Unit : 3 Context free Grammar (CFG)

Ques

1) Derivation of a string from given CFG
2) Checking of ambiguity of CFG.
3) Simplification of CFG.
   a) ~~Illumination~~ Elimination of ϵ productions
   b) Elimination of unit productions
   c) Elimination of useless symbols.
4) Normal forms
   a) CNF (chomsky normal form)
   b) Greibach normal form (GNF)
   c) -
5) Applications of CFG
   chomsky Hiearchy

There are four different types of grammar

① unrestricted grammar (Type 0)
② context sensitive grammar (Type 1)
③ context free grammar (Type 2)
④ Regular grammar (Type 3)

context-free grammer

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

where V = set of non-terminals.

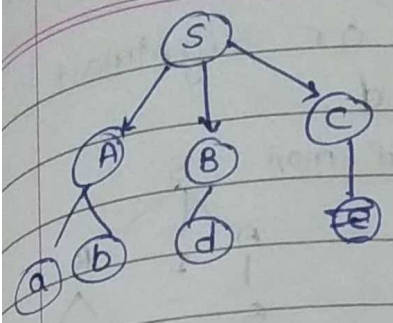
Note : Generally non-terminals are shown by capital letters.

T = set of terminals [leaf node]

Note- terminals are represented by other than capital letters.

S = { starting non-terminals (variables) }

P = productions



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

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

$S = \{S\}$

$P = \{ S \to A|B|C \quad A \to a|b \quad B \to d \quad C \to e$
$S \to AB|C \quad A \to ab \quad C \to \epsilon$

All the grammars are defined as above only the difference is in production rule (P)

Derivation of string from CFG

$S \to A1B$

$A \to 0A/\epsilon$

$B \to 0B/1B/\epsilon$

Give the left most and right most derivation of the string 1001

Left most :- Always replace the first non-terminal after the arrow.

Right most :- Always replace the non-terminal which is at the last

1001

(Left-Most)

$S \to A1B$

$S \to 1B \quad [A \to \epsilon]$

$S \to 10B \quad [B \to 0B]$

$S \to 100B \quad [B \to 0B]$

$S \to 1001B \quad [B \to 1B]$

$S \to 1001 \quad [B \to \epsilon]$

(Right-most)

$S \to A1B$

$S \to A10B \quad [B \to 0B]$

$S \to A100B \quad [B \to 0B]$

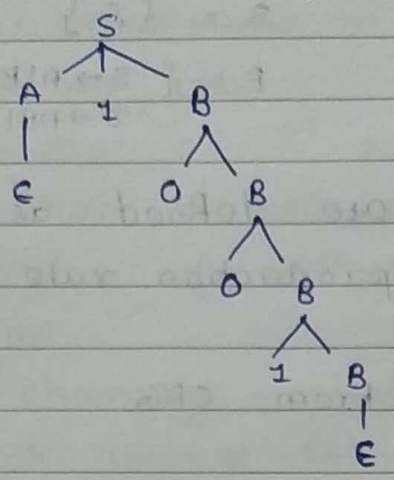
$S \to A1001B \quad [B \to 1B]$

$S \to A1001 \quad [B \to \epsilon]$

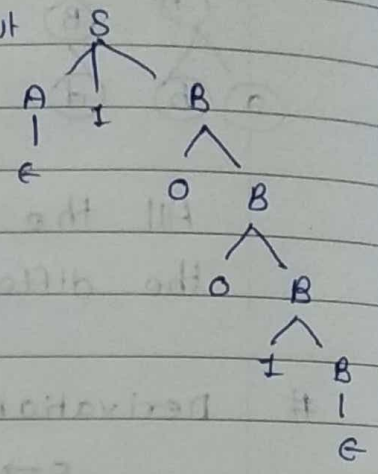
$S \to 1001$

Sometimes in a question leftmost or rightmost ~~deriv~~ parsed tree may be asked.

(left-most)



Right-most



NOTE :- If leftmost or rightmost word is used in a que and parsed tree is also asked then always derive the string by horizontal method [sentential form] and then draw parsed tree

Q $S \to XY$

$X \to YY / a$

$Y \to XY / b$

string = "aabbb"

$S \to XY$	$S \to XY$
$S \to aY$ $(X \to a)$	$S \to YYY$ $(X \to YY)$
$S \to aXY$ $(Y \to XY)$ $(X \to a)$	$S \to XYYY$ $(Y \to XY)$
$S \to aaY$ $(Y \to XY)$	$S \to aYYY$ $(X \to a)$
$S \to aaXY$ $(X \to YY)$	$S \to aXYYY$ $(X \to XY)$
$S \to aaYYY$ $(Y \to b)$	$S \to aaYYY$ $(X \to a)$
$S \to aabYY$ $(Y \to b)$	$S \to aabYY$ $(Y \to b)$
$S \to aabbY$ $(Y \to b)$	$S \to aabbY$ $(Y \to b)$
$S \to aabbb$	$S \to aabbb$ $(Y \to b)$

For the same string by using method either left most or rightmost if more than one solution is available then given CFG is called ambiguous grammer

S → XY
S → X XY {Y → YY}
S → X X X b
S → X Y Y b
S → X Y b b
S → X X Y b b
S → X X b b b
S → a a b b b

check whether following CFG is ambiguous.

S → XY
X → YY / a
Y → XY / b

string =

1. E → E + T / T
T → T * F / F
F → (E) / a / b

string = (a+b * a + b) (a+b)* a+b

E → E+T / T
E → (E)

simplification of CFG.

To simplify CFG, following steps are executed in a given sequence

1) Elimination of ϵ productions.
2) Elimination of unit productions.
3) Elimination of useless non-terminals.

1) Elimination of ϵ productions

Q.1) $S \to ABAC$
$A \to aA \mid \epsilon$
$B \to bB \mid \epsilon$
$C \to d$

step: I) Find out nullable non-terminals
$A \to \epsilon$, $B \to \epsilon$ hence nullable non-terminals are $N/s = \{A, B\}$

step: II) Elimination of nullable non-terminals
$S \to ABAC \mid BAC \mid AAC \mid ABC \mid AC \mid BC \mid C$
$A \to aA \mid a$
$B \to bB \mid b$
$C \to d$

Q.2) $S \to ABA$
$A \to aA \mid \epsilon$
$B \to bB \mid \epsilon$
$C \to d$

$A \to \epsilon$, $B \to \epsilon$, $S \to ABA$ hence nullable's are
$N/s = \{A, B, S\}$

$S \rightarrow ABA \mid BA \mid AA \mid AB \mid A \mid B \mid$
$A \rightarrow aA \mid a$
$B \rightarrow bB \mid b$
$c \rightarrow d$

(Q.3) $S \rightarrow AaA$
$A \rightarrow Sb \mid bCC \mid \epsilon$
$C \rightarrow ccabb \mid abb$

Null non-terminals = N's = $\{A\}$

$S \rightarrow AaA \mid aA \mid Aa \mid a$
$A \rightarrow Sb \mid bcc$
$C \rightarrow cc \mid abb$

@# Elimination of unit productions.

$S \rightarrow ABA$
$A \rightarrow aA \mid \epsilon$
$B \rightarrow bB \mid \epsilon$
$C \rightarrow d$

exact one non-terminal

To remove unit productions we must first eliminate ϵ productions first

$N's = \{A, B, S\}$

$S \rightarrow ABA \mid BA \mid AA \mid AB \mid A \mid B \mid$
$A \rightarrow aA \mid a$
$b \rightarrow bB \mid b$
$C \rightarrow d$

unit production eliminations.
unit productions are $S \rightarrow A$ and $S \rightarrow B$
To remove the unit production replace that non-terminal by it's productions

$S \rightarrow ABA \mid BA \mid AA \mid AB \mid aA \mid a \mid bB \mid b$

$A \to aA \mid a$

$B \to bB \mid b$

$C \to d$

Q. $S \to AB$

$A \to a$

$B \to C \mid b$

$C \to d D$

$D \to E$

$E \to a$

unit productions $B \to C$, $C \to D$, $D \to E$

① $D \to E$

$D \to a$

② $C \to D$

$C \to a$

③ $B \to c \mid b$

$B \to a \mid b$

④ $S \to AB$

⑤ $E \to a$

⑥ $A \to a$

Q. # Elimination of useless non-terminals.

There are two types of useless non-terminals

1) Non-generating

$S \to AB \mid AC$

$A \to aA \mid a$

$B \to bB$

$C \to d$

$B \to bB$ → the Non-terminal which is responsible for not getting a string is non-generating

$S \rightarrow AC$

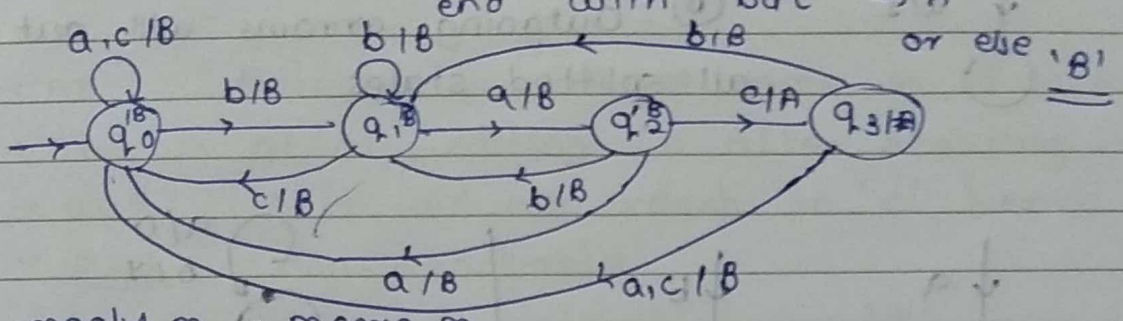
$A \rightarrow aA \mid a$

~~$B \rightarrow bB$~~

$c \rightarrow d$

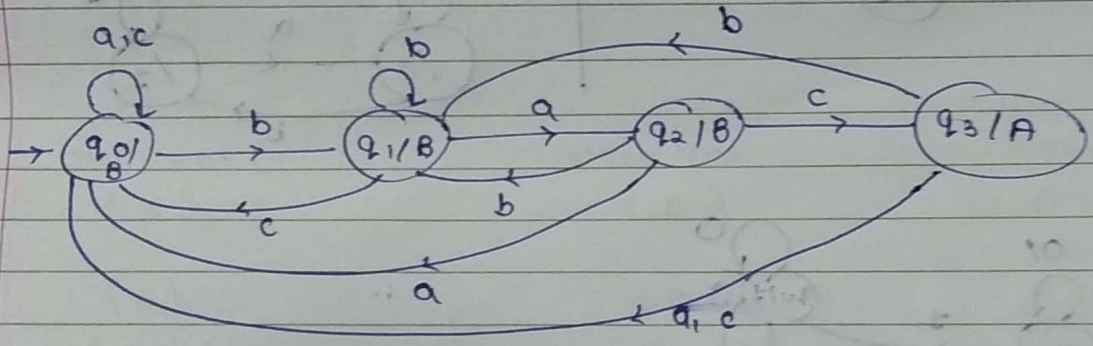
mealy to moore machines.

end with ; bac → A or else 'B'

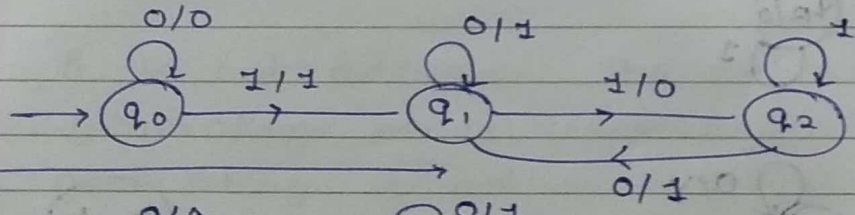


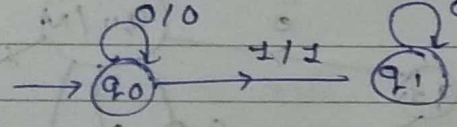
① mealy m - moore m (Not Reduced)

② mealy m - moore (Reduced)



Q.

① 

② 

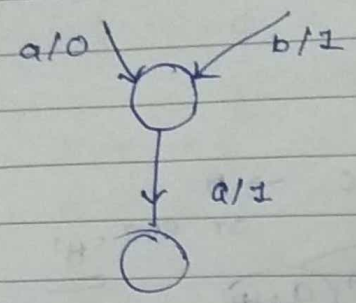
* NOTE :- check all incoming arrows to a particular state and find out number of outputs

① If no. of output is 1 then keep that state as it is

② If incoming arrows have multiple outputs then

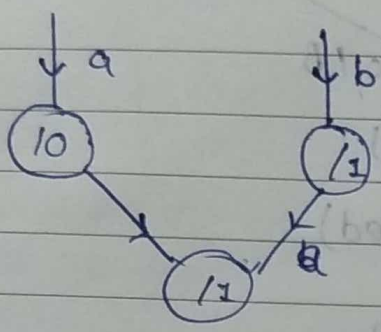
split that state into no. of states such that one one state one output.

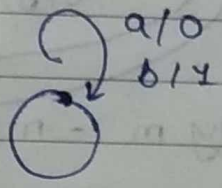
for ex-

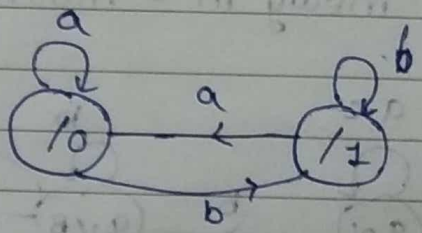


ⓐ Incoming transitions will be go into only one state of splitted state.

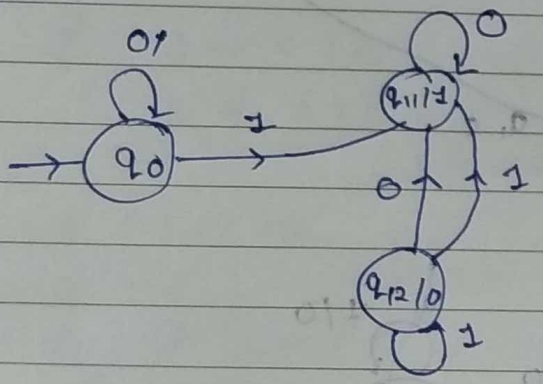
ⓑ Outgoing arrows will out from all splitted states.



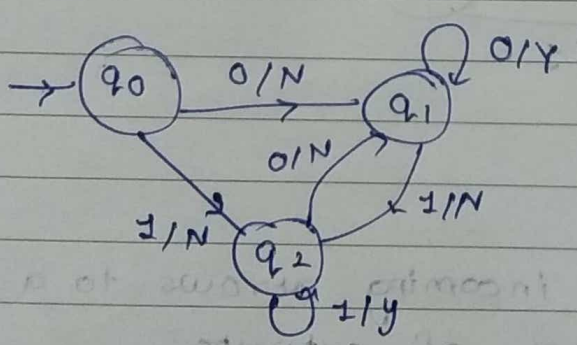


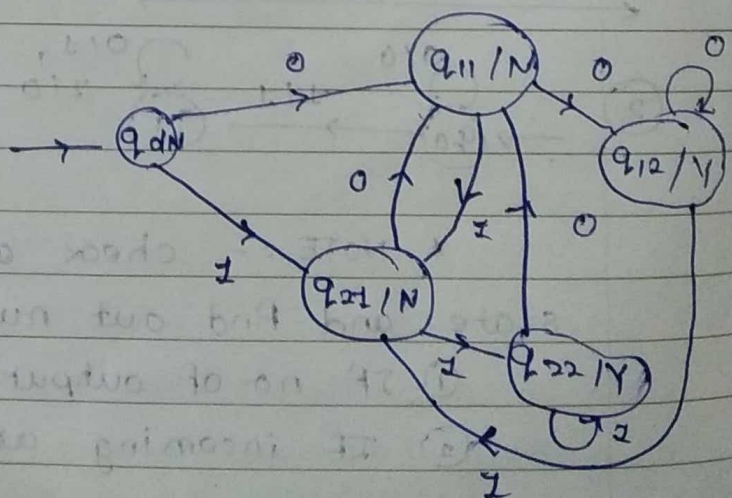


Q.



Q.





Elimination of useless (continue)

② Non-reachable

$S \to AB$
$A \to 0A \mid 0$
$B \to 0B \mid 01$
$C \to 0 \mid 11$

$S \to AB$
$S \to 0AB$
$S \to 00B$
$S \to 001$

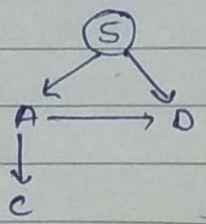
NOTE: C will not come in the derivation of any string i.e after starting with S it will not reach to 'C' hence C is non-reachable

Ex- $S \to AB \mid BC \mid AD$
$A \to BD \mid CD$
$C \to aD \mid AC$
$D \to b$

old var	New Var	Reason
$\{\phi\}$	$\{D\}$	$D \to b$
$\{D\}$	$\{C, D\}$	$C \to aD$
$\{C, D\}$	$\{A, C, D\}$	$A \to CD$
$\{A, C, D\}$	$\{S, A, C, D\}$	$S \to AD$

B is not added in the table hence it is non-generating hence grammar after removing non-generating symbol is ~~conte~~

$S \to AD$
$A \to CD$
$C \to aD \mid AC$
$D \to b$



There is no non-reachable terminal hence grammar after removing useless non-termi is

$S \rightarrow AD$
$A \rightarrow CD$
$C \rightarrow aD / AC$
$D \rightarrow b$

Q. $S \rightarrow AB / BC$
$A \rightarrow aAB / bAC$
$B \rightarrow bC / Ab$
$C \rightarrow b$

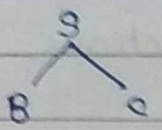
emoval Non-generating symbols.

old Var	New Var	Reason
$\{\phi\}$	$\{C\}$	$C \rightarrow b$
$\{C\}$	$\{C, B\}$ $\{B, C\}$	$B \rightarrow bC$
$\{B, C\}$	$\{S, B, C\}$	$S \rightarrow BC$

hence A is not added hence it is non-generating symbol. hence grammar after removing non-generating symbol (Var) is

$S \rightarrow BC$
$B \rightarrow bC$
$C \rightarrow b$

② Non-reachable symbols



hence there is no non-reachable symbol
grammar after removing useless symbol is :

$S \to BC$
$B \to bC$
$C \to b$

Q. $S \to AB \mid BC$
$A \to 0A \mid 0I \mid IB$
$B \to AB \mid BC$
$C \to 0 \mid I \mid 0C$

N	old var	New var	Reason
	~~{ϕ}~~	~~{C}~~	$C \to 0$
	~~{C}~~	~~{A,C}~~ ~~{A,C}~~	
	{ϕ}	{A,C}	$C \to 0$ $A \to 0I$

S, B are non-generating symbols.
NOTE :- If starting symbol is useless then whole grammar is useless.

Common example
Simplification of context free grammar

Q. Simplify the following CFG
$S \to ASB \mid \epsilon$
$A \to aAS \mid a$
$B \to SbS \mid A \mid bb$

step: I) Elimination of ε productions.

$S \to \epsilon$

Nullable NT = {S}

① $S \to ASB \mid AB$

$A \to aAS \mid a \mid aA$

$B \to SbS \mid A \mid bb$

$bs \mid Sb \mid b$

step- II)

$S \to ASB \mid AB$

$A \to aAS \mid a \mid aA$

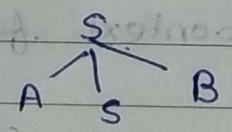
$B \to SbS \mid aAS \mid a \mid aA \mid bb \mid bS \mid Sb \mid b$

step- III)

non-generating:

~~old~~ new var	new var	~~old~~ var	reason
{ϕ}	{B, A}		B → b, A → a
{B, A}	{S, B, A}		S → AB

non-reachable



no non-reachable non terminal.

hence simplified grammar is

$S \to ASB \mid AB$

$A \to aAS \mid a \mid aA$

$B \to SbS \mid aAS \mid a \mid aA \mid bb \mid bS \mid Sb \mid b$

step: III)

Q. S → PQP
P → 0P | ∈
Q → 1Q | ∈

step: I) Removal of ∈ productions

P → ∈ , Q → ∈ , S → ∈

Nullable NT = { P, Q, S }

S → PQP | QP | PQ | PP | P | Q
P → 0P | 0
Q → 1Q | 1

step: II) Elimination of unit productions.

S → P S → Q

S → PQP | QP | PP | PQ | P | Q
replace P → 0P | 0
Q → 1Q | 1
S → 0P | 0 | 1Q | 1 | PQP | QP | PP | PQ | 0P | 0 | 1Q | 1
P → 0P | 0
Q → 1Q → 1

step: III)

non-generating

old var	new var	reason
{ ∅ }	{ S, P, Q }	S → 0 P → 0 Q → 1

non-reachable

There is no reachable non-terminal

There are two types of normal form

① chomsky Normal form (CNF):

If Every production rule of simplified CFG is in the following form then it is said to be in CNF.

$NT \rightarrow NT \cdot NT$

OR

$NT \rightarrow T$

Q. convert the following CFG into CNF

$S \rightarrow aAb \mid bAa$
$A \rightarrow aBA \mid bB \mid a$
$B \rightarrow b$

Simplification of CFG

① Removal or elimination of ϵ productions. → NA
② Removal of unit production → NA
③ Removal of useless symbols → NA

Hence given CFG is simplified.

hence convertion of simplified CFG into CNF

NOTE :- The production rules which are already in CNF will remain unchanged

(1) $S \to aAb \mid bAa$

Terminal — Non-Terminal

NOTE : wherever combination of T and NT convert that production into Non-Terminals only by adding new productions but that added new production must be in CNF.

Add $T_1 \to a$

$T_2 \to b$

$S \to T_1AT_2 \mid T_2AT_1$

Add

$T_3 \to T_1A$

$T_4 \to T_2A$

$S \to T_3T_2 \mid T_4T_1$

$A \to T_1BA \mid T_2B \mid a$

$B \to b$

Add

$T_5 \to T_1B$

hence $\boxed{A \to T_5A \mid T_2B \mid a}$

$\underline{B \to b}$

Hence CNF is

$S \to T_3T_2 \mid T_4T_1$

$A \to T_5A \mid T_2B \mid a$

$B \to b$

$T_1 \to a$

$T_2 \to b$

$T_5 \to T_1B$

$T_3 \to T_1A$

$T_4 \to T_2A$

Q. $S \to PQP \mid QP \mid PP \mid PQ \mid OP \mid 0 \mid 1Q \mid 1$

$P \to 0P \mid 0$

$Q \to 1Q \mid 1$

$S \to PQP \mid QP \mid PP \mid PQ \mid OP \mid 0 \mid 1Q \mid 1$

$T_1 \to PQ$

$T_2 \to 0$

$T_3 \to 1$

~~$S \to PQP \mid Q$~~ $S \to T_1P \mid QP \mid PP \mid PQ \mid T_2P \mid 0 \mid T_3Q \mid 1$

$P \to 0P \mid 0$

$P \to T_2P \mid 0$

$Q \to 1Q \mid 1$

$Q \to T_3Q \mid 1$

hence CNF is

$S \to T_1P \mid QP \mid PP \mid PQ \mid T_2P \mid 0 \mid T_3Q \mid 1$

$P \to T_2P \mid 0$

$Q \to T_3Q \mid 1$

$T_1 \to PQ$

$T_2 \to 0$

$T_3 \to 1$

Q.

Greibach Normal form.
(GNF)

If each production rule of simplified CFG is in the following format then it is said to be in GNF

$NT \to T.(NT)^*$

for ex

$S \to a \mid aA \mid aAB \mid aABC.$

Example

$P \rightarrow QR \mid pqR$

$Q \rightarrow p \mid qR$

$R \rightarrow q \mid qRR$

Given grammar is simplified.

Rules :

1) If terminal occurs at second position or onwards then add new rules like CNF

Ad ① $P \rightarrow QR \mid pqR$

Add $T_1 \rightarrow q$

$P \rightarrow QR \mid pT_1R$

② $Q \rightarrow p \mid qR$

③ $R \rightarrow q \mid qRR$

2) If non-terminal occurs at first position in any production rule then replace that non-terminal by it's production rules.

① $P \rightarrow QR \mid pqR$

$P \rightarrow pR \mid qRR \mid pT_1R$

② $Q \rightarrow p \mid qR$

③ $R \rightarrow q \mid qRR$

Hence GNF is;

① $P \rightarrow pR \mid qRR \mid pT_1R$

② $Q \rightarrow p \mid qR$

③ $R \rightarrow q \mid qRR$

$T_1 \rightarrow q$

Q. ~~S → AA | 0~~

S → AA | 0

A → SS | 1

* if L.H.S symbol comes at 1st position then it is left recursion

S → AA | 0

S → SSA | 1A | 0

S → AASA | 0SA | 1A | 0

→ If such a type of situation occurs it is called as left recursion.

Rules for left recursion.
(Removal of left recursion)

① Divide the no. of productions into two categories as α and β where

α = no. of productions which contains left recursion.

β = no. of productions which doesn't contains left recursion.

S → AA | 0

S → S $\underbrace{SA}_{\alpha}$ | $\underbrace{1A}_{\beta_1}$ | $\underbrace{0}_{\beta_2}$

② Given production rule is splitted as follows.

L.H.S → all the betas (β_s) with L.H.S' & β_s

L.H.S' → all the alphas (α_s) with L.H.S' & α_s

S → 1AS' | 0S' | 1A | 0

S' → SASS' | SA

S' → 1AS'AS' | 0S'AS' | 1AAS' | 0AS'

| 1AS'A | 0S'A | 1AA | 0A

#

$A \to SS \mid 1$

$A \to 1AS'S \mid 0S'AS' \; 0S'S \mid 1AS \mid 0S \mid 1$

Hence GNF is

$S \to 1AS' \mid 0S' \mid 1A \mid 0$

$S' \to 1AS'AS' \mid 0S'AS' \mid 1AAS' \mid 0AS' \mid 1AS'A \mid 0S'A \mid 1AA \mid 0A$

$A \to 1AS'S \mid 0S'S \mid 1AS \mid 0S \mid 1$

Q. $S \to AB$

$A \to BSB \mid BB \mid b$

$B \to aAb \mid a$

~~$S \to BSBB \mid BBB \mid bB$~~

Rule I)

① $S \to AB$

② $A \to BSB \mid BB \mid b$

③ $B \to aAb \mid a$

$T_1 \to b$

$B \to aAT_1 \mid a$

Rule - II)

① ~~$S \to AB$~~

~~$S \to BSBB \mid BBB \mid bB$~~

~~$S \to a$~~

$A \to BSB \mid BB \mid b$

$A \to aAT_1SB \mid aSB \mid aAT_1B \mid aB \mid b$

② $S \to AB$

$S \to aAT_1SBB \mid aSBB \mid aAT_1BB \mid aBB \mid bB$

hence GNF is

$S \to aAT_1SBB \mid aSBB \mid aAT_1BB \mid aBB \mid bB$

$A \to aAT_1SB \mid aSB \mid aAT_1B \mid aB \mid b$

$B \to aAT_1 \mid a$

Automata theory ~~Un~~ MTE pattern

Q.1 Unit 1 — 15 (solve any 3 out of 4)
Q.2 Unit 2 — 15 (solve any 3 out of 4)
Q.3 unit 3 — 20 (solve any 4 out of 5)

context free grammar for a language.

Questions on recursion.

Q.1) construct a context free grammar for the following language.
~~no of a's equal to no. of b's~~
① $L = \{a^n b^n \mid n \text{ is greater than or equal to } a\}$ $n \geq a$

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

$S \rightarrow aSb \mid \epsilon$

$S \rightarrow aSb$
$\rightarrow ab$

asb
aasbb
aabb ($S \rightarrow \epsilon$)

② $L = \{a^n b^n \mid n \geq 1\}$

$S \rightarrow aS_1b$
$S_1 \rightarrow aS_1b \mid \epsilon$

$S \rightarrow aS_1b$
$S \rightarrow aaS_1bb$
$S \rightarrow aabb$ ($S_1 \rightarrow \epsilon$)

Q.2) construct a CFG for a language over a,b where no. of a's equal to no. of b's
Soln

$S \rightarrow aSb \mid bSa \mid \epsilon$

asb

ab S ab

ab bS a ab

ab b a ab

* Definition must write