20/10/2020

FLAT - Unit III

CFG → Accepted by PDA → has stack as memory component

↓

Type 2 Grammar

CFG = {V, T, P, S}

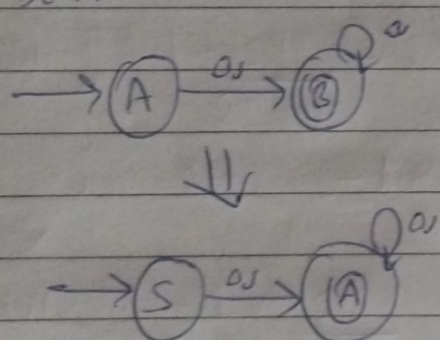
Variables, Terminal, Production, Start symbol.

Variables ↓ uppercase (A, B)

Terminal ↓ lowercase, digits, symbols etc.

✗ $L = \{a, aa, aaa, \ldots\}$, $\Sigma = \{a\}$

→ DFA:



$$\boxed{\begin{array}{l} S \to aA \\ A \to aA \\ A \to \epsilon \end{array}}$$

→ add ε for final state

Check 'a':

$S \to aA$

$\to a\epsilon$

$\to a$

Check 'aa':

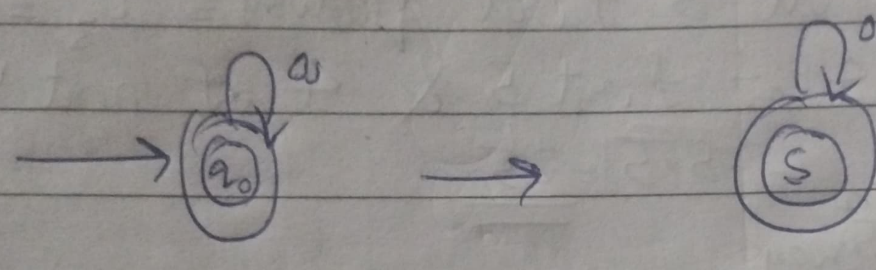
$S \to aA$

$\to aaA$

$\to aa\epsilon$

$\to aa$

27/10/2020 (Missed, saw later) FLAT Unit-3

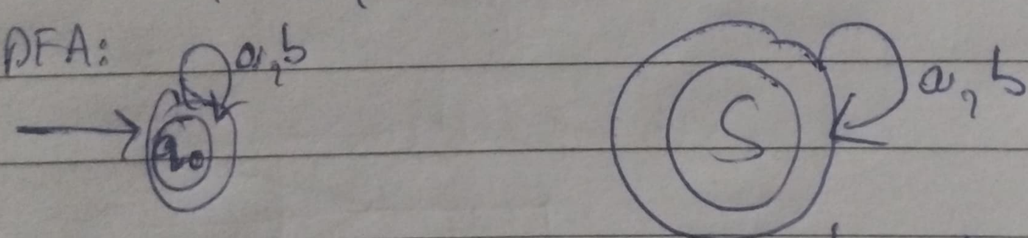


$\delta(q_0, a) = q_0$ $S \rightarrow aS \mid \epsilon$

* Obtain grammar to generate a string consisting of any no. of a's & b's.

→ $L = \{\epsilon, a, b, ab, ba, aba, aaa, baa, \ldots\} = \{(a+b)^*\}$

DFA:



$S \rightarrow aS \mid bS \mid \epsilon$ or $S \rightarrow aS$

$S \rightarrow bS$

$S \rightarrow \epsilon$

For ab

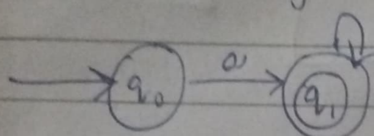
$S \to aS$

$S \to abS$

$S \to ab$ $\quad$ {S ~~replaced~~ ($S \to ab\epsilon$ equivalent to ab)

* Grammar to generate strings consisting of atleast 1 a, over $\Sigma = \{a\}$

→



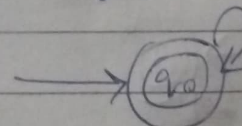
$L = \{a, aa, aaa, \ldots\}$ = ~~{aa*}~~ $= \{a^n \mid n >= 1\}$

S A		S
$S \to aA$	or	$S \to a$
$A \to aA$		$S \to aS$ } $S \to aS \mid a$
$A \to \epsilon$		
$V = \{S, A\}$ $T = \{a, \epsilon\}$		

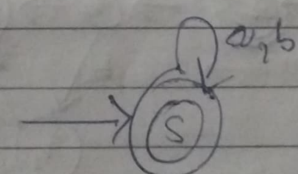
10/11/2020

* Obtaining grammar to generate any no. of a's & b's.

→



$\delta(q_0, a) = \delta(q_0, b) = q_0$



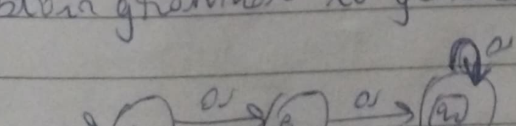
$S \to aS \mid bS \mid \epsilon$

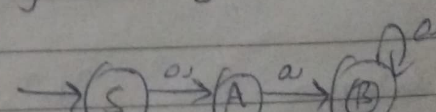
$V = \{S\}$

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

* Obtain grammar to generate string consisting of atleast 2 a's.

→





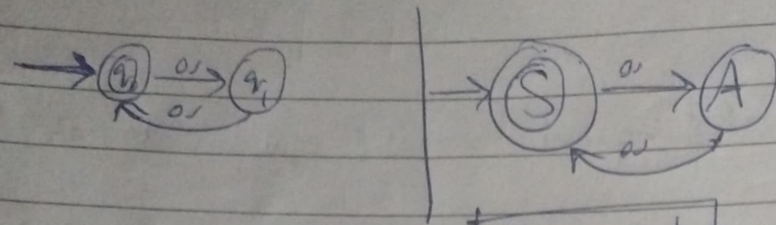
$S \to aA$

$A \to aB$ $\quad$ or $S \to aS \mid aa$

$B \to aB \mid \epsilon$

* Grammar for $L = \{n_a(w) \bmod 2 = 0\}$

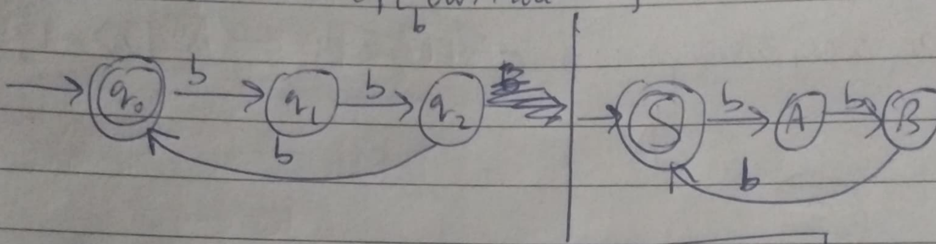
→



$\boxed{\begin{array}{l} S \to aA \mid \epsilon \\ A \to aS \end{array}}$ or $\boxed{S \to \epsilon \mid aaS}$

* Grammar for $L = \{n_b(w) \bmod 3 = 0\}$

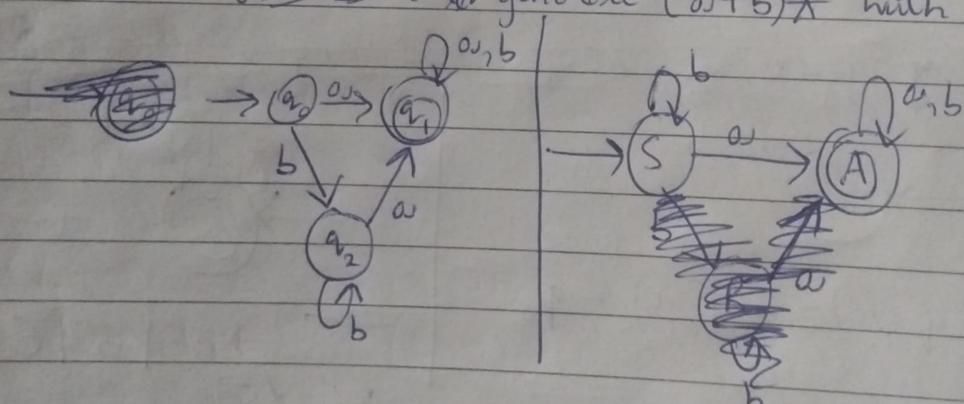
→



$S \to bbbS \mid \epsilon$ or $\boxed{\begin{array}{l} S \to bA \mid \epsilon \\ A \to bB \\ B \to bS \end{array}}$

* Grammar ~~for L =~~ to generate $(a+b)^*$ with atleast 1 a.

→



$S \to aA \mid bS$

$A \to aA \mid bA \mid \epsilon$

~~$B \to aA \mid bB$~~

* Grammar for $a^n b^n$, $n >= 0$.

→ $L = \{\epsilon, ab, aabb, aaabbb, \ldots\}$

$\boxed{S \to \epsilon \mid aSb}$

Date

if $n>=1$,
$L = \{ab, aabb, aaabbb, \dots\}$

$$\boxed{S \to ab \mid aSb}$$

* Grammar for $\{a^{n+1} b^n, n>=0$.

→ $L = \{a, aab, aaabb, \dots\}$

$$S \to a \mid aSb$$

for $a^n b^{n+1}, n>=0$
$L = \{b, abb, aabbb, \dots\}$

$$S \to b \mid aSb$$

* Grammar for $L = \{a^{2n} b^n \mid n>=0\}$

→ $L = \{\epsilon, aab, aaaabb, \dots\}$

$$S \to \epsilon \mid aaSb$$

* Grammar for $L = \{a^n b^{2n} \mid n>=0\}$

→ $S \to \epsilon \mid aSbb$

* Grammar for $L = \{0^m 1^m 2^n \mid m>=1, n>=0\}$

→ $S \to AB$
$A \to 01 \mid 0A1$
$B \to \epsilon \mid 2B$

* Grammar for $(a+b)^*$
→ $S \rightarrow \epsilon \mid aS \mid bS$

* Grammar for $(a+b)^* ab (a+b)^*$
→ $S \rightarrow AabA$
$A \rightarrow \epsilon \mid aA \mid bA$

* Grammar for $(a+b)^* ab$
→ $S \rightarrow Aab$
$A \rightarrow \epsilon \mid aA \mid bA$

* Grammar for $01(0+1)^*$
→ $S \rightarrow 01A$
$A \rightarrow \epsilon \mid 0A \mid 1A$

* Consider the grammar:
$E \rightarrow E+E \mid E-E \mid E*E \mid E/E \mid id$
~~...~~ string of terminals id+id*id
derive the
→ LMD → left most derivation (LM Variable)
RMD → right most derivation (RM Variable)

$E \rightarrow E+E$ (LMD)
$\rightarrow id+E$ ($E \rightarrow id$)
$\rightarrow id+E*E$ ($E \rightarrow E*E$)
$\rightarrow id+id*E$ ($E \rightarrow id$)
$\rightarrow id+id*id$ (")

Derivation is process of obtaining string of terminals.

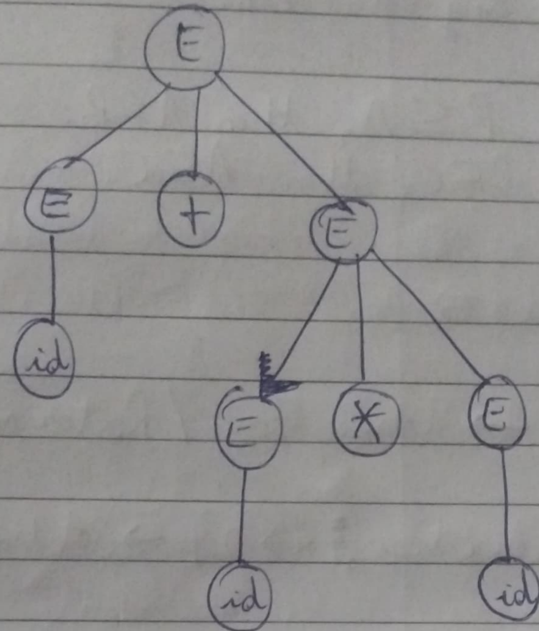
* $S \rightarrow aCa$
$C \rightarrow aCa \mid b$
→ $a^n b a^n$

Date

☆ Parse Tree:-

└→ Derivation Tree (May be leftmost or rightmost)

For id+id*id:



Read L to R excluding E. → To get yield of tree.

If grammar has more than 1 parse tree for same string of terminals, its ambiguous grammar

17/11/2020

1) Construct LMD parse tree & RMD parse tree for given string of terminals & if the parse trees are different, then grammar is ambiguous.

2) Apply LMD for some string w two times & construct parse trees & compare them. If they're different, grammar is ambiguous.

3) ~~Sta~~ (2) replace LMD with RMD.

17/11/2020

★ Check whether given grammar is ambiguous or not:-

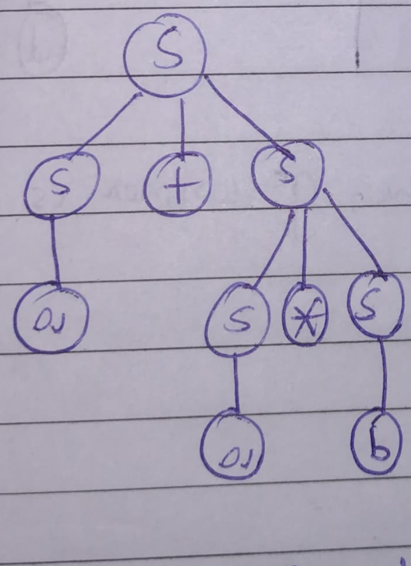
1) $S \rightarrow S+S$
$S \rightarrow S*S$
$S \rightarrow a|b$
for $\boxed{a+a*b}$

→ $a+a*b$

Applying LMD:

$S \xrightarrow{lm} S+S$
$\rightarrow a+S$ ~~(S → S*S)~~
$\rightarrow a+S*S$
$\rightarrow a+a*S$
$\rightarrow a+a*b$

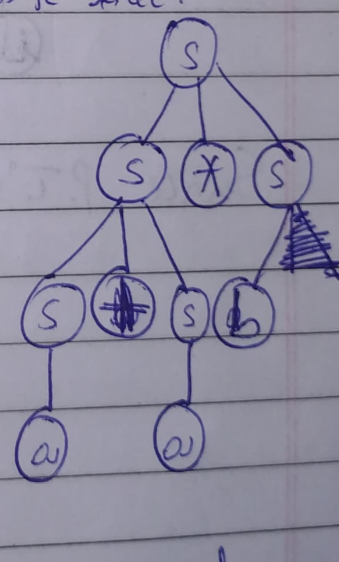
Parse tree:



Applying LMD:

$S \xrightarrow{lm} S*S$
$\rightarrow S+S*S$
$\rightarrow a+S*S$
$\rightarrow a+a*S$
$\rightarrow a+a*b$

Parse tree:



Since parse trees are different, grammar is ambiguous.

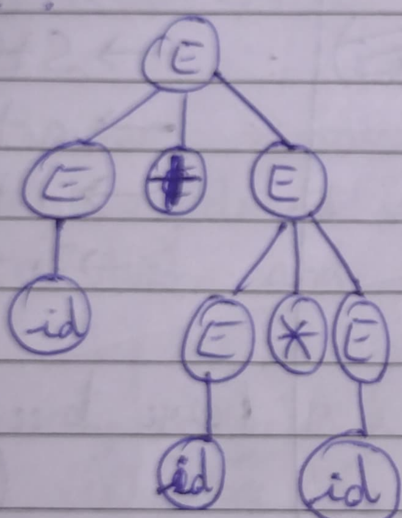
P.T.O.

2) $E \rightarrow E+E \mid E*E$
$E \rightarrow id$ for $id+id*id$

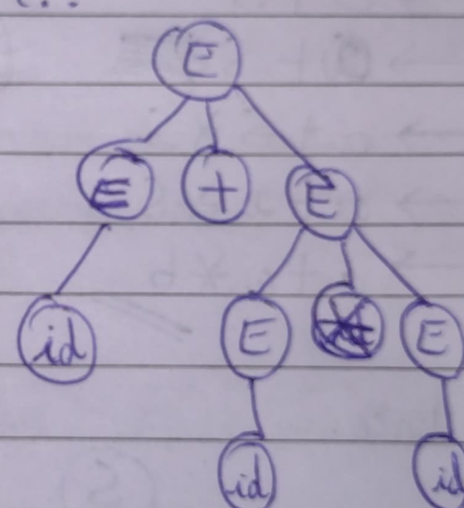
$\rightarrow$ $id+id*id$

Applying LMD:	Applying RMD:
$E \xrightarrow{lm} E+E$	$E \xrightarrow{rm} E+E$
$\rightarrow \cancel{E}\,id+E$	$\rightarrow E+E*E$
$\rightarrow id+E*E$	$\rightarrow E+E*id$
$\rightarrow id+id*E$	$\rightarrow E+id*id$
$\rightarrow id+id*id$	$\rightarrow id+id*id$

P.T. :



P.T. :



Since P.T.'s are same, grammor is not ambiguous.

✪ Normal forms:-

CFG

L | R
↓ | X (No restrictions)
1 S
↓
Non-terminal

1) CNF (Chomsky Normal Form)
2) GNF (Greibach " ")

1) CNF : A Grammar is in Chomsky normal form if & only if all of grammar productions are of form $A \to a \mid BC$

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

$T = \{a\}$

2) GNF : Grammar is in GNF if all productions are of the form : $A \rightarrow a\alpha$, $\alpha \in (V \cup T)^*$

begin with terminal

* Generate a grammar for given language:

$L = \{a^n b^{n-3} \mid n \geq 3\}$

→ $L = \{aaa, aaaab, aaaaabb, \ldots\}$

$S \rightarrow aaa \mid aSb$ or $\boxed{\begin{array}{l} S \rightarrow aaaA \\ A \rightarrow aAb \mid \epsilon \end{array}}$

* $L = \{a^{n+2} b^m \mid n \geq 0, m > n\}$

→ ~~L = {aa, aab, aabb, ..., aaab, aaabb, aaabbb, ...}~~

$L = \{aabb^*, aaabbb^*, aaaabbbb^*, \ldots\}$

~~$S \rightarrow aaA$~~

~~$A \rightarrow \epsilon \mid aAb \mid b$~~

$S \rightarrow aAB$

$A \rightarrow aAb \mid ab$

$B \rightarrow bB \mid \epsilon$

20/11/2020 FLAT

☆ Examples:-

1) Given:

$E \rightarrow E+T \mid T$

$T \rightarrow T*F \mid F$

$F \rightarrow (E) \mid a$

perform LMD & RMD for $w = a+a*a$.

→ LMD:-

$E \rightarrow E+T$

$\rightarrow T+T$

$\rightarrow F+T$

$\rightarrow a+T$

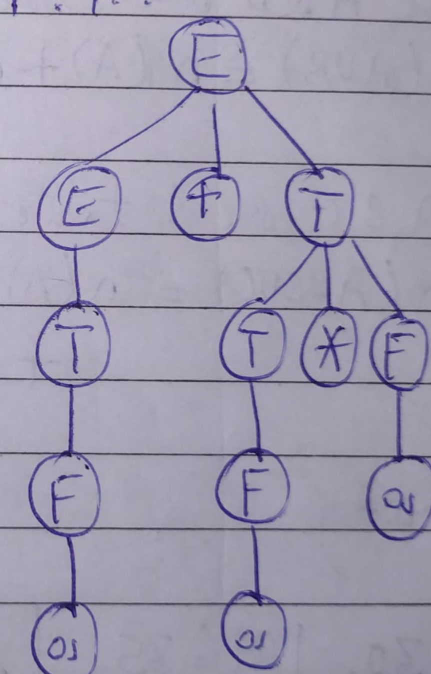
$\rightarrow a+T*F$

$\rightarrow a+F*F$

$\rightarrow a+a*F$

$\rightarrow \underline{a+a*a}$

P.T. :-



~~RMD:-~~

$\cancel{E \rightarrow T}$

$\cancel{\rightarrow T*F}$

$\cancel{\rightarrow T*a}$

$\cancel{\rightarrow T*F}$

RMD:–

$E \to E+T$

$E \to E+T*F$

$\to E+T*a$

$\to E+F*a$

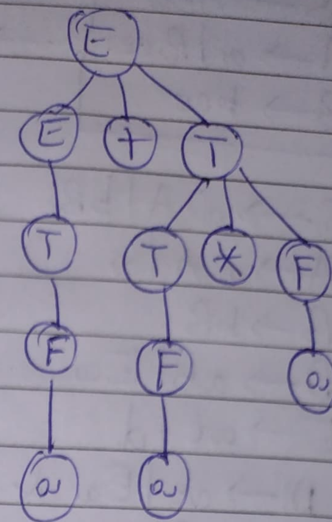
$\to E+a*a$

$\to T+a*a$

$\to F+a*a$

$\to a+a*a$

P.T.:–



☆ Simplification Of CFG:–

Step 1) Eliminate useless symbols & productions.

Step 2) " unit productions.

Step 3) " epsilon " .

1) Eliminating useless symbols & productions:–

↓

non-generating symbols,
not reachable from the start symbol.
& non-gen. symbols.

✱ Eliminate useless symbols for the grammar:–

i) $S \to ABa \mid BC$

$B \to bcc$

$C \to CA$

$D \to E$

$A \to aC \mid Bee \mid a$

$E \to e$

→ $C \to CA$ does not produce string of ~~symbols~~ terminals.

Hence, C is non-generating symbol.

$S \rightarrow ABa$
$A \rightarrow a \mid Bcc$
$B \rightarrow bcc$

→ Simplified grammar.

ii) $S \rightarrow aA \mid bB$
$A \rightarrow aA \mid a$
$B \rightarrow bB$
$D \rightarrow ab \mid Ea$
$E \rightarrow aC \mid d$

→ $D \rightarrow ab \mid Ea$ & $E \rightarrow aC \mid d$ are useless.
$B \rightarrow bB$ is non generating.

∴ $\boxed{\begin{array}{l} S \rightarrow aA \\ A \rightarrow aA \mid a \end{array}}$

OV (Old var)	NV (New Var)	P (Prods.)
ϕ	A, D, E	$A \rightarrow a$, $D \rightarrow ab$, $E \rightarrow d$
A, D, E	A, D, E, S	$S \rightarrow aA$, $A \rightarrow aA$, $D \rightarrow Ea$
A, D, E, S	A, D, E, S	~~$S \rightarrow aA$~~ ~~$A \rightarrow a \mid aA$~~ ~~$D \rightarrow ab \mid Ea$~~ ~~$E \rightarrow d$~~

$S \rightarrow aA$
$A \rightarrow aA \mid a$
$D \rightarrow ab \mid Ea$
$E \rightarrow d$

} Eliminate ⇒ $\boxed{\begin{array}{l} S \rightarrow aA \\ A \rightarrow aA \mid a \end{array}}$ → Answer

iii) $S \rightarrow AB \mid a \mid b$
$A \rightarrow aa \mid \epsilon \mid aA$
$B \rightarrow \epsilon \mid bB \mid b$

→ Variable producing ϵ is called Nullable variable.
Here, A & B produce ϵ directly. S produces ϵ indirectly via AB.
∴ All are nullable.

Unit productions example:

$S \rightarrow AB \mid a \mid b$

$\left.\begin{array}{l} A \rightarrow B \\ B \rightarrow C \end{array}\right\}$ unit productions.

$C \rightarrow cC \mid d$

23/11/2020

☆ Simplification Of Grammar:-

i) $S \rightarrow aA \mid a \mid Bb \mid cC$
$A \rightarrow aB$
$B \rightarrow a \mid Aa$
$C \rightarrow cCD$
$D \rightarrow ddd$

→

OV	NV	P
ϕ	S, B, D	$S \rightarrow a$, $B \rightarrow a$, $D \rightarrow ddd$
S, B, D	S, B, D, A	$S \rightarrow Bb$, $A \rightarrow aB$
S, B, D, A	S, B, D, A	$S \rightarrow aA$, $B \rightarrow Aa$
S, B, D, A	S, B, D, A	

Date

Now, we have:

$S \rightarrow a \mid Bb \mid aA$

$A \rightarrow aB$

$B \rightarrow a \mid Aa$

$D \rightarrow ddd$

Since D is not reachable from start symbol, eliminate D.

$\therefore$
$$\boxed{\begin{array}{l} S \rightarrow a \mid Bb \mid aA \\ A \rightarrow aB \\ B \rightarrow a \mid Aa \end{array}}$$

ii) Eliminate epsilon productions:-

$S \longrightarrow ABCa \mid bD$

$A \longrightarrow BC \mid b$

$B \longrightarrow b \mid \epsilon$

$C \rightarrow c \mid \epsilon$

$D \rightarrow d$

① ϵ prods.
② unit "
③ useless symbols.

Follow the order to simplify.

OV	NV	Prods.
ϕ	B, C	$B \rightarrow \epsilon$, $C \rightarrow \epsilon$
B, C	B, C, A	$A \rightarrow BC$
B, C, A	B, C, A	

Nullable variable list = {A, B, C}

Substituting A as ϵ in ABCa
↓

$S \rightarrow ABCa \mid BCa \mid ACa \mid ABa \mid Aa \mid Ba \mid Ca \mid a \mid bD$
$A \rightarrow BC \mid B \mid C \mid b$
$B \rightarrow b$
$C \rightarrow c$
$D \rightarrow d$

iii) Eliminate ϵ-prodn.:-
$S \rightarrow BAAB$
$A \rightarrow 0A2 \mid 2A0 \mid \epsilon$
$B \rightarrow AB \mid 1B \mid \epsilon$

OV	NV	Prodn.
ϕ	A, B	$A \rightarrow \epsilon$, $B \rightarrow \epsilon$
A, B	A, B, S	$S \rightarrow BAAB$, $B \rightarrow AB$
A, B, S	A, B, S	

Nullable var. list = {A, B, S}

$S \rightarrow BAAB \mid AAB \mid BAB \mid BAA \mid AA \mid BB \mid A \mid B \mid BA \mid AB$
$A \rightarrow 02 \mid 20 \mid 0A2 \mid 2A0$
$B \rightarrow B \mid A \mid AB \mid 1B \mid 1$

* Eliminating unit productions:-

i) $S \rightarrow ABC \mid aA \mid bB$
$\left.\begin{array}{l} A \rightarrow B \\ B \rightarrow C \end{array}\right\}$ unit prodn.
$C \rightarrow c \mid d$

P. T. O.

First, seperate unit prodn. from non-unit prodn.'s
Then, for unit prodn.'s, construct dependency graph.

$$A \to B \to C$$

Examples:-

i) $S \to AB$

$A \to a$

$B \to C \mid b$

$C \to D$

$D \to E \mid bc$

$E \to d \mid Ab$

→ Non-unit prodn.s :-

$S \to AB$, $A \to a$, $B \to b$, $D \to bc$, $E \to d \mid Ab$

Unit prodn.'s:-

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

$$B \to C \to D \to E$$

Now:

$$\boxed{\begin{array}{l} D \to d \mid Ab \mid bc \\ C \to d \mid Ab \mid bc \\ B \to d \mid Ab \mid b \mid bc \\ S \to AB \\ A \to a \\ E \to d \mid Ab \end{array}}$$

ii) $S \to A0 \mid B$
$B \to A \mid 11$
$A \to 0 \mid 12 \mid B$

→ Non unit prodn.'s:

$S \to A0$
$B \to 11$
$A \to 0 \mid 12$

Unit prodn.'s

$S \to B$
$B \to A$
$A \to B$

Dependency graph:-

$$\begin{tikzcd} S \arrow[r] & B \arrow[r, bend left] & A \arrow[l, bend left] \end{tikzcd}$$

Now:-

$B \to 11 \mid 0 \mid 12$
$A \to 11 \mid 0 \mid 12$
$S \to A0 \mid 11 \mid 0 \mid 12$

iii) $S \to Aa \mid B \mid Ca$
$B \to aB \mid b$
$C \to Db \mid D$
$D \to E \mid d$
$E \to ab$

→ Unit prodn.'s:-

$S \to B$
$C \to D$
$D \to E$

$$S \to B \qquad C \to D \to E$$

Now:-

$S \to Aa \mid Ca \mid aB \mid b$
$D \to ab \mid d$
$C \to Db \mid ab \mid d$
$B \to aB \mid b$
$E \to ab$

☆ Normal Forms:-

1) CNF → Chomsky NF
2) GNF → Greibach NF

CNF:-
$A \to BC$ or $A \to a$ type prodn.s

~~CNF~~ For converting to CNF, first eliminate ϵ prodn.'s, unit prodn.'s & useless symbols & prodn.'s.

* Convert given grammar into CNF:-

i) $S \to 0A \mid 1B$
$A \to 0AA \mid 1S \mid 1$
$B \to 1BB \mid 0S \mid 0$

→ $A \to 1$
$B \to 0$

$S \to 0A$ replace 0 by B_0 $S \to B_0A$
$B_0 \to 0$

$S \to 1B$ replace 1 by B_1 $S \to B_1B$
$B_1 \to 1$

$A \to 0AA$ replace 0 by B_0 $A \to B_0AA$
$B_0 \to 0$

$A \to 1S$ replace 1 by B_1 $A \to B_1S$
$B_1 \to 1$

$B \to 1BB$ replace 1 by B_1 $B \to B_1BB$
$B_1 \to 1$

$B \to 0S$ replace 0 by B_0 $B \to B_0 S$

$B_0 \to 0$

$A \to B_0 AA$ replace $B_0 A$ by D_0 $A \to D_0 A$

$D_0 \to B_0 A$

$B \to B_1 BB$ replace $B_1 B$ by D_1 $B \to D_1 B$

$D_1 \to B_1 B$

$\therefore$ We get:-

$S \to B_0 A \mid B_1 B$

$B_0 \to 0$

$B_1 \to 1$

$A \to 1 \mid B_1 S \mid D_0 A$

$B \to 0 \mid B_0 S \mid D_1 B$

$D_0 \to B_0 A$

$D_1 \to B_1 B$