Unit Production Removal

The unit production are the productions in which one non-terminal symbol observes another single non-terminal symbols.

Eg:- $A \to B$; $B \to C$
A, B are unit production.

To optimise the grammer, we need to remove the unit productions.

If $A \to B$ is a unit production and $B \to x_1, x_2\ x_3 \ldots x_n$, then while removing $A \to B$ production, we should add a rule.

$$A \to x_1, x_2, \ldots x_n$$

Eg:-

$A \to B$	equivalents	$A \to a$
$B \to C$	=	$B \to a$
$C \to D$		$C \to a$
$D \to a$		$D \to a$.

Eg:- Remove all the unit productions for the given grammer.

(answer)	(question)
$S \to 0A \mid 1B \mid 01$	$S \to 0A \mid 1B \mid C$
$A \to 0S \mid 00$	$B \to 1 \mid A$
$B \to 1 \mid 0S \mid 00$	$A \to 0S \mid 00$
$C \to 01$	$C \to 01$

*****

$S \to AB$	$S \to AB$
$A \to a$	$A \to a$
$B \to c \mid b$	$B \to Ab \mid bc \mid a \mid b$
$C \to D$	$C \to Ab \mid bc \mid a$
$D \to E \mid bc$	$D \to Ab \mid bc \mid a$
$E \to a \mid Ab$	$E \to a \mid Ab$

③

$S \to AB$
$A \to a$
$B \to c$
$C \to D$
$D \to b$

$S \to AB$
$A \to a$
$B \to b$
$C \to b$
$D \to b$

1) define the grammer
2) 4 tuple reprsentation for given grammer
3) define unit production
4) Formula for unit production
5) 4 tuple representation for required grammer
6) $L(G) = L(G')$

---

④ Remove the unit productions for the given grammer

$E \to E+T / T$, $T \to T*F / F$; $F \to (E) / id$

$E \to E+T \mid T*F \mid (E) \mid id$

$T \to T*F \mid (E) \mid id$ [∵ $F \to (E)$ is not unit production]

$F \to (E) \mid id$

---

⑤ $S \to ABA \mid AB \mid BA \mid AA \mid A \mid B$,

$A \to aA / a$

$B \to bB / b$

$S \to ABA / AB / BA / AA / aA / a / bB / b$

$A \to aA / a$

$B \to bB / b$

Removal of useless symbols

A symbol can be useless if it does not appear on the right hand side of production and it does not take part in the derivation of any string that symbol is useless.

Eg: $T \to aaB \mid abA \mid aaT$, $A \to aA$, $B \to ab/b$
$C \to ad$.

The variable 'C' is useless, because it is never occured in the right hand side of any production. And it has never occured in derivation of aabba. So we it is a useless symbol.

Therefore, after eliminating the useless production, the grammer should be

$T \to aaB / abA / aaT$, $A \to aA$; $B \to ab/b$.

→ A symbol is useful when it appears on the right hand side in the production and it generates some terminal string.

If it does not derives a terminal string, it is called as a useless symbol.

① $S \to A11B \mid 11A$

$S \to 1B \mid 11$

$A \to 0$

$B \to BB$

remove the useless variable ((or) non terminal) symbol

1. For any grammer the starting variable is useful variable, even if it does not occurs in right hand side of production.
2. Then check the variable which is ~~not~~ in right side of production and the variable should ~~be~~ produce the terminal symbol. If the conditions satisfy, it is not useless ~~Symbol~~ variable.
3. Then eliminate the complete string when useless variable is there.

Sol:- S, A is useful symbol and B is useless symbol.

$S \to$ ~~A~~ $11A$

$S \to 11$

$A \to 0$

[∵ we eliminated A11B and 1B completely]

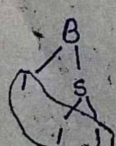
---

② $S \to A11B \mid 11A$, $S \to 1B \mid 11$, $A \to 0$, $B \to BB \mid S$

S is useful

A is useful

B is useful ∵



B derives terminal symbol.

---

8. $S \to aABbC \mid aB \mid bC$.

③ $S \to AB \mid CA$

$A \to a$

$B \to BC \mid AB$

$C \to aB \mid b$

remove useless variables

C, A, S is useful

B is useless symbol [∵ terminal symbol is not produced].

Now, $S \to CA$, $A \to a$, $C \to b$

---

④ $E \to E+T \mid T$, $T \to T*F \mid F$, $F \to (E) \mid id$

reduce the autometa into minimized autometa.

(1) No ϵ production

(2) Unit production

$E \to E+T \mid T*F \mid (E) \mid id$

$T \to T*F \mid (E) \mid id$

$F \to (E) \mid id$

(3) removal of useless production

No useless variables

∴ $E \to E+T \mid T*F \mid (E) \mid id$

$T \to T*F \mid (E) \mid id$

$F \to (E) \mid id$.

(4) $S \to aA/bB$; remove the useless symbols in the given grammar

$A \to aA/a$

$B \to bB$

$D \to ab/Ea$

$E \to ac/d$

$S \to aA$

B, D are useless grammar

$S \to aA/\cancel{bB}$ $\quad$ $S \to aA$

$A \to aA/a \Rightarrow A \to aA/a$

$E \to ac/d$ $\quad$ [eliminate E also]

---

(5) Minimize the context free grammar

$S \to aS/A/C$

$A \to a$

$B \to aa$

$C \to aCb$

No ϵ productions, unit production

$S \to aS/a/aCb$

$A \to a$

$B \to aa$

$C \to aCb$ (unit production removal)

here A, B, C are useless symbols

∴ The minimized context free grammar is

$S \to aS/a/\cancel{aCb}$

∴ $S \to aS/a$

---

(6) Minimize the context free grammar.

$S \to aA/a/Bb/cC$, $A \to aB$; $B \to a/Aa$,

$C \to cCD$, $D \to ddd$

i) No ϵ-productions

ii) No unit productions

iii) Here useless symbols are C and ; so remove

$S \to aA/a/Bb$, $A \to aB$ $\quad B \to a/Aa$,

$D \to ddd$

Now D is also useless symbol

∴ $S \to aA/a/Bb$, $A \to aB$ $\quad B \to a/Aa$

is the minimized context free grammar.

---

(7) Minimize the context free grammar

$S \to AB/a$, $A \to Bc/d$, $B \to aB/C$, $C \to aC/B$

i) No ϵ-productions

ii) No unit productions

Since you will get unit production; so do like this

$\boxed{C \to aC/aB}$

$S \to AB/a$

$A \to Bc/d$

$B \to aB/aC$

$C \to aC/aB$

Removal of useless variable.

here B and C are useless variable

$S \to AB/a$

$A \to b$

$S \to a$

$A \to b$

Furthermore A is useless variable

$S \to a$

Minimization of Finite Automata
(Myhil Nerode Theorem)

Input :- DFA

Output :- Minimized DFA

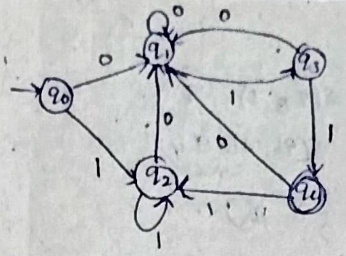
Step 1:- Draw a table for all pairs of states (P, Q)

Step 2:- Consider every pair (P, Q) in the DFA and mark P∈F, & Q∉F or vice versa and mark them.

Step 3:- Repeat step until we cannot mark any more states.

If there is unmarked pairs (P, Q) mark it if the pair δ(P,a), δ(Q,a) is marked.

Step 4:- Combine all unmarked pairs and make them as single state



{x → repeated pairs}

	q_0	q_1	q_2	q_3	q_4
q_0	X	X	X	X	X
q_1		X	X	X	X
q_2			X	X	X
q_3				X	X
q_4					X

	q_0	q_1	q_2	q_3	q_4
q_0					
q_1					

	q_0	q_1	q_2	q_3
q_1	X			
q_2		X		
q_3	X	X	X	
q_4	X	X	X	X

X → since q_4 is final and remaining is non final state

$\delta(q_0, q_3)$

$\delta(q_0,0) = q_1$	$\delta(q_0,1) = q_2$
$\delta(q_3,0) = q_1$	$\delta(q_3,1) = q_4$

So mark (q_0, q_3)

$\delta(q_1\ q_3)$

$\delta(q_1,0) = q_1$	$\delta(q_1,1) = q_3$
$\delta(q_3,0) = q_1$	$\delta(q_3,1) = q_4$

So mark (q_1, q_3)

$\delta(q_2\ q_3)$

$\delta(q_2,0) = q_1$	$\delta(q_2,1) = q_2$
$\delta(q_3,0) = q_1$	$\delta(q_3,1) = q_4$

So mark $(q_2\ q_3)$

$\delta(q_0, q_2)$

$\delta(q_0,0) = q_1$	$\delta(q_0,1) = q_2$
$\delta(q_2,0) = q_1$	$\delta(q_2,1) = q_2$

So dont mark

$\delta(q_1\ q_2)$

$\delta(q_1,0) = q_1$	$\delta(q_1,1) = q_3$
$\delta(q_2\ 0)\ q_1$	$\delta(q_2,1) = q_2$

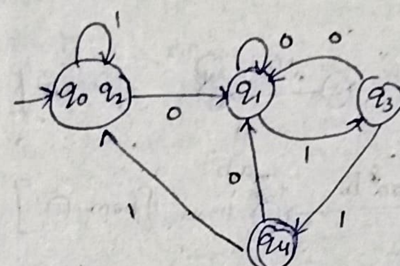
So mark $(q_1\ q_2)$

$\delta(q_0, q_1)$

$\delta(q_0,0) = q_1$	$\delta(q_0,1) = q_2$
$\delta(q_1,0) = q_1$	$\delta(q_1,1) = q_1$

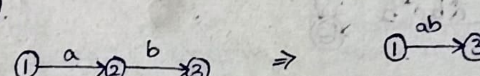
So mark $\delta(q_0\ q_1)$

Unmarked state is $q_0 q_2$ ⇒ so make as 1 state

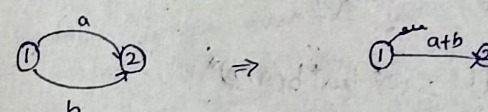


FA to RE using (state elimination method)

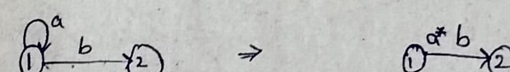
①.



②

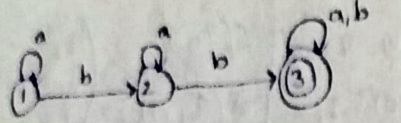


③



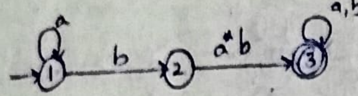
** We are eliminating all the states other than final and intial state.

5

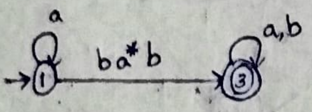


should eliminate all the intermediate state

removing :

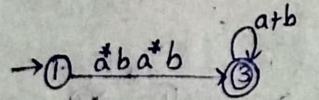


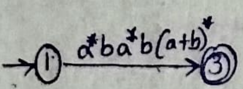
[from ②]



[from ①]

Since self loops are there we have to remove them





RE:- $a^* ba^* b(a+b)^*$

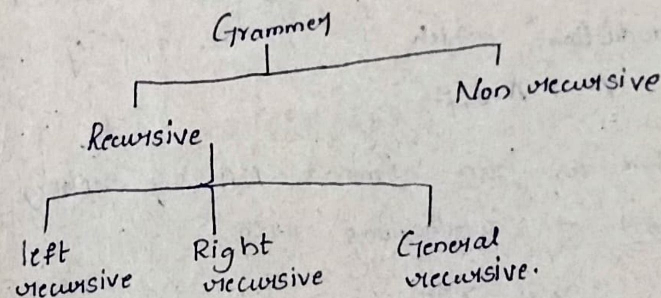
Types of grammer

Classification of grammer based on number of parse trees.

1) Ambiguous grammer
2) Unambigous grammer

* Classification of grammer based on number of strings

1) Recursive grammer
2) Non - recursive grammers



Recursive

A variable which exist in both left and right hand sides of the production, then that grammer is called recursive grammer

eg:- $S \rightarrow Sa/a$

Left recursive

A production of grammer is said to be left recursive grammer if the left most variable of its RHS is same as the value which

$B \to b | \epsilon$

$A \to BB | aA | ab$

- A grammer containing a product having left recursion is called left recursive grammer.
- Therefore left recursion have to be eliminated from the given grammer.

Elemenation of left recursion grammer.

→ Left recursion is eliminated by converting the grammer into a right reccursive grammer

→ If the left recursive grammer contains a production which is in the form of

$A \to A\alpha / \beta$.

Then we can eliminate L.R by replacing the pair of productions with

$A \to \beta A'$

$A' \to \alpha A' / \epsilon$

→ If the left recursive grammer contains a product which is in the form of

$A \to A\alpha_1 | A\alpha_2 / \dots A\alpha_n / \beta$

Then we can eliminate LR by replacing the pair of production with

$A \to \beta A'$

$A' \to \alpha_1 A' / \alpha_2 A' / \dots \alpha A' / \epsilon$

→ If the LRG contains a production which is in the form of

$A \to A\alpha_1 | A\alpha_2 | \dots A\alpha_n / \beta_1 / \beta_2 / B_3 \dots / \beta_n$

Then we can eliminate LR by replacing the pair of production with

$A \to \beta_1 A' | \beta_2 A' \dots / \beta_n A'$

$A' \to \alpha_1 A' / \alpha_2 A' / \dots / \alpha_n A' / \epsilon$

① $S \to Sa / Sb / Sab / c$

$S \to cS'$

$S' \to aS' | bS' / abS' / \epsilon$.

② $S \to Sa / a$

$S \to aS'$

$S' \to aS' / \epsilon$

③ $S \to Se | Sf / Sg / ae / bc$

$S \to aeS' | bcS'$

$S' \to eS' | fS' | gS' | \epsilon$

④ $A \to ABd / Aa / a$; $B \to Be / db$

$B \to bB'$ $B' \to eB' / \epsilon$

$A \to aA'$; $A' \to BdA' / aA' / \epsilon$

(5) $E \to E+T/T \; ; \; T \to T*F/F \; ; \; F \to (E)/id$

$E \to E'T$ $\quad T \to FT'$ $\quad F \to (E)/id$ (no change)

$E' \to +TE'/\epsilon$ $\quad T' \to *FT'/\epsilon$

(6) $E \to E+E/E*E/a$

$E \to aE'$ $\quad E' \to +EE'/*EE'/\epsilon$

(7) $S \to (L)/a$ $\quad L \to L,S/S$

$S \to (L)/a$ $\quad L \to SL'$

$L' \to ,SL'/\epsilon$

(8) $S \to S0S1S/01$ $\quad$ (9) $S \to A$

$S \to 01S'$

$S' \to 0S1SS'/\epsilon$

(9) $S \to A$

$A \to Ad/Ac/aB/ac$

$B \to bBc/f$

$S \to a$

$A \to aBA'/acA'$

$A' \to dA'/cA'/\epsilon$

$B \to bBc/f$

(10) $A \to A\alpha/\beta$

$A \to \beta A'$

$A' \to A\alpha A'/\epsilon$

(11) $A \to Ba/Aa/c$

$B \to Bb/Ab/d$

$A \to cA'/BaA'$

$A' \to aA'/\epsilon$

$B \to AbB'/dB'$

$B' \to bB'/\epsilon$

2) Right Recursive Grammer

A production of grammer is said to have a right recursion if the right most variable of RHS is same as the variable of its LHS.

→ If a grammer contains at least one production having right recursion is called as Right recursive grammer.

eg:- $S \to aS/a$

→ Right recursion doesn't create any problem for the top down parsers. Therefore there is no need of eliminate right recursion from the grammer.

3) General Recursive grammer

A recursion which is neither left nor right recursion is called as general recursion.

→ If the grammer contains general recursive production then it is called as general recursive grammer.

eg: $S \rightarrow aS/S$ right (no need to eliminate)

$S \rightarrow Sa/ab$ left

$S \rightarrow aSb/ab$ General (no need to eliminate)

Grammer with Common Prefixes

→ If RHS of more than one production starts with the same symbol then such grammer is called as grammer with common prefixes.

eg:- $A \rightarrow B\alpha_1 / B\alpha_2 / \dots / B\alpha_n$

→ This kind of grammer creates a problem of construction of top-down parsers.

→ Top down parsers cannot decide which production must be choosen to parse the given string.

→ To remove this cofusion we use left factoring.

Left factoring.

→ Left factoring is a process by which the grammer with common prefixing is transformed to make it useful for top down parces.

$A \rightarrow B\alpha_1 / B\alpha_2 / \dots B\alpha_n / \beta$.

$A \rightarrow BA' / \beta$

$A' \rightarrow \alpha_1 / \alpha_2 / \dots / \alpha_n$

→ In left factoring

i) We make one production for each common prefixes.

ii) The common prefixes may be a terminal, non terminal (or) combination of both.

iii) Rest of the derivation is added by new productions.

iv) The grammer obtained after the process of left factory is called as LFG.

① $S \rightarrow iEtS / iEtSeS / a$; $E \rightarrow b$

$S \rightarrow iEtSS' / a$

$S' \rightarrow eS / \epsilon$ $S' \rightarrow \epsilon / eS$ (epsilon)

$E \rightarrow b$

② $A \rightarrow aAB / aBC / aAC$

$A \rightarrow aA'$

$A' \rightarrow AB / BC / AC$

$A \rightarrow aA'$

$A' \rightarrow AA'' / BC$
$A'' \rightarrow B / C$] again LFG.

③ $S \rightarrow bSSaaS / bSSaSb / bSb / a$

$S \rightarrow bSSaS' / bSb / a$
$S' \rightarrow aS / Sb$ }

$S \to bSS \mid a$
$S' \to aS \mid Sb$
$S'' \to SaS' \mid b$

(4) $S \to aSSbS \mid aSaSb \mid abb \mid b$

$S \to aS' \mid b$
$S' \to SSbS \mid SaSb \mid bb$ } 1

$S \to aS' \mid b$
$S' \to SS'' \mid bb$
$S'' \to SbS \mid aSb$ } 2

(5) $S \to a \mid ab \mid abc \mid abcd$

$S \to aS'$
$S' \to \epsilon \mid b \mid bc \mid bcd$ } 1

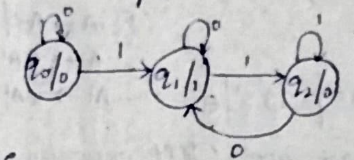
$S \to aS'$
$S' \to bS'' \mid \epsilon$
$S'' \to \epsilon \mid c \mid cd$ } 2

$S \to aS'$
$S' \to bS'' \mid \epsilon$
$S'' \to cS''' \mid \epsilon$
$S''' \to \epsilon \mid d$ } 3

(6) $S \to aAd \mid aB$; $A \to a \mid ab$; $B \to ccd \mid ddc$
$S \to aS'$; $S' \to Ad \mid B$

$A \to a \mid ab$
$B \to ccd \mid ddc$

Design melay and moore machines for 2's complement of binary number.
When we get 1 from right to left then all the remaining after '1' will becomes complement.



Take
1010

$\to q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2$
outputs: 0, 1, 1, 0

101100

$\to q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_1 \xrightarrow{1} q_3$
outputs: 0, 0, 1, 0, 1, 0

010100

→ In this problem we are reading the input string from LSB to MSB. So in that order take. If input is 0, we have to take all those symbols upto the symbol as complemented

(f) $S \rightarrow Sa \mid SaA \mid SaAB$

$A \rightarrow abS \mid ab$

$B \rightarrow Bab \mid Bb \mid Bbb$

(i) eliminate all left recursive

(ii) Minimization of CFG

$S \rightarrow Sa \mid SaA \mid SaAB \mid \epsilon$

[$\because A \rightarrow A\alpha \mid \beta$ then
$A \rightarrow \beta A'$
$A' \rightarrow \alpha A' \mid \epsilon$]

$S \rightarrow \epsilon S'$

$S' \rightarrow aS' \mid aAS' \mid aABS' \mid \epsilon$

If there again apply left recursion

$\therefore$ no left recursion

Now left factoring;

$S \rightarrow \epsilon S'$

$S' \rightarrow aS' \mid aAS' \mid aABS' \mid \epsilon$

[$\because A \rightarrow \beta\alpha_1 \mid \beta\alpha_2 \ldots \mid \beta$
$A \rightarrow \beta A' \mid \beta$
$A' \rightarrow \alpha_1 \mid \alpha_2 \mid \alpha_3 \ldots$]

$S' \rightarrow aS'' \mid \epsilon$

$S'' \rightarrow S' \mid AS' \mid ABS'$

$\because$ no left recursion

$\because$ there is left factoring.

[$\because A \rightarrow \beta\alpha_1 \mid \beta\alpha_2 \mid \beta$
$A \rightarrow \beta A' \mid \beta$
$A' \rightarrow \alpha_1 \mid \alpha_2 \mid \ldots$]

$S'' \rightarrow AS''' \mid S'$

$S''' \rightarrow S' \mid BS'$

$\therefore S \rightarrow \epsilon S'$

$S' \rightarrow aS'' \mid \epsilon$

$S'' \rightarrow AS''' \mid S'$

$S''' \rightarrow S' \mid BS'$ $\rightarrow$ [If in the form $S''' \rightarrow S' \mid S'B$ then we have to apply left factoring]

[No left recursion, left factoring].

Now Minimization,

$S' \rightarrow \epsilon$

$S \rightarrow S'$

$S' \rightarrow aS''$

CNF (CHOMSKY Normal Form)

$S \to \epsilon$ (starting symbol derives ϵ)

$C \to AB$ — rules of CNF (non terminal derives only two non terminals)

$B \to a$ (non terminal derives only one terminal symbol)

$S \to aAB$
$A \to aBb$
$B \to b$

$S \to CB$
$C \to DA$
$D \to a$

$A \to aBb$
$A \to aE$
$A \to Bb$
$A \to BF$
$F \to b$

① Convert all the productions of CFG into CNF

$S \to aAD$, $A \to aB / bAB$, $B \to b$, $D \to d$

i. No ϵ production
ii. No unit production
iii. No useless symbols.

So it is minimized CFG.

To convert into CNF.

$S \to aAD$
$S \to A'D$
$A' \to aA$
$A' \to B'A$
$B' \to a$

$A \to aB / bAB$
$A \to B' / B''B$
$B' \to aB$
$B' \to B'''B$
$B''' \to a$
$B'' \to bA$
$B'' \to B^{iv}A$
$B^{iv} \to b$

$B \to b$,
$D \to d$

② $S \to 1A / 0B$, $A \to 1AA / 0S / 0$, $B \to 0BB / 1S / 1$

Reduced CFG

Now

$S \to CA / DB$
$C \to 1$
$D \to 0$

$A \to EA / GS / 0$
$E \to FA$
$F \to 1$
$G \to 0$

$B \to HB / IS / 1$
$H \to JB$
$J \to 0$
$I \to 1$

(or) directly

$S \to CA / DB$
$A \to EA / DS / 0$
$E \to CA$
$B \to HB / CS / 1$
$H \to DB$

$C \to 1$
$D \to 0$

③ $S \to aS / ABa$; $A \to BaA / B$; $B \to bC / b / \epsilon$; $C \to aA / aC$

Reduced CFG

$S \to DS / AE$
$E \to BD$
$A \to EA / DF /$

$D \to a$
$F \to b$

$A \to \epsilon$

$B \to \epsilon$

$S \to aS \mid ABa \mid Aa$

$A \to BaA \mid aA \mid B \mid \epsilon$

$B \to aB\epsilon \mid aC \mid b$

$C \to aA \mid aC$

$S \to aS \mid ABa \mid Ba \mid Aa \mid a$

$A \to BaA \mid Ba \mid aA \mid a \mid B$

$B \to aBC \mid aC \mid b$

$C \to aA \mid a \mid aC$

Unit production removal

$S \to aS \mid ABa \mid Ba \mid Aa \mid a$

$A \to BaA \mid Ba \mid aA \mid a \mid aBC \mid aC \mid b$

$B \to aBC \mid aC \mid b$

$C \to aA \mid a \mid aC$

Now

$S \to DS \mid AE \mid BD \mid AD \mid a$ $\{D \to a\}$

$E \to BD$

$A \to EA \mid BD \mid DA \mid FC \mid DC \mid b$

$F \to DB$

$B \to BC \mid DC \mid b$

$C \to DA \mid a \mid DC$

$D \to a$

③ $S \to aAB,\ A \to BaC,\ B \to b \mid ABa,\ C \to d$

No ϵ production, unit production, usless symbols

so it is minimized CFG

Now CNF

$S \to DB$

$D \to EA$

$A \to BF$

$F \to EC$

$B \to b \mid AG$

$G \to BE$

$C \to d$

$E \to a$

$E \to a$

② $S \to a \mid aA \mid B,\ A \to aBB \mid \epsilon,\ B \to Aa \mid b.$

$A \to \epsilon$

$S \to a \mid aA \mid a \mid B$

$A \to aBB$

$B \to Aa \mid a \mid b$

unit productions.

$S \to a \mid aA \mid a \mid aBB\ Aa \mid a \mid b$

$A \to aBB$

$B \to Aa \mid a \mid b$

∴ Minimized CFG

$S \to a \mid CA \mid AC \cancel{a} \mid b$ $C \to a$

$A \to DB$

$D \to CB$

$B \to AC \mid a \mid b$

$C \to a$

GNF (Greibach Normal Form)

rules:- A CFG is said to be in GNF; if all its productions satisfies any one of three rules. [Starting symbol derives ϵ]

i) $S \rightarrow \epsilon$

ii) $A \rightarrow a AB$ [non terminal derives a terminal followed by any number of non terminal symbols].

iii) $A \rightarrow a$. [non terminal derives only one terminal symbol].

If CFG satisfies these rules, that CFG is called GNF,

GNF can also be defined as,

In $A \rightarrow a\alpha$,

a is the terminal symbol, A is the non-terminal symbol, α is 0 (or) more occurrence of any ~~cont~~ combination of non terminal symbol. along with $S \rightarrow \epsilon$.

① $S \rightarrow aSA \mid ab$; $A \rightarrow a/b \mid Sa \mid AS$ · Check weather the productions is in GNF or not.

$S \rightarrow aSA \mid ~~ab~~$ ⊗ because not in forms $A \rightarrow a\alpha$ where α is 0 (or) more occurance of Non-terminal symbols.

$A \rightarrow a \mid ~~b~~ \mid ~~Sa~~ \mid AS$ ⊗

$S \rightarrow aSA$ ✓

$S \rightarrow ab$ ⊗

$A \rightarrow a$ ✓

$A \rightarrow b$

$A \rightarrow Sa$ X

$A \rightarrow AS$ X

Steps to convert into GNF

1. Reduced into CFG (removal of, ϵ, unit, useless).
2. Convert CFG in ~~GNF~~ CNF.
3. Then ~~replace~~ & rename the production with the increasing order of the subscript.

Procedure to convert a given CFG into GNF.

1. Simplify the given Context Free Grammer [removal of ϵ- production, unit production, useless symbols].
2. Convert the simplified grammer to CNF.
3. Rename all the variables with $A_1, A_2, A_3, \dots A_n$
4. Choose a production such that left hand side variable subscript is greater than right hand side starting variable subscript.
5. Then apply Lemma 1 (or) Lemma 2 according to production.
6. Repeat applying Lemma 1 (or) Lemma 2 for all the productions of that grammer in GNF.

Lemma1 (Substitution rule)

Principal:- $A \rightarrow B\alpha$

$B \rightarrow \beta_1 \mid \beta_2 \mid \dots \beta_n$

$A \rightarrow \beta_1\alpha \mid \beta_2\alpha \mid \dots \mid \beta_n\alpha$

eg: $A \to Bc|a|b$

$B \to a|b$

$A \to ac|bc|a|b$

Lemma 2 (Elimination of Left Recursion).

Principle eg: ① $A \to A\alpha|B$

then $A \to BA'$

$A' \to \alpha A'|\epsilon$

$\Rightarrow$ $A \to BA'|B$ $[\because A' \to \epsilon]$

$A' \to \alpha A'|\alpha$

② $A \to A\alpha_1|A\alpha_2|\ldots\beta_1|\beta_2|\ldots$

then $A \to B_1A'|B_2A'|\ldots|B_1|B_2|B_3\ldots$

$A' \to \alpha_1A'|\alpha_2A'|\alpha_3A'|\ldots|\alpha_1|\alpha_2|\alpha_3\ldots$

① $S \to AA|a$, $A \to SS|b$

Convert CFG into GNF

$S \to AB1|0$, $A \to 00S|B|\epsilon$, $B \to 1A1$

Removal of ϵ-production.

$A \to \epsilon$

$S \to AB1|B1|0$, $A \to 00S|B$, $B \to 1A1|11$

Removal of unit production

$S \to AB1|B1|0$, $A \to 00S|1A1|11$, $B \to 1A1|11$

No useless symbols

$\therefore$ This is minimized CFG.

Now CFG to CNF.

$S \to AC|\cancel{AD}|0$. $C \to BD$

$A \to \cancel{EES}$ $D \to 1$

$E \to 00$ $E \to GG$

$S \to AC|BD|0$ $F \to BA$

$A \to ES|FD|DD$ $G \to 0$

$B \to FD|DD$

(still there)

① $S \rightarrow AB$, $A \rightarrow a$. $B \rightarrow C/b$. $C \rightarrow a$

Convert into GNF

Unit production

$S \rightarrow AB$, $A \rightarrow a$, $B \rightarrow a/b$, $C \rightarrow a$

Useless Symbol

$S \rightarrow AB$, $A \rightarrow a$, $B \rightarrow a/b$,

∴ Minimized grammer

$S \rightarrow AB$, $A \rightarrow a$, $B \rightarrow a/b$

is in CNF.

Now applying lemma 1

$S \rightarrow aB$, $A \rightarrow a$, $B \rightarrow a/b$

is is GNF.

② $E \rightarrow E+T/T$, $T \rightarrow T*F/F$, $F \rightarrow a$

Convert in GNF

Unit production

$E \rightarrow E+T/T*F/a$

$T \rightarrow T*F/a$

$F \rightarrow a$

Now $E \rightarrow EA/TC/a$

$T \rightarrow TC/a$

$F \rightarrow a$

$A \rightarrow BT$

$B \rightarrow +$

$C \rightarrow DF$

$D \rightarrow *$

Is in CNF

Now By looking, apply lemma 2, lemma 1

$T \rightarrow TC/a$

$T \rightarrow aG/a$ (In GNF)

$G \rightarrow CG/C$

$G \rightarrow DFG/DF$

$G \rightarrow *FG/*F$ [In GNF]

$E \rightarrow EA/TC/a$

$E \rightarrow TCH/aH/TC/a$

$H \rightarrow AH/A$

$E \rightarrow aGCH/aCH/aGC/aC/a/aH$ [In GNF].

$H \rightarrow AH/A$

$H \rightarrow BTH/BT$

$H \rightarrow +TH/+T$ [In GNF].

$F \rightarrow a$ [In GNF]

$A \rightarrow BT$

$A \rightarrow +T$ [In GNF]

$C \rightarrow DF$

$C \rightarrow *F$

[In GNF].