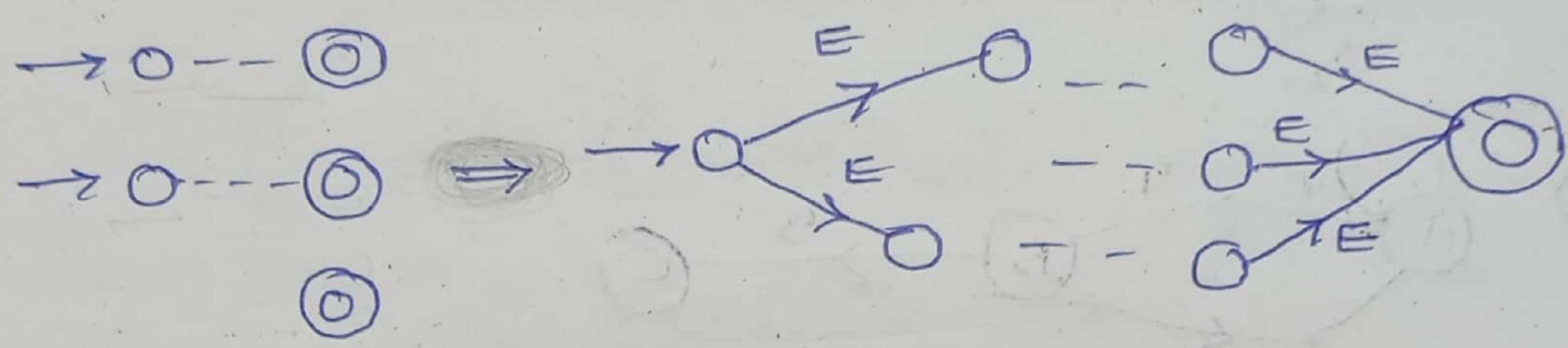
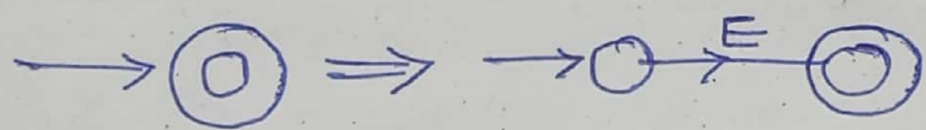
⇒ F.A → R.E

2) State-Elimination Method :-

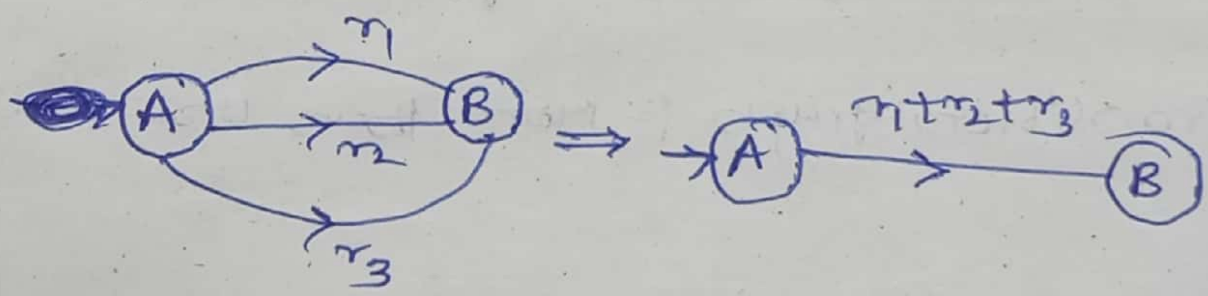
Step 1 :- Simplify in such a way that the transition graph contains only one initial state and one final state.



Step 2 :- Simplify the transition graph in such a way it has different initial and Final states



Step 3 :-



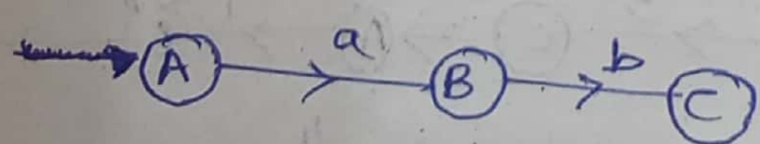
Parallel edges (going in same direction)

Step 4 :-

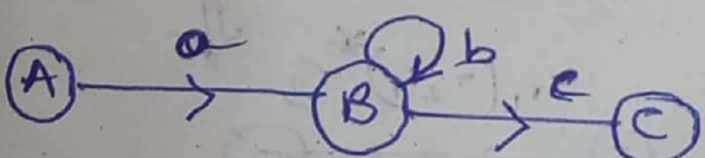
State Elimination :-

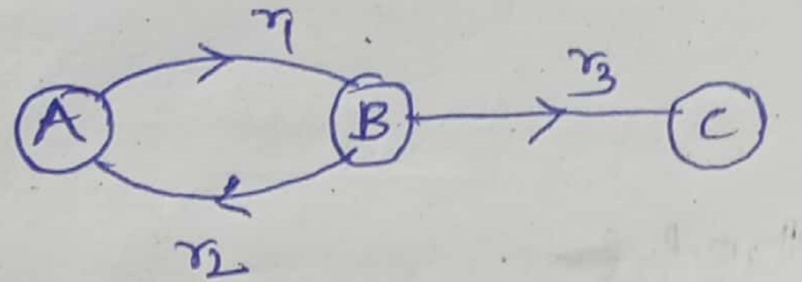
Serial edges

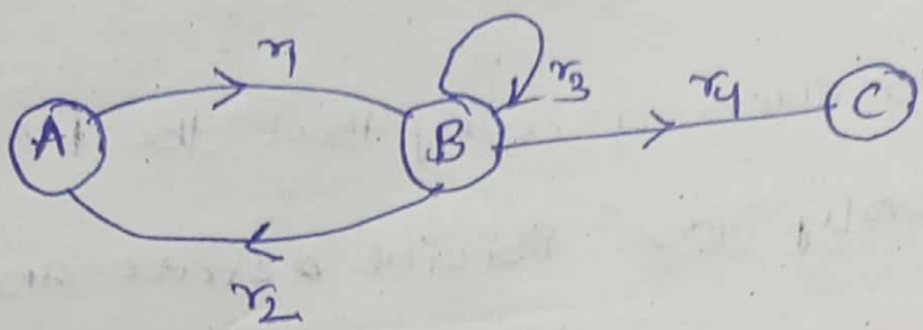
a)

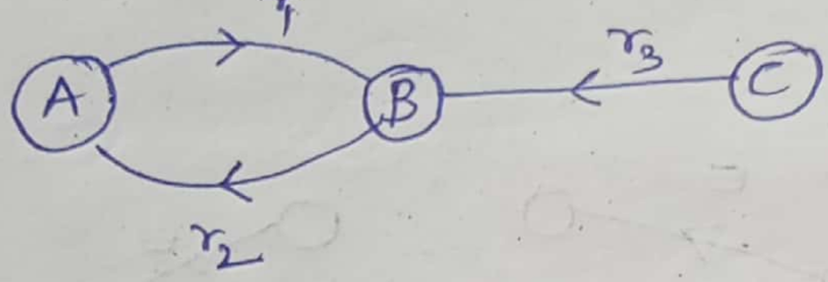


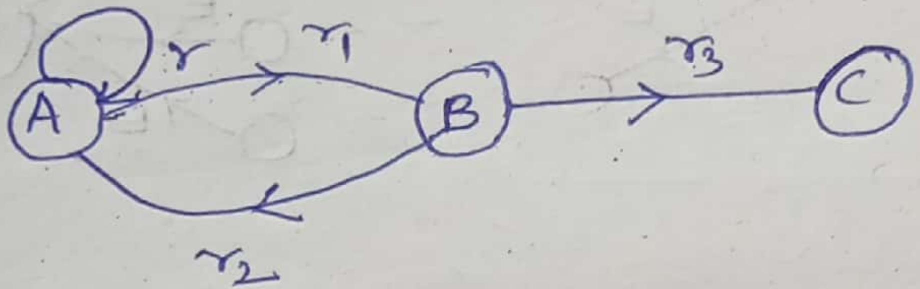
b)

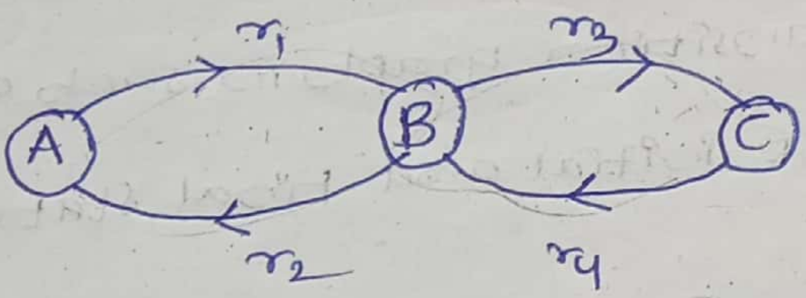


c) 

d) 

e) 

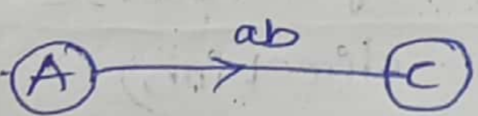
f) 

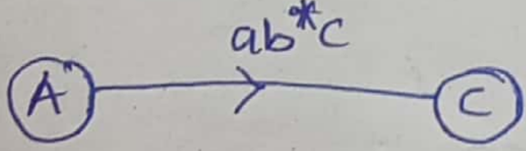
g) 

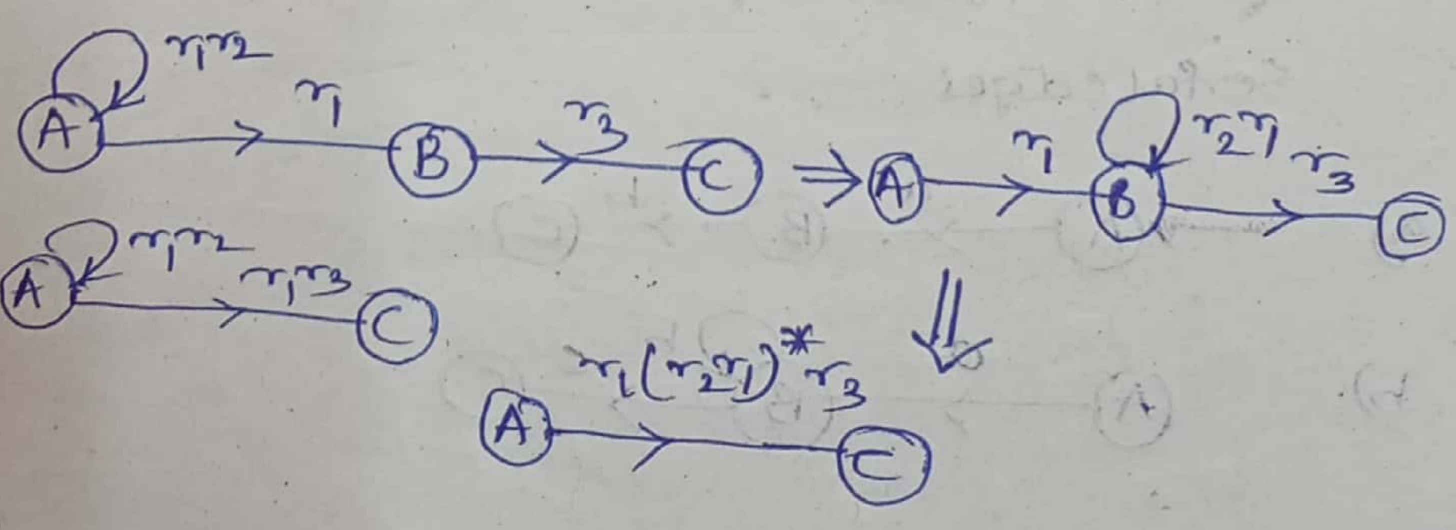
Transition state := only one state

Transition graph := More than one state.

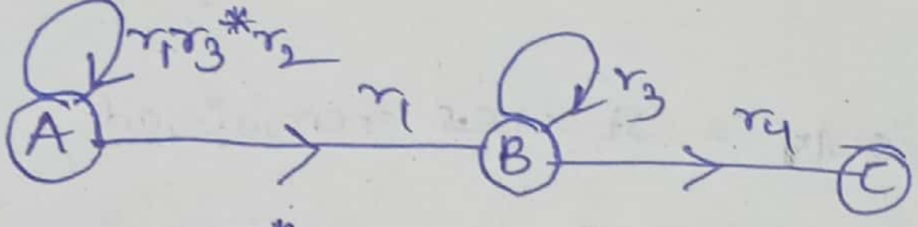
Answers :=

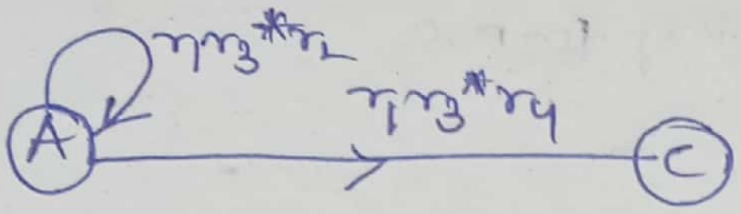
a) 

b) 

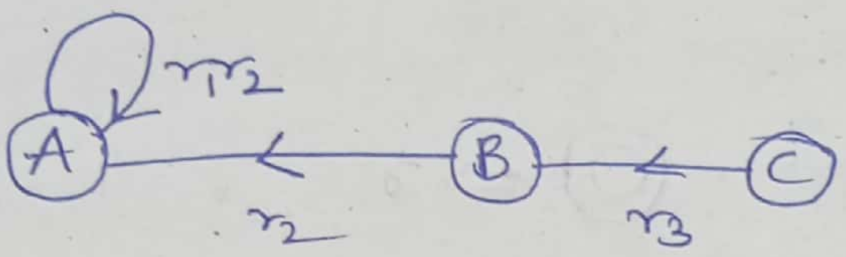
c) 

d)

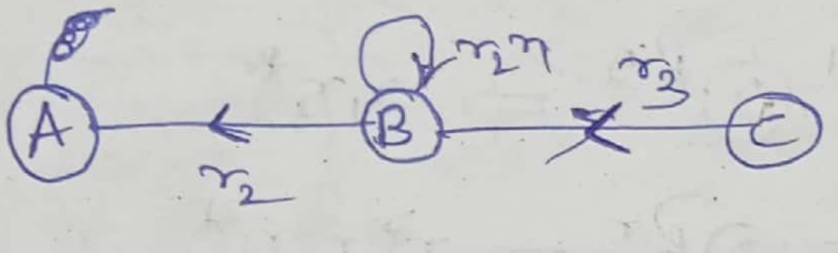

A
B
C
r1r3*r2
r1
r3
r4


A
C
r1r3*r2
r1r3*r4

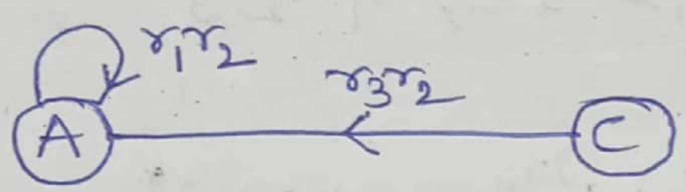
e)


A
B
C
r1r2
r2
r3

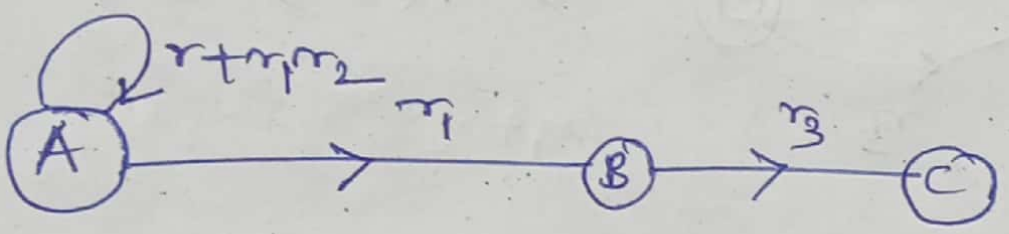
(or)


A
B
C
r2r1
r3
r2

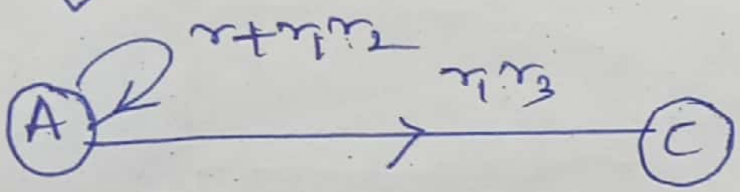
⟹


A
C
r1r2
r3r2

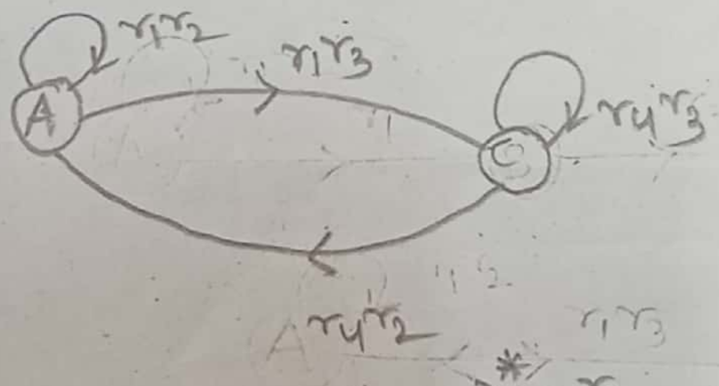
f)

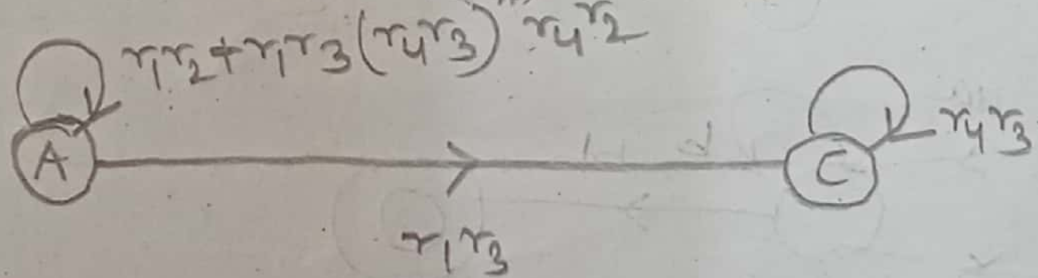

A
B
C
r+r1r2
r1
r3

⇓


A
C
r+r1r2
r1r3

g)

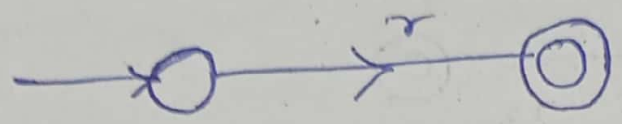

A
C
r1r2
r1r3
r4r3
r4r2


A
C
r1r2+r1r3(r4r3)*r4r2
r4r3
r1r3

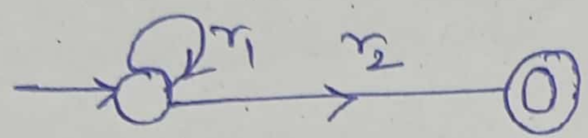
~~Step 5~~ :-

2) repeat step 4 only 2 states remained in one of the following forms.

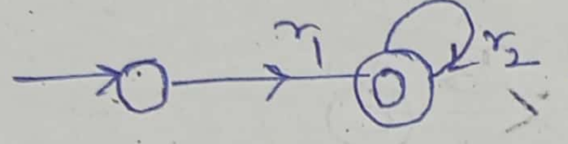
Step 5 :-

a) 

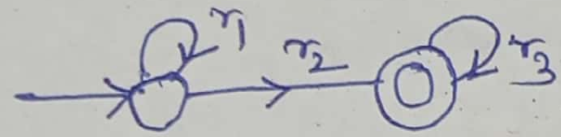
$\Rightarrow r$

b) 

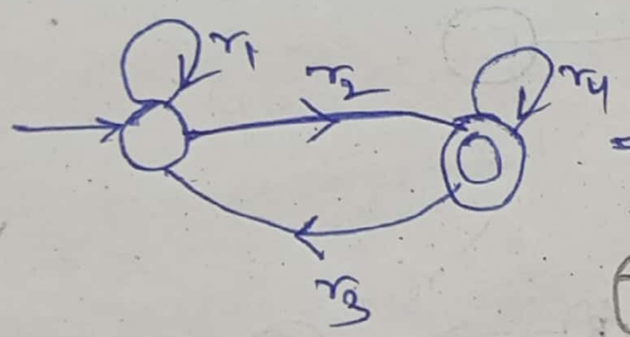
$\Rightarrow r_1^* r_2$

c) 

$\Rightarrow r_1 r_2^*$

d) 

$\Rightarrow r_1^* r_2 r_3^*$

e) 

$\Rightarrow \cancel{r_1^*(r_2 r_3)^* r_2 r_4^*}$

$(r_1 + r_2 r_4^* r_3)^* r_2 r_4^*$

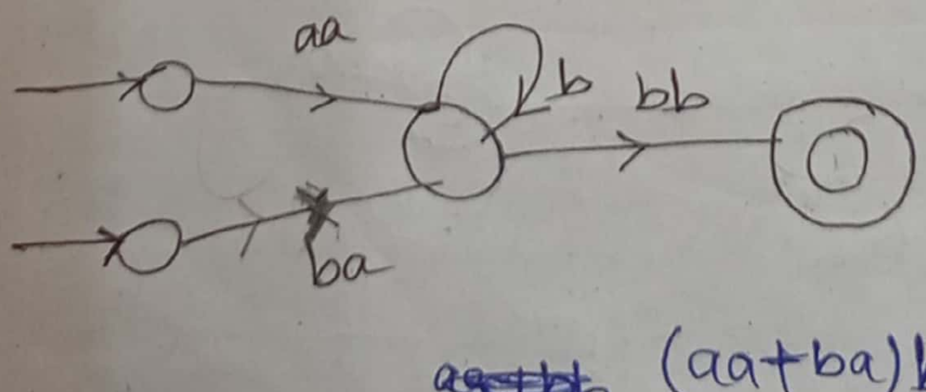
* Precedence order :-

⇒ Kleene closure ↑

⇒ concatenation

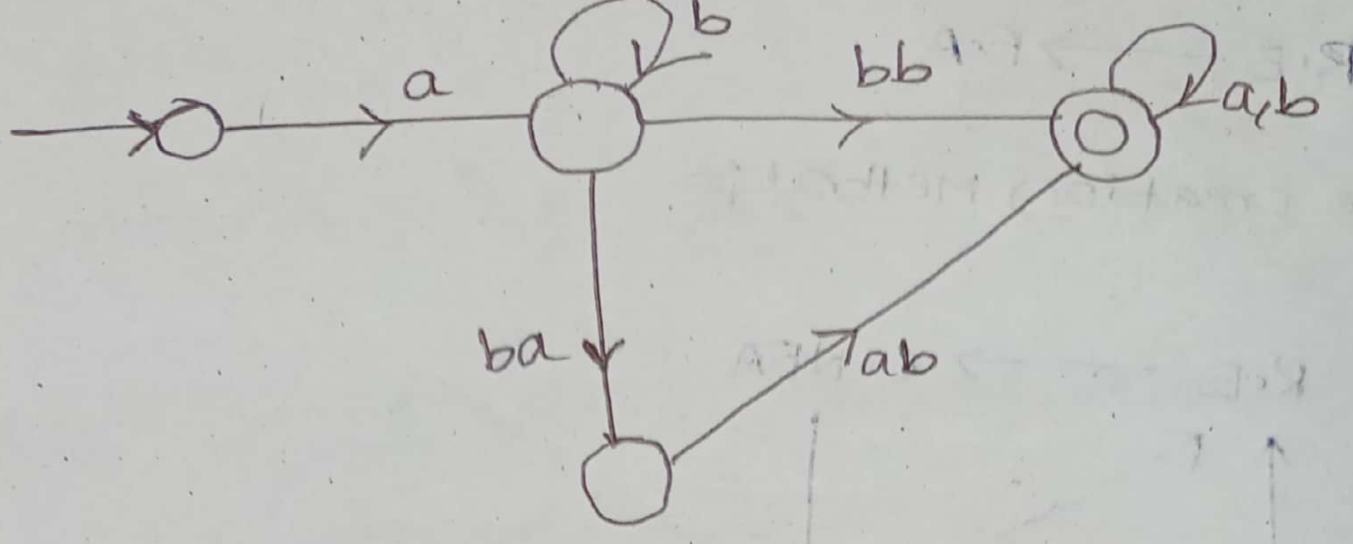
⇒ union ↓

H/W

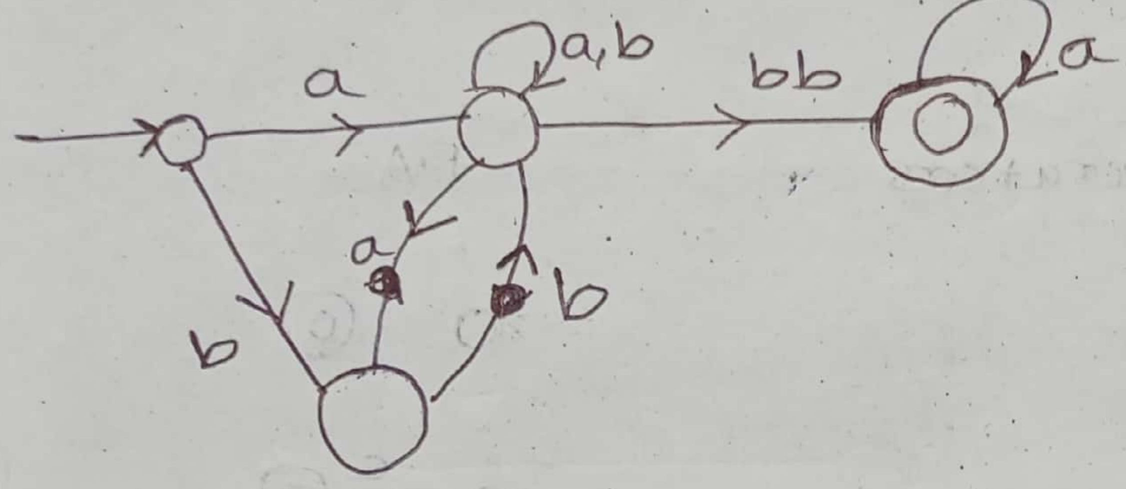
1) 

~~aa+bb~~ $(aa+ba)b^*bb$

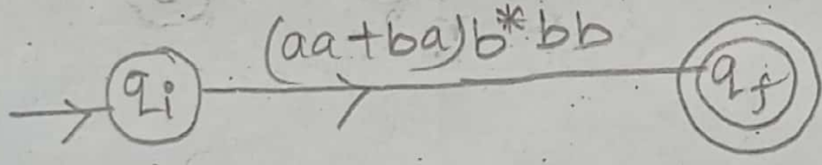
2)



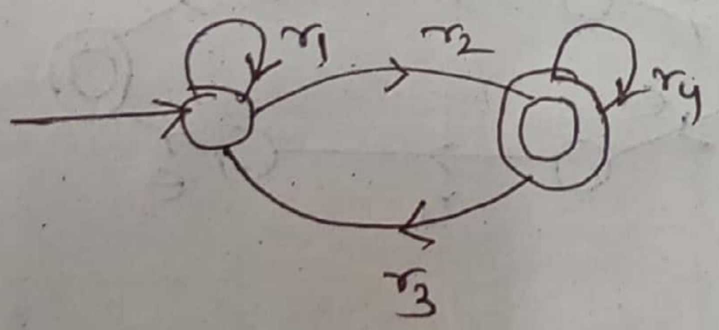
3)



$\Rightarrow$ 1)



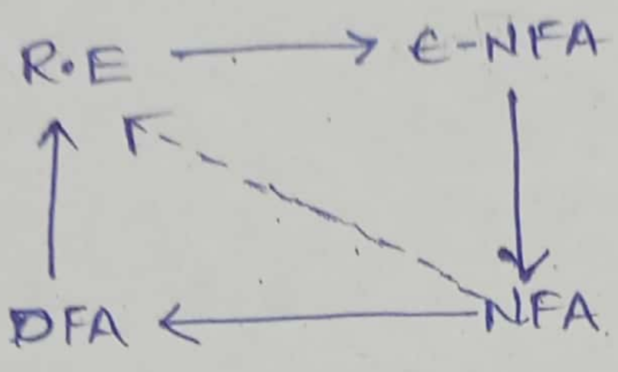
4)



$$\Rightarrow \left(r_1 + r_2 r_4^* r_3\right)^* r_2 r_4^*$$

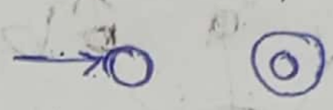
⇒ R.E ⟶ F.A

State creation Method :=

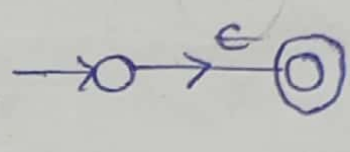


R.E without operators F.A.

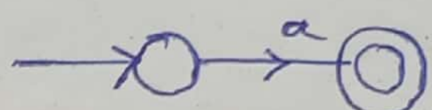
$r = \phi$



$r = \epsilon$

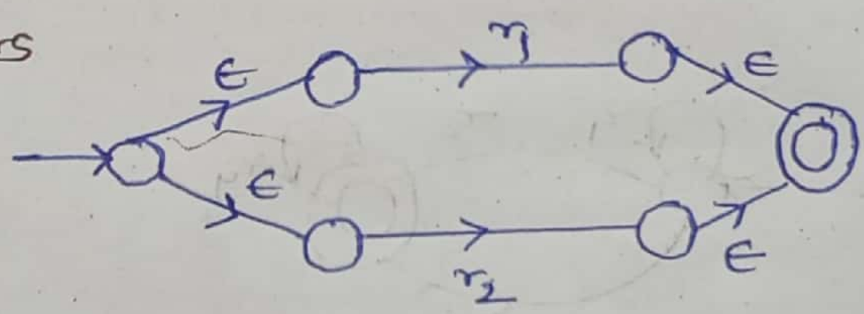


$r = a$

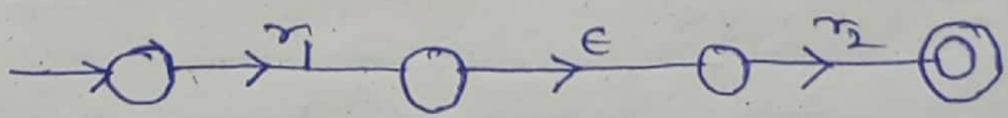


RE with operators

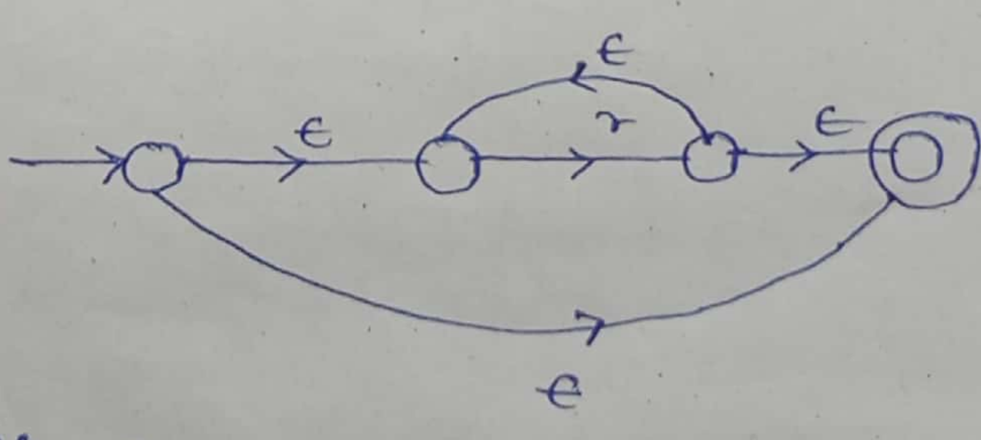
1) $r_1 + r_2$



2) $r_1 r_2$

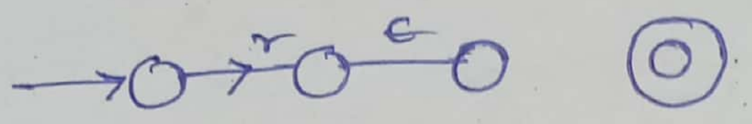


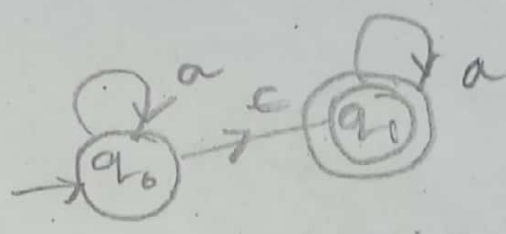
3) r^*



* NFA which has ε Transition is "ε-NFA".

$\Rightarrow r\phi = \phi$

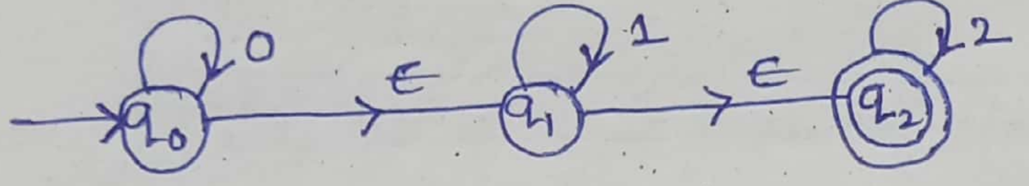




$\Rightarrow \phi$ is Annihilator over concatenation.

* **ε-NFA** :- NFA which has ε Transitions.

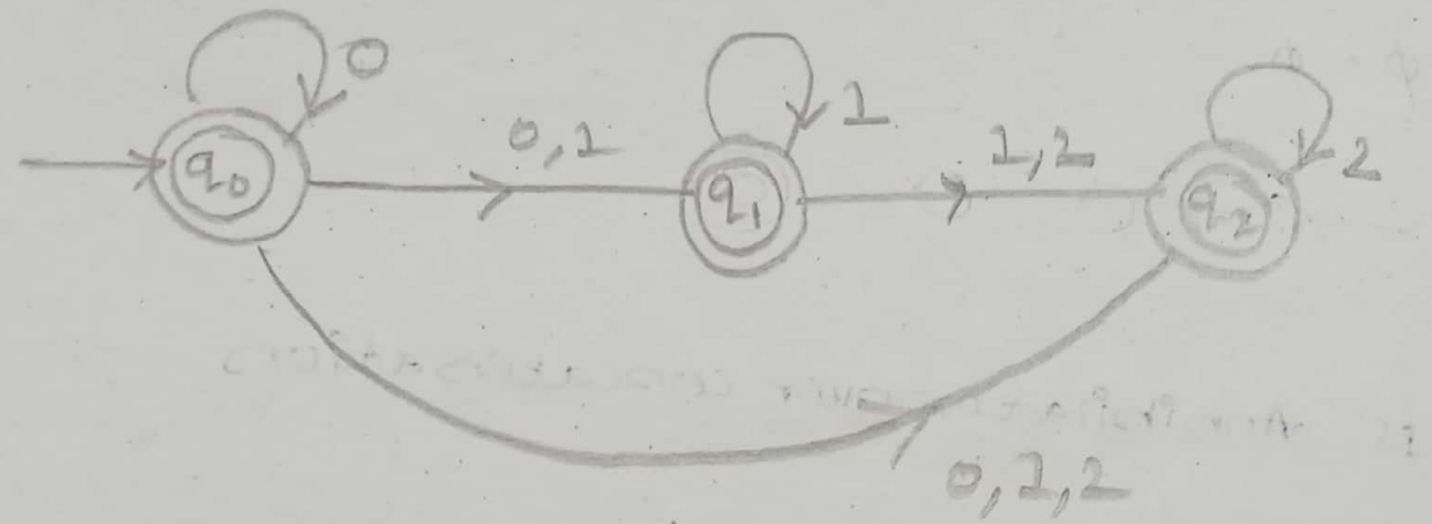
$\Rightarrow$ ε-NFA to NFA conversion :-

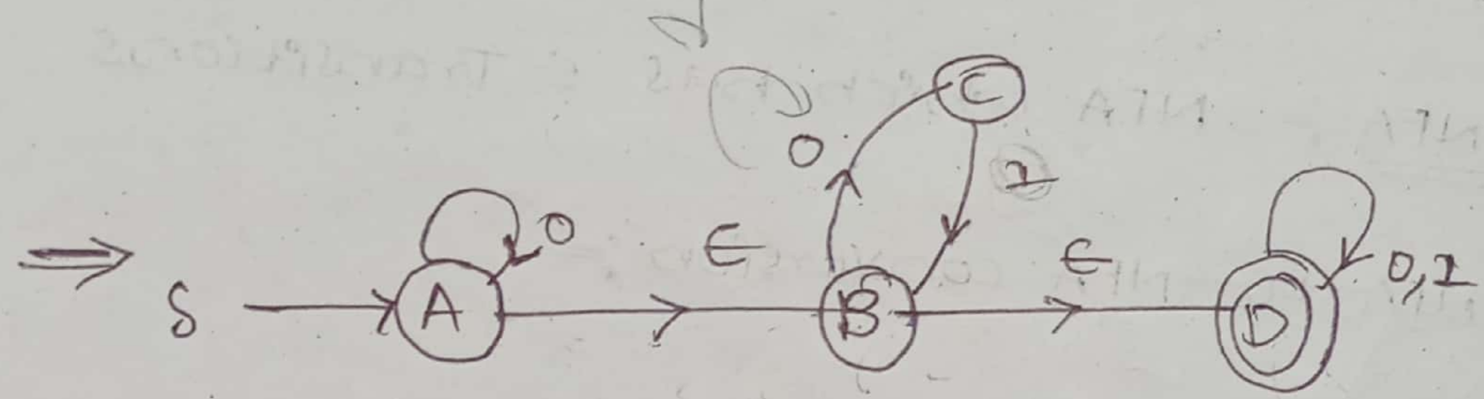


$\Rightarrow$ ε is not part of the i/p symbol

	0	1	2
$\rightarrow$ * q_0	$q_0q_1q_2$	q_1q_2	q_2
* q_1	—	q_1q_2	q_2
* q_2	—	—	q_2

$\Rightarrow$ The states from which on reading epsilon (ε) Transitions ~~are there to final state~~ if we reach the final state then that state which read the ε is considered as final state.





δ'	0	1
→ * A	~~ABCD~~ ABD	D
* B	C,D	D
C	–	BD
* D	D	D

* ϵ-closure(q) :-

Set of all reachable states ~~to which q has~~ ~~ϵ-Transition~~ from q on reading ϵ

$\Rightarrow$ $\epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$

$\epsilon\text{-closure}(q_1) = \{q_1 q_2\}$

$\Rightarrow$ $\boxed{\cancel{\delta'(q_0, 0) = \epsilon\text{-closure}(\delta(q_0, 0))}}$ ~~***~~

$\Rightarrow$ $\boxed{\delta'(q_0, 0) = \epsilon\text{-closure}(\delta(\cancel{\text{closure}}\ \epsilon\text{-closure}(q_0), 0))}$

$\boxed{\delta'(q_0, 0) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_0), 0))}$

$\delta'(A, 0) = \epsilon\text{-closure}(A)$

$= \{A, B, D\}$

$\delta(A, B, D), 0$

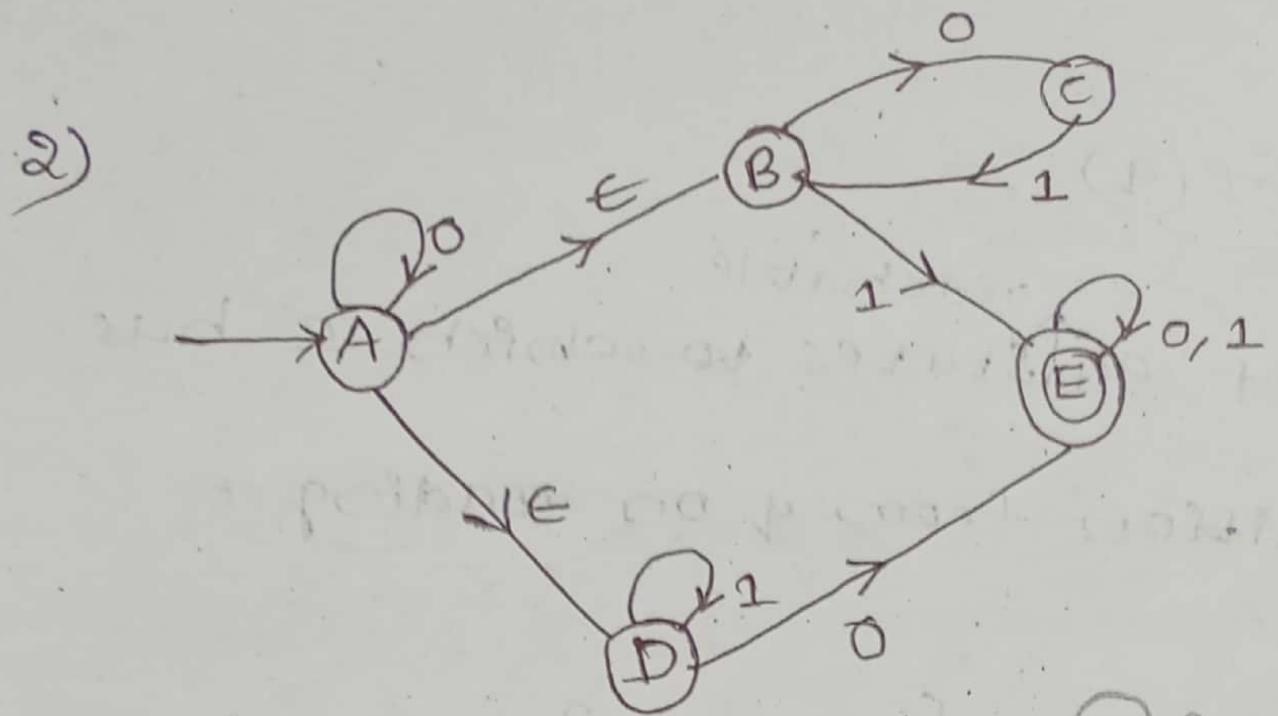
$\Rightarrow \delta(A, 0) \cup \delta(B, 0) \cup \delta(D, 0)$

$= \{A, C, D\}$

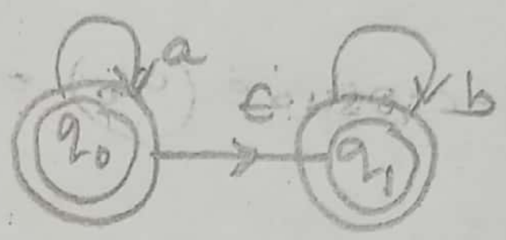
$\epsilon\text{-closure}(A, C, D)$

$= \{A, B, D\} \cup \{C\} \cup \{D\} = \{A, B, C, D\}$

2)



H|w



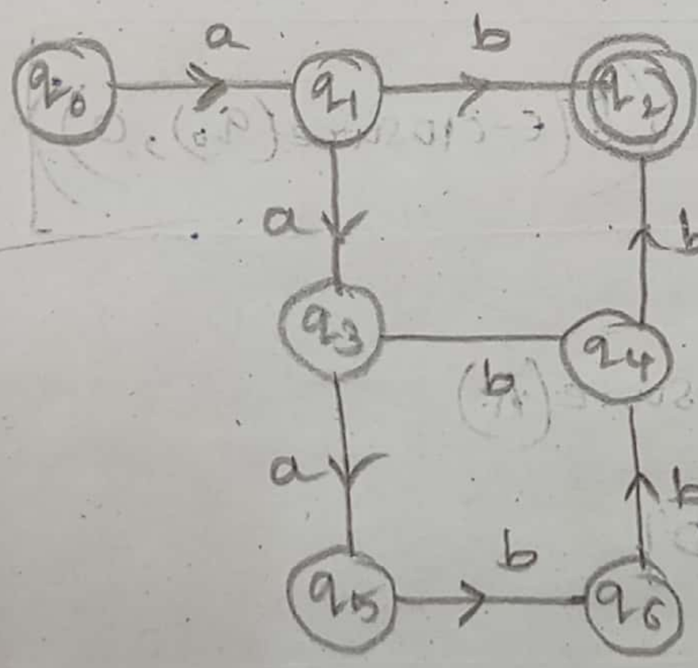
$L_1 = a^m b^n \mid m, n \geq 0\}$

R.L

$L_2 = a^m b^m \mid 1 \leq m \leq 3\}$

$L_3 = \{a^m b^m \mid 1 \leq m \leq 100\}$

$L_2 = \{a^m b^m \mid 1 \leq m \leq 3\}$



$L_4 = \{a^m b^m \mid m \geq 1\}$ Non Regular Language
Because, we cannot
Express with finite
number of
States

$\frac{99}{198}$

$\frac{198 + 3}{201}$

Regular Languages :-

A Language for which a regular expression is written | a Finite Automata is constructed.

⇒ closure Properties of Regular Languages.

- $L_5 = \{a^m b^n \mid m \geq n\}$

Along with the pattern when memorizing concept comes then Finite Automata is not sufficient.

$\boxed{m, n \geq 1}$

Here, in the above, pattern is considered and we need to memorize the m. So, it is non-Regular Language

If the Language itself is Finite ~~then~~ even ~~if the strings~~ whatever may be the condition, even if there is memorizing concept also, it is a regular Language (upper Limits ~~FA~~ exists)

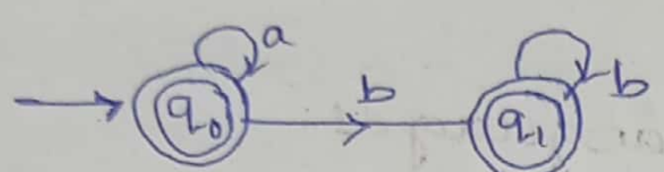
⇒ All infinite ~~strings are~~ Languages may be Regular | may not be Regular.

⇒ All Finite Languages are always Regular.

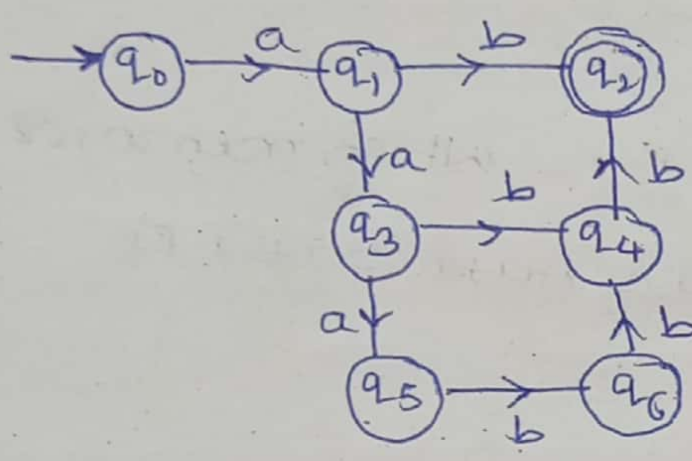
Regular Language :=

A Language for which finite automata can be constructed.

$L_1 = \{a^m b^n \mid m, n \geq 0\}$



$L_2 = \{a^m b^m \mid 1 \leq m \leq 3\}$



$L_3 = \{a^m b^m \mid 1 \leq m \leq 100\}$

Regular Language

(100 is countable)

$L_4 = \{a^m b^m \mid m \geq 1\}$

Not regular

Not upper Limit for m

$L_5 = \{a^m b^n \mid m \geq n\}$

Not regular Language

$L_6 = \{a^p \mid p \text{ is Prime but } p < 100\}$

Not all infinite set are not regular Languages (all finite are regular Languages)

Ex:= $L = \{a^m b^n \mid m, n \geq 0\}$

Minimization of DFA :-

	0	1
→ A	F	B
B	C	G
* C	C	A
D	G	C
E	F	H
F	G	C
G	F	G
H	C	G

1) zero Equivalent state

S_0: {A, B, C, D, E, F, G, H}

differentiate between final and non-final

Level 1 : {C} {A, B, D, E, F, G, H}

on '0'

→ on 1 they are Equivalent

Level 2 : {C} {B, H} {A, D, E, F, G}

If on all the i/p's the behaviour is same then they are not seperated.

on '1'

Level 3 : {C} {B, H} {D, F} {A, E, G}

↳ not differentiable.

{A, E, G} on '0'

Level 4 : {C} {B, H} {D, F} {A, E} {G}

A and E on 1 going to B, H respectively But B, H are Equivalent. So A, E are not differentiated.

Initial state : {A, E}

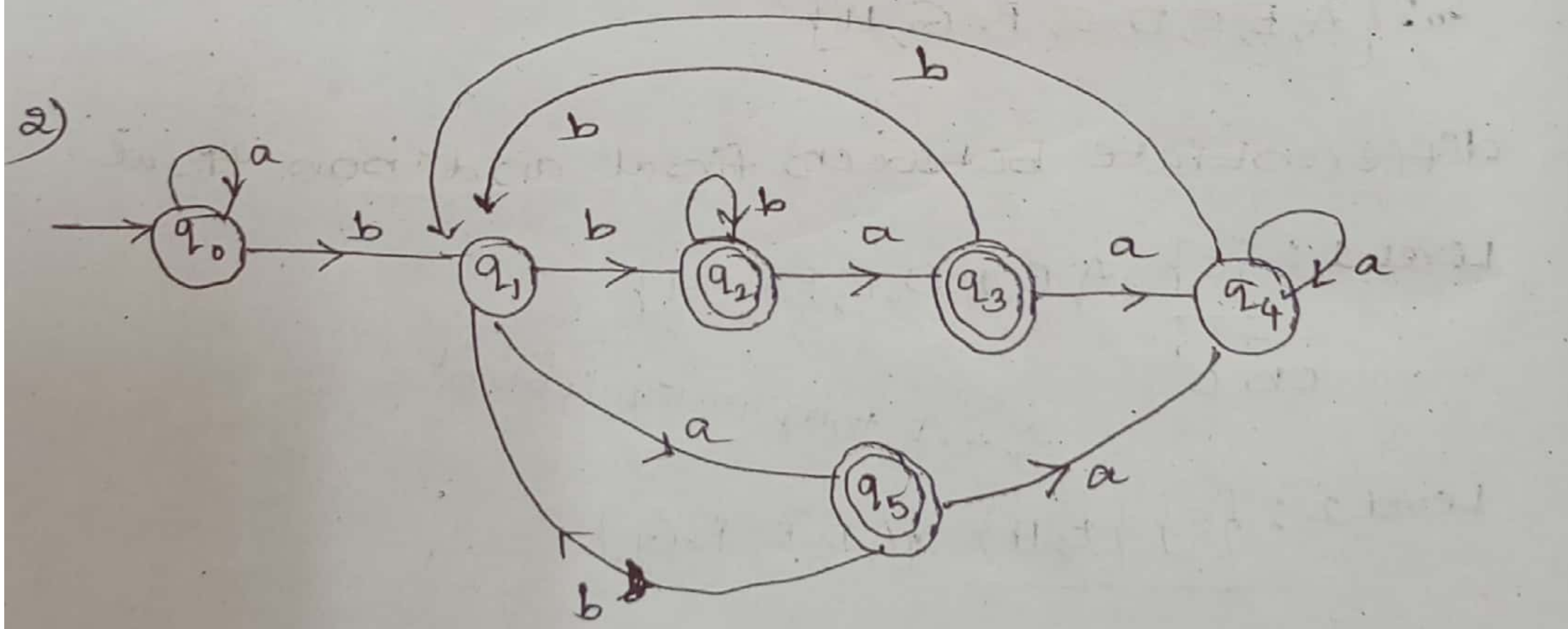
Final state : {C}

2) Transition Table :-

	0	1
$\rightarrow$ $S_0 \rightarrow$ [A, E]	[D, F]	[B, H]
S_1 [B, H]	[C]	[G]
S_2 * [C]	[C]	[A, E]
S_3 [D, F]	[G]	[C]
S_4 [G]	[A, E]	[G]

Automata

2)



1 : $\{q_2, q_3, q_5\}$ $\{q_0, q_1, q_4\}$

on 'a'

2 : $\{q_2\}$ $\{q_3, q_5\}$ $\{q_5\}$ $\{q_0, q_4\}$

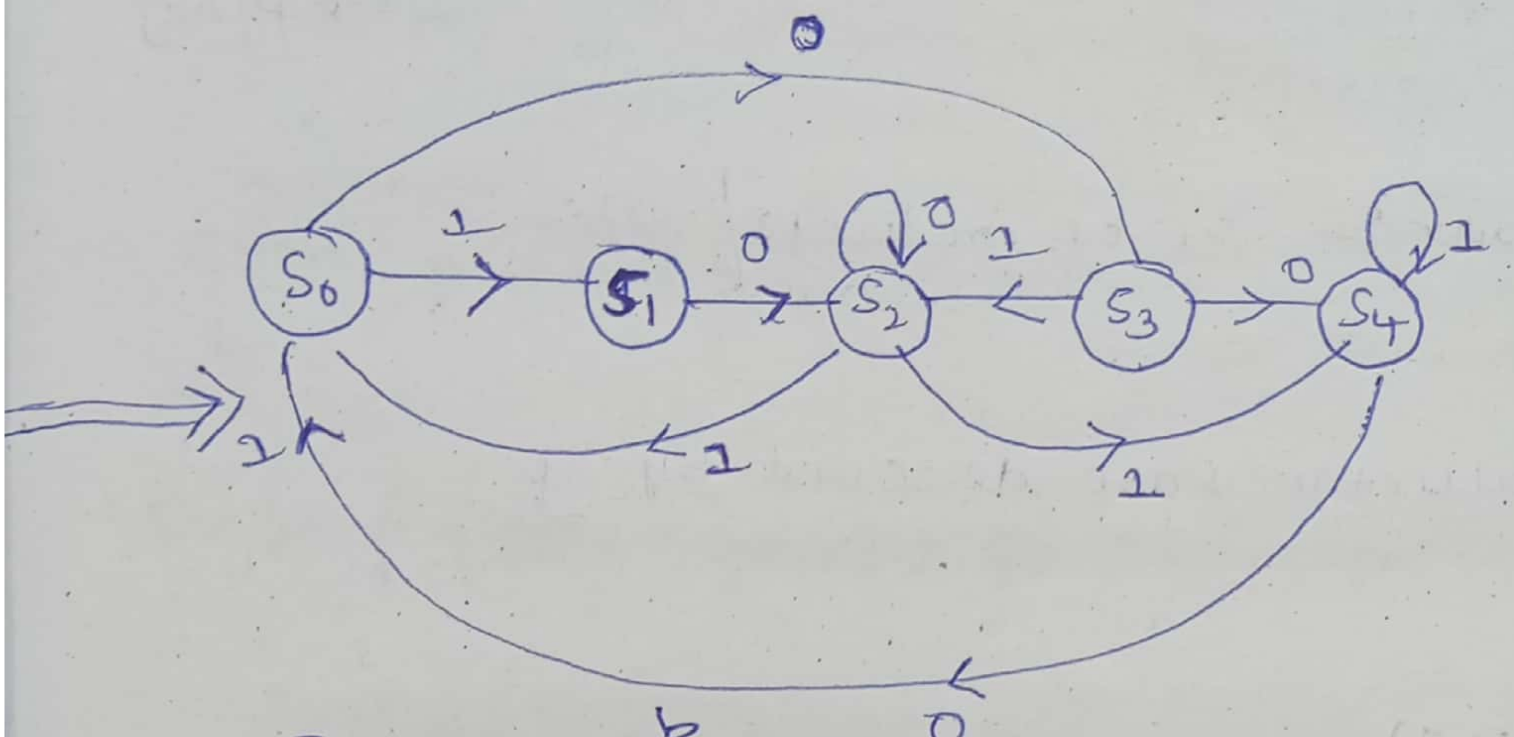
on 'b'

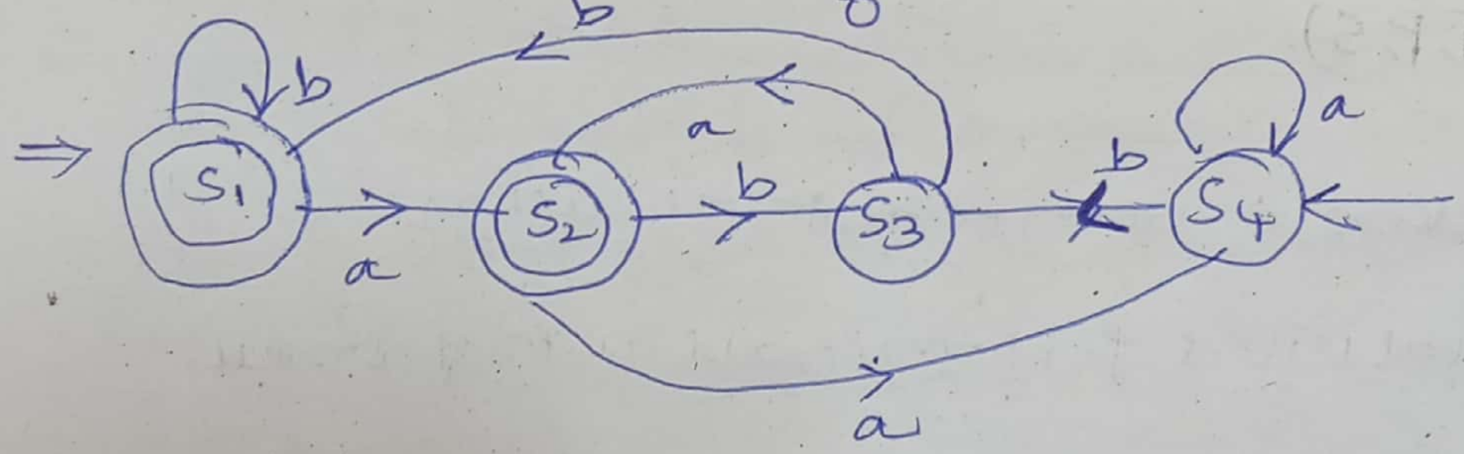
$\rightarrow S_1$ $\rightarrow S_2$ $\rightarrow S_3$ $\rightarrow S_4$

3 : $\{q_2\}$ $\{q_3, q_5\}$ $\{q_1\}$ $\{q_0 q_4\}$

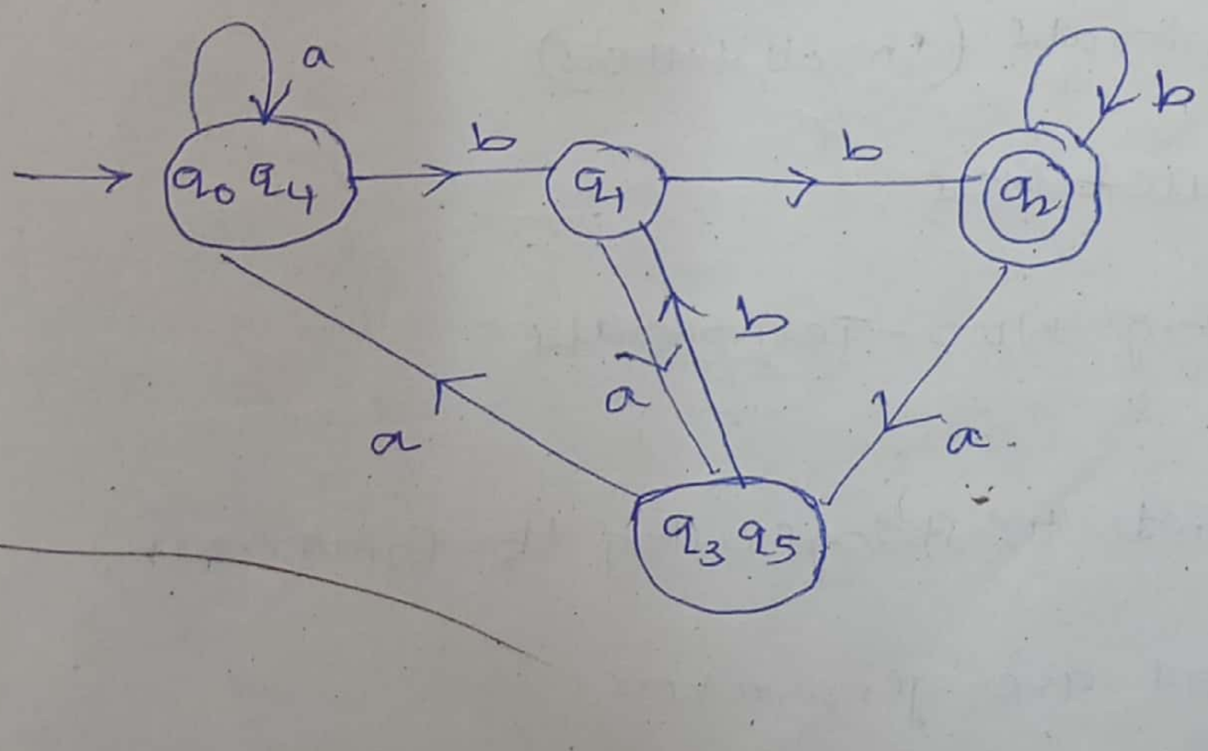
	a	b
$\rightarrow q_0$	q_0	q_1
q_1	q_5	q_2
$* q_2$	q_3	q_2
$* q_3$	q_4	q_1
q_4	q_4	q_1
$* q_5$	q_4	q_1

	a	b
S_1 [q_2]	[q_3, q_5]	[q_2]
S_2 [q_3, q_5]	[q_0, q_4]	[q_1]
S_3 [q_1]	[q_3, q_5]	[q_2]
S_4 [q_0, q_4]	[q_0, q_4]	[q_1]





⇒



* Context Free Grammar (CFG)

2) PDA → FA + stack memory

⇕

1) CFG

⇕

3) CFL

S → Abc

A → n | p

C → p | adj

⇒ Grammar := set of productions / rules

⇒ Productions are denoted by "p"

⇒ (N, T, P, S)

⇒ we denote non terminals using capital letters & terminals using small letters

N : non-Terminals (capital letters)

T : Terminals (small letters)

P : productions

S : Starting Non-Terminals

⇒ we generate the strings using the Grammar.

⇒ Grammars are generators

⇒ Automata's are recognizers

$\Rightarrow \alpha \rightarrow \beta$

where, α is non-Terminal (single n.T)

β is $(T+NT)^*$ (Terminal & non-Terminal & including ϵ and also Combination) of T+NT

$\Rightarrow$ Every ~~product~~ non-Terminals have production.

Ex:- $(\{S, A, B\}, \{a, b\}, P, S)$

$P: S \rightarrow AB$

$A \rightarrow aa \mid bb \Rightarrow A \rightarrow aa$

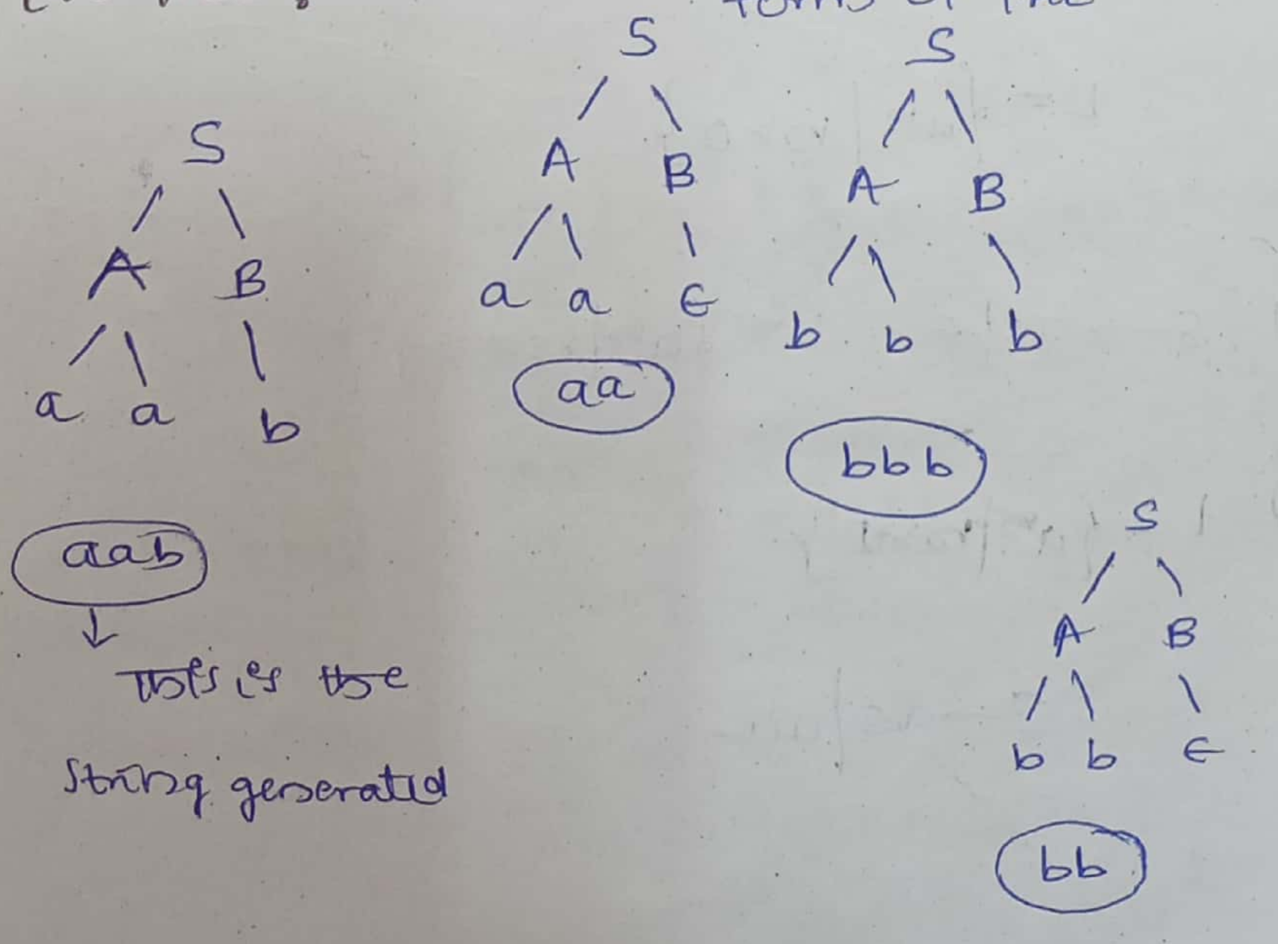
$A \rightarrow bb$

$B \rightarrow b \mid \epsilon$

$\Rightarrow B \rightarrow b$

$B \rightarrow \epsilon$

$\Rightarrow$ Parse Tree :- Derivation represented in the Form of Tree



Derivation :=

$S \to AB$

aaB

aab

↳ all are Terminals

⇒ LMD = {Left most Derivation}

always the left most non-Terminal substituted First.

⇒ RMD = {Right most Derivation}

always the Right most non-Terminal substituted First

Eg:= 1) $S \to aS \mid a$

$S \to a$ $\quad S \to aS$ $\quad S \to aS$

$\quad aa \quad aaS$

$\quad aaS$

$L = \{a^n \mid n > 0\}$

2) $S \to aS \mid \epsilon$ $\quad L = \{a^n \mid n \geq 0\}$

3) $L = \{a^n \mid n > 1\}$

$S \to aS \mid aa$

4) $L = \{a^{2n} \mid n \geq 0\}$

$S \rightarrow aaS \mid \epsilon$

⇒ Every Regular Language is context Free Language.

5) $L_5 = \{a^m b^n \mid m, n \geq 0\}$

6) $L_6 = \{a^m b^n \mid m, n \geq 1\}$

7) $L_7 = \{a^m b^n \mid m = n, m, n \geq 0\}$

8) $L_8 = \{a^m b^n \mid m > n, m, n > 0\}$

9) $L_9 = \{a^m b^n \mid m < n, m, n > 0\}$

10) $L_{10} = \{w \mid w \in \{a,b\}^*$, w has equal no. of a's and b's$\}$

11) $L_{11} = \{wcw^R \mid w \in \{a,b\}^*$, c is constant character, w^R is reverse of w$\}$

5) $S \rightarrow aS \mid B$

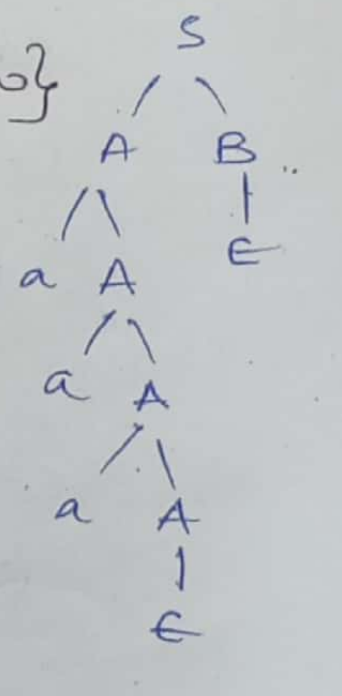
$B \rightarrow bB \mid \epsilon$

(or)

$S \rightarrow AB$

$A \rightarrow aA \mid \epsilon$

$B \rightarrow bB \mid \epsilon$



3) $S \rightarrow aS \mid bS \mid \epsilon$ $(a+b)^*$ (Any string over a, b). Σ^*

4) $S \rightarrow$ ~~as~~ $abS \mid \epsilon$ $(ab)^n$

$\Rightarrow$ $S \rightarrow aS \mid b$ (a^*b) (Any number of a's ending with b.

L_6)

$S \rightarrow AB$
$A \rightarrow aA \mid a$ (or) $S \rightarrow aS \mid aB$
$B \rightarrow bB \mid b$ $B \rightarrow bB \mid b$

L_7) $S \rightarrow aSb \mid \epsilon$

$L_7 = \{a^m b^n \mid m = n,\ m, n \geq 1\}$

$S \rightarrow aSb \mid ab$

L_8)

$S \rightarrow AB$
$A \rightarrow aA \mid a$
$B \rightarrow aBb \mid \epsilon$

(or)

$S \rightarrow aSb \mid aS \mid a$

(or)

$S \rightarrow aSB \mid a$
$B \rightarrow b \mid \epsilon$

$\{a^m b^n \mid m > n,\ m, n \geq 0\}$

$\Rightarrow$ $S \rightarrow aab \mid aS \mid aSb$ $\forall$ $\{m > n,\ m, n > 0\}$

(or)

$S \rightarrow aSB \mid aab$
$B \rightarrow b \mid \epsilon$

Scanned with CamScanner

(or)

$S \rightarrow aaAb$

$A \rightarrow aAb \mid aA \mid a$

(or)

$S \rightarrow AB$

$A \rightarrow aA \mid a$

$B \rightarrow aBb \mid ab.$

L9) $S \rightarrow aSb \mid Sb \mid abb.$

L10) $S \rightarrow aSbS \mid bSaS \mid \epsilon$

L11) $S \rightarrow aSa \mid bSb \mid c$ abcba

L12) $\{ a^m b^m c^n d^n \mid m, n \geq 1 \}$

L13) $\{wcw^R \mid w \in \{a,b\}^*\}$ (length is even)

(no odd length strings)

L14) $\{w \mid w$ is a palindrome over $\{a,b\}.\}$

(Both even & odd covers)

L13) $S \rightarrow aSa \mid bSb \mid \epsilon$

L14) $S \rightarrow aSa \mid bSb \mid a \mid b \mid \epsilon$

L12) $S \rightarrow AB$

A → will generate equal no. of a's & b's

B → will generate equal no. of c's & d's

$S \rightarrow AB$

$A \rightarrow aAb \mid ab$

$B \rightarrow cBd \mid cd$

Scanned with CamScanner

L(15) = $\{a^m b^n c^n d^m \mid m, n \geq 0\}$

L(15) $S \rightarrow aSd \mid A$

$A \rightarrow bAc \mid \epsilon$

L(16) = $\{a^m b^n c^m d^n \mid m, n \geq 0\}$

{not context Free Grammar}

$\Rightarrow$ (V, Σ, P, S)

$V \rightarrow$ non Terminals

$\Sigma \rightarrow$ Terminals

$G: (\{E\} \{+, *, a\}, P, E)$

$E \rightarrow E+E$

$\mid E*E$

$\mid E-E$

$\mid E/E$

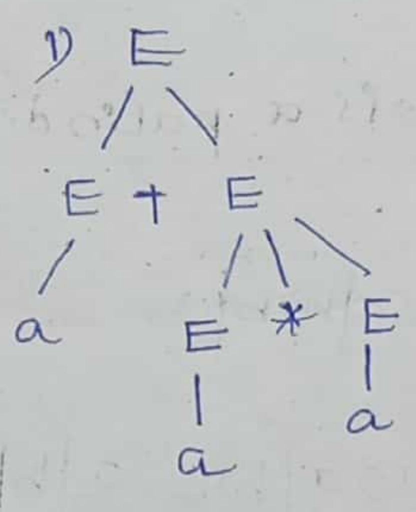
$\mid a$

$\boxed{a+a*a}$

↳ construct Parse Tree?

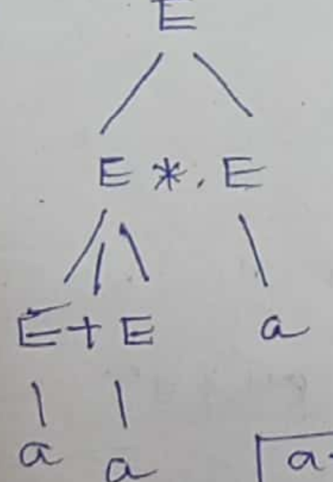
Parse Tree:-

i)



$\boxed{a+a*a}$

ii)



$\boxed{a+a*a}$

$\boxed{\text{If then else Grammar}}$

⇒ For a single string if there exists more than one parse Tree then the Grammar is said to be "Ambigous Grammar".

⇒ **LMD** :-

$E \to E+E$
$a+E$
$a+E*E$
$a+a*E$
$\boxed{a+a*a}$

⇒ **RMD** :-

$E \to E+E$
$E+E*E$
$E+E*a$
$E+a*a$
$\boxed{a+a*a}$

* **Assignment** :-

1) Simplification of CFG? →
2) chomsky normal Form Grammar (CNF).
3) Greibach normal Form (GNF)

$S \to aSB \mid \epsilon$
$B \to b$
↳ unit production
removing unnecessary
ϵ-symbols
unreachable from
Starting symbol
nongenerated

⇒ $S \to aB \mid bA$

$A \to a \mid aS \mid bAA$

$B \to b \mid bS \mid aBB$

String "aaabbabbba"

construct parse Tree

LMD

RMD.

⇒ **Push Down Automata (PDA)**

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

Q: Finite set of states

Σ: input alphabet

Γ: stack Alphabet

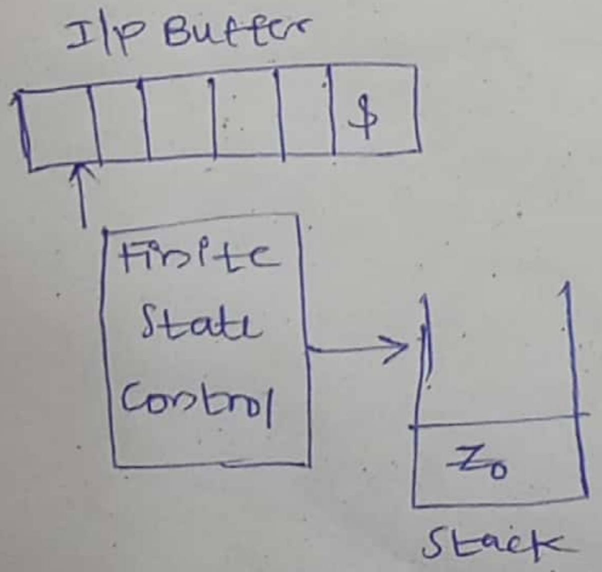
δ: transition Function

$\delta : Q \times \Sigma \times \Gamma \to (Q, \Gamma^*)$

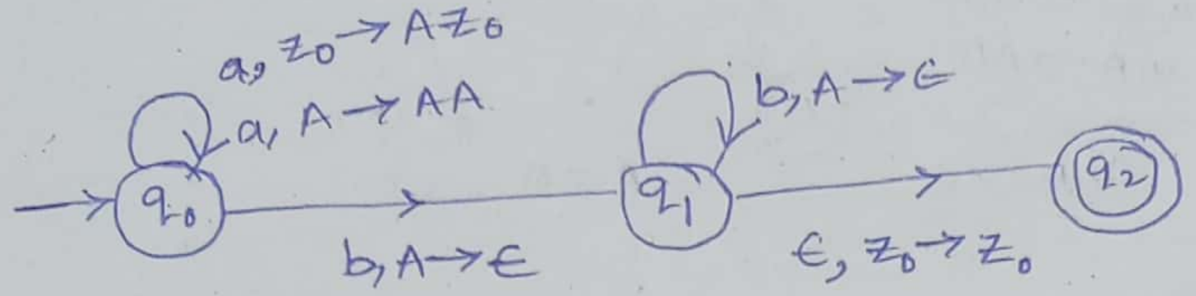
q_0: Initial state

Z_0: Initial stack symbol / bottom of stack symbol.

F: Finite state.



$\Rightarrow L_1 = \{a^n b^n \mid n > 0\}$



$\delta(q_0, a, Z_0) \rightarrow (q_0, AZ_0)$

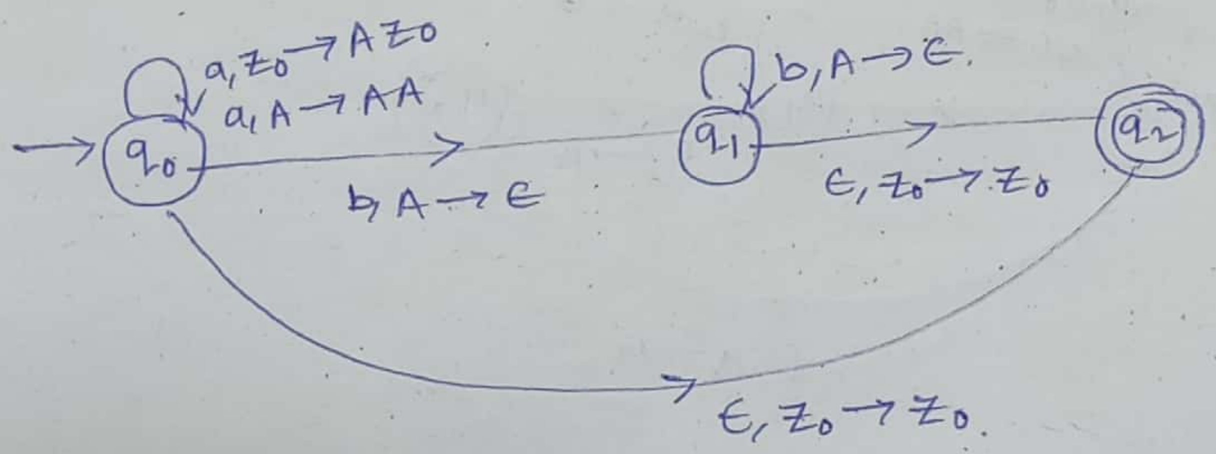
$\delta(q_0, a, A) \rightarrow (q_0, AA)$

$\delta(q_0, b, A) \rightarrow (q_1, \epsilon)$

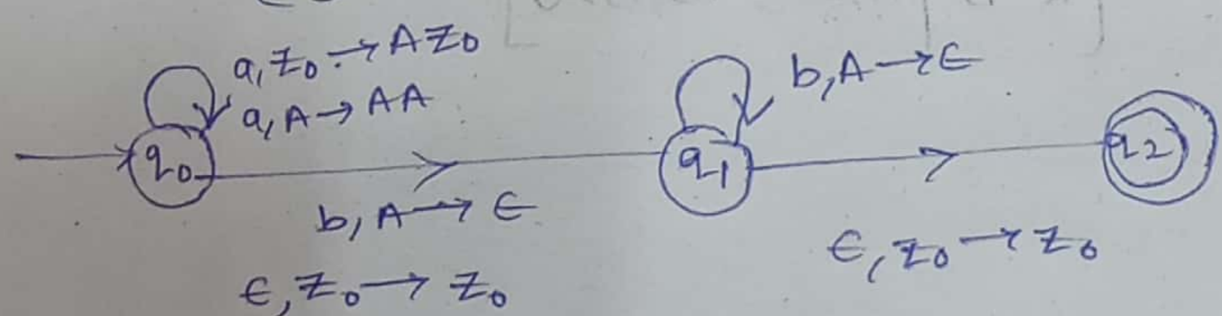
$\delta(q_1, b, A) \rightarrow (q_1, \epsilon)$

$\delta(q_1, \epsilon, Z_0) \rightarrow (q_2, Z_0)$

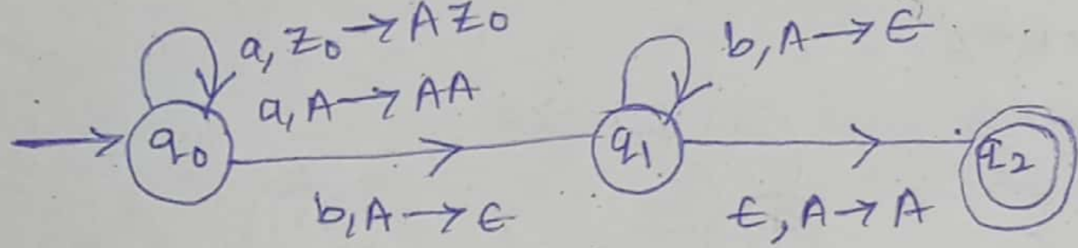
$\Rightarrow L_2 = \{a^n b^n \mid n \geq 0\}$



(or)



$\Rightarrow L_3 = \{ a^m b^n \mid m > n,\ m,n > 0 \}$



$$\delta(q_0, a, Z_0) \to (q_0, AZ_0)$$

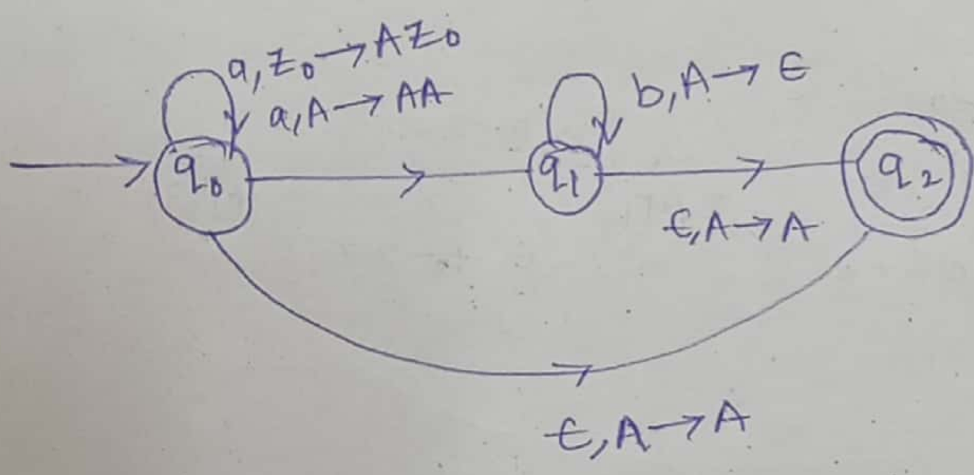
$$\delta(q_0, a, A) \to (q_0, AA)$$

$$\delta(q_0, b, A) \to (q_1, \epsilon)$$

$$\delta(q_1, b, A) \to (q_1, \epsilon)$$

$$\delta(q_1, \epsilon, A) \to (q_2, A)$$

$\Rightarrow L_4 = \{ a^m b^n \mid m > n,\ m,n \geq 0 \}$



$\Rightarrow L_5 = \{ a^m b^n \mid m < n,\ m,n > 0 \}$