Regular expression :-

The languages accepted by finite Automata are easily described by simple expressions called Regular expressions. Regular expressions are useful for representing certain sets of strings in a algebric fashion.

operators of regular expression

(i) + : union
(ii) . : concatenation

$\}$ both are binary operator.

(iii) + : Positive/transitive closure
(iv) * : Kleene closure

$\}$ both are unary operator.

ex. $(0+1)$ is a regular expression, which is union of 0 and 1

(0.1) is a regular expression which is concatenation of 0 and 1

$(0+1)^+$ is a regular expression which is Positive closure of $(0+1)$

$(0+1)^*$ is a regular expression which is Kleene closure of $(0+1)$

order of Precedence (in increasing order)

$+ < \cdot < \left.\begin{matrix}+\\ *\end{matrix}\right\}$ same Precedence

$\uparrow$ union, $\uparrow$ concatenation

<u>Regular set</u> : A regular set is the set represented by a regular expression.

or

Set of strings, generated by regular expression is called regular set.

ex.

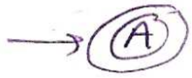
Regular expression		Regular set
$0+1.0$	=	$\{0, 10\}$
$(0+1).(1+0)$	=	$\{00, 01, 10, 11\}$
$(0+1).\varepsilon.0$	=	$\{00, 10\}$
$(0+\varepsilon).1$	=	$\{01, 1\}$
0	=	$\{0\}$

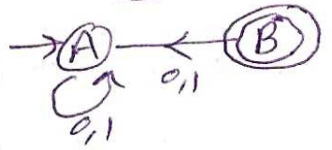
Note: * ε is a regular expression which generat reg. set $\{\varepsilon\}$

* ϕ is a reg. expression which generate reg. set $\{\}$

here $|\varepsilon| = 1$, $|\phi| = 0$

* every set generated by any regular expression, there exist some finite state m/c to accept it.

* DFA for R.E. ε is 

* DFA for R.E ϕ is 

<u>Kleen closure (*)</u> : Let r is a regular expression which represent a regular set L

$$r\ (R.E.) \Rightarrow L\ (R.S.)$$

then r^* is also regular expression which represent a regular set L^*

where $L^* = \bigcup_{i=0}^{\infty} L^i$ here $L^0 = \{\varepsilon\}$
$L^K = L^{K-1} . L^1$

<u>Positive closure (+)</u> : Let r is a regular expression ~~where~~ which represent a regular set L
then r^+ is also regular expression which represent a set L^+.

where $L^+ = \bigcup_{i=1}^{\infty} L^i$ $L^K = L^{K-1} . L^1$

ex. Let $r = 0 \Rightarrow L = \{0\}$

then $r^* = L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \ldots$

$L^0 = \{\varepsilon\}$
$L^1 = \{0\}$
$L^2 = L^1 . L^1 = \{0\}.\{0\} = \{00\}$
$L^3 = L^2 . L^1 = \{00\}.\{0\} = 000$
$\vdots$

$L^* = \{\varepsilon, 0, 00, 000, \ldots\}$

ex: Let $r = (0+1) \Rightarrow L = \{0, 1\}$

$r^* \Rightarrow L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \cup \ldots$

$L^0 = \{\varepsilon\}$. $L^1 = \{0, 1\}$, $L^2 = \{0,1\}.\{0,1\} = \{00, 01, 10, 11\}$

$L^3 = \{00, 01, 10, 11\}.\{0,1\} = \{000, 001, 010, 011, 100, 101, 110, 111\}$

$(0+1)^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, \ldots\}$

$(0+1)^+ = (0+1)^* - \{\varepsilon\}$

or

$r^* = r^+ \cup \{\varepsilon\}$

ex. $(0+01)^+ = ?$

$= \{\underbrace{0, 01}_{L^1}, \underbrace{001, 00, 010, 01}_{L^2}; \underbrace{000, 0001, 0010, 00101, 0100, 0101, 01001, 01010; 010101}_{L^3}, \ldots\ldots\}$

= all possible combination of 0 & 01.

Q find all 4 length string in $(0+01)^+$

$= \{0000, 0001, 0010, 0100, 0101\}$

Q find a set of all 4 length string in $(a+b)\,c^*d^+$

$= \{acdd, accd, addd, bccd, bcdd, bddd\}$

Q which strings belong the R.E. $\underline{(a+b)^* b^+ c^* d^+ (e+f)^+}$

(i) abdf ✓ (ii) bbcde ✓ (iii) abcf ✗

(iv) accdf ✗ (v) abcccdddee ✓

Algebraic Properties of Regular expression

(*) Let L, M, N are regular expression then

(i) $L + M = M + L$ (commutative law for union)

(ii) $(L + M) + N = L + (M + N)$ (associative ~~commutative~~ law for union)

(iii) $(LM)N = L(MN)$ (associative law for concatenation)

(iv) Regular expressions does not follow the commutative Property for concatenation.

(v) $\phi + L = L + \phi = L$ (ϕ is the identity for union)

(vi) $\varepsilon L = L\varepsilon = L$ (ε is the identity for concatenation)

(vii) $\phi L = L\phi = \phi$ (ϕ is the annihilator for concatenation)

(viii) $L(M+N) = LM + LN$ (Left distributive law ~~for~~ of concatenation over union)

$(M + N)L = ML + NL$ (Right " " " " " " " ")

(ix) $L + L = L$ (idempotent Law)

(x) $(L^*)^* = L^*$

(xi) $(\phi)^* = \varepsilon$

(xii) $\varepsilon^* = \varepsilon$

(xiii) $L^+ = LL^*$

(xiv) $L^* = L^+ + \varepsilon$

Q ~~f~~ write regular exp. for the fallowing language.

(I) all string ending in 01

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

(II) all string not ending in 01 $= (0+1)^*(11+00+10)$

(iii) contain 101 as substring $= (0+1)^* 101 (0+1)^*$

(IV) whose 4^{Th} symbol from RHS is 1

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

(V) ~~not contain 100~~
even No. of a fallowed by odd No. of b, $L = \{a^{2n} b^{2m+1}, n,m \geq$

$$(aa)^*.(bb)^*.b$$

(VI) no Pair of consecutive 0

$$(1+01)^*(0+\varepsilon)$$

(VII) $L = \{a^n b^m : n+m \text{ is even}\}$

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

(VIII) $L = \{w : |w| \bmod 3 = 0 \quad w \in (a,b)^*\}$

$$= ((a+b)3)^*$$

(IX)

$L = \{w \in (a,b)^* : n_a(w) \bmod 3 = 0\}$

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

Identities of Regular Expression:

(I) $\phi + R = R + \phi = R$

(ii) $\phi R = R\phi = \phi$

(iii) $\varepsilon R = R\varepsilon = R$

(IV) $\varepsilon^* = \varepsilon$, $\phi^* = \varepsilon$

(V) $R + R = R$

(VI) $R^* R^* = R^*$

(VII) $R^* R = R R^*$

(VIII) $(R^*)^* = R^*$

(IX) $\varepsilon + RR^* = \varepsilon + R^* R = R^*$

(X) $(PQ)^* P = P(QP)^*$

(XI) $(P+Q)^* = (P^* Q^*)^* = (P^* + Q^*)^*$

(XII) $(P+Q)R = PR + QR$, $R(P+Q) = RP + RQ$

Q. Prove that

$$(1+100^*) + (1+100^*)(0+10^*)(0+10^*)^* = 10^*(0+10^*)^*$$

$LHS = (1+100^*) + (1+100^*)(0+10^*)(0+10^*)^*$ by using I_{12}

$= (1+100^*)(\varepsilon + (0+10^*)(0+10^*)^*)$

$= (1+100^*)(0+10^*)^*$ by using I_9

$= 1(\varepsilon + 00^*)(0+10^*)^*$ by using I_{12}

$= 1(0^*)(0+10^*)^*$ by using I_9

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

$=$ R.H.S.

Q Prove that

$$10 + (1010)^*(\varepsilon + (1010)^*) = 10 + (1010)^*$$

$$\begin{aligned} LHS &= 10 + (1010)^*(\varepsilon + (1010)^*) \\ &= 10 + (1010)^*\varepsilon + (1010)^*(1010)^* \\ &= 10 + (1010)^* + (1010)^* \\ &= 10 + (1010)^* \end{aligned}$$

Q.

Conversion of Regular expression to Finite Automata.

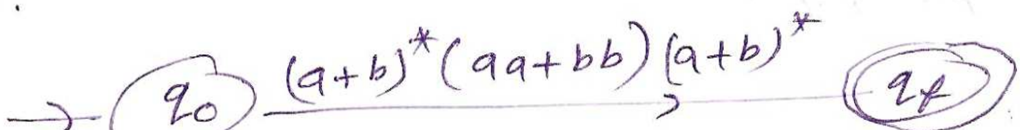
(i) R.E. to NFA with ε

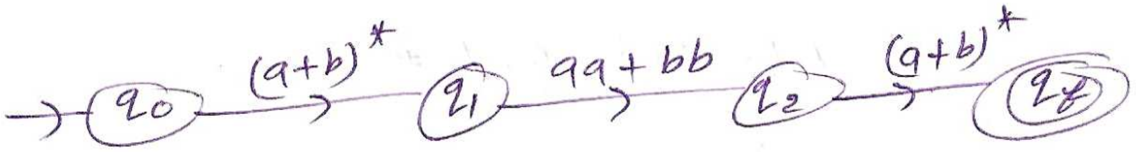
(ii) R.E. to DFA

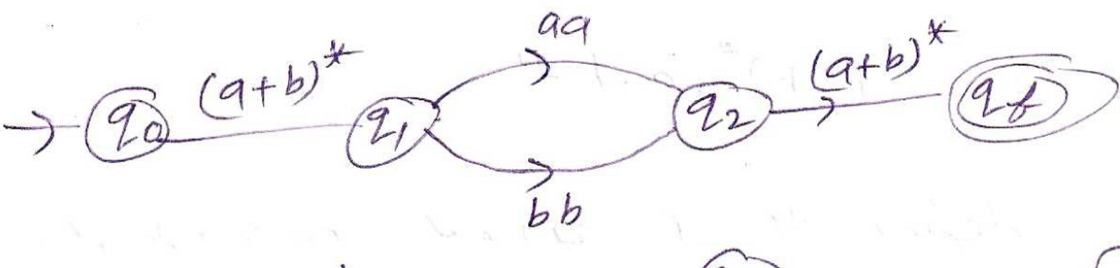
(i) R.E. to NFA with ε

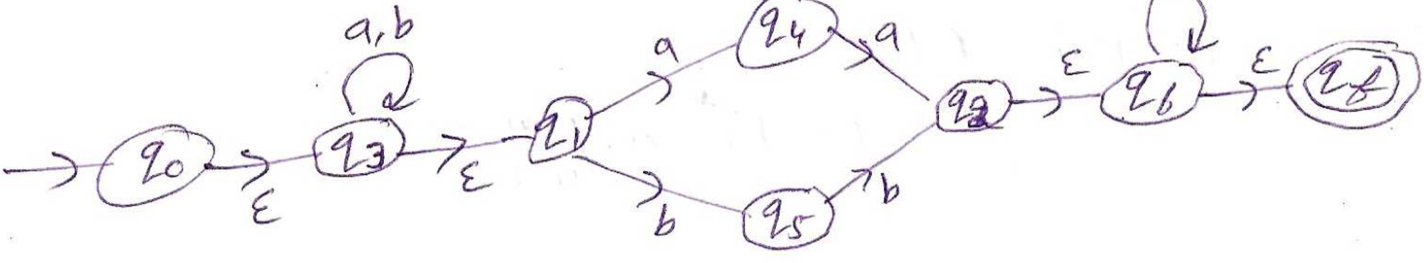
Q. $(a+b)^*(aa+bb)(a+b)^*$

Solution:







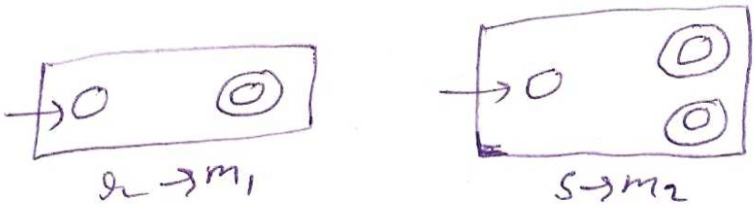


Let r and s be regular expression then

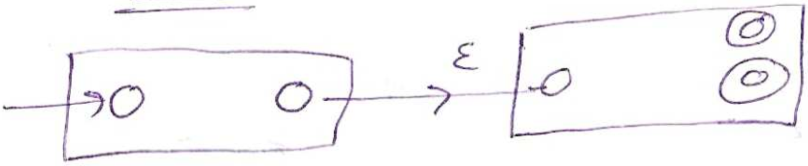
(i) $r.s$ and $s.r$ are also regular expression.

(ii) $r+s$ and $s+r$ are also regular expression.

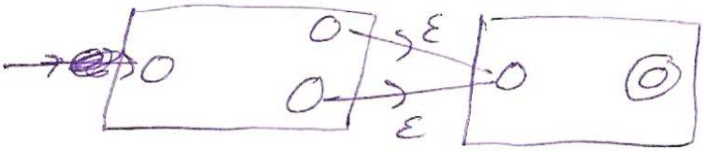
Let



then $r.s$



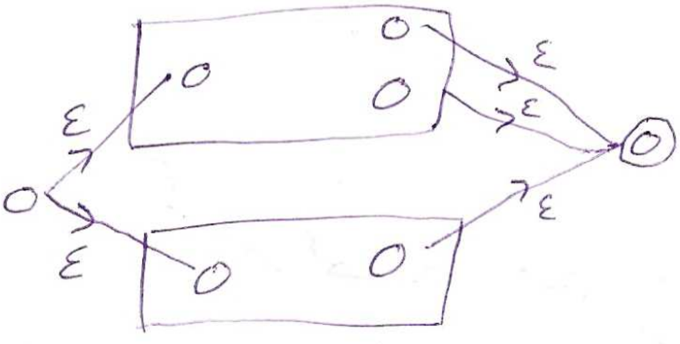
$s.r$



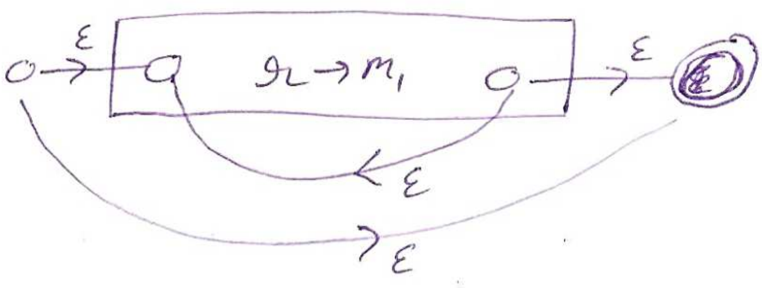
$r+s$

or

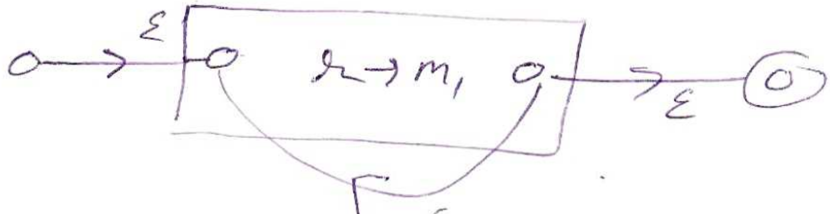
$s+r$



r^*

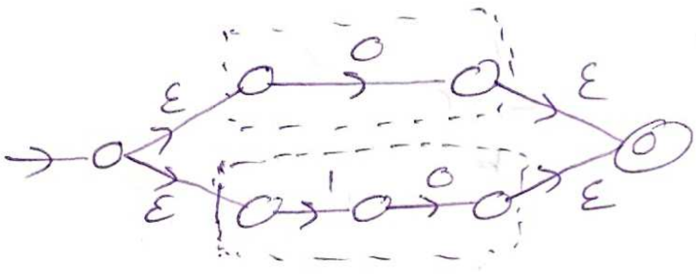


r^+

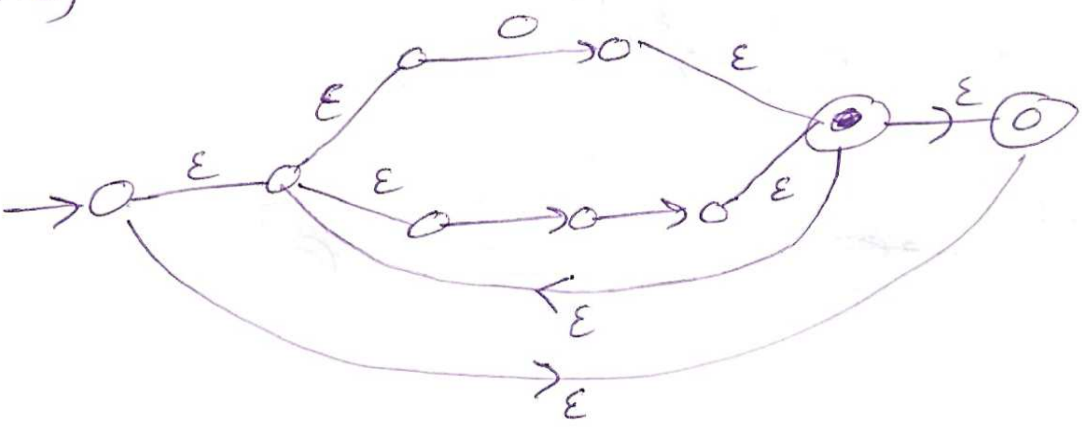


er

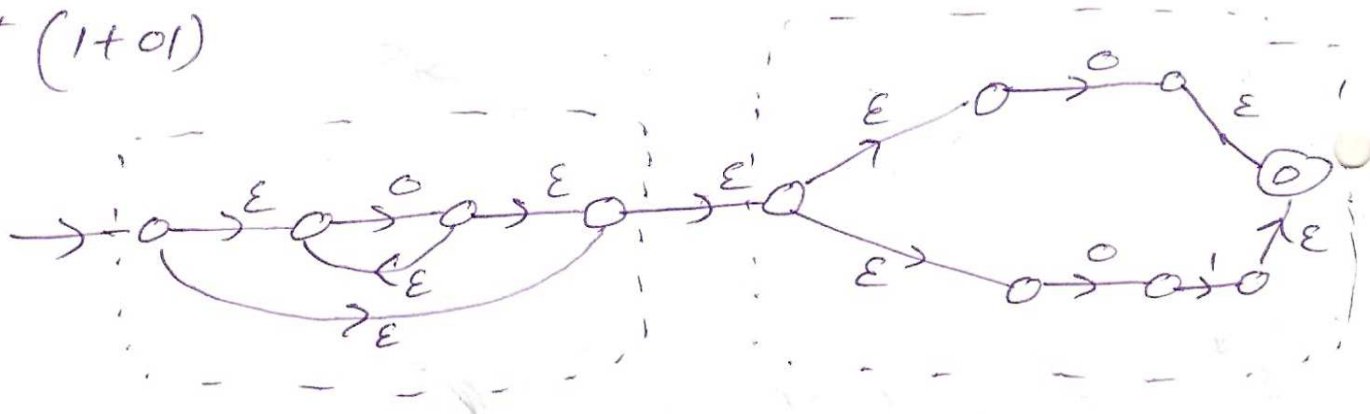
NFA for R.E. $0+1.0$



$(0+1.0)^*$



$0^*(1+01)$



(II) R.E. to DFA :

First Pos : The set of the positions of the first symbals of the strings generated by the R.E.

Fallow Pos(P) : set of the Positions of the Next symbols just after the Position P, in the strings generated by R.E.

Step 1 : concatinate an operator # so all the strings of R.E. ends with #.

ex. $(a+b)^*.a.b$

$\Rightarrow (a+b)^*.a.b\#$

Step 2 : define the Posiations for each operator used in R.E.

$$\underset{1}{(a}+\underset{2}{b)^*}.\underset{3}{a}.\underset{4}{b}.\underset{5}{\#}$$

Step 3 : find First Pos and Fallow Pos set.

~~Find~~ First Pos = {1, 2, 3}

Follow Pos (1) = {1, 2, 3}

Fallow Pos (2) = {1, 2, 3}

Fallow Pos (3) = {4}

Fallow Pos (4) = {5}

fallow Pos (5) = ϕ

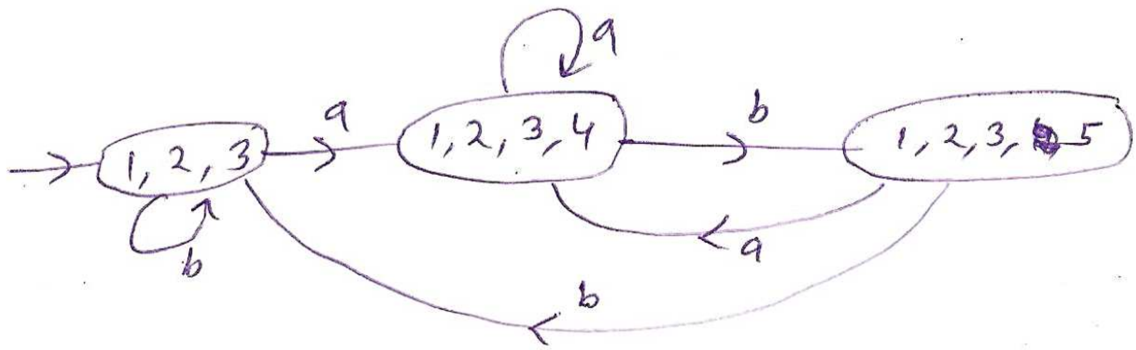
step4: Now make DFA, in which initial state is First pos

step 5: show the transection for each alphabet from initial state.

* in this initial state a belongs the position 1 and 3 so after a read we reach the state which will generate by union of fallow pos of 1 and 3.

  and same for b.

* show the all transection for all newly generate state.



step 6: make all those states final, which have the position of #.