = \{0, 1\}$ $S = S$

$P' = \{S \to C_0 A \mid C_1 B$

$A \to C_1 S \mid C_0 D_1 \mid 1$

$B \to C_0 S \mid C_1 D_2 \mid 0$

$D_1 \to A\,A$
$D_2 \to B\,B$
$C_0 \to 0$
$C_1 \to 1\}$

PROBLEM 5

Reduce the following grammar G to CNF. G is

$S \to a\,A\,D$
$A \to a\,B \mid b\,A\,B$
$B \to b$
$D \to d$

SOLUTION

Step 1: No useless symbols, $\in$ - productions and unit productions.

$B \to b$

$D \to d$, already in CNF.

Step 2: Replace terminals by variables

$S \to a\,A\,D$, is replaced by $S \to C_a\,A\,D,\ C_a \to a$
$A \to a\,B$, is replaced by $A \to C_a\,B,\ C_a \to a$
$A \to b\,A\,B$, is replaced by $A \to C_b\,A\,B,\ C_b \to b$

Step 3: Restrict the number of variables on the RHS by two

$S \to C_a\,A\,D$, is replaced by $S \to C_a\,D_1,\ D_1 \to A\,D$
$A \to C_b\,A\,B$, is replaced by $A \to C_b\,D_2,\ D_2 \to A\,B$

$\therefore$ The equivalent grammar in CNF $\boldsymbol{G' = (V', T, P', S)}$

Where $V' = \{S, A, B, D, D_1, D_2, C_a, C_b\}$

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

$S = S$

$P' = \{S \to C_a\,D_1$
$A \to C_a\,B \mid C_b\,D_2$
$D_1 \to A\,D$
$D_2 \to A\,B$
$B \to b$
$D \to d$
$C_a \to a$
$C_b \to b\}$

PROBLEM 6

Find a grammar in CNF equivalent to

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

$A \to a\,A \mid a$

$B \to b\,B \mid b$

SOLUTION

Step 1: No useless symbols, $\in$ - productions and unit productions.

$A \to a$

$B \to b$, already in CNF

Step 2: Replace terminals by variables

$S \to a\,A\,b\,B$ is replaced by

$S \to C_a\,A\,C_b\,B$

$C_a \to a$

$C_b \to b$

$A \to a\,A$, is replaced by $A \to C_a\,A$, $C_a \to a$

$B \to b\,B$, is replaced by $B \to C_b\,B$, $C_b \to b$

Step 3: Restrict the number of variables on the RHS by two.

$S \to C_a\,A\,C_b\,B$, is replaced by $S \to C_a\,D_1$

$D_1 \to A\,D_2$

$D_2 \to C_b\,B$

∴ **The equivalent grammar in CNF $G' = (V', T, P', S)$**

Where $V' = \{S, A, B, D_1, D_2, C_a, C_b\}$ $T = \{a, b\}$ $S = S$

$P' = \{S \to C_a\,D_1$

$A \to C_a\,A \mid a$

$B \to C_b\,B \mid b$

$D_1 \to A\,D_2$

$D_2 \to C_b\,B$

$C_a \to a$

$C_b \to b$

PROBLEM 7

Find a grammar in CNF equivalent to the grammar

$S \to \sim S \mid [S + S] \mid p \mid q$

where $T = \{\sim, [, +,], p, q\}$

SOLUTION

Step 1: No useless symbols, $\in$ - productions and unit productions.

$S \to p$

$S \to q$, already in CNF

Step 2: Replace terminals by variables.

$S \to \sim S$, is replaced by $S \to C_1 S, \quad C_1 \to \sim$

$S \to [S + S]$, is replaced by $S \to C_2 S C_3 S C_4$

$C_2 \to [$

$C_3 \to +$

$C_4 \to]$

Step 3: Restrict the number of variables on the RHS by two.

$S \to C_2 S C_3 S C_4$, is replaced by $S \to C_2 D_1$

$D_1 \to S D_2$

$D_2 \to C_3 D_3$

$D_3 \to S C_4$

$\therefore$ **The equivalent grammar in CNF $G' = (V', T, P', S)$**

Where $V' = \{S, D_1, D_2, D_3, C_1, C_2, C_3, C_4\}$

$T = \{\sim, [, +,], p, q\}$

$S = S$

$P' = \{S \to C_1 S \mid C_2 D_1 \mid p \mid q$

$D_1 \to S D_2 \quad D_2 \to C_3 D_3 \quad D_3 \to S C_4$

$C_1 \to \sim \quad C_2 \to [\quad C_3 \to + \quad C_4 \to]\}$

PROBLEM 8

Convert the given grammar to CNF

$S \to AB \mid CA$

$B \to BC \mid AB$

$A \to a$

$C \to aB \mid b$

SOLUTION

Step 1: No useless symbols, $\in$ - productions and unit productions.

$S \to AB \mid CA, \quad A \to a$

$B \to BC \mid AB, \quad C \to b$, already in CNF

Step 2: Replace terminals by variables $C \to a\,B$, is replaced by $C \to C_a B, C_a \to a$

Step 3: Not applicable

The equivalent grammar in CNF $G' = (V', T, P', S)$

where $V' = \{S, A, B, C_a\}$ $\quad T = \{a, b\}$ $\quad S = S$

$P' = \{S \to A\,B \mid C\,A$

$B \to B\,C \mid A\,B$

$A \to a$

$C \to C_a B \mid b$

$C_a \to a\}$

PROBLEM 9

Write the equivalent CNF, for the following grammar.

$S \to A\,B$

$A \to a\,a\,b$

$B \to a\,A\,C$

SOLUTION

Step 1: No useless symbols, $\in$ - productions and unit productions

$S \to A\,B$, already in CNF

Step 2: Replace terminals by variables.

$A \to a\,a\,b$, is replaced by $\quad A \to C_a C_a C_b$, $C_a \to a$, $C_b \to b$

$B \to a\,A\,C$, is replaced by $\quad B \to C_a A\,C$, $C_a \to a$

Step 3: Restrict the number of variables on the RHS by two

$A \to C_a C_a C_b$, is replaced by $\quad A \to C_a D_1$, $D_1 \to C_a C_b$

$B \to C_a A\,C$, is replaced by $\quad B \to C_a D_2$, $D_2 \to A\,C$

The equivalent grammar in CNF $G' = (V, T, P', S)$

where $V' = \{S, A, B, D_1, D_2, C_a, C_b\}$ $\quad T = \{a, b\}$ $\quad S = S$

$P' = \{S \to A\,B$

$A \to C_a D_1$

$B \to C_a D_2$

$D_1 \to C_a C_b$

$D_2 \to A\,C$

$C_a \to a$

$C_b \to b\}$

Problem 10

Reduce the grammar, into CNF

$S \rightarrow A0B$

$A \rightarrow AA \mid 0S \mid 0$

$B \rightarrow 0BB \mid 1S \mid 1$

Solution

Step 1: No unit productions, $\in$ - productions and useless symbols.

$A \rightarrow AA \mid 0$

$B \rightarrow 1$, already in CNF

Step 2: Replace terminals by variables.

$S \rightarrow A0B$, is replaced by $S \rightarrow AC_0B,\ C_0 \rightarrow 0$

$A \rightarrow 0S$, is replaced by $A \rightarrow C_0S,\ C_0 \rightarrow 0$

$B \rightarrow 0BB$, is replaced by $B \rightarrow C_0BB,\ C_0 \rightarrow 0$

$B \rightarrow 1S$, is replaced by $B \rightarrow C_1S,\ C_1 \rightarrow 1$

Step 3: Restrict the number of variables on the RHS by two.

$S \rightarrow AC_0B$ is replaced by $S \rightarrow AD_1,\ D_1 \rightarrow C_0B$

$B \rightarrow C_0BB$ is replaced by $B \rightarrow C_0D_2,\ D_2 \rightarrow BB$

$\therefore$ **The equivalent grammar in CNF $G' = (V', T, P', S)$**

where $V' = \{S, A, B, D_1, D_2, C_0, C_1\}$

$T = \{0, 1\}$

$S = S$

$P' = \{S \rightarrow AD_1$

$A \rightarrow AA \mid 0 \mid C_0S$

$B \rightarrow C_1S \mid C_0D_2 \mid 1$

$D_1 \rightarrow CB$

$D_2 \rightarrow BB$

$C_0 \rightarrow 0$

$C_1 \rightarrow 1\}$

4.9 Greibach Normal Form (GNF)

In CNF there is restriction on the number of symbols on the right hand side of the production (i.e.,) it should be either two variables or one terminal.

In GNF there is no restriction on the length of RHS of a production but restrictions occur on the position in which terminals and variables appear.