Regular Expression :-

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

$b = \{b\}$

$a = \{a\}$

- ϕ, ϵ are RE

  ϕ — $\{\}$, ϵ — $\{\epsilon\}$

- If V_1 is RE and V_2 is RE then, $V_1 + V_2$ is also RE

  $$\left(L(V_1+V_2) = L(V_1) \cup L(V_2) \right)$$

- If R is Reg Exp (RE) then (R) is also RE

- If is RE then R^* is also RE

eg:- ✓ $(\phi^* + \epsilon + a)^* \cdot a^*$ (RE)

✓ $(\phi + \epsilon)^*$ (RE)

✗ $(\phi^* + \epsilon)^* \cdot ab + ab +$ (Not RE)

✓ $(a)^* + (\phi)^*$ (RE)

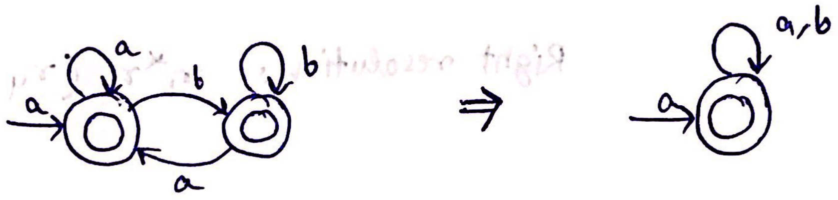
⇒ Machine to RE :-

Steps :-

(i) NFA/ DFA doesn't matter for finding RE.

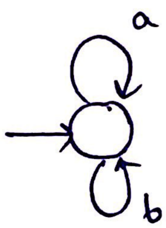
(ii) Remove Non-reachable states. and trap states.

(iii)

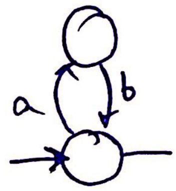


Both {a,b} I/P
are accepted

(iv)

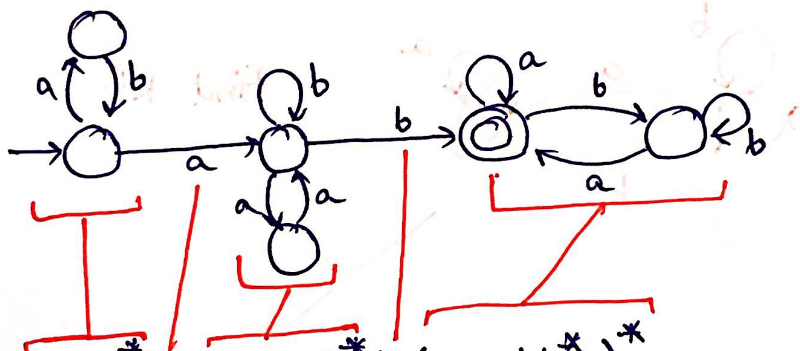


RE = $(a+b)^*$



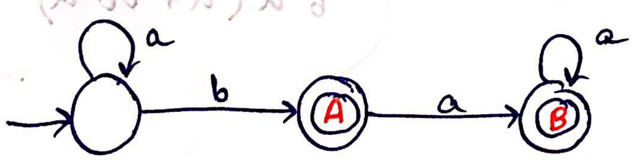
$(ab)^*$

(v)



$(ab)^* a (b+aa)^* b (a+bb^*a)^*$

(vi) Simplify the RE



RE = $\gamma_A + \gamma_B$

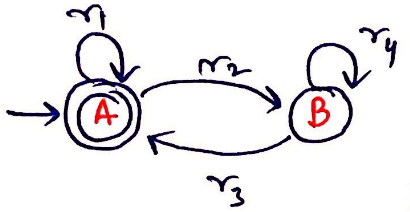
$\gamma_A = a^* b$

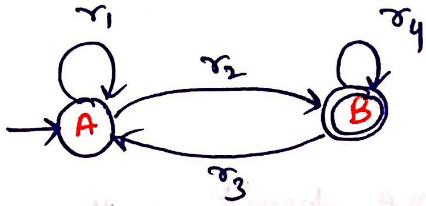
$\gamma_B = \gamma_A \cdot aa^*$

RE = $a^*b + \gamma_A aa^* = a^*b(\epsilon + aa^*)$

$= \underline{a^* b a^*}$

(vii)



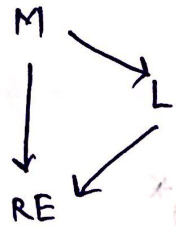


$r_A = (r_1 + r_2 r_4^* r_3)^*$

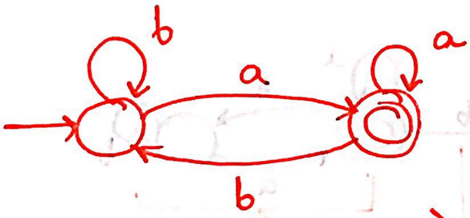
Left resolution : $(r_1 + r_2 r_4^* r_3)^* r_2 r_4^*$

Right resolution : $r_1^* r_2 (r_4 + r_3 r_1^* r_2)^*$

(viii)



Q.



Find RE

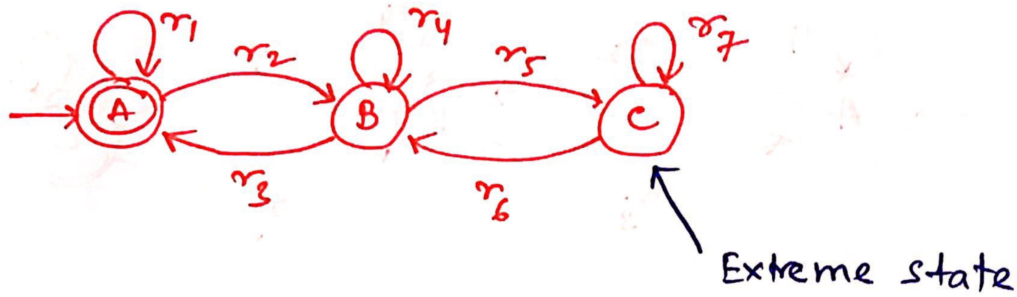
Left Resolution

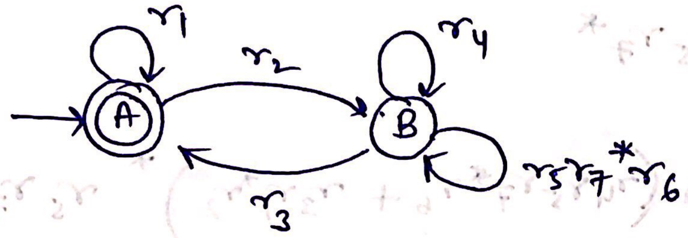
$((b + aa^*b)^* aa^*)$

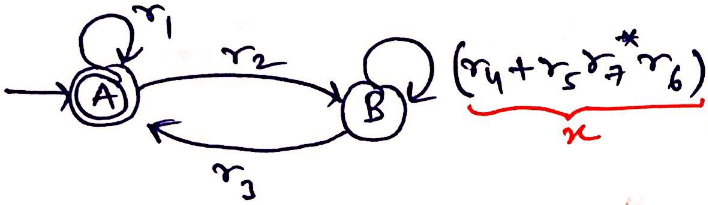
Right Resolution

$(b^*a(a + bb^*a)^*)$

Q.



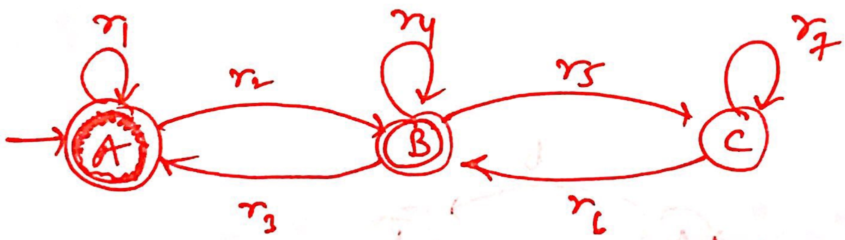


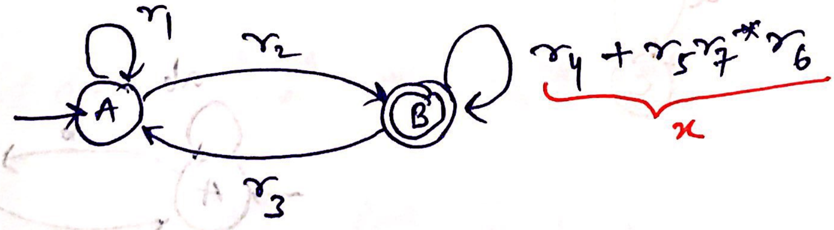


$$r_A = r_1 + r_2 r_B^* r_3$$

$$r_A = \left(r_1 + r_2 (r_4 + r_5 r_7^* r_6)^* r_3\right)^*$$

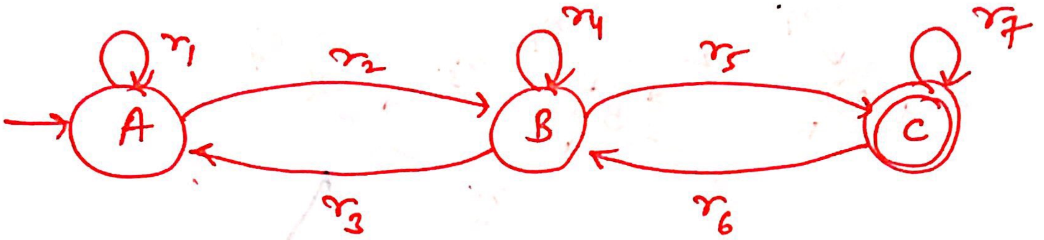
Q.





Right resolution : $r_B = r_1^* r_2 \left(r_4 + r_5 r_7^* r_6 + r_3 r_1^* r_2 \right)^*$

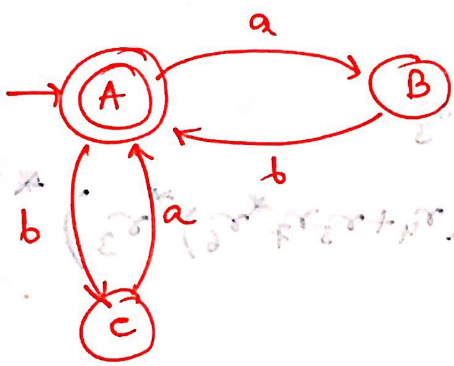
Q.



$$r_c = r_B r_5 r_7^*$$

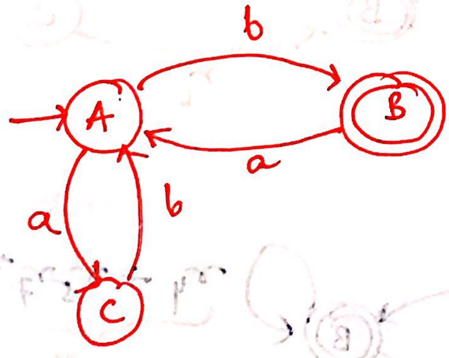
$$= r_1^* r_2 \left(r_4 r_5 r_7^* r_6 + r_3 r_1^* r_2 \right)^* r_5 r_7^*$$

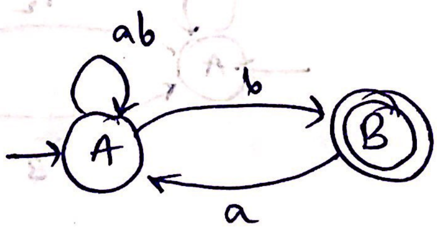
Q.



$(ab + ba)^*$

Q.

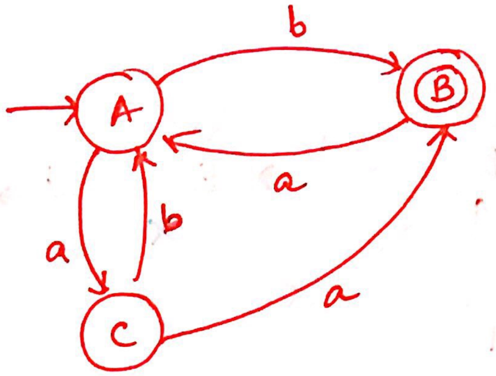


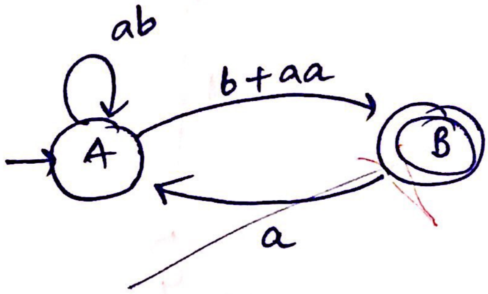


LR : $(ab + ba)^* b$

RR : $(ab)^* b \left(a (ab)^* b \right)^*$

Q.

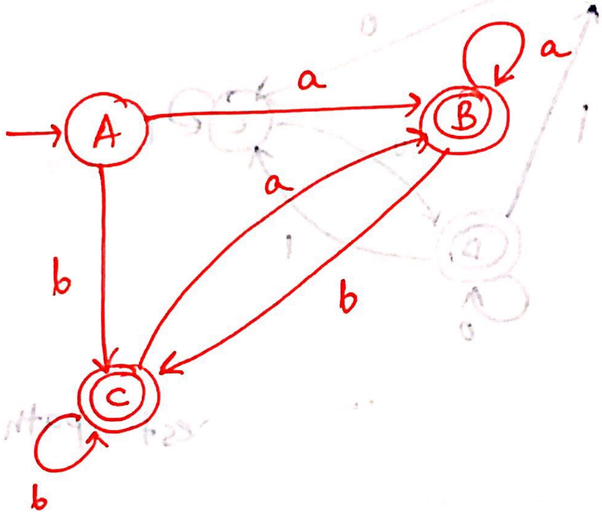




LR : $(ab + (b+aa)a)^* \quad (b+aa)$

RR : $(ab)^* (b+aa) (a(ab)^* (b+aa))^*$

Q.



$\underline{\underline{(a+b)^+}}$

$B = a(a+bb^*a)^* + b(b+aa^*b)^* aa^*$

$C = a(a+bb^*a)^* bb^* + b(b+aa^*b)^*$

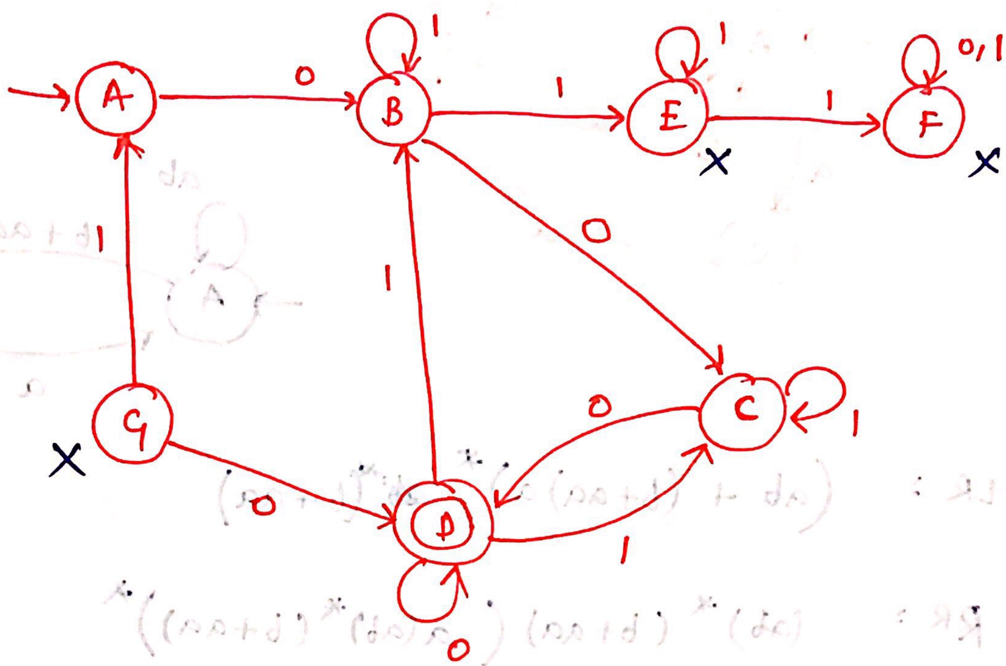
$B+C = a(a+bb^*a)^*(\epsilon+bb^*) + b(b+aa^*b)^*(aa^*+\epsilon)$

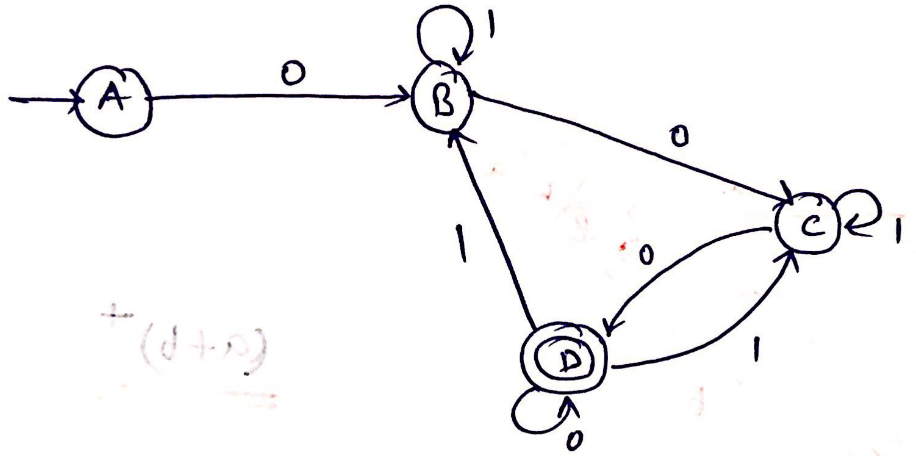
$= a(a+bb^*a)\, b^* + b(b+aa^*b)\, a^*$

$= a((bb^*+\epsilon)a)\, b^* + b((aa^*+\epsilon)b)\, a^*$

$= (a(b^*a)b^* + b(a^*b)a^*)$

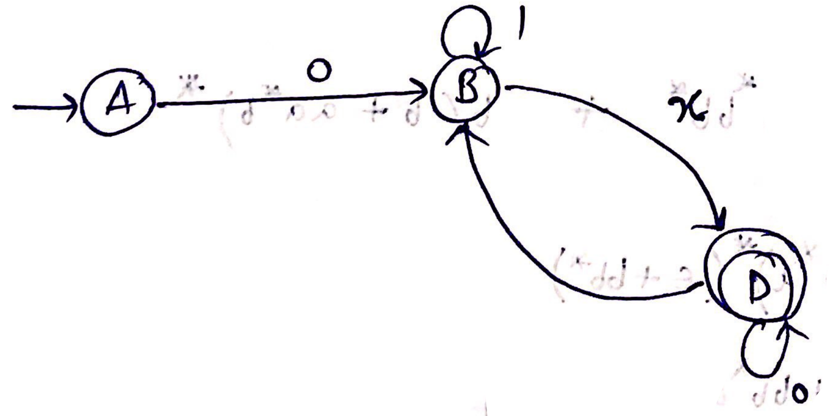
Q.





Delete c state and write longest path for B→C→D

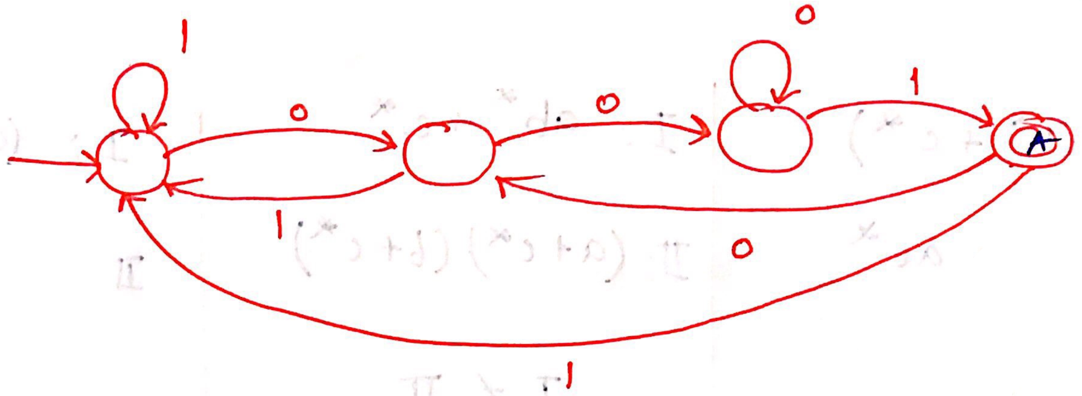
i.e $\underbrace{0(1+00^*1)^*}_{x}0$



$r_D = 0(1+x0^*1)^* x0^*$ LR

$r_D = 01^*x(0+11^*x)^*$ RR

Q.



$(r_A = (0+1)^* 001)$

⇒ Properties of RE :-

1. **commutative** : $r+s = s+r$

$r.s \neq s.r$ — $a^*b \neq ba^*$, $aa^* = a^*a$

2. **Associative** : $r+(s+t) = (r+s)+t$

$r.(s.t) = (r.s).t$

3. **Distributive** : $r(s+t) = r.s + rt$

$r+st \neq (r+s)(r+t)$

eg:-

I : $a(b^* + c^*)$	I: $ab^* + c^*$	I: $(a+b)(a+c)$
II: $ab^* + ac^*$	II: $(a+c^*)(b+c^*)$	II: $a+bc$
$I = II$	$I \neq II$	$I \neq II$

4. Identity : $r + \phi = r = \phi + r$

$r \cdot \epsilon = r = \epsilon \cdot r$

eg:-

$r + \epsilon \neq r$ if 'ϵ' $\notin \{r\}$; $r + \epsilon = r$ if 'ϵ' $\in \{r\}$

5. Properties of *, +, · :

$\epsilon^* = \epsilon$ $\quad$ $\epsilon^+ = \epsilon$

* $\phi^* = \epsilon$ $\quad$ $\phi^+ = \phi$

$\epsilon\epsilon = \epsilon$ $\quad$ $r + r = r$ $\quad$ $(r^*)^* = r^*$

$\epsilon \cdot \phi = \phi$ $\quad$ $rr \neq r$
$= r^2$ $\quad$ $(r^+)^+ = r^+$

$\phi \cdot \phi = \phi$ $\quad$ $(r^+)^* = r^*$

- $r^* \cdot r^+ = r^+$
- $r^+ \cdot r^+ = rr^+ = rrr^*$

eg:- $(ab^*)^*$ $\qquad (a + ab + abb + abb + \dots)^*$

(i) aabab ✓

(ii) ababba ✓

(iii) abbabbb ✓

(iv) b ✗

- $\epsilon + rr^* = r^*$

eg:-

I : $\epsilon + ab^*b(ab^*b)^*$ $\qquad$ I = II

II : ab^*b

eg:-

I : $\epsilon + a + aaa^*$ $\qquad$ I = III ≠ II

II : aa^*

III : a^*

- $p(qp)^* = (pq)^*p$ $\qquad$ where p and q are RE

$p(qr)^* \neq (pq)^*r$ $\qquad$ where p, q and R are RE

$(pq)^*q \neq p(qq)^*$

$(r^* + s^*)^* = (r+s)^* = (r^* s^*)^*$

$\Updownarrow$

$(s^* r^*)^*$

$(r+s)^* = (r+s)^* r^*$

$\neq (r+s)^* r$

Arden's Theorem :-

Let consider a recurrence relation $(r = p + qr)$

then it has a unique solution : $(\epsilon \notin L(q))$

$$(r = q^* p)$$

OR for $r = p + rq \rightarrow (r = pq^*)$

eg:-

$A = (1+01) + 1A + 01B$

$B = \epsilon + 01B + 1C$

$C = 10B + 0$

$B = \epsilon + 01B + 1(10B + 0)$

$B = \epsilon + 01B + 110B + 10$

$(B = \underset{p}{(\epsilon + 10)} + \underset{q}{(01 + 110)} B)$ Apply Arden's theorem

$$\boxed{B = (01+110)^*(\epsilon+10)}$$

$$\boxed{C = 10\left((01+110)^*(\epsilon+10)\right) + 0}$$

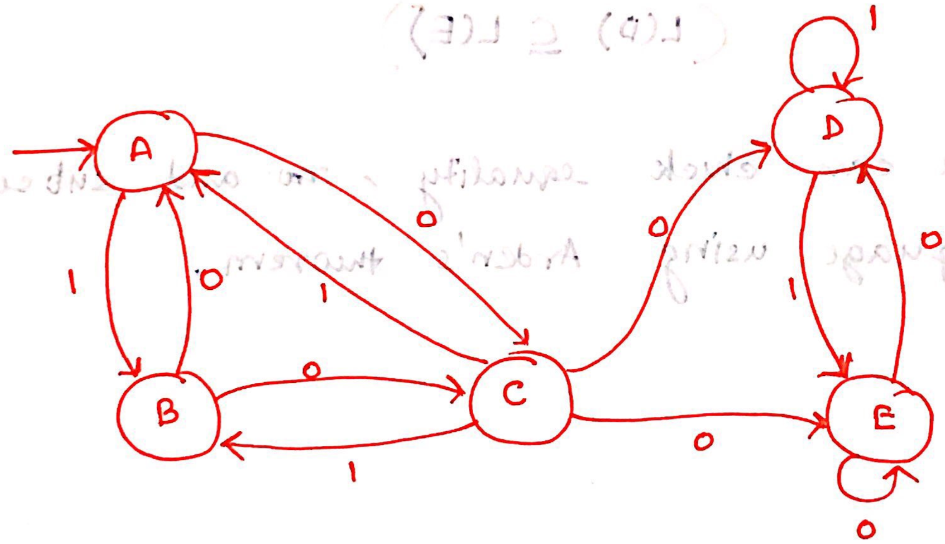
$$A = (1+01) + 1A + 01(01+110)^*(\epsilon+10)$$

$$A = \underbrace{(1+01) + \left(01(01+110)^*(\epsilon+10)\right)}_{p} + \underbrace{1}_{q}A$$

<u>Apply Arden's theorem:-</u>

$$\boxed{A = \overset{q^*}{1^*}\overset{p}{\left((1+01) + \left(01(01+110)^*(\epsilon+10)\right)\right)}}$$

Q.



If D is final state then L(D) is language accepted by FA, similarly L(E) for E as final state.

Find:

(a) ✓ $L_D = L_E$

(b) $L_D \subseteq L_E$

(c) $L_E \subseteq L_D$

(d) $L_D \neq L_E$

Make eq. for D and E only using Arden's theorem:

$D = C0 + D1 + E0$

since both D and E eq. are same

$E = C0 + D1 + E0$

so, $(L(D) = L(E))$

NOTE:

if. $D = C0 + D1 + E0$

$E = C0 + D1$

$(L(D) \subseteq L(E))$

So, we can check equality, ~~no~~ and subset of language using Arden's theorem.

Regular Grammer :-

Grammer should be either Left Linear or right linear, but both should not be occur at simuntaneously.

Eg.

$V \longrightarrow T^*V + T^*$ RL

$V \longrightarrow VT^* + T^*$ LL

eg:-

$S \rightarrow a$

$S \rightarrow \epsilon$

$S \rightarrow A|B|aba$

as $S \rightarrow A$, $S \rightarrow B$ it can be LL or RL so it is

Regular grammer

$S \rightarrow aA \mid Ba$

$S \rightarrow aA$ RL

$S \rightarrow Ba$ ∴ LL

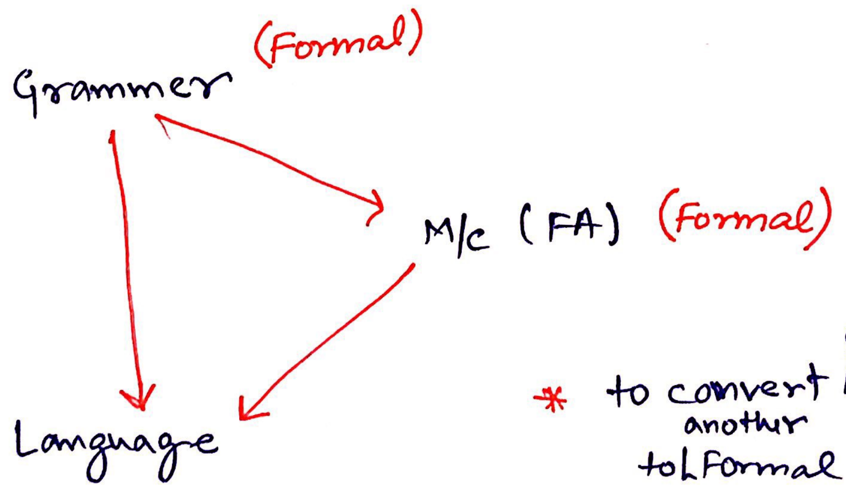
Both occur same time

Not Regular grammer

NOTE:-

(Regular grammer $\longrightarrow$ Right Linear $\longrightarrow$ FA)

Algorithm only converts RL lang to FA.



* to convert one Formal to another Formal form, there always exist an algorithm.

Some standard Regular Grammers :-

Language	Grammer
ϕ	$S \to S$
ϵ	$S \to \epsilon$
a	$S \to a$
$a+b$	$S \to a \mid b$
a^*	$S \to aS \mid \epsilon$ or $S \to Sa \mid \epsilon$ or $S \to a \mid SS \mid \epsilon$ NOT REGULAR GRAMMER
$a^* + b^*$	$S \to S_1 \mid S_2$ $S_1 \to aS_1 \mid \epsilon$ $S_2 \to bS_2 \mid \epsilon$
$(ab)^*$	$S \to abS \mid \epsilon$ or $S \to Sab \mid \epsilon$ or $S \to ab \mid SS \mid \epsilon$ NOT REGULAR GRAMMER
(aa)	$S \to AA$ $A \to a$

$a^* \cdot b^*$

$S \rightarrow AB$

$A \rightarrow aA \mid \epsilon$

$B \rightarrow bB \mid \epsilon$

CFG

S -> aB | €

B -> aB | bB | €

REG

$(a+b)^*$

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

or

$S \rightarrow Sa \mid Sb \mid \epsilon$ LL

or

$S \rightarrow a \mid b \mid SS \mid \epsilon$ NON-REGULAR

$(a+b)^* x$

$S \rightarrow aS \mid bS \mid x$

$x(a+b)^*$

$S \rightarrow Sa \mid Sb \mid x$

⇒ Grammer to M/c to Lang :-

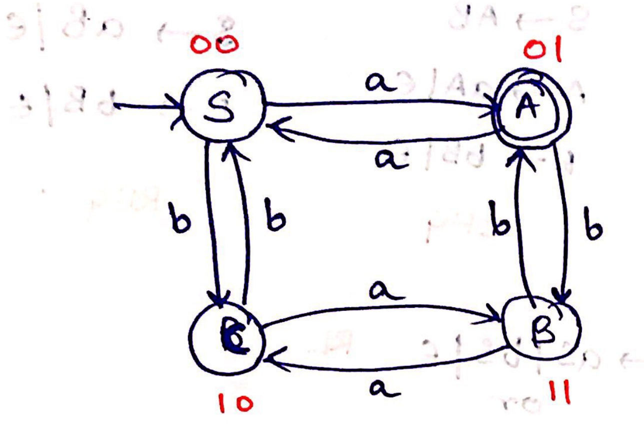
$S \rightarrow aA \mid bC$

* $A \rightarrow aS \mid bB \mid \epsilon$

$B \rightarrow aC \mid bA$

$C \rightarrow aB \mid bS$

Right Linear and $A \rightarrow \epsilon$ so A is final state.



MOD machine

$L = \{ a \bmod 2 = 1 \ \& \ b \bmod 2 = 0 \}$

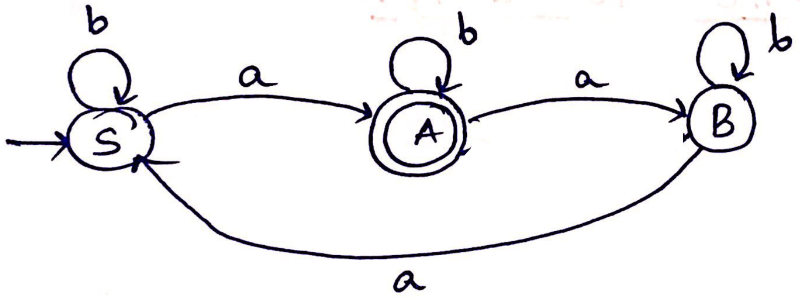
Q.

$S \rightarrow aA \mid bS$

$A \rightarrow aB \mid bA$

$B \rightarrow aS \mid bB$

$A \rightarrow \epsilon$ — Final state



$L = \{ a \bmod 3 = 1 \}$

or

$L = \{ a^n \mid n = 3x+1 \}$

$(V \rightarrow T^{*}V)$

$A \rightarrow B$ $\qquad$ $(A) \xrightarrow{\epsilon} (B)$

$A \rightarrow aB$ $\qquad$ $(A) \xrightarrow{a} (B)$

$A \rightarrow abcB$ $\qquad$ $(A) \xrightarrow{a} (X_1) \xrightarrow{b} (X_2) \xrightarrow{c} (B)$

Algorithms in FA's :-

1. NFA $\longrightarrow$ DFA (Subset construction)
2. DFA $\longrightarrow$ min DFA (Partition algo)
3. NFA with ϵ-moves $\longrightarrow$ NFA without ϵ-moves (ϵ-closure algo)

(i) Subset-construction :-

NFA $\longrightarrow$ DFA

$(Q, \Sigma, \delta, q_0, F)$ $\qquad$ $(Q', \Sigma', \delta', q_0', F')$

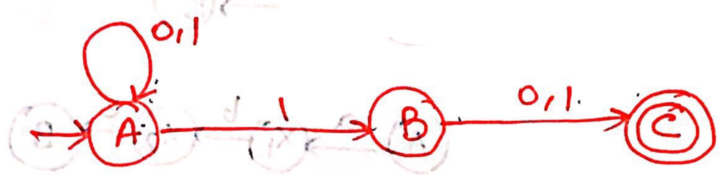
$(Q' \subseteq 2^{Q})$ $\quad$ $(\Sigma' = \Sigma)$ $\qquad$ $(q_0' = \{q_0\})$

$F' \subseteq Q' \subseteq 2^{Q}$

$(F' \subseteq 2^{Q})$

- It is not necessary to get minimal DFA from this Method.

eg:- $(0+1)^* 1 (0+1)$



NFA

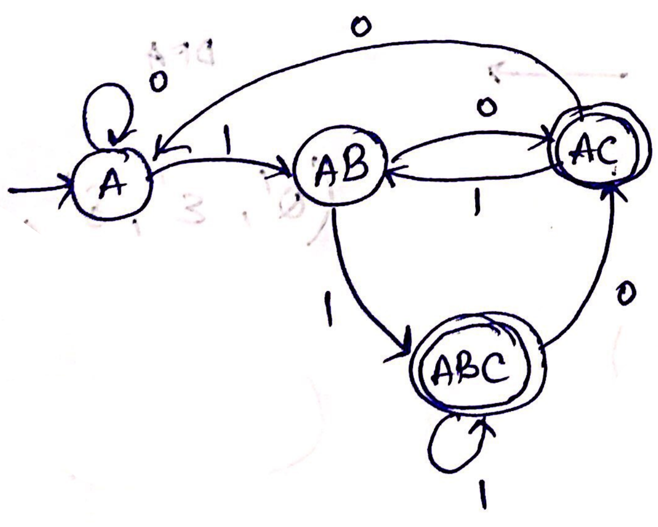
Q \ Σ	0	1
→ A	A	{A, B}
B	C	C
* C	∅	∅

final state

DFA

Q \ Σ	0	1
→ A	A	AB
AB	AC	ABC
* AC	A	AB
* ABC	AC	ABC

final state (contain C)



(DFA)

Limitation:-

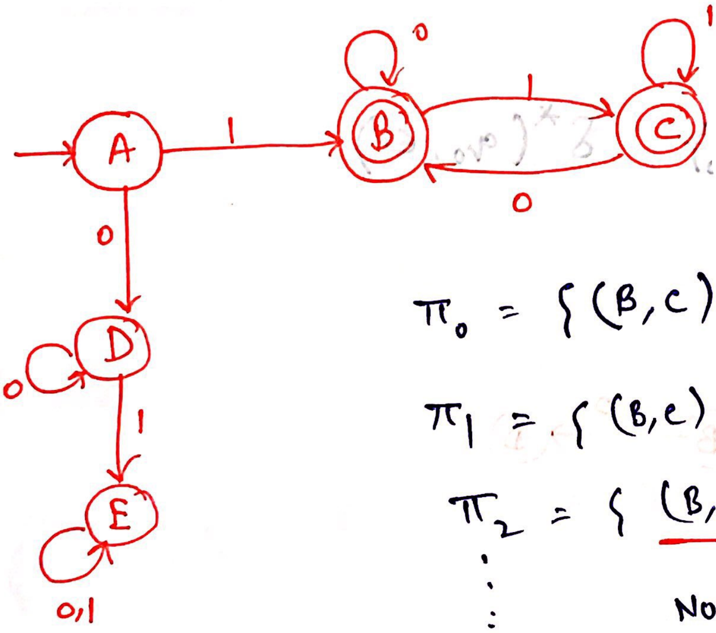
- It only take NFA without ϵ move as I/P.

(ii) Partition algorithm :-

- $A \equiv B$ iff $\forall\ w \in \Sigma^*$ $\delta^*(A, w)$ $\delta^*(B, w)$

Non final

final



$\pi_0 = \{(B, C), (A, D, E)\}$

$\pi_1 = \{(B, C), (A), (D, E)\}$

$\pi_2 = \{(B, C), (A), (D, E)\}$

No change STOP

for π_0, $|w| \leq 0$

$\delta^*(A, \epsilon) \rightarrow A$ $\delta^*(D, \epsilon) \rightarrow D$

(NF)

so, $(A \equiv D)$

NOTE:- Remove unreachable state from NFA ~~without~~ before applying this algorithm.

similarly do for all states

$\pi_0 \Rightarrow$

$A \xrightarrow{0} D$

$*B \xrightarrow{0} *B$

$*C \xrightarrow{0} *B$

$D \xrightarrow{0} D$

$E \xrightarrow{0} E$

$\pi_1 \Rightarrow$

$A \xrightarrow{1} *B$

$*B \xrightarrow{1} *C$

$*C \xrightarrow{1} *C$

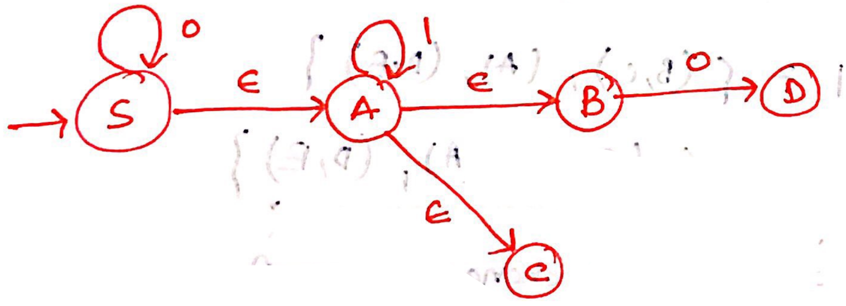
$D \xrightarrow{1} E$

$E \xrightarrow{1} E$

$\{(A, D, E), (B, C)\}$ $\{(B, C), (A), (D, E)\}$

(iii) ϵ - closure algorithm :-

$$\epsilon\text{-closure}(q_0) = \delta^*(q_0, \epsilon)$$



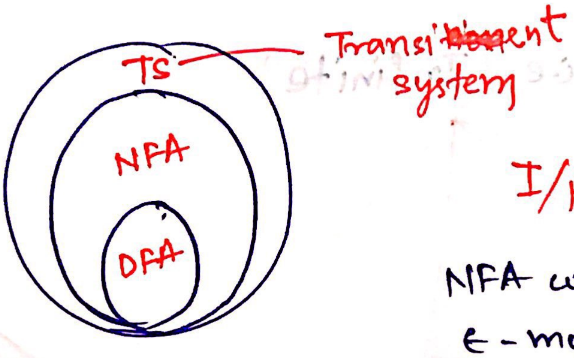
ϵ - closure(S) = $\{S, A, B, C\}$

ϵ - closure (A) = $\{A, B, C\}$

ϵ - closure (B) = $\{B\}$

ϵ - closure (C) = $\{C\}$

ϵ - closure (D) = $\{D\}$



I/p		O/p
NFA with ϵ-moves	$\longrightarrow$	TS without ϵ-moves
		$\downarrow$
		NFA without ϵ-move

- Number of states remains same
- δ (transition) are added i.e edges to NFA were increased

⇒ Identify Regular, CFL or CSL :-

↳ Finite : Yes $\longrightarrow$ Regular

eg:- $\{100, 010, 001\}$

$\{0^n 1^n \mid n \leq 10\}$; $\{0^n 1^n \mid n \geq 10\}$ CFL

$\{0^{n^2} \mid n \leq 10\}$; $\{0^{n^2} \mid n \geq 10\}$ CSL

finite $(n \leq 10)$

↳ Finite : ? → No i.e (Infinite)

- I — Non-Linear power
  - $0^{n^2}, 0^{n^3}, 0^{n!}, 0^{2^n} \ldots$ with $n \geq 0$ → CSL
- II — +/− or ×/÷ relation
  - (+/−) → CFL
  - (×/÷) → CSL
- III — Counting comparision
  - 0 cmp → REG
  - 1 cmp → CFL
  - 2 cmp → CSL
- IV — Type of cmp
  - Linear → CFL
  - Non-Linear → CSL

eg:- **I:** $L = \{ 0^n 1^n 2^m \mid n \leq 10 ; m \geq 0 \}$

it is <u>regular</u> as $0^n 1^n$ is finite set as $n \leq 10$ and 2^m (m is independent & linear)

II: $L = \{ a^m b^n c^p \mid m + n = p \}$

it is <u>not regular</u> as there is dependency b/w m, n, p. ($m + n = p$)

III:

$L_1 = \{ a^n \} \longrightarrow$ no cmp REG

$L_2 = \{ a^n b^n \text{ or } a^n b^{2n} \} \longrightarrow (n_a = n_b)$ CFL

$L_3 = \{ a^n b^n c^n \text{ or } a^n b^{2n} c^{3n} \} \longrightarrow (n_a = n_b = n_c)$ CSL

IV:

$L_1 = \{ a^m b^n \mid m = n+10 \text{ or } m-n=10 \}$

Linear comparision CFL

$L_2 = \{ a^m b^n \mid m = 10n \text{ or } m/n = 10 \}$

$(k_1 n + k_2) \longleftarrow$ Linear comparision then CFL

V:

$L_1 = \{ a^m b^n \}$

- $m+n = 10$ FINITE → REG
- $m+n \leq 10$ FINITE → REG
- $m+n \geq 10$ INFINITE → REG

NOTE: $\overline{0^* 1^*} = (0+1)^* 1 0 (0+1)^*$

10 Aug 2019