# Regular Expressions & Languages

An Expression which is constructed over the $\Sigma$ using the operators $*, \cdot, +$ is called as Regular Expression.

Ex: ① $r = 0+1$

② $r = (10)^*$

③ $r = 0^* \cdot 1^*$

④ $r = 1^*0 + 0^*1$

⑤ $r = (1^*0)^* + 1^*0^*$

**Regular operators:** The operators $*, \cdot, +$ are called as Regular operators.

$+ \rightarrow$ union

$\cdot \rightarrow$ Concatenation

$* \rightarrow$ Kleen closure

**Order of precedence:**

$* > \cdot > +$

1 2 3

Ex: $r = (a + b \cdot a)^*$

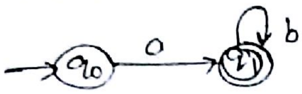
$* \rightarrow$ ①, $\cdot \rightarrow$ ②, $+ \rightarrow$ ③

**Note:** If $r$ is any regular Expression then $L(r)$ is the Language generated by R.E 'R'

| SNO | R.E | RL |
|---|---|---|
| 1. | $r = \phi$ | $L = \{\}$ |
| 2. | $r = \epsilon$ | $L = \{\epsilon\}$ |
| 3. | $r = a$ | $L = \{a\}$ |
| 4. | $r = ab$ | $L = \{ab\}$ |
| 5. | $r = a+b$ | $L = \{a, b\}$ |
| 6. | $r = a+b+c$ | $L = \{a, b, c\}$ |
| 7. | $r = w_1 + w_2 + \cdots w_n$ | $L = \{w_1, w_2 \ldots w_n\}$ |
| 8. | $r = (a+b) \cdot a$ | $L = \{aa, ba\}$ |
| 9. | $r = a^*$ | $L = \{\epsilon, a, aa, aaa \ldots\}$ |

② Every language generated by RE will be accepted by [illegible]

<u>Ex:</u> $r = ab^*$ $L = \{a, ab, abb, abbb \dots\}$




<u>Identity Rules:</u>

① Every Finite language is regular

$$L = \{W_1, W_2, \dots W_n\}$$

$$r = W_1 + W_2 + \dots W_n$$

② If $r$ is a regular Expression then both $r^*$, $r^+$ are also R.E

$$r^* = \epsilon, r, rr, rrr \dots$$

$$r^+ = r, rr, rrr \dots$$

③ If $r = \phi$ then

$$r^* = \epsilon,$$

$$r^+ = \phi$$

④ If $r = \epsilon$ then $r^* = \epsilon$, $r^+ = \epsilon$

⑤ If $r_1, r_2$ be the two R.E, then both $r_1 + r_2$, $r_1 \cdot r_2$ are also regular

$$L(r_1 + r_2) = L(r_1) + L(r_2)$$

$$L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2)$$

⑥ If $r_1, r_2$ be the two R.E then

$$(r_1 + r_2)^* = (r_1^* + r_2^*)^*$$
$$= (r_1 + r_2^*)^*$$
$$= (r_1^* + r_2)^*$$
$$= (r_1^* \cdot r_2^*)^*$$
$$= (r_2^* \cdot r_1^*)^*$$

⑦ If $r_1, r_2$ be the two R.E then

$$r_1 (r_2 r_1)^* = (r_1 r_2)^* r_1$$

Two regular expressions $r_1$ and $r_2$ are equal iff

$$\boxed{L(r_1) = L(r_2)}$$

(a)

$r^* = \{\epsilon, r, rr, \ldots\}$

$r^+ = \{r, rr, rrr, \ldots\}$

(i) $r^+ \subset r^*$

(ii) $r^+ \cup r^* = r^*$

(iii) $r^+ \cap r^* = r^+$

(iv) $r^* \cdot r^+ = r^+$

(v) $(r^*)^* = r^*$

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

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

(viii) $r^* \cdot r^* = r^*$

(ix) $r^+ \cdot r^+ = rr^+$

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

(Q) ① Simplify the following regular expressions.

① $(0+\epsilon)^* = 0^*$

② $(0^* + 0^+)^* = (0^+)^* = 0^*$

③ $(1^*0^* + 1)^* = (1^*0^* + 1^*)^*$

$= (1^*(0^* + \epsilon))^*$

$= (1^*0^*)^*$

$= (1+0)^*$

④ $(10^* + 10)^*$

$= (10^*)^*$

$= (10^*)^*$

(5) $r = (1^*0)^* + 0^*1 + 1^*$

(6) $(1^*0)^* 0^* + (01)^*$

$= (1^*0)^* + (01)^*$

(7) $r = (0^* + 1)^* + (10)^* 0$

$= (0+1)^* + (10)^* 0$

$= (0+1)^*$

(8) $r = (0^* + 1^* 0)^*$

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

$= (1^* 0)^*$

(9) $r = (0^* + 1^+ 0^*)^* + (01^*)^*$

$= (0^*(\epsilon + 1^+))^* + (01^*)^*$

$= (0^* 1^*)^* + (01^*)^*$

$= (0+1)^* + (01^*)^*$

$= (0+1)^*$

(10) $r = ((1^*0 + 0^*)^*)^+$

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

$= (1^*0)^*$

Construction of R.E

① Construct the R.E that generates all strings of a's & b's

(i) including $\epsilon$

(ii) Excluding $\epsilon$

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

① $\Sigma^* = (a+b)^*$

② $\Sigma^+ = (a+b)^+$

② Construct the R.E that generates all the strings of a's & b's where every string contains substring aba

(i) Starts with ab  (ii) ends with ba  (iii) contains substring aba

Sol: (i)

$w = ab x$

$= ab(a+b)^*$

(ii)

$w = x ba$

$= (a+b)^* ba$

(iii)

$w = x aba x$

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

③ (i) Starts & ends with a

(ii) starts & ends with same symbol

(iii) Starts & ends with different symbol

(i) Sol:

$w = a x a$

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

(ii)

$w = a x a + b x b$

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

(iii) $w = a\,x\,b + b\,x\,a$

$a(a+b)^{*}b + b(a+b)^{*}a$

(4) Construct the R.E that generates all the strings of a's & b's where

(i) 3rd symbol from left end a

(ii) 4th symbol from right end is b

(i) Sol: $w = x\ x\ a\ x$

$= (a+b)(a+b)\,a\,(a+b)^{*}$

(ii) Sol: $w = (a+b)^{*}\ \underset{4}{b}(a+b)(a+b)(a+b)$

(8) (*) Construct the R.E that generates all the strings of a's & b's, where the Length of the string is

(i) Exactly 3 (ii) at most 3 (iii) atleast 3 (iv) even (v) odd (vi) $\equiv 2 \pmod 3$

(i) Sol: $\Sigma = \{a,b\}$

$w = x\ x\ x$

$= (a+b)(a+b)(a+b)$

(ii) $w = \epsilon + (a+b) + (a+b)(a+b) + (a+b)(a+b)(a+b)$

(iii) $w = ((a+b)(a+b)(a+b))(a+b)^{*}$

(iv) $|w| = \text{even} = 0 \pmod 2$

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

(v) $|w| = \text{odd} = 1 \pmod 2$

$= (a+b)\,[(a+b)^{2}]^{*}$

(vi) $|w| = 2 \pmod 3$ $[\epsilon, 3, 6, 9 \dots]$

$= (a+b)^{2}\,[(a+b)^{3}]^{*}$

Construct R.E for the following Languages

1. $L = \{ a^n \mid n \geq 0 \}$

$r = a^*$

2. $L = \{ a^n \mid n \geq 1 \}$

$r = a^+$

3. $L = \{ a^m b^n \mid m \geq 0, n \geq 0 \}$

$r = a^* . b^*$

4. $L = \{ a^m b^n \mid m \geq 1, n \geq 1 \}$

$r = a^+ . b^+$

5. $L = \{ a^m b^n \mid m \geq 0, n \geq 1 \}$

$r = a^* b^+$

6. Construct R.E that generates all the strings of a's & b's such that length of the string is even

(i) The strings starts with a and length of the string is odd.

(ii) String starts with b and

(i) $\Sigma = \{a, b\}$

$W = a\underline{x}$ and $|W| = \text{even}$

$= a\,[(a+b)^2]^* . \cancel{(a+b)}\,(a+b)$

(ii) $\Sigma = \{a, b\}$

$W = b\underline{x}$ and $|W| = \text{odd}$

$= b\,[(a+b)^2]^*$

(Q) Construct the R.E for the following language

① $L = \{a^m b^n \mid m+n = \text{even}\}$

② $L = \{a^m b^n \mid m+n = \text{odd}\}$

$m+n = \text{even}$

$m,n \to \text{even}$ $\quad$ $m,n \to \text{odd}$

$m = 2x$

$n = 2y$

$a^{2x} \cdot b^{2y}$

$(aa)^x \cdot (bb)^y$

$(aa)^* \cdot (bb)^*$

$m = 2x+1$

$n = 2y+1$

$a^{2x+1} \cdot b^{2y+1}$

$(aa)^x \cdot a\ (bb)^y \cdot b$

$(aa)^* a\ (bb)^* b$

$\Rightarrow (aa)^* (bb)^* + (aa)^* a\ (bb)^* b$

$m+n = \text{odd}$

$m = \text{even}$, $n = \text{odd}$ $\quad$ $m = \text{odd}$, $n = \text{even}$

$m = 2x$

$n = 2y+1$

$a^{2x}\ b^{2y+1}$

$(aa)^x\ (bb)^y\ b$

$m = 2x+1$

$n = 2y$

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

Eb: Construct the R.E that generate all the strings of a's & b's where no. of a's in a string

(i) Exactly 3 $\quad$ (ii) atmost 3 $\quad$ (iii) atleast 3 $\quad$ (iv) even $\quad$ (v) odd $\quad$ (vi) $\equiv 1 (\text{mod}$

(i) Exactly 3

$W = x\,a\,x\,a\,x\,a\,x$

$R = b^* a b^* a b^* a b^*$

(ii) atmost 3

$W = x(a+\epsilon)\,x\,(a+\epsilon)\,x\,(a+\epsilon)$

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

(iii) atleast 3

$W = x\ a\ x\ a\ x\ a\ x$

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

(iv) even $= |w_a|$

$\equiv 0 (\text{mod } 2)$

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

(v) odd $= |w_a|$

$= b^* (a2)^* \cdot a \cdot b^*$

## Identity Rules:

1. $\epsilon R = R\epsilon = R$

2. $\epsilon^* = \epsilon$

3. $(\phi)^* = \epsilon$

4. $\phi R = R\phi = \phi$

5. $\phi + R = R$

6. $R + R = R$

7. $RR^* = R^*R = R^+$

8. $(R^*)^* = R^*$

9. $\epsilon + RR^* = R^*$

10. $(P+Q)R = PR + QR$

11. $(P+Q)^* = (P^*Q^*)$
    $= (P^* + Q^*)^*$

12. $R^*(\epsilon + R) = (\epsilon + R)R^* = R^*$

13. $(R+\epsilon)^* = R^*$

14. $\epsilon + R^* = R^*$

15. $(PQ)^*P = P(QP)^*$

16. $R^*R + R = R^*R$

## Conversion of FA to RE

1. Arden's Theorem
2. State Elimination Method

**Arden's Theorem:**

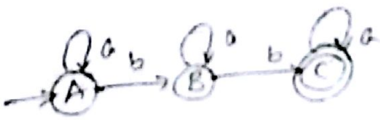
Let P, Q, R be the three RE such that $R = Q + RP$ then

(i) $R = Q + RP$ has unique solution

$\boxed{R = QP^*}$ if P is free from $\epsilon$

(ii) $R = Q + RP$ has infinite many solutions if P contains $\epsilon$

Ex 1:



$A = Aa + \epsilon$

$A = \epsilon a^*$

$A = a^*$

$B = \cancel{Bb}\ Ab + Ba$

$B = a^*b + Ba$

$B = a^*ba^*$

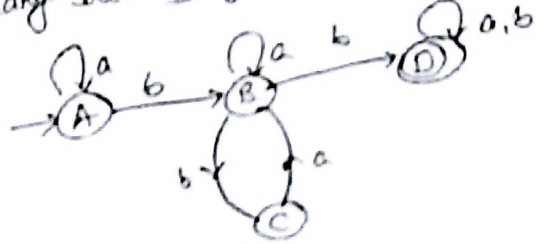
$C = Bb + Ca$

$C = a^*ba^*b + Ca$

$= a^*ba^*ba^*$

$\therefore R.E = a^*ba^*ba^*$

Ex 2:



$A = Aa + \epsilon$

$= a^*$

$B = Ab + Ba + Ca$

$\begin{cases} C = Bb \\ A = a^* \end{cases}$

$B = a^*b + Ba + Bba$

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

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

$D = \cancel{Ba}$

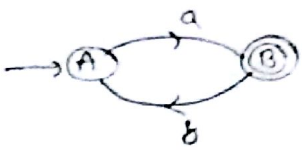
$= Bb + D(a+b)$

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

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

$\therefore RE = a^*b(a+ba)^*(a+b)^*$

③



$A = \epsilon + Bb$

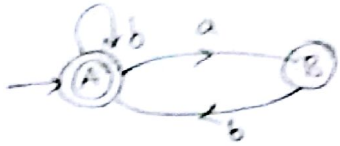
$B = Aa$

$A = \epsilon + Aab$

$A = (ab)^*$

$\Rightarrow B = (ab)^* a$

④



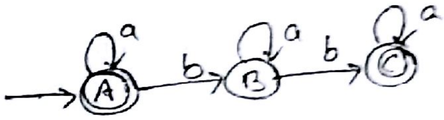
$A = Ab + Bb + \epsilon$

$= \epsilon + Ab + Aab$

$= \epsilon + A(b+ab)$

$= (b+ab)^*$

⑤



$A = \epsilon + Aa$

$A = a^*$

$B = Ab + Ba$

$= a^* b + Ba$

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

$C = Bb + Ca$

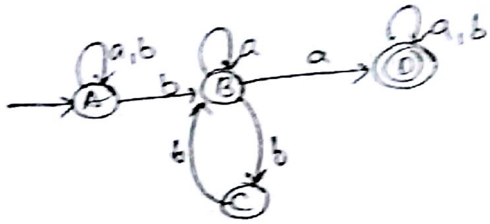
$= (a^* b) a^* b + Ca$

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

$r = A + C$

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

⑥



$A = \epsilon + A(a+b)$

$= (a+b)^*$

$B = Ab + Ba + Cb$

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

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

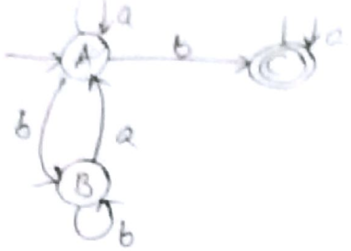
$= (a+b)^* b (a+bb)^*$

$D = Ba + D(a+b)$

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

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

(7)




$A = \epsilon + Aa + Bba$

$= \epsilon + Aa + Abb^*a$

$= \epsilon + (a + bb^*a)A$

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

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

$C = Ab + Ca$

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

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

$= [a(\epsilon + b^+)]^* ba^*$

$= (ab^*)^* ba^*$

(8)


$A = \epsilon + Aa + Bb$

$B = Ab + Ba$

$= Aba^*$

$A = \epsilon + Aa + (Aba^*)b$

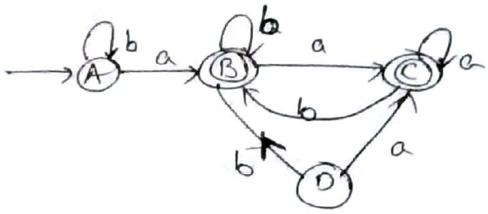
$= \epsilon + A(a+ba^*b)$

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

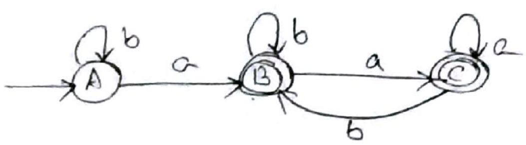
$B = Aba^*$

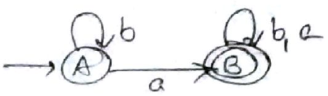
$= (a+ba^*b)^* ba^*$

(9)




Remove unreachable state




$A = \epsilon + Ab$

$= b^*$

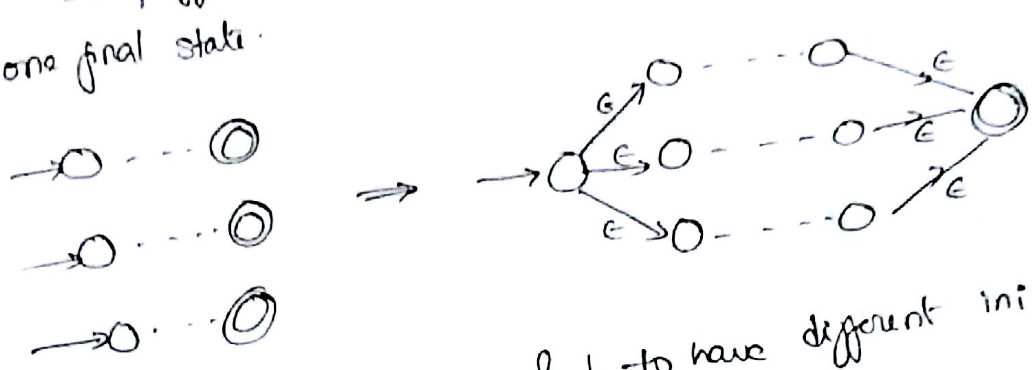
$B = Aa + (a+b)B$

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

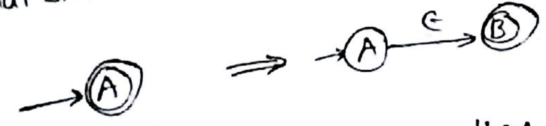
$R = b^*a(a+b)^*$

## State Elimination Method:

**Step I:** Simplify the transistion graph which have only one initial state and one final state.

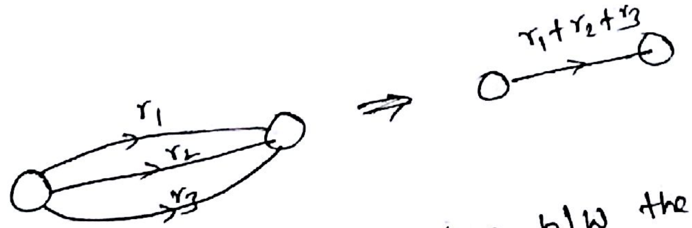



**Step II:** Simplify the transistion graph to have different initial and Final states



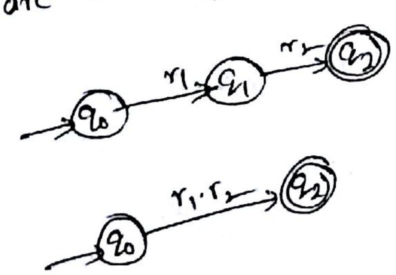

**Step III:** If there exists more than one edge in the same direction, then they are called parallel edges.

Parallel edges can be combined together in to single edge by using '+' operator.

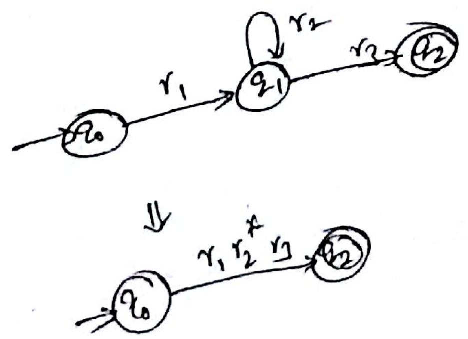



**Step IV:** Apply concatenation operation, b/w the expressions $r_1, r_2 \ldots r_n$ if they are in the following form in a given FA.

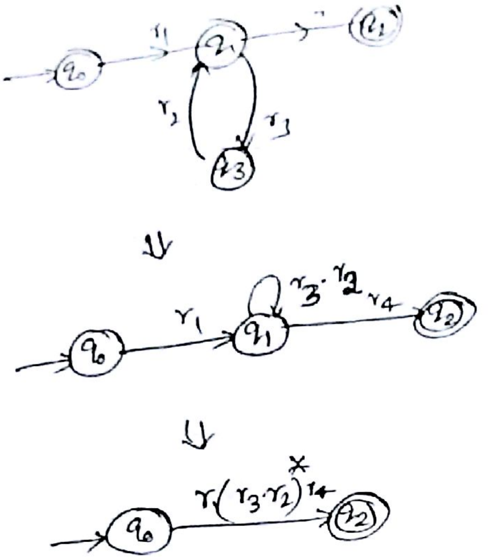
**EX 1:**



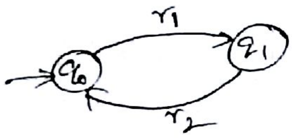

**EX 2:**



3

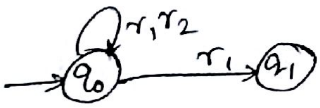


4

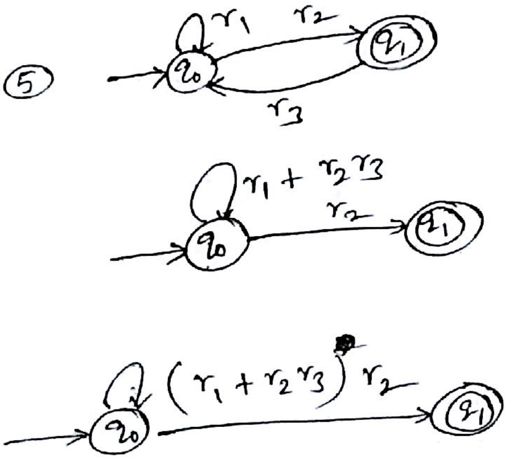


$q_0 \xrightarrow{r_1} q_1 \xrightarrow{r_2} q_0$

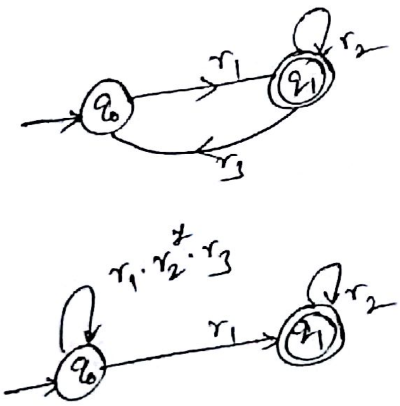
$q_0 \xrightarrow{r_1 r_2} q_0$



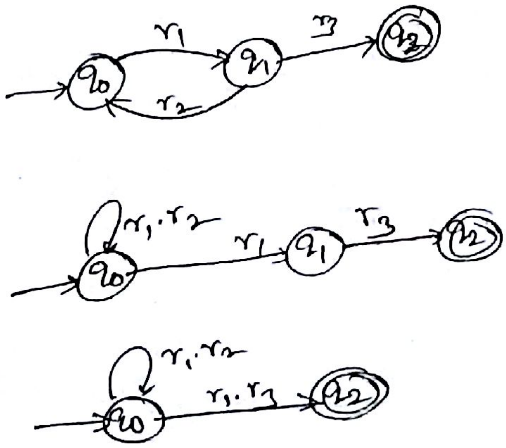
5



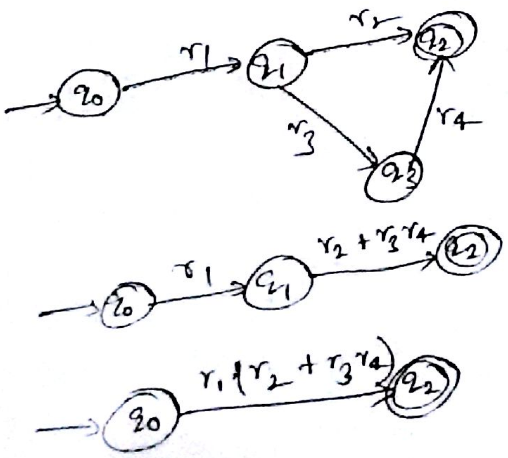
6



7

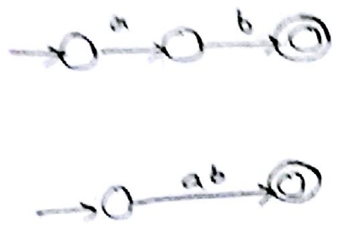


8



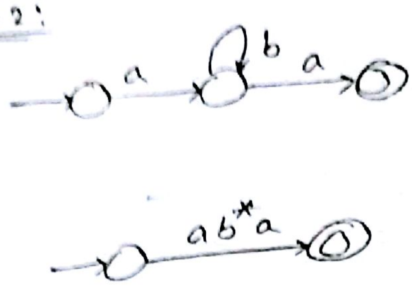
Step 3: Continue the process of state elimination ...... until it contains one initial and one final state.

Ex 1:




$R = ab$

Ex 2:



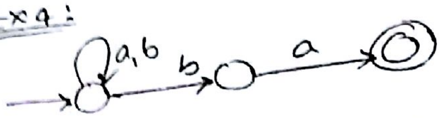

$R = ab^*a$

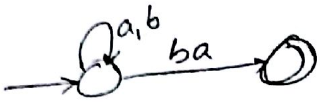
Ex 3:




$R = b(aa)^* b$

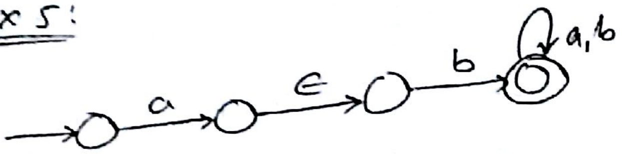
Ex 4:




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

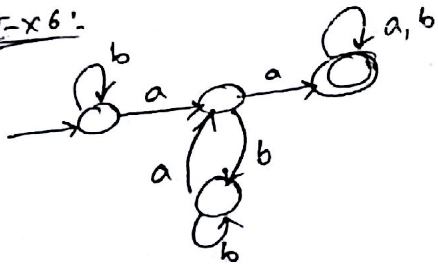
Ex 5:



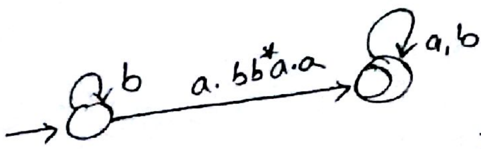

⇓


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

Ex 6:

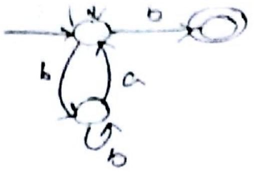


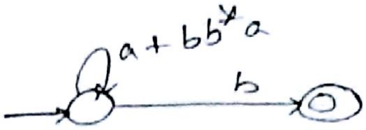

⇓


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

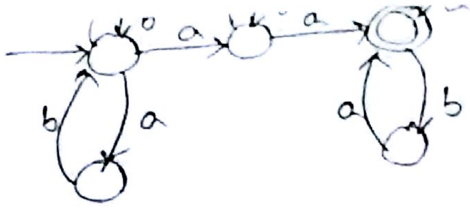
(7)

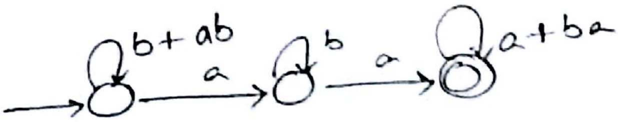


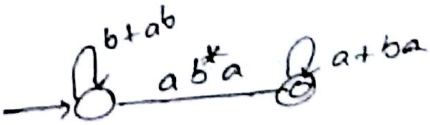




$(a+bb^*a)^*b$

(8)

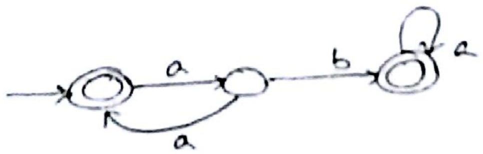


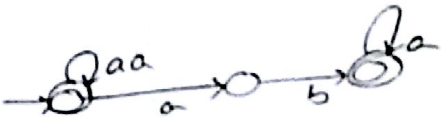



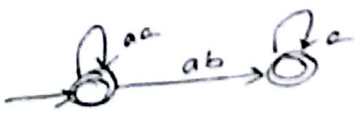




$(b+ab)^*\, ab^*a\, (a+ba)^*$

(9)



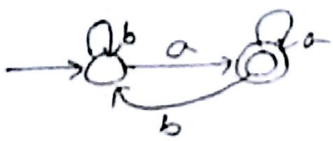



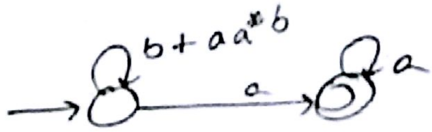




$R = \boxed{(aa)^* + (aa)^* aba^*}$

$(aa)^*(\epsilon + aba^*)$

(10)




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

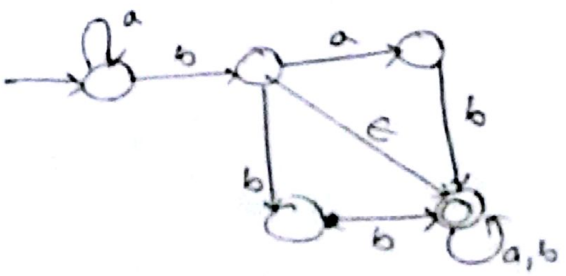
$= (b + a^+b)^* a^+$

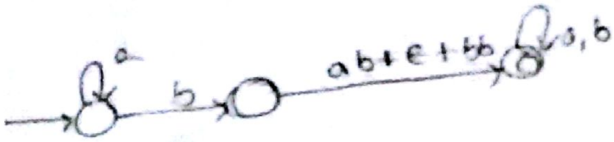
$= (b(\epsilon + a^+))^* a^+$

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

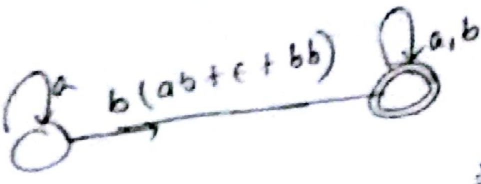
$= \boxed{(ba^*)^* a^+}$

(11)



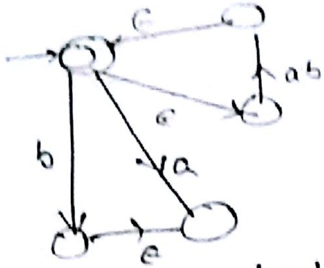




$\Rightarrow$



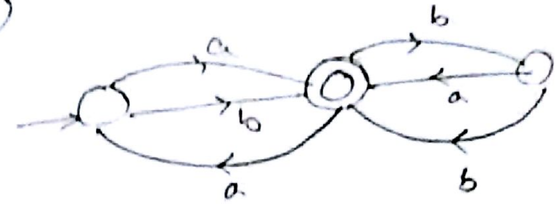

$R = a^*\, b(ab+\epsilon+bb)(a+b)^*$

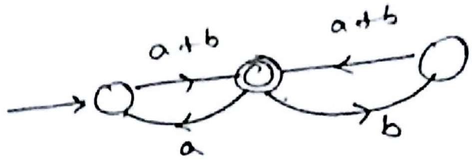
(12)

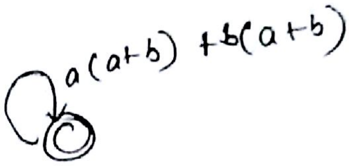




(13)




$R = [a(a+b) + b(a+b)]^*$

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

$= [(a+b)^2]^*$

## Conversion of RE to FA:

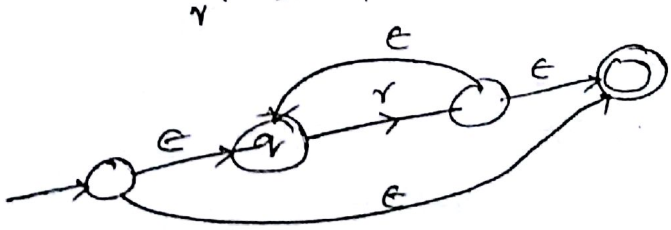
1. Method of Synthesis
2. Method of Decomposition

## Method of Synthesis: $(*, \cdot, +)$

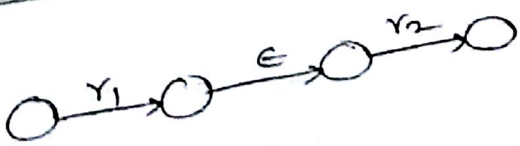
1. Kleen closure: $(r^*)$

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

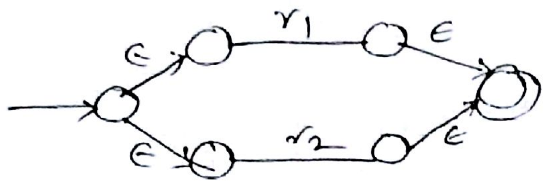
$r^+ = r, rr, rrr \ldots$



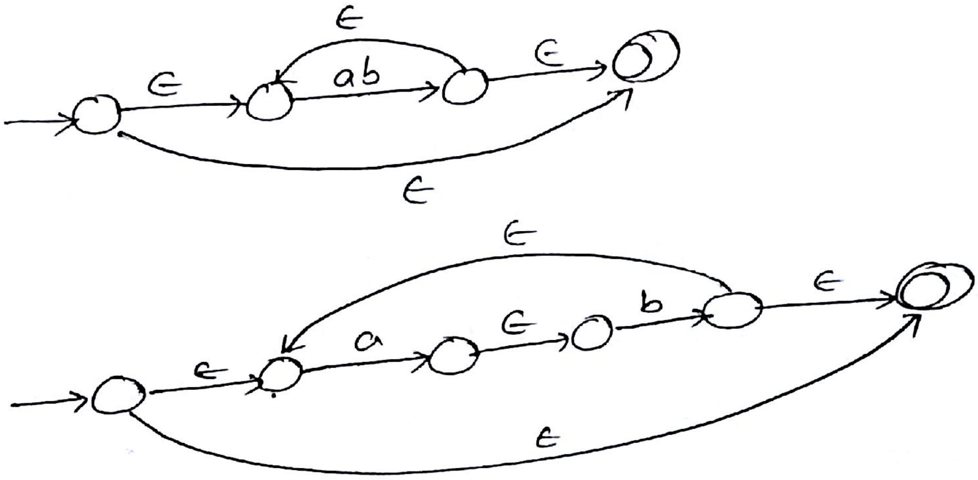

2. Concatenation: $(r_1 \cdot r_2)$



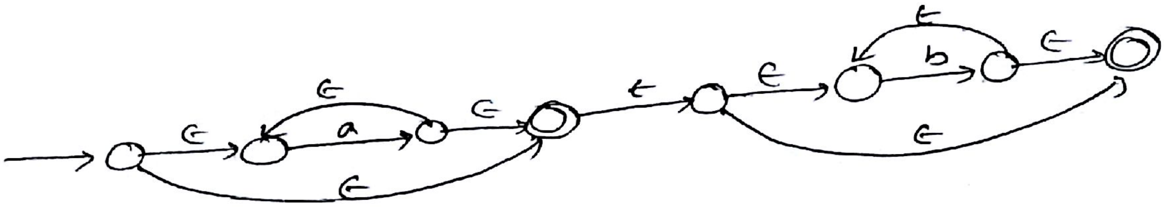
③ **Union** $(r_1 + r_2)$



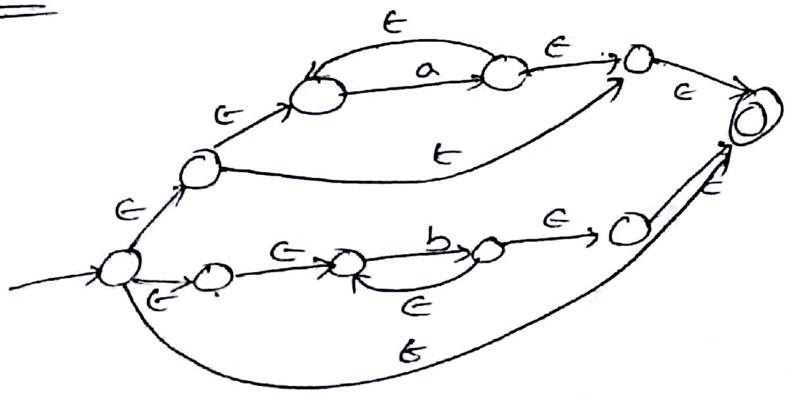

**Ex1:** $r = (ab)^*$



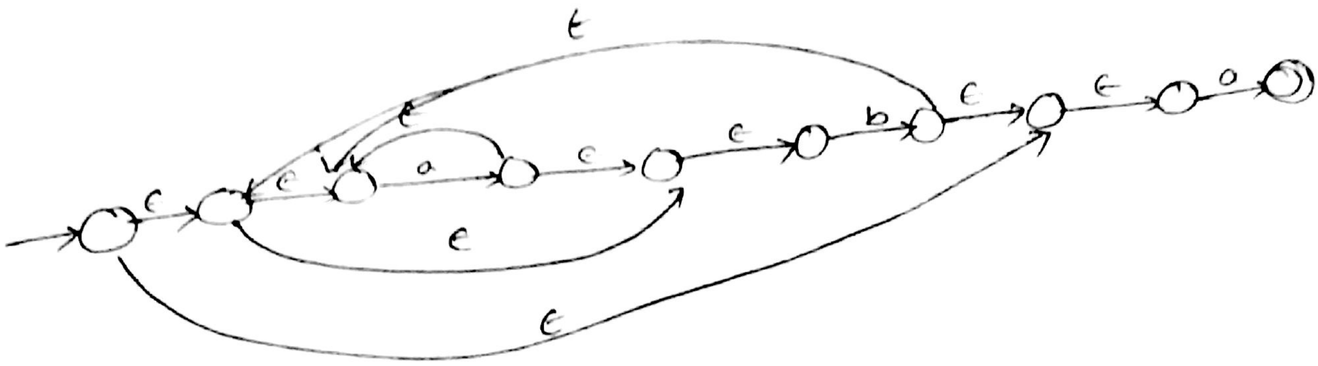

**Ex2:** $r = a^* . b^*$



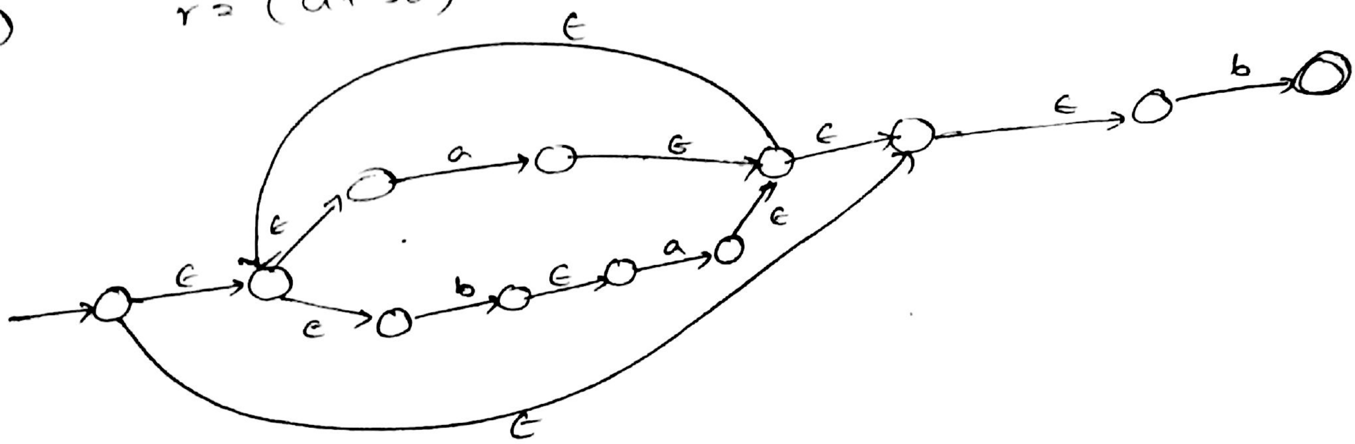

**Ex3:** $r = a^* + b^*$



(4) $r = (a^* b)^* a$




(5) $r = (a + ba)^* b$




<u>Method of decomposition:</u>

(1) This technique is also called as state creation method.

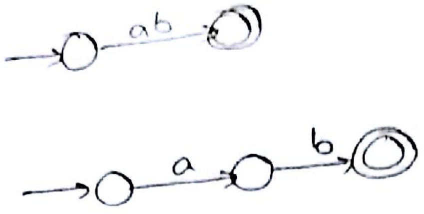
(2) If r is a regular Expression then start the process by NFA with 2 states and 'r' is the edge label.

(3) split the RE into Symbol & create the State.

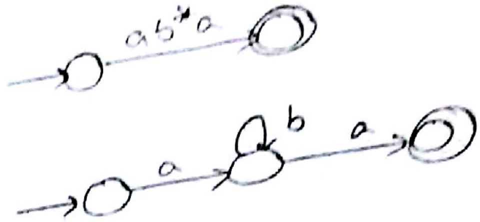
(4) Continue the process of state creation untill entire RE is decomposed in to symbol.

LA

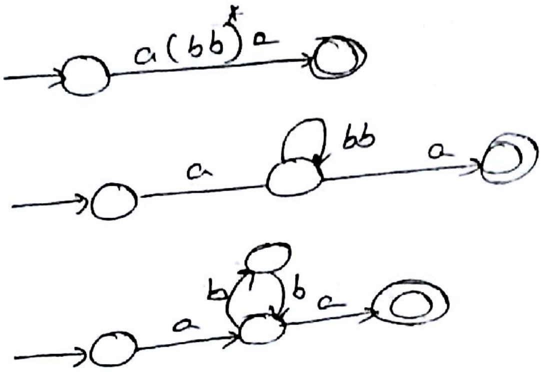
① $r = ab$



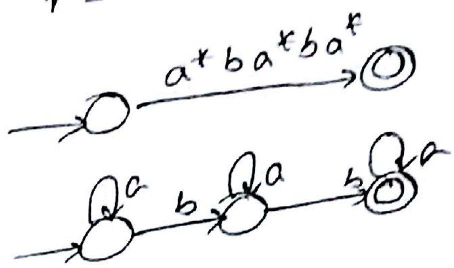

② $r = ab^*a$



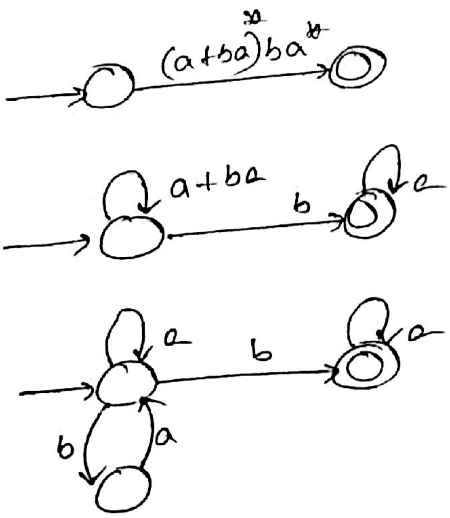

③ $r = a(bb)^*a$



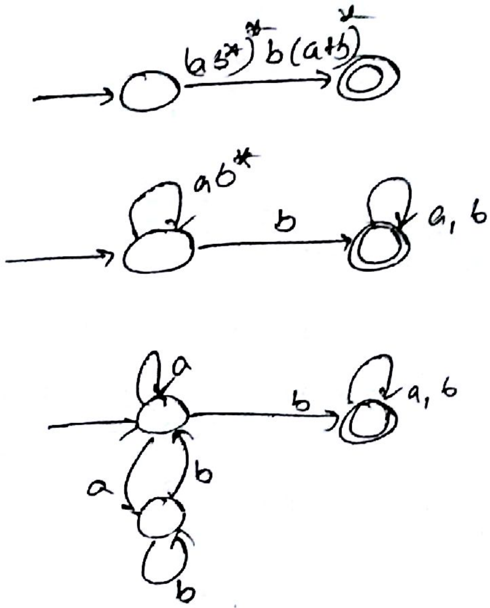

④ $r = a^*ba^*ba^*$



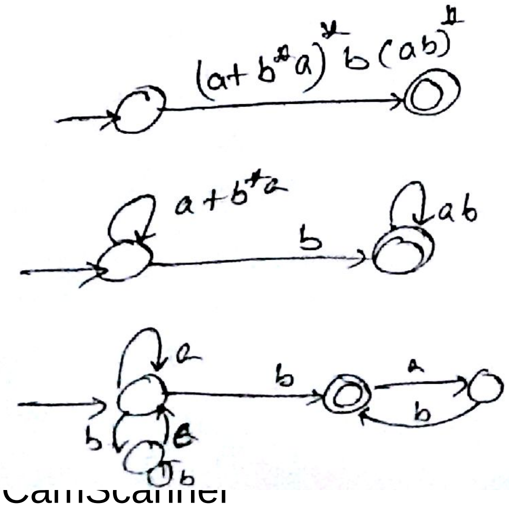

⑤ $r = (a+ba)^*ba^*$



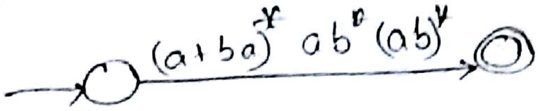

⑥ $r = (ab^*)^*b(a+b)^*$

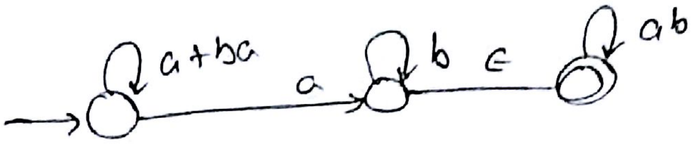


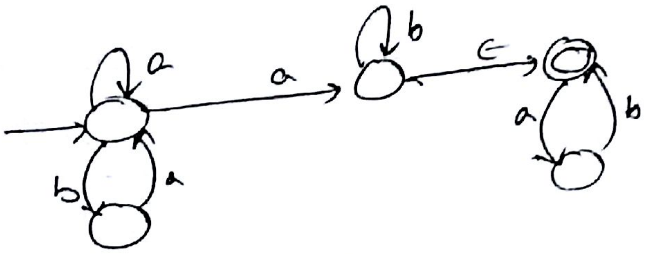

⑦ $r = (a+b^*a)^*b(ab)^*$



(8) $r = (a+ba)^{n} ab^{*} (ab)^{*}$




**Pumping Lemma:** If L is any Regular Language and $z \in L$ such that $|z| \geq n$ then,

$$uv^{i}w \in L \ \forall \ i \geq 0$$

$$z = uvw$$

$$|z| \geq |uv|$$

① Pumping Lemma is used ~~for~~ to prove, some of the Language is NRL.

② For pumping Lemma i/p is NRL and o/p is NRL.

③ Every Regular Language satisfies pumping Lemma property.

④ The Language which does not satisfy pumping Lemma property is not Regular.

**Process of pumping lemma:**

Given that L is a Non-Regular Language.

① Assume that L is regular

② Choose $z \in L \Rightarrow |z| \geq n$ (pumping lemma Const)

(3) Split z into 3 parts

z

u v w

$|uv| \leq |z|$

$|v| \neq 0$

(4) There exists atleast one value for 'i' such that $uv^iw \notin L$, then L is a Non-Regular Language.

Ex1: Prove that $L = \{a^nb^n \mid n \geq 1\}$ is NRL.

Let $z \in L$

$z = a^k b^k$ , $|z| = 2k$

$a^{k-1}$ $a$ $b^k$

u v w

$|uv| \leq |z|$

$|v| \neq 0$

$uv^iw = a^{k-1} . a . b^k$ for $i = 1$

for $i = 2$

$\Rightarrow a^{k-1} . a^2 . b^k$

$\Rightarrow a^{k+1} . b^k \notin L$ $((k+1) > k)$

$\boxed{\therefore\ uv^iw \notin L}$

Ex 2: $L = \{a^mb^n \mid m < n\}$ is NRL

Let $z \in L$

$z = a^k b^{k+1}$

u v w

$a^{k-1}$ $a$ $b^{k+1}$

$|uv| \leq |z|$

$|v| \neq 0$

Scanned by CamScanner

$uv^iw \Rightarrow a^{k-1} . a . b^{k+1} \in L$

for $i=2$

$a^{k-1} . a^2 . b^{k+1}$

$\boxed{a^{k+1} . b^{k+1} \notin L}$

(3) prove that $L = \{a^{n^2} \mid n \geq 1\}$ is NRL

$z \in L$

$z = a^{n^2}$

$a^{n^2-1} \quad a \quad \epsilon$

then $uv^iw = a^{n^2-1} . a^i . \epsilon$

$= a^{n^2-1} . a^2 . \epsilon$ for $i=2$

$= \boxed{a^{n^2+1}} \notin L$

$\therefore L$ is NRL

<u>closure property of R.E:</u>

Regular Language satisfy the closure property w.r.t following operators.

1. Complement
2. kleen closure
3. positive closure
4. Reversal
5. * Prefix operator
6. Concatenation

(7) union

(8) Intersection

(9) Difference

(10) Quotient operator

(11) Homomorphism

(12) Substitution

(13) Inverse Homomorphism

**Quotient operator:**

If $L_1$ & $L_2$ be the two Regular Language then

$\frac{L_1}{L_2} = \{x \mid xy \in L_1, y \in L_2\}$ is also Regular language.

**Ex 1:** $\frac{L_1}{L_2} = \frac{010}{0} = 01$

**Ex 2:** $L_1 = 0^* 1 0^* 1$

$L_2 = 010^*$

$\frac{1010}{10} = 10$

**Ex 3:** $\frac{0110}{01} = \phi$

**Ex 4:** $\frac{101}{101} = \epsilon$

**Ex 5:** $\frac{101}{\epsilon} = 101$

**Subtitution:** Substitution is a mapping from $\Sigma$ to $\Delta$ such that the symbols from $\Sigma$ is replaced by a R.L of another alphabet $\Delta$.

$S: \Sigma \rightarrow \Delta$ such that

$S(a) = L$

$a \downarrow \Sigma$, $L \downarrow$ RL over $\Delta$

**Ex:** $\Sigma = \{a, b\}$ $\Delta = \{0, 1\}$

$S: \Sigma \rightarrow \Delta$

$S(a) = 0^*$

$S(b) = (10)^*$

Note: If L is a RL over the alphabet $\Sigma$ and S is a substitution $\Sigma$, then S(L) is also a RL over the alphabet $\Delta$.

$$S(\phi) = \phi$$
$$S(x^*) = (S(x))^*$$
$$S(\epsilon) = \epsilon$$
$$S(x_1 + x_2 + \cdots + x_n) = S(x_1) + S(x_2) + \cdots + S(x_n)$$
$$S(x_1 \cdot x_2 \cdots x_n) = S(x_1) \cdot S(x_2) \cdots S(x_n)$$

① $\Sigma = \{0,1\}$ $\Delta = \{a,b\}$

$S(0) = a^*$

$S(1) = b^* a$

$L = \{01, 10\}$

$S(L) = \{S(01), S(10)\}$

$= \{S(0) \cdot S(1), S(1) \cdot S(0)\}$

$= a^* b^* a + b^* a a^*$

② $L = 0^* 1$

$S(L) = S(0^* 1)$

$= S(0^*) \cdot S(1)$

$= (S(0))^* \cdot S(1)$

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

$= a^* b^* a$

Homomorphism: Homomorphism is a kind of substitution from $\Sigma$ to $\Delta$, where the symbols of $\Sigma$ is replaced by single string of another alphabet.

$h: \Sigma \to \Delta$

$h(a) = x$

$a \downarrow \Sigma$, $x \downarrow \Delta^*$

Ex: $\Sigma = \{a,b\}$, $\Delta = \{0,1\}$

$h(a) = 01$ $h(b) = 100$

Note: If L is a Regular Language over the alphabet $\Sigma$ and h is a homomorphism from $\Sigma$ to $\Delta$, then h(L) is also Regular Language over alphabet $\Delta$.

Ex 1: $\Sigma = \{0,1\}$ $\Delta = \{a,b\}$

$h(0) = a$

$h(1) = ab$

$L = \{01, 10, 11\}$

$h(L) = \{h(01), h(10), h(11)\}$

$= \{aab, aba, abab\}$

Ex 2:

$L = 0^* 1$

$h(L) = h(0^* 1)$

$= (h(0))^* h(1)$

$= a^* ab$

$= a^+ b$

Ex 3:

$L = (10^*)^*$

$h(L) = (h(10^*))^*$

$= [h(1) \cdot (h(0))^*]^*$

$= (ab \cdot a^*)^*$

Inverse Homomorphism:

- Let $h: \Sigma \to \Delta$ is a homomorphism and $L$ is a RL over the alphabet $\Delta$, then $h^{-1}(L)$ is the set of all the strings whose image exist.

If $L$ is a Regular Language over the alphabet $\Delta$, then $h^{-1}(L)$ is also RL over the alphabet $\Sigma$.

Ex 1: $\Sigma = \{0,1\}$ $\Delta = \{a,b\}$

$h(0) = a$

$h(1) = ba$

(i) $L = \{ab, a, baa\}$
$\phi$ $0$ $10$

$h^{-1}(L) = \{0, 10\}$

(ii) $L = \{aba, baa, aaab\}$
$01$ $10$ $\phi$

$h^{-1}(L) = \{01, 10\}$

(iii) $L = a^* b$

$h^{-1}(L) = \phi$

(iv) $L = a^* ba$

$h^{-1}(L) = 0^* 1$

(v) $L = (ab)^* a$

$= a(ba)^*$

$= 0(1)^* = 01^*$