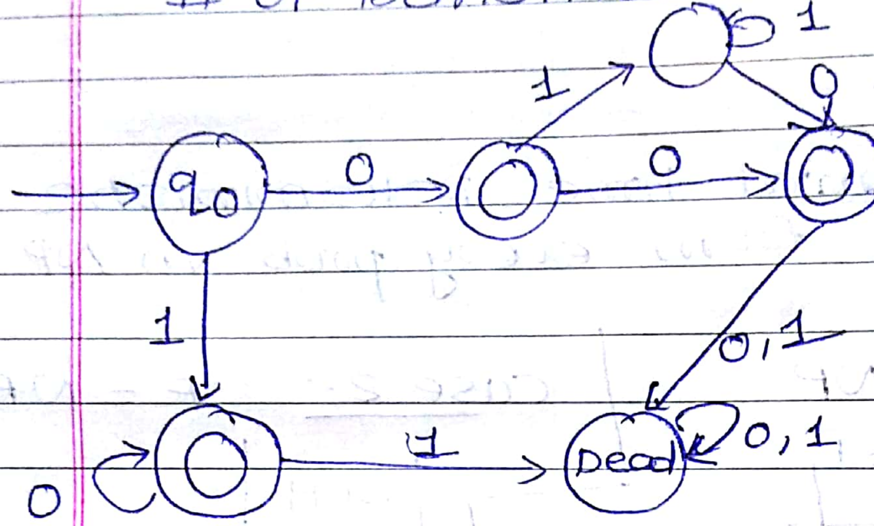
1. RE to DFA construction

$0 + 10^* + 01^*0$

In DFA,

\# of transitions = \# of I/P symbols.



2. NFA - DFA conversion using subset construction method

↳ For complex system, it is easier to construct NFA than DFA.

↳ One can implement NFA in software but it is computationally expensive

↳ complex system

↓

construct NFA

↓

convert into DFA

↓

Implement DFA in s/w which is computationally less expensive (as no parallelism)

Example:-

Transition Table for NFA

δ	a	b
Initial state $\rightarrow q_0$	q_0	q_1
q_1	q_0q_1	q_2
$\rightarrow * q_2$ accepting state	q_1	q_0q_1

NFA: $(Q, \Sigma, \delta, q_0, F)$

$Q: \{q_0, q_1, q_2\}$

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

$q_0: \{q_0\}$

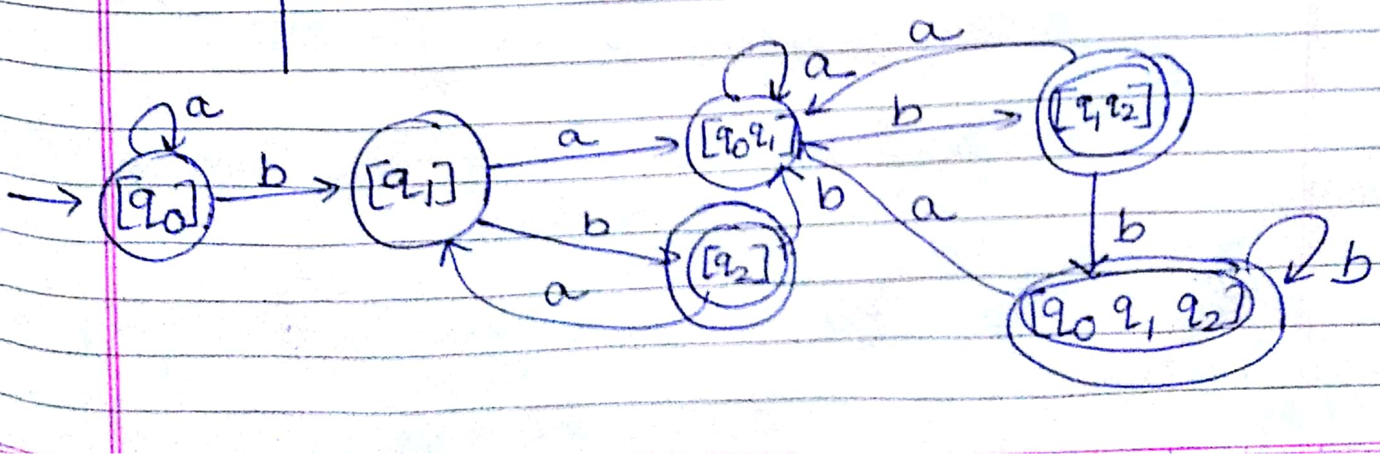
$F: \{q_2\}$

Transition Table for DFA

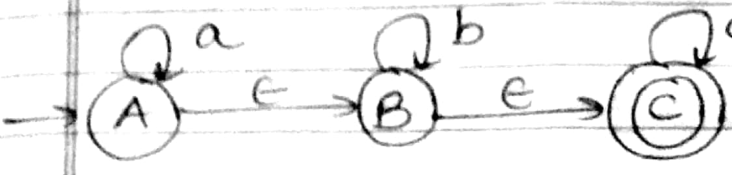
# of states in DFA: 2^Q

$\{q_0\}, \{q_1\}, \{q_2\}, \{q_0q_1\}, \{q_0q_2\}$ $\{q_1, q_2\}, \{q_0, q_1, q_2\}, \phi$

δ	a	b
$\rightarrow [q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_0q_1]$	$[q_2]$
$[q_0q_1]$	$[q_0q_1]$	$[q_1, q_2]$
$* [q_2]$	$[q_1]$	$[q_0q_1]$
$* [q_1, q_2]$	$[q_0q_1]$	$[q_0q_1, q_2]$
$* [q_0q_1, q_2]$	$[q_0q_1]$	$[q_0q_1, q_2]$



3. NFA-Λ to NFA



$\Lambda(A) = \{A, B, C\}$

↳ Null closure of A is a set of states that can be Reachable from A by ~~without reading anything~~ Reading ϵ.

↳ Null closure of any state is that state + states that can be Reachable from A.

$\Lambda(B) = \{B, C\}$

$\Lambda(C) = \{C\}$

Transition Table for NFA

\# of states in NFA = # of states in NFA-Λ

	a	b	c
A	$\{A, B, C\}$	$\{B, C\}$	$\{C\}$
B	ϕ	$\{B, C\}$	$\{C\}$
C	ϕ	ϕ	$\{C\}$

(1) $\delta^*(A, a) = \Lambda\left(\bigcup_{r \in \delta^*(A, \Lambda)} \delta(r, a)\right)$ ← put values

$\delta^*(A, \Lambda) = \Lambda(A) = \{A, B, C\}$

$= \Lambda\left(\delta(A, a) \cup \delta(B, a) \cup \delta(C, a)\right)$

$= \Lambda\left(\{A\} \cup \phi \cup \phi\right)$

$= \Lambda(A) = \{A, B, C\}$

(2) $\delta^*(A, b) = \Lambda\left(\bigcup_{r \in \delta^*(A,\Lambda)} \delta(r, b)\right)$ $\delta^*(A,\Lambda) = \{A,B,C\}$

$= \Lambda\left(\delta(A,b) \cup \delta(B,b) \cup \delta(C,b)\right)$

$= \Lambda\left(\phi \cup \{b\} \cup \phi\right)$

$= \Lambda(\{b\}) = \{b, c\}$

(3) $\delta^*(A, c) = \Lambda\left(\bigcup_{r \in \delta^*(A,\Lambda)} \delta(r, c)\right)$

$= \Lambda\left(\delta(A,c) \cup \delta(B,c) \cup \delta(C,c)\right)$

$= \Lambda\left(\phi \cup \phi \cup \{c\}\right)$

$= \Lambda(\{c\})$

Date: / /

(4) $\delta^*(B,a) = \Lambda\left(\bigcup_{r \in \delta^*(B,\Lambda)} \delta(r,a)\right)$ $\qquad \delta^*(B,\Lambda) = \Lambda(B) = \{B,C\}$

$= \Lambda\left(\delta(B,a) \cup \delta(C,a)\right)$

$= \Lambda(\phi \cup \phi) = \phi$

(5) $\delta^*(B,b) = \Lambda\left(\bigcup_{r \in \delta^*(B,\Lambda)} \delta(r,b)\right)$

$= \Lambda\left(\delta(B,b) \cup \delta(C,b)\right)$

$= \Lambda\left(\{B\} \cup \phi\right)$

$= \Lambda(\{B\}) = \{B,C\}$

(6) $\delta^*(B,c) = \Lambda\left(\bigcup_{r \in \delta^*(B,\Lambda)} \delta(r,c)\right)$

$= \Lambda\left(\delta(B,c) \cup \delta(C,c)\right)$

$= \Lambda\left(\phi \cup \{C\}\right) = \Lambda(\{C\})$

$= \{C\}$

(7) $\delta^*(c,a) = \Lambda\left(\bigcup_{r \in \delta^*(c,\Lambda)} \delta(r,a)\right)$

$\delta^*(c,\Lambda) = \Lambda(c) = \{c\}$

$= \Lambda\left(\delta(c,a)\right) = \phi$

(8) $\delta^*(c,b) = \Lambda\left(\bigcup_{r \in \delta^*(c,\Lambda)} \delta(r,b)\right)$

$= \Lambda\left(\delta(c,b)\right)$

$= \phi$

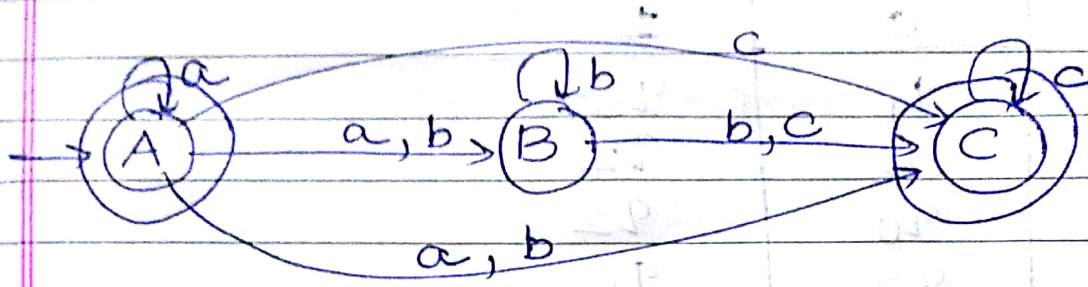
(9) $\delta^*(c,c) = \Lambda\left(\bigcup_{r \in \delta^*(c,\Lambda)} \delta(r,c)\right)$

$= \Lambda\left(\delta(c,c)\right)$

$= \Lambda(c)$

$= \{c\}$

Transition Diagram for NFA



Initial state of NFA = initial state of NFA-Λ.
Accepting state of NFA = Accepting state of NFA-Λ
(But if ~~NFA~~ ϵ is a valid string, then initial state of NFA is also accepting state
Here ϵ is valid string, so A is also an accepting state.

4. Minimization of DFA

		a	b
Initial state	→ q_0	q_1	q_5
	q_1	q_6	q_2
Accepting state	* q_2	q_0	q_2
	q_3	q_2	q_6
	q_4	q_7	q_5
	q_5	q_2	q_6
	q_6	q_6	q_4
	q_7	q_6	q_2

There is no state from which we reach to q_3.
∴ q_3 is not reachable.

Modified Transition Table

	0	1
$\rightarrow q_0$	q_1	q_5
q_1	q_6	q_2
$* q_2$	q_0	q_2
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

Find 0 equivalent state

[q_0 q_1 q_4 q_5 q_6 q_7] [q_2]

1 equivalent state (check previous states and transitions

[q_0, q_4 q_6] [q_1, q_7] [q_5] [q_2]

2 equivalent state

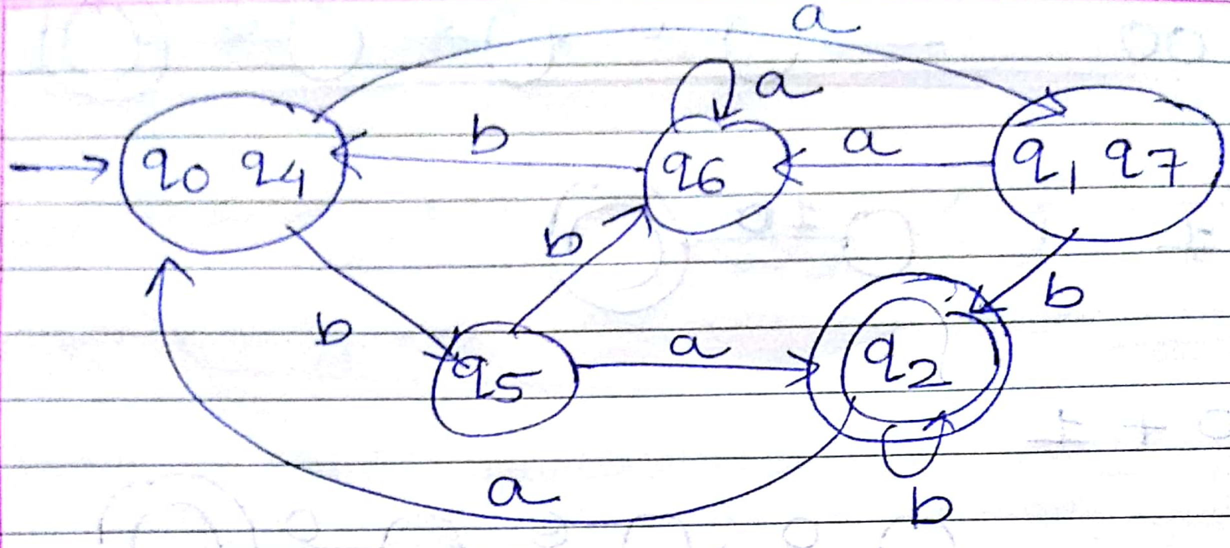
[q_0 q_4] [q_6] [q_1 q_7] [q_5] [q_2]

3 equivalent state.

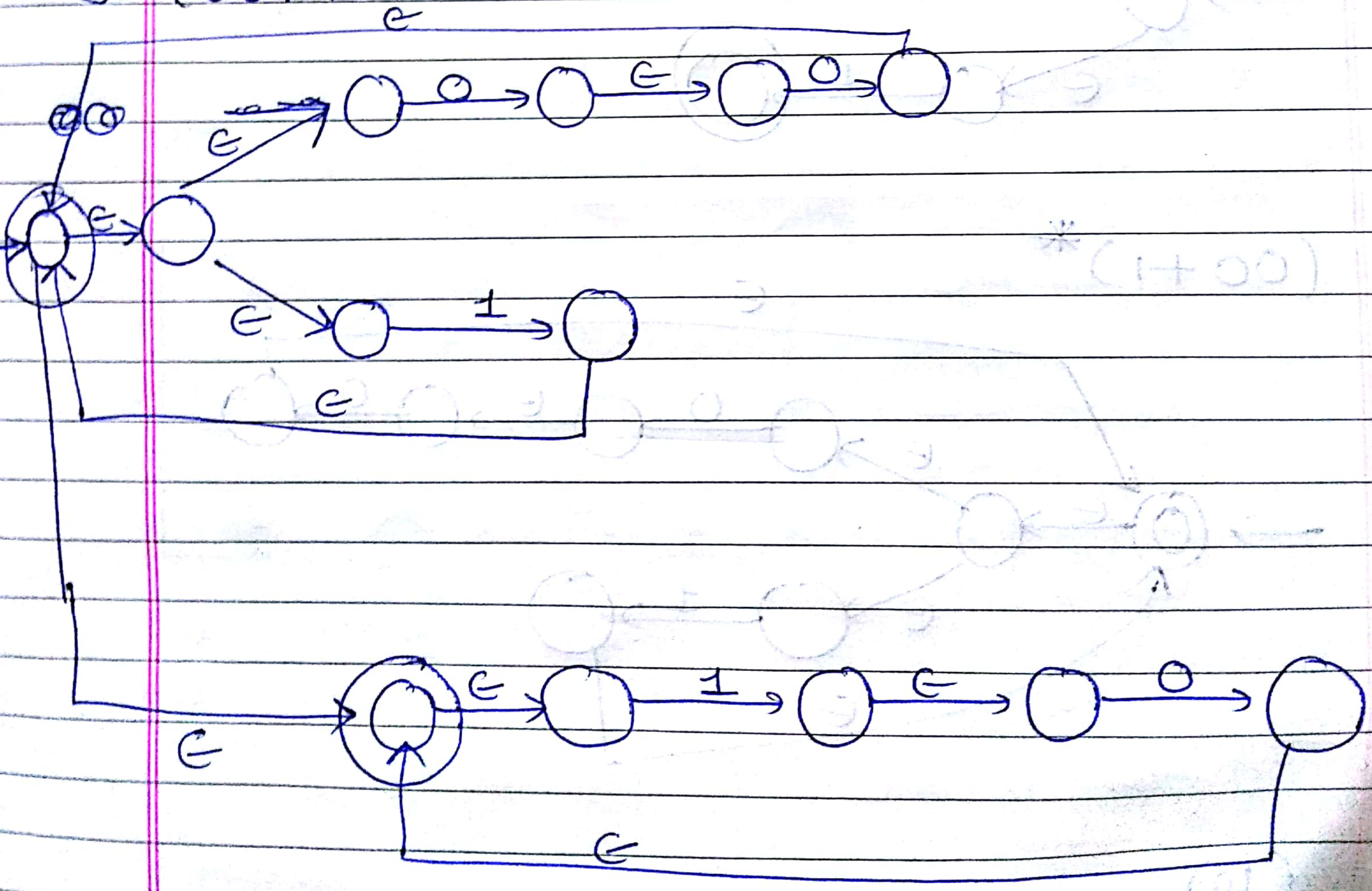
[q_0 q_4] [q_6] [q_1, q_7] [q_5] [q_2]

same states ↓ stop.

8 states are ~~conv~~ minimized to 5 States.

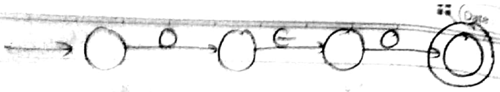


5. $(00+1)^* (10)^*$

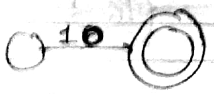


Scanned by CamScanner

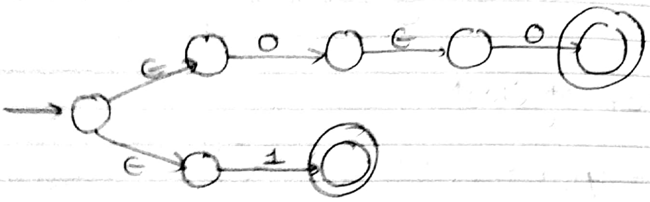
00



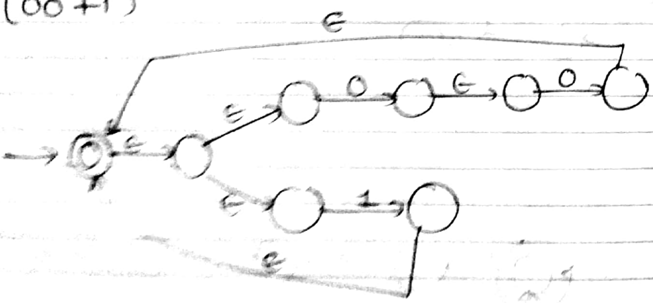
1



00 + 1

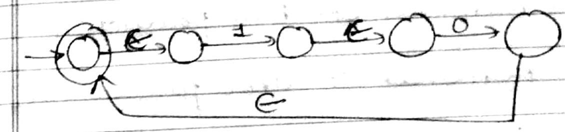


$(00+1)^*$

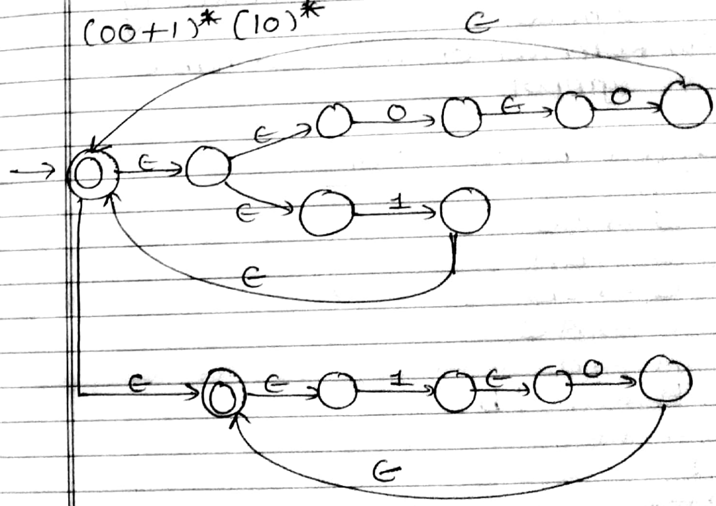


(10)

$(10)^*$



$(00+1)^* (10)^*$



Date: / /

6. Simplification / Reduced form of CFG

Step 1:- Removal of null production

Step 2 :- Removal of unit production

Step 3:- Removal of useless symbol.
(a) Removal of non-generating symbol
(b) Removal of non-reachable symbol

Reduced below CFG

$S \rightarrow AC$
$S \rightarrow BA$
$C \rightarrow CB$
$C \rightarrow AC$
$A \rightarrow a$
$B \rightarrow aC \mid b$

Step 1:-
There is NO null production

Step 2:-
There is NO unit production

Step 3:- Removal of useless symbol
(a) Removal of non-generating symbol
C is non-generating symbol as does not produce any terminal string.

$S \rightarrow BA$
$A \rightarrow a$
$B \rightarrow b$

(b) Find non-Reachable symbol
↳ But here all Non-terminals are reachable from starting non-terminal

Reduced form of CFG

$S \rightarrow BA$
$A \rightarrow a$
$B \rightarrow b$

7. Removal of Null-production

$S \rightarrow aA$
$A \rightarrow b / \epsilon$

$A \rightarrow \epsilon$ is Null production.

Step 1:- If $A \rightarrow \epsilon$ is a production to be eliminated, then look all productions whose right side contains A.

Step 2:- Replace each occurrence of A in each of the production to obtain the non-ϵ production.

Step 3:- These non-null productions must be added to grammar to keep the language generating power the same.

Note:- If ϵ is in the language set generated from G, Null-production cannot be Removed.

$S \rightarrow aA \mid a$
$A \rightarrow b$

2. $S \rightarrow aX \mid bX$
$X \rightarrow a \mid b \mid \epsilon$

$S \rightarrow aX \mid bX \mid a \mid b$
$X \rightarrow a \mid b$

3. $S \rightarrow aS \mid A$
$A \rightarrow aA \mid \epsilon$

↳ $A \rightarrow \epsilon$ is a null production.
↳ But in the language set generated by G, there is a null string which can be generated by following way
$S \rightarrow A \rightarrow \epsilon$
∴ Null string is in the language set so null production cannot be removed.

4. $S \rightarrow ABaC$
$A \rightarrow BC$
$B \rightarrow b \mid \epsilon$
$C \rightarrow D \mid \epsilon$
$D \rightarrow d$

$S \rightarrow ABaC \mid AaC \mid ABa \mid Aa$
$A \rightarrow BC \mid C \mid B \mid \epsilon$
$B \rightarrow b$
$C \rightarrow D$
$D \rightarrow d$

$B \rightarrow \epsilon$ and $C \rightarrow \epsilon$ is eliminated

$S \rightarrow ABaC \mid AaC \mid ABa \mid Aa \mid BaC \mid aC \mid Ba \mid a$

$A \rightarrow BC \mid C \mid B$

$B \rightarrow b$

$C \rightarrow D$

$D \rightarrow d$

8. Removal of Unit production

$S \rightarrow AB$

$A \rightarrow E$ ✓ Unit

$B \rightarrow C$ ✓ Unit

$C \rightarrow D$ ✓ Unit

$D \rightarrow b$

$E \rightarrow a$

$\left.\begin{matrix} \text{Unit} \\ \text{Unit} \\ \text{Unit} \end{matrix}\right\}$ Remove it

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow b$

$C \rightarrow b$

$D \rightarrow b$

$E \rightarrow a$

9. CFG → CNF

CFG is said to be in CNF if

N.T → string of exactly two N.T.

N.T → single terminal

Following are the steps for converting CFG → CNF.

Step 1:-

1. Elimination of ∈ production
2. Elimination of Unit production
3. Elimination of Useless Symbol

Step 2:-

convert it into CNF.

- $S \rightarrow bA \mid aB$
  $A \rightarrow bAA \mid aS \mid a$
  $B \rightarrow aBB \mid bS \mid a$

Step 1:-

NO ∈ production

NO unit production

All symbols are generating and Reachable.

Step 2:-

only $\left.\begin{array}{l} A \rightarrow a \\ B \rightarrow a \end{array}\right\}$ are in CNF.

Scanned by CamScanner

Replace terminal 'a' by C_a and 'b' by C_b

$C_a \rightarrow a$
$C_b \rightarrow b$

$S \rightarrow C_b A \mid C_a B$
$A \rightarrow C_b AA \mid C_a S \mid a$
$B \rightarrow C_a BB \mid C_b S \mid a$

convert it into CNF.

$S \rightarrow C_b A \mid C_a B$
$A \rightarrow C_b D \mid C_a S \mid a$
$D \rightarrow AA$
$B \rightarrow C_a E \mid C_b S \mid a$
$E \rightarrow BB$
$C_a \rightarrow a$
$C_b \rightarrow b$

- CFG → CNF

$E \rightarrow E + E$
$E \rightarrow E * E$
$E \rightarrow id$ ⟶ only in CNF.

Step 1:- No null and unit production and there is not any useless symbol.

Step 2:- convert it into CNF.

$\Sigma = \{+, *, id\}$

$E \rightarrow id$ is in CNF. Introduce N.T. for + and *.

$C \rightarrow +$, $D \rightarrow *$

New productions:

$E \rightarrow ECE$
$E \rightarrow EDE$
$E \rightarrow id$
$C \rightarrow +$
$D \rightarrow *$

Make above productions in CNF.

$E \rightarrow EX$
$X \rightarrow CE$
$E \rightarrow EY$
$Y \rightarrow DE$
$E \rightarrow id$
$C \rightarrow +$
$D \rightarrow *$