Regular Expression

The languages accepted by finite automata are easily described by simple expressions called regular expressions

Let Σ denotes the character set a, b c... z. The regular expression is defined by the following rules.

1) Every character or terminal or sigma is a regular expression.

2) Null string ϵ is a regular expression.

3) If R_1 and R_2 are regular expression then $R_1 . R_2$ is also a regular expression.

eg: If $R_1 = ab$

$R_2 = bc$

Then $R_1 . R_2 = abbc$ is also a regular expression.

4) The union of 2 regular expressions R_1 & R_2 written $R_1 + R_2$ or R_1 / R_2 is also a regular expression

If $R_1 = ab$ and $R_2 = bc$

$R_1 + R_2 = ab + bc$

5) The iteration of a closure of a regular expression are written as R^* is also a regular expression.

eg: $R = ab$

$R^* = (ab)^*$

6) If R is a regular expression then R^+ is also a regular expression.

7) If R is a regular expression. The (R) also regular expression.

8) Any combination of the preceding rule is also a regular expression.

Regular set

A regular expression generates a set of strings, Any set containing regular expression are represented by it is called a regular set.

If $\Sigma = \{a, b\}$, then ~~RE~~ RE

1) a denotes the set $\{a\}$

2) $a+b$ denotes the set $\{a, b\}$

3) $a \cdot b$ " $\{a, b\}$

4) a^+ " $\{a\varepsilon, a, aa, aaa, \ldots\}$

5) $(a+b)^+$ " $\{a, b\}^+ = \{\varepsilon, ~~ab~~, abab$

$= \{\varepsilon, a, b, ab, ab, aab, abb, aabab, \ldots\}$

Q. Describe the following sets of strings by regular expression.

a) { ~~1~~ 101 }

b) { ababa }

c) { 01, 10 }

d) { Λ, ab }

e) { abb, a, b, bba }

f) { Λ, 0, 00, 000, ... }

g) { 1, 11, 111, ... }

→ (a) { 101 }

= 101

(b) { ababa }

= ababa

(c) { 01, 10 }

= 01 + 10

(d) { Λ, ab }

= Λ + ab

(e) { abb, a, b, bba }

= abb + a + b + bba

(f) { Λ, 0, 00, 000, ... }

= 0^*

(9) {1, 11, 111, ... }
$= 1^+$

Q. Describe the following sets by regular expressions:

(1) L_1 = set of all strings of 0's & 1's ending in 00

$\to$ $(0+1)^*00$

(2) L_2 = set of all strings of 0's & 1's beginning with zero and ending with 1.

$\to$ $0(0+1)^*1$

(3) L_3 = set of all strings of 0's and 1's with at least 2 consecutive zeros

$\to$ $(0+1)^*00(0+1)^*$

(4) L = set of all strings of 0's and 1's begining with 1 and not having 2 consecutive 0's

{1, 11, 111, 10, 1010, 101 ... }

(5) L =

$\to$ ~~1(0+1)*~~ ~~1(01)*~~

$(1+10)^+$

($(1+01)^+$ not possible)

(5) L = any no. of zeros followed by any no of ones followed by any no. of 2s.

→ ~~0+1+2~~ $0^* 1^* 2^*$

(6) L = every string contains alternating zeros and ones.

01010, 010, 1010, 1010

→ $\cancel{0^* 1^* 0^* 1^*}$ $\cancel{(01)^* (01)^*}$

$\cancel{0(10)^* + 0 + 0 + (01)^* 0 +}$ $+ 0(10)^* + 0(10)^*1$

$1(01)^* + 1(01)^* 0 + 0(10)^* + 0(10)^* 1$

(7) L = every strings begins with 00 and ends with 11

→ $00(0+1)^* 11$

(8) L = Every string in $\Sigma = \{a, b, c\}$ contains a substring ccc

→ $(a+b+c)^* ccc (a+b+c)^* +$
$ccc(a+b+c)^* + (a+b+c)^* ccc$

(9) Wr
L =
⇒

(10) Wr
lan
→

Q.1

Q-2)

(9) Write regular expression for language
$L = \{a^n b^m \mid n \geq 4, m \leq 3\}$

$\Rightarrow$ $\cancel{a^4 (a^*) b + b^2 + b^3} +$
$+ \cancel{a^4(a^*)b + a^4(a)^* b^2 + a^4(a)^* b^3}$.

$a^4 a^* (b^3 + b^2 + b^1 + \varepsilon)$ ✓

(10) Write regular expression for language $L = \{a^n b^m \mid (n+m) \text{ is even}\}$

$\rightarrow$ $\cancel{a^2 b^2 (a+b)^* (a^2 b^2 + a^3 b)}$

$\cancel{(a^2 b^2 + ab)^*}$

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

Q.1 Write regular expression over the alphabet $\Sigma = \{a, b, c\}$ contains atleast 1 a & atleast 1 b.

Q.2) Write regular expression for the set of strings of 0's and 1's whose 10th symbol from the ~~of~~ right end is one.

Q.3) Write the regular expression for the set of strings for equal no of 0's & 1's such that in every prefix the no. of zero, differ from no. of ~~a~~ 1's at most one.

Q4) Write regular expression over alphabet $\{a, b\}$ for the set of strings with even of no. of a's followed by odd no. of b's i.e for the language $L = \{a^{2n} b^{2m+1}, n \geq 0, m \geq 0\}$

→ (1) $(a+b+c)^* a (a+b+c)^* b (a+b+c)^* +$
$(a+b+c)^* b (a+b+c)^* a (a+b+c)^*$

(2) $(0+1)^* 1 (0+1)^9$

0101010 ✓
101010 ✓
1001 ✓
1100 ✗

(3) $(01+10)^*$

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

Q5) Write regular expression for the language $L = \{a b^n w : n \geq 3, w \in (a, b)^+\}$

Q6) Write
langua

Q7) L =

(5)

(6)

(7)

La

ex

1) Con
ex

2)

Q6) write regular expression for the language $\{ w : |w| \bmod 3 = 0 ;\ w \in (a,b)^* \}$

Q7) $L = \{ w \in (a,b)^*,\ n_a(w) \bmod 3 = 0 \}$.

→ (5) $ab^3 b^* (a+b)^+$

(6) $\cancel{ab^3(ab^3)}\ [(a+b)^3]^*$

(7) $\cancel{(a.3)^* b^*}\ (b^* a b^* a b^* a b^*)^*$

Languages associated with regular expression:

1) Constants, ϵ and ϕ are regular expression denoting the languages

$L(\epsilon) = \{\epsilon\}$

$L(\phi) = \phi$

2) If 'a' is any symbol, then 'a' is a regular expression. This expression denotes the language $\{a\}$.

$L(a) = \{a\}$

3) If ~~R1~~ ~~R2~~ R_1 & R_2 are regular expressions then, $R_1 + R_2$ is a regular expression denoting the union of $L(R_1)$ & $L(R_2)$.

$\cancel{L(R_1) + L(R_2) = L(R_1) \cup L(B_2)}$

or

$L(R_1 + R_2) = L(R_1) \cup L(R_2)$

4) If R_1 & R_2 are 2 regular expression, then $R_1 R_2$ is a regular expression denoting the concatenation of $L(R_1)$ & ~~R~~ $L(R_2)$

5) If R is a regular expression, then R^* is also a regular expression denoting the closure ~~L(R*)~~ $L(R^*)$

$L(R^*) = (L(R))^*$

6) If R is a regular expression, then (R) ~~is~~ is also a regular expe.

$L((R)) = L(R)$

Q1) Find $L(a^* (a+b))$

→ $L(\underbrace{a^*}_{R_1} * \underbrace{(a+b)}_{R_2})$

$L(a^*) \cdot L(a+b) = L(a^*) \cdot L(a) \cup L(b)$

$= \{\varepsilon, a, aa, aaa, \ldots\} \cdot \{a\} \cup \{b\}$

= {ε

=

Q2) L((a

→ L(

=

Pre

op

1)

2)

3)

4)

Id

I

$= \{\varepsilon, a, aa, aaa\} \cdot \{a, b\}$ $\quad \varepsilon a = a$, $\varepsilon b = b$

$= \{a, b, aa, ab, aaa, aab, \ldots\}$

Q2) $h((a+b)^* (a+bb))$

$\Rightarrow h(a+b)^* \cdot h(a+bb)$

$= h(a+b)^* \cdot h(a) \cup h(bb)$

$= \cancel{h(a+b)^*}\ h(a+b)^* \cdot h\{a, bb\}$

$= \{\varepsilon, a, b, aa, ab, ba, bb \ldots\}\{a, bb\}$

$= \{a, bb, aa, abb, ba, bbb\}$

Precedance of Regular expression operators

1) ( ) - highest
2) $*$
3) concantenation of dot (.) operator
4) Union (+ operators)

Idententies of regular expression.

$I_1 : \phi + R = R$

$I_2 : \phi R = R\phi = \phi$

I_3 : $\Lambda R = R\Lambda = R$

I_4 : $\Lambda^* = \Lambda$ and $\phi^* = \Lambda$

I_5 : $R + R = R$

I_6 : $R^* R^* = R^*$

I_7 : $RR^* = R^* R$

I_8 : $(R^*)^* = R^*$

I_9 : $\Lambda + RR^* = R^* = \Lambda + R^* R$

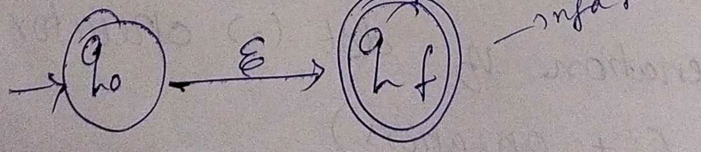
I_{10} : $(PQ)^* P = P(QP)^*$

I_{11} : $(P+Q)^* = (P^* Q^*)^* = (P^* + Q^*)^*$

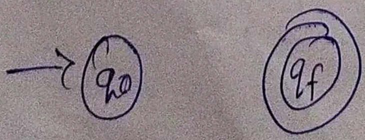
I_{12} : $(P+Q)R = PR + QR$ and $R(P+Q) = RP + RQ$.

NFA with ε moves and Regular expressions

1) For ε, construct NFA



$R = \phi$



2) For a,

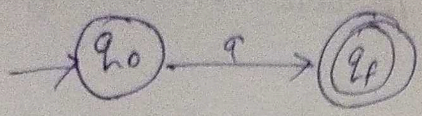
3) Suppose
for the
NFA

(a)

b) NFA

c) N

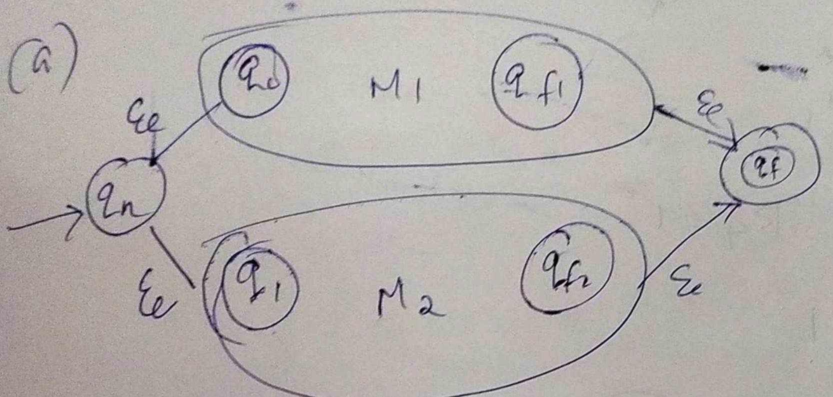
2) For a,



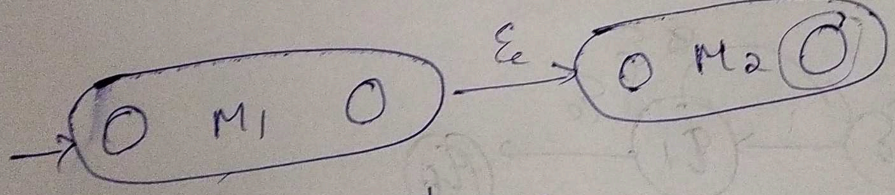
3) Suppose M_1 & M_2 are the NFAs for the regular expression R_1 and R_2.

NFA for union.

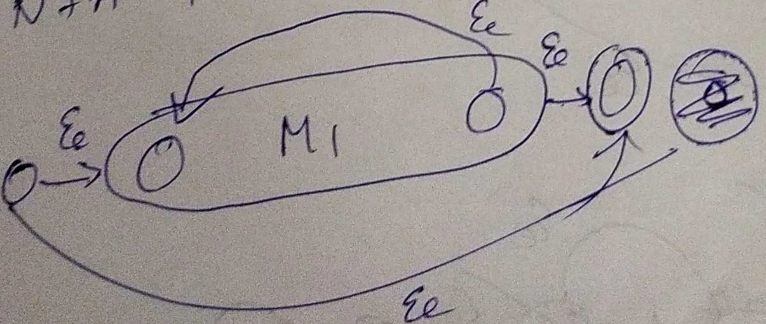
(a)



b) NFA for concatenation,



c) NFA for closure



0 or more occurance.

~~3) for~~
(d) for (R), we use m1 itself as the NFA

Q Construct an ~~A~~ NFA with ϵ transition for the regular expression 01^*+1

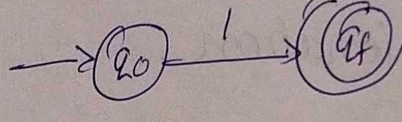
→ $\underbrace{01^*}_{R_1} + \underbrace{1}_{R_2}$

$R_1 = \underbrace{0}_{R_3}\underbrace{1^*}_{R_4}$

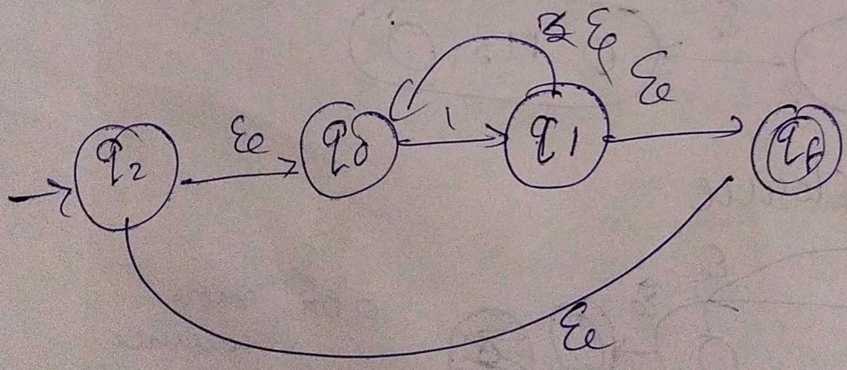
$R_2 = 1$

$R_3 = 0$ $R_4 = 1^*$

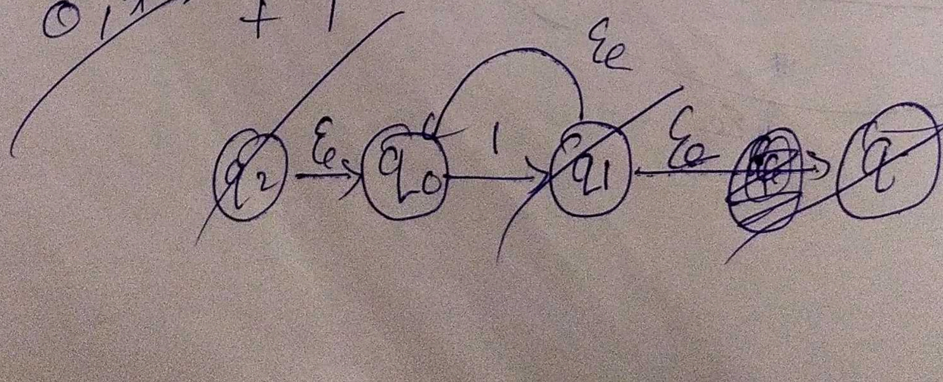
NFA for 1



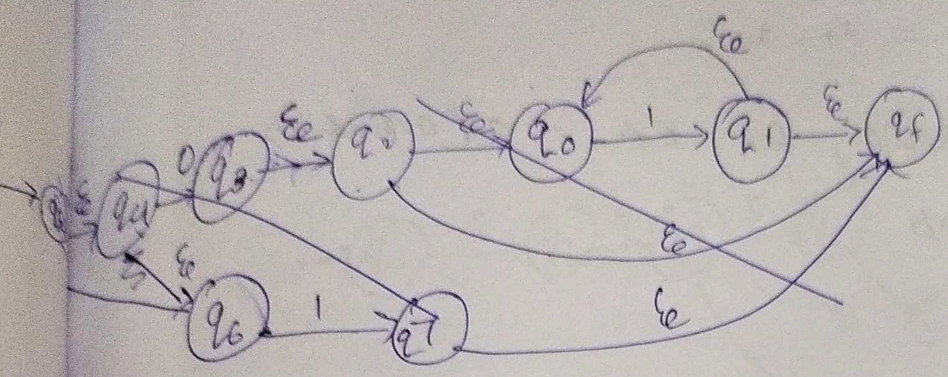
NFA for 1^*

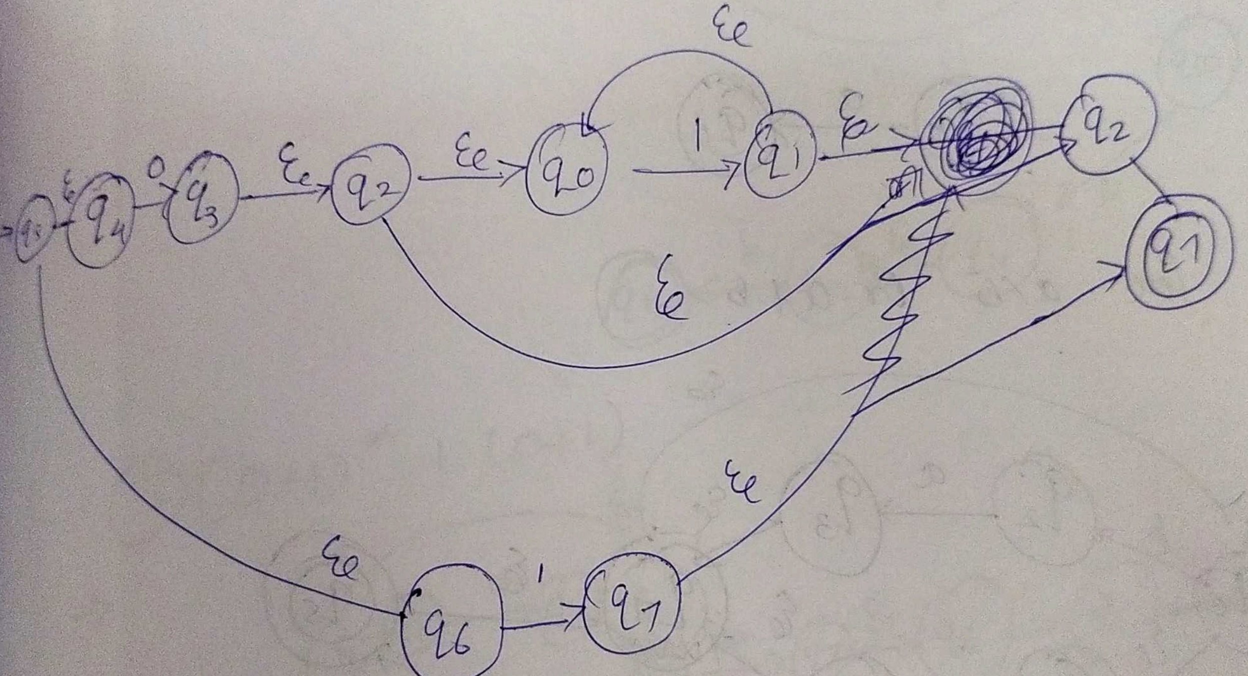


$01^* + 1$



01* + 1





Q construct NFA for regular expression

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

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

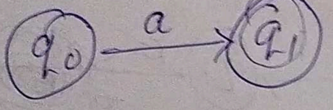
(a) ~~$r = (0+1)^* 1 (0+1)$~~

$= r_1 \cdot r_2 ..$

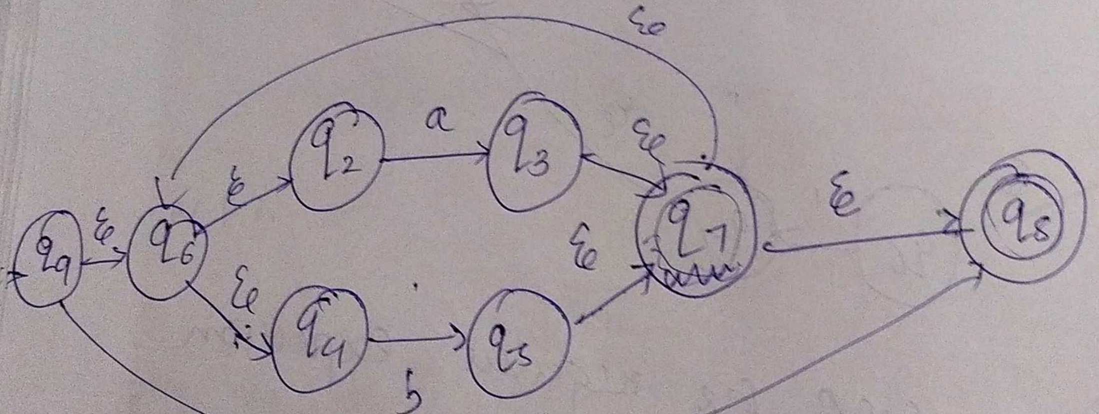
Q Construct NFA for RE $a(a/b)^* bb$

→ (a) $a(a/b)^* bb$

For a,

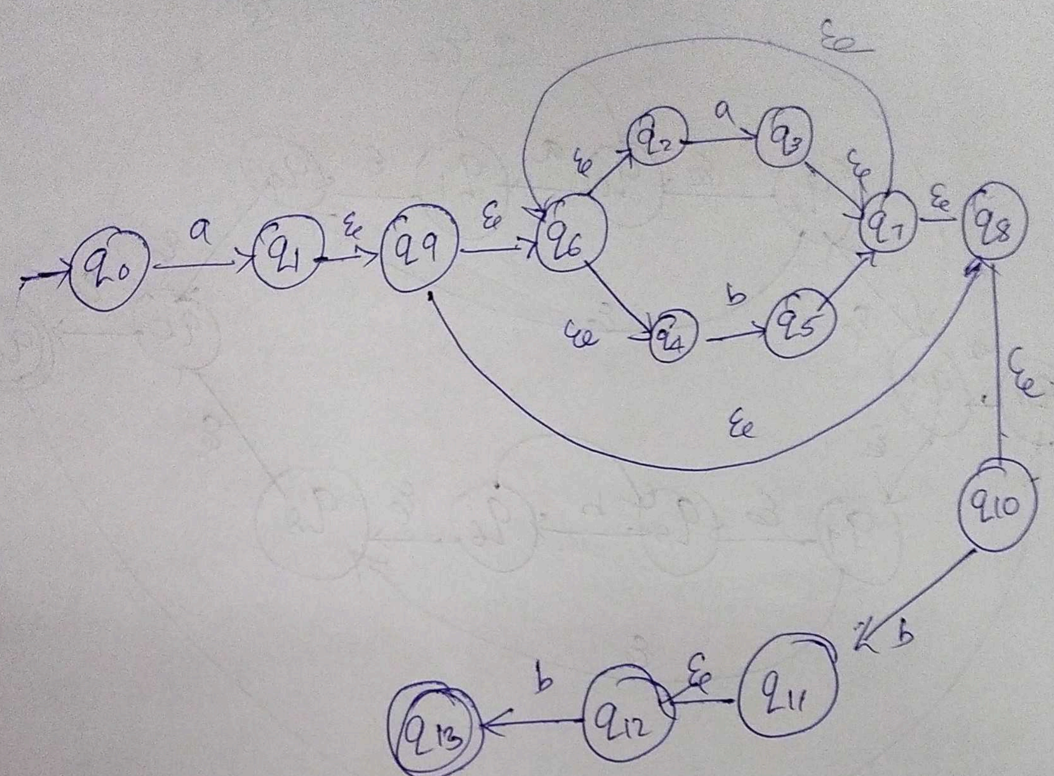


for a/b or a+b

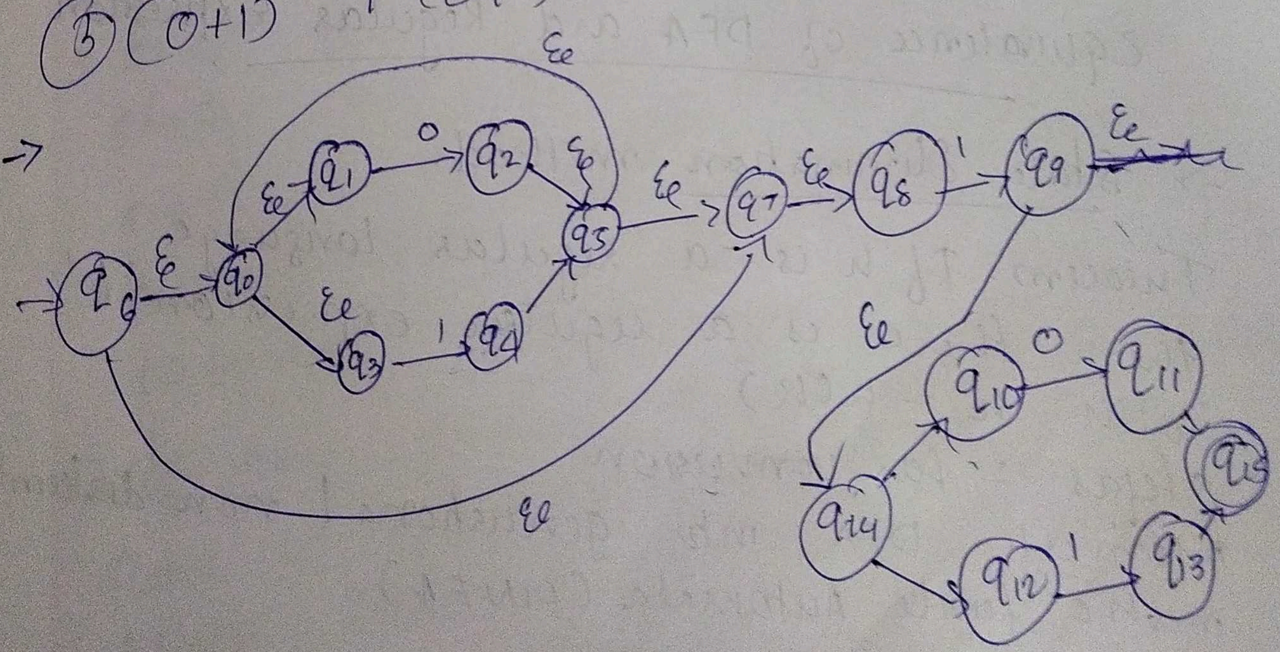


for bb. ε

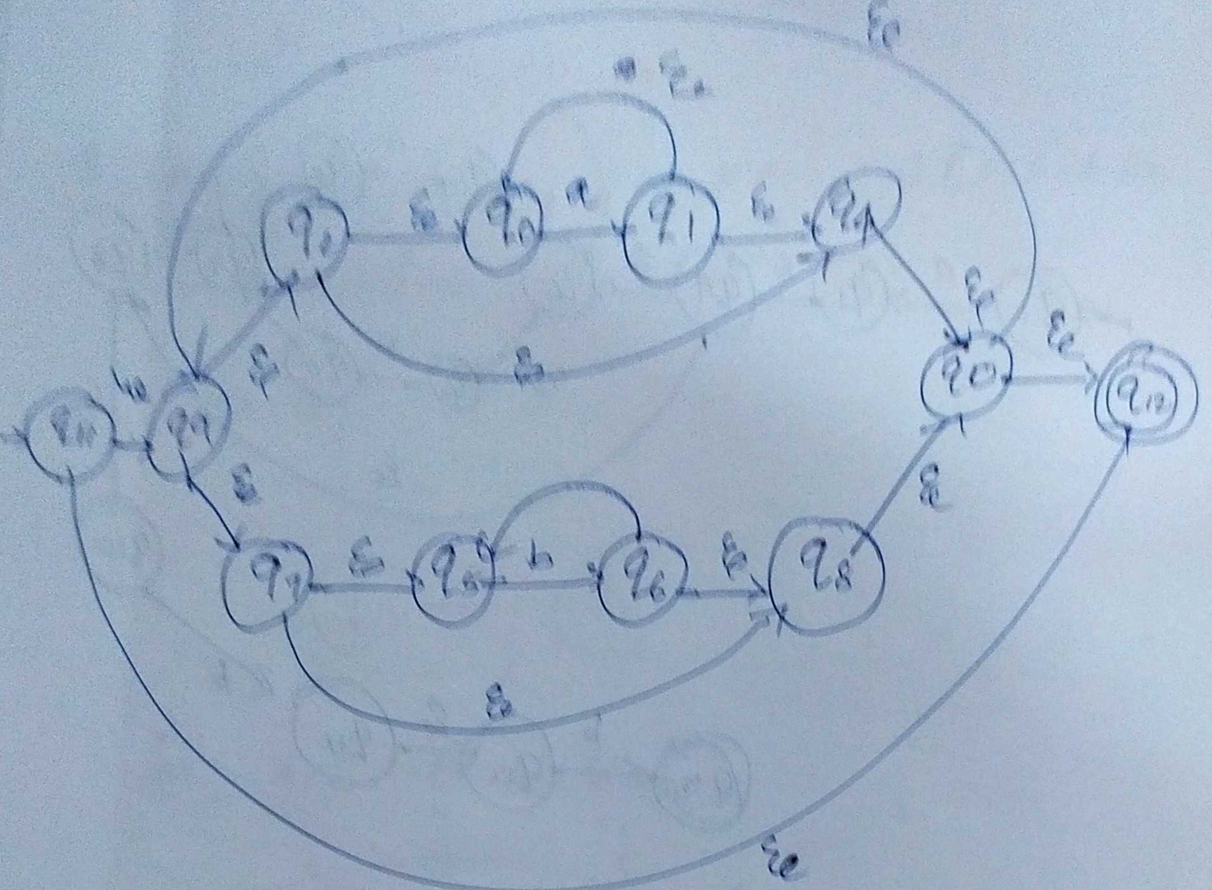
$a(a+b)^* bb$



③ $(0+1)^* 1 (0+1)$



(8) $(a^*/b^*)^+$



Equivalence of DFA and Regular Expression

→ State Elimination method.

Theorem: If L is a regular language then there is a regular expression 'R' with L = L(R)

steps : for conversion

1) convert DFA into generalised non-deterministic finite Automata (GNFA)

1) => GNFA is NFA with transitions being labelled by R.E

GNFA should have a special form that means the following conditions.

1) The start state has transition arrows going to every other ~~state~~ stage but no arrows ~~to~~ coming in from any other state. This can be done by adding a new start state ~~with~~ with ϵ moves with ϵ arrow to all the start states

2) There is only a single accepting state and it has arrows comming in from every other state but no arrows going to any other ~~of~~ state. Also accept state is not the same as start state

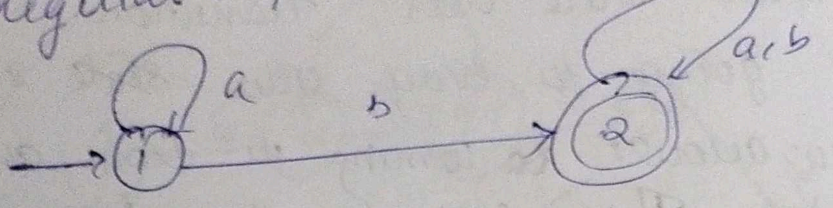
3) Except from the start an accept from every states, one arrow goes from every state to every other state and also from each state to itself. (~~to be~~ labelled as ϕ).

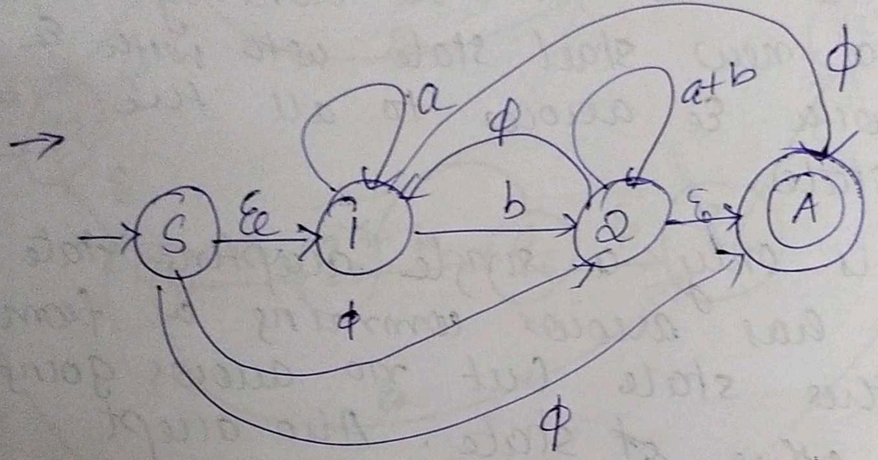
4) If any arrows ~~a~~ have multiple labels, then it is replaced by a single arrow whose label is the union of previous labels.

Step 2:

GNFA obtained is converted into regular expressions. This is done by removing states one at a time in

any order ~~Rep~~ replacing the
labels of edges with more complicated
regular expression

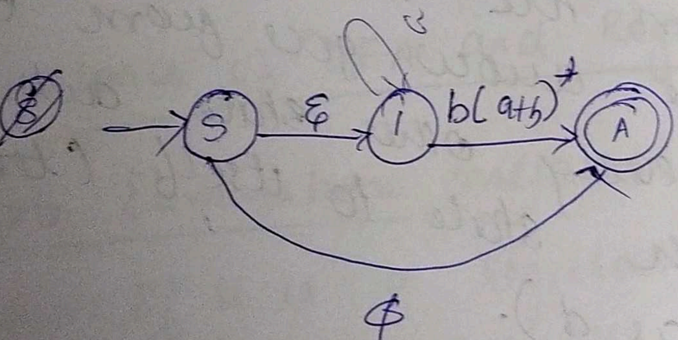


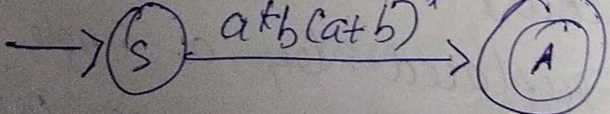


$1 \rightarrow A = \phi + b\cdot(a+b)^*$

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

Removing state 2.





Using formula (Kleen's formula constructor).

RE, $R_{ij}^{K} = R_{ij}^{K-1} + R_{iK}^{K-1}\left[R_{KK}^{K-1}\right]^{*} R_{Kj}^{K-1}$

$R \rightarrow RE$

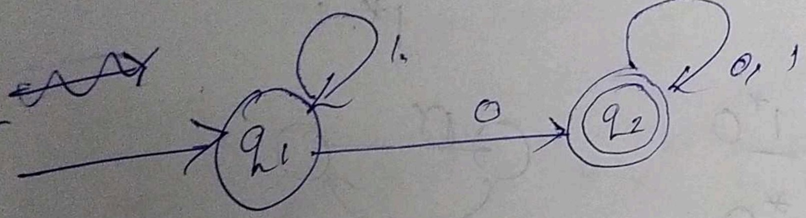
$i \rightarrow$ initial state

$j \rightarrow$ final state

$K \rightarrow$ intermediate states.

Intermediate zero means direct transition..

eg:-



$i = 1$
$j = 2$
$K = 2$

RE =,

$$R_{12}^{2} = R_{12}^{1} + R_{12}^{1}\left[R_{22}^{1}\right]^{*} R_{22}^{1}$$

$$R_{12}^{1} = R_{12}^{0} + R_{11}^{0}\left[R_{11}^{0}\right]^{*} R_{12}^{0}$$

$$R_{22}^{1} = R_{22}^{0} + R_{21}^{0}\left[R_{11}^{0}\right]^{*} R_{12}^{0}$$

Equations for direct transition, (from DFA)

$$R_{11}^{0} = \cancel{(\epsilon+1)} \quad \epsilon + 1$$

$$R_{12}^{0} = 0$$

$$R_{22}^{0} = \epsilon + \cancel{\epsilon}\ 0 + 1$$

$$R_{21}^{0} = \phi$$

$R_{12}^{1} = 0 + (\varepsilon_e+1)[\varepsilon_e+1]^{*}(0)$

$(\varepsilon_e+1)^* \Rightarrow (\varepsilon_e, 1, 1\varepsilon_e, 11\varepsilon_e, \varepsilon_e 011 \ldots$

$= \cancel{0 + (\varepsilon_e+1)}$

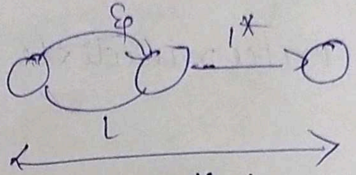
$\hookrightarrow (\varepsilon_e, 11, 111, \ldots)$

$\hookrightarrow 1^*.$

$= \cancel{0 + \varepsilon_f}$

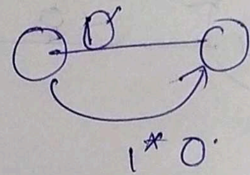
$R_{12}^{1} = 0 + \underline{(\varepsilon_e+1)1^*}0$

$(\varepsilon_e+1)1^* \rightarrow$



$1^*.$

$R_{12}^{1} = 0 + \underline{1^*}0$



$R_{12}^{1} = \underline{1^*0}$

$R_{22}^{1} = R_{22}^{0} + R_{21}^{0}[R_{11}^{0}]^{*} R_{12}^{0}$

$= (\varepsilon_e+0+1) + \phi(\varepsilon_e+1)^*0$

$(\phi * RE = \phi)$

$= (\varepsilon_e+0+1) + \phi$

$= \underline{\varepsilon_e+0+1}$

RG,

$R_{12}^{2} = R_{12}^{1} + R_{12}^{1}(R_{22}^{1})^{*} R_{22}^{1}$

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

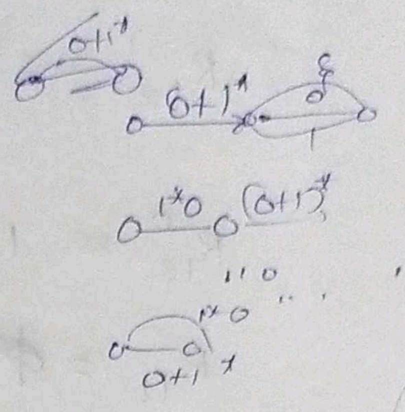
$= \cancel{\phi}1^*0 + 1^*0\,\underline{(0+1)^*(\varepsilon_e+0+1)}$

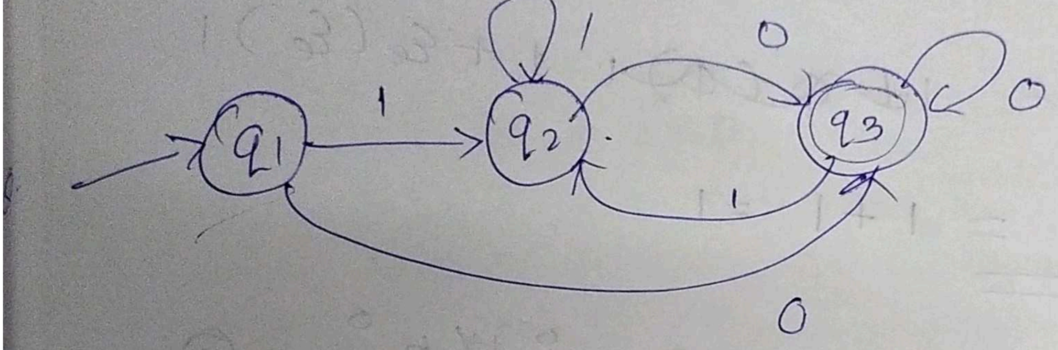
$= 1^*0 + 1^*0(0+1)^*$

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

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

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





$\Rightarrow R_{ij}^{k} = R_{ij}^{k-1} + R_{ik}^{k-1}[R_{kk}^{k-1}]^* R_{kj}^{k-1}$

$i = 1$

$j = 3$

$k = 3$

$R_{13}^{3} = R_{13}^{2} + R_{13}^{2}[R_{33}^{2}]^* R_{33}^{2}$ — (1)

$R_{13}^{2} = R_{13}^{1} + R_{12}^{1}[R_{22}^{1}]^* R_{23}^{1}$ — (2)

$R_{13}^{1} = R_{13}^{0} + R_{11}^{0}[R_{11}^{0}]^*[R_{13}^{0}]$ — (3)

$R_{13}^{0} = 0$

$R_{11}^{0} = \cancel{\epsilon}\ \phi$

(3) $\Rightarrow R_{13}^{1} = 0 + \overset{\epsilon}{\cancel{\phi}}(\overset{\epsilon^*}{\cancel{\phi}}) \times 0 = 0 + 0$

$R_{13}^{1} = 0$

$$R_{12}^{1} = R_{12}^{0} + R_{11}^{0} [R_{11}^{0}]^{*} R_{12}^{0} \quad \text{--- } 4$$

$R_{12}^{0} = 1$

$R_{11} = \cancel{\phi}\ \varepsilon$

$4 \Rightarrow$

$R_{12}^{1} = \cancel{1 + \phi(\phi)^{*}1}\ 1 + \varepsilon_e(\varepsilon_e)^{*}1$

$R_{12}^{1} = 1 + 1 = 1$

$$R_{22}^{1} = R_{22}^{0} + R_{21}^{0} [R_{11}^{0}]^{*} R_{12}^{0} \quad \text{--- } 5$$

$R_{22}^{0} = \varepsilon_e + 1$

$R_{21}^{0} = \phi$

$R_{11}^{0} = \varepsilon_e$

$R_{12}^{0} = 1$

$5 \Rightarrow$

$R_{22}^{1} = \varepsilon_e + 1 + \phi[\varepsilon_e]^{*}1$

$= \varepsilon_e + 1 + \phi$

$R_{22}^{1} = \varepsilon_e + 1$

$$R_{23}^{1} = R_{23}^{0} + R_{21}^{0} [R_{11}^{0}]^{*} R_{13}^{0} \quad -(6)$$

$R_{23}^{0} = 0$

$R_{21}^{0} = \phi$

$R_{11}^{0} = \varepsilon$

$R_{13}^{0} = 0$

(6) $\Rightarrow$ $R_{23} = 0 + \phi[\varepsilon]^{*} \times 0$

$= \underline{\underline{0}}$

(2) $\Rightarrow$ $R_{13}^{2} = R_{13}^{1} + R_{12}^{1} [R_{22}^{1}]^{*} R_{23}^{1}$

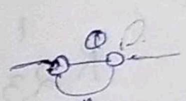
$= 0 + 1 [\varepsilon + 1]^{*} 0$

$= 0 + 1 [\varepsilon + 1]^{*} 0$

$= 0 + 1 \cdot 1^{*} 0$

$= \cancel{0 + 1^{*} 0}$

$\cancel{1^{*} 0}$



$1^{*}0$ $\quad \varepsilon \cdot 0$

ε $\quad \varepsilon .$

$\varepsilon . 0$

$= 0$

$$R_{33}^{2} = R_{33}^{1} + R_{32}^{1} [R_{22}^{1}]^{*} R_{23}^{1} \quad -(7)$$

$R_{33}^{1} = R_{33}^{0} + R_{31}^{0} [R_{11}^{0}]^{*} R_{13}^{0}$

$R_{33}^{0} = 0 + \varepsilon \qquad R_{31}^{0} = \phi$

$R_{11}^{0} = \varepsilon \qquad R_{13}^{0} = 0$

$R_{33}^{1} = 0 + \varepsilon + \phi [\varepsilon]^{*} 0 = 0 + \varepsilon$

$$R_{32}^{1} = R_{32}^{0} + R_{31}^{0} [R_{11}^{0}]^{*} R_{12}^{0} \quad -8$$

$$R_{32}^{0} = 1 \qquad R_{11}^{0} = \varepsilon$$

$$R_{31}^{0} = \phi \qquad R_{12} = 1$$

(8) =>

$$R_{32}^{1} = 1 + \phi [\varepsilon_e]^{*} 1$$

$$= 1 + \phi$$

$$= \underline{\underline{1}}$$

* $R_{22}^{1} = \varepsilon_e + 1$

$R_{23}^{1} = 0$

$$R_{33}^{2} = R_{33}^{1} + R_{32}^{1} [R_{22}]^{*} R_{23}$$

$$= 0 + \varepsilon_e + 1 [\varepsilon_e + 1]^{*} 0$$

$$= 0 + \varepsilon_e + 1 . 1^{*} 0 \qquad \varepsilon 0$$

$$= 0 + \varepsilon_e + 1 1^{*} 0$$

$$= \underline{\underline{\varepsilon_e + 1^{*} 0}}$$

$$R_{13}^{3} = R_{13}^{2} + R_{13}^{2} [R_{33}^{2}]^{*} R_{33}^{2}$$

$$= 1^{*} 0 + 1^{*} 0 [\varepsilon_e + 1^{*} 0]^{*} (\varepsilon_e + 1^{*} 0)$$

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

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

$$R_{13}^{3} = R_{13}^{2} + R_{13}^{2}\left(R_{33}^{2}\right)^{*} R_{33}^{2}$$

$$= (0+11^*0) + (0+11^*0)(0+\varepsilon+11^*0)^{*}(0+\varepsilon+11^*0)$$

$$= (0+11^*0) + (0+11^*0)[0+11^*0)^{*}(0+\varepsilon+11^*0)$$

$$= (0+11^*0) + (0+11^*0)(0+11^*0)^{*}$$

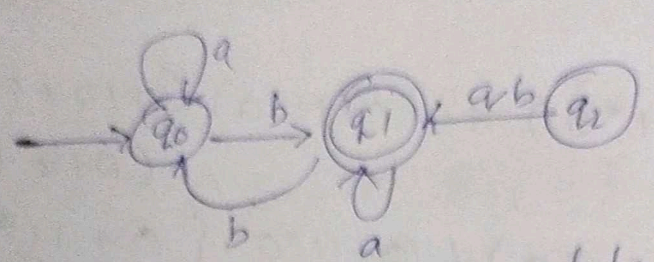
$$= (0+11^*0)(0+11^*0)^{*}$$

Minimisation or Optimisation of DFA

Minimisation or optimisation of DFA refers to deleting those states of DFA whose presence or absence ~~in~~ DFA does not the language accepted by the automata. Hence these states can be eliminated from the automata without affecting the language accepted by the automata

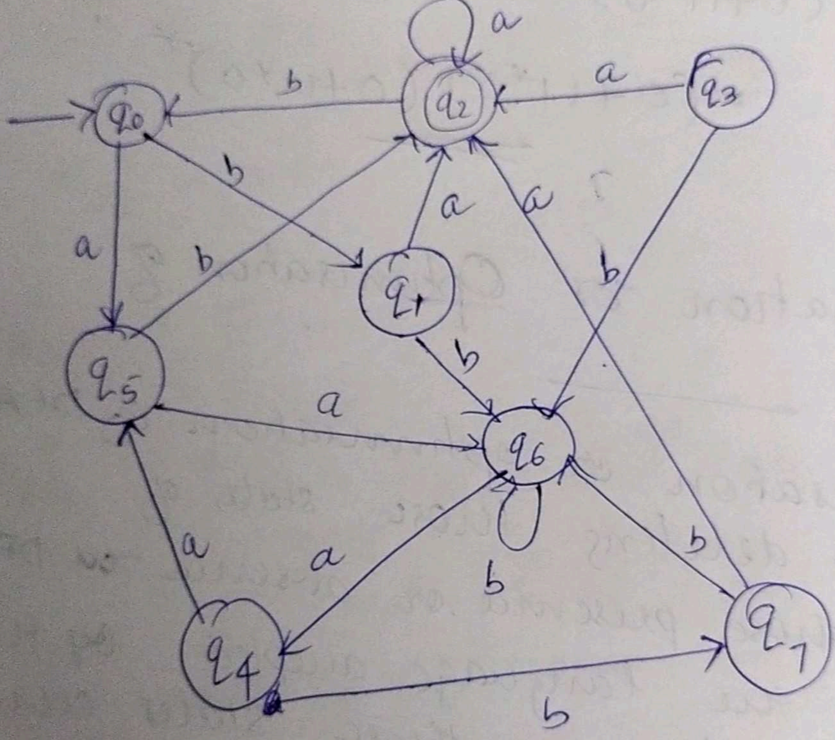
Unreachable states or inaccessible states

All those states which can never be reached from initial ~~states~~ state are called inaccessible state



Here q_2 is the inaccessible state.

Q. Minimise the given DFA



Step 1

Remove the unreachable states from the DFA. Here q_3 is the unreachable state

Step 2

Draw the transition table for rest of states

Q/Σ	a	b
q_0	q_5	q_1
q_1	q_2	q_6
q_2	q_2	q_0
q_4	q_5	q_7
q_5	q_6	q_2
q_6	q_4	q_6
q_7	q_2	q_6

Step3
Divide the rows of the transition table into 2 sets. One set contains only those rows which starts from ~~no~~ non-final states. Other set contains only those row which starts from final states.

set 1

Q/Σ	a	b
q_0	q_2	q_1
q_1	q_2	q_2
q_4	q_5	q_7
q_5	q_6	q_2
q_6	q_4	q_6
q_7	q_2	q_6

set 2

~~Set~~ Q/Σ	a	b
q_2	q_2	q_0

Step-4

Check for similar rows in both the sets. skip 1 & replace.

Q/Σ	a	b
q_0	q_5	q_1
q_1	q_2	q_6
q_4	q_5	q_7 q_1
q_5	q_6	q_2
q_6	q_4 q_0	q_6
q_7	q_2	q_2

=>

Q/Σ	a	b
$\rightarrow q_0$	q_5	q_1
q_1	q_2	q_6
q_5	q_6	q_2
q_6	q_0	q_6

Similarly apply this to set 2

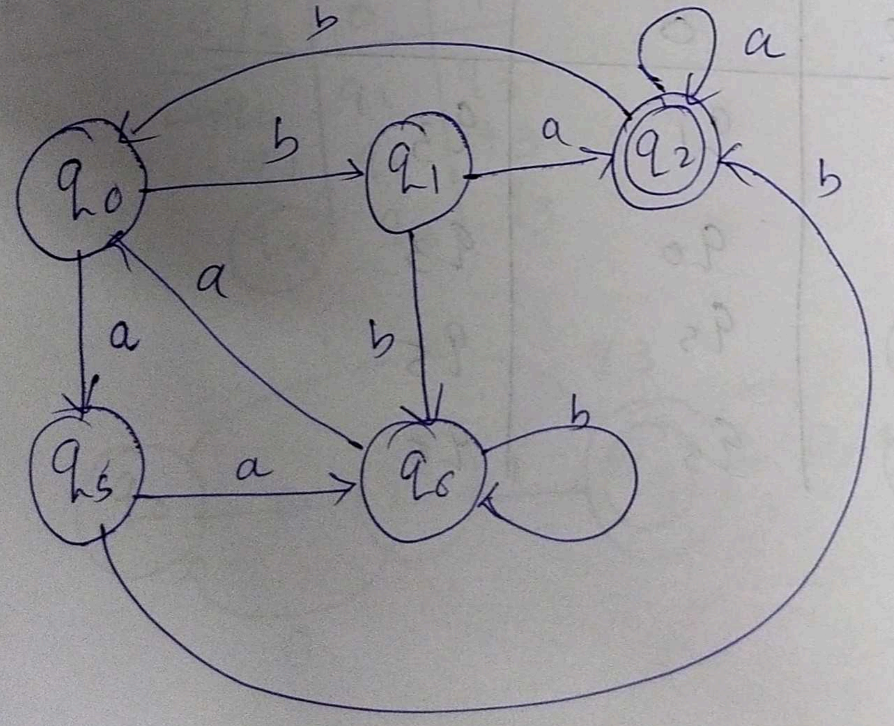
Q/ε	a	b
q_2	q_2	q_0

So, set 1 & set 2 is minimized.

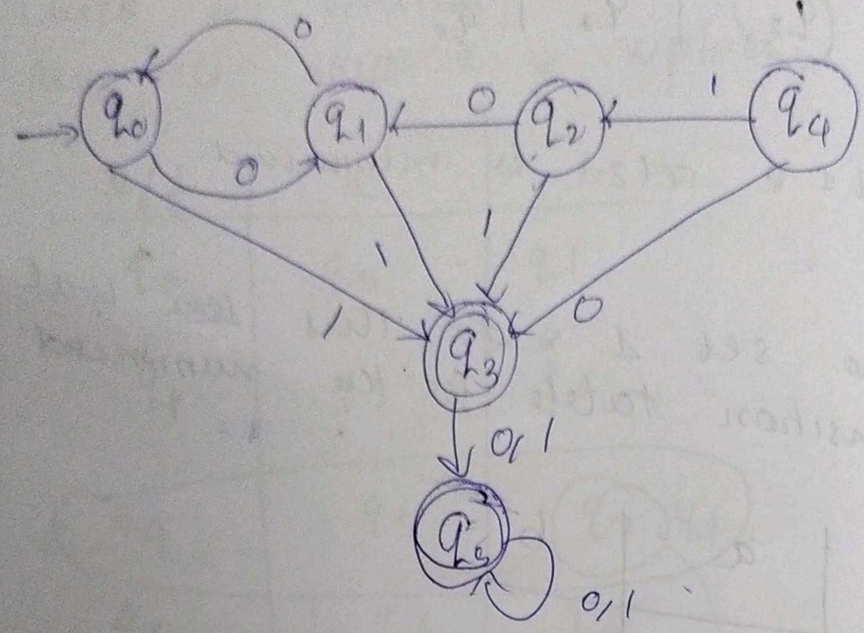
Step 5

Combine set 1 & 2. This ~~the~~ will be the transition table for the minimized DFA.

Q/ε	a	b
$\rightarrow q_0$	q_5	q_1
q_1	q_2	q_6
q_2	q_2	q_0
q_5	q_6	q_2
q_6	q_0	q_6



Q Minimize the given DFA



→ Step 1

~~Remove q2,q4, (unreachable.)~~

Remove q_4, q_2 (unreachable)

Step 2

ε/a	0	1
→ q_0	q_1	q_3
q_1	q_0	q_3
q_3	q_5	q_5
q_5	q_5	q_5

Step 3

Set 1

ε/Q	0	1
→q0	q1	q3
q1	q0	q3

Set 2

ε/Q	0	1
q3	q5 q3	q5 q3
q5	q5	q5

x

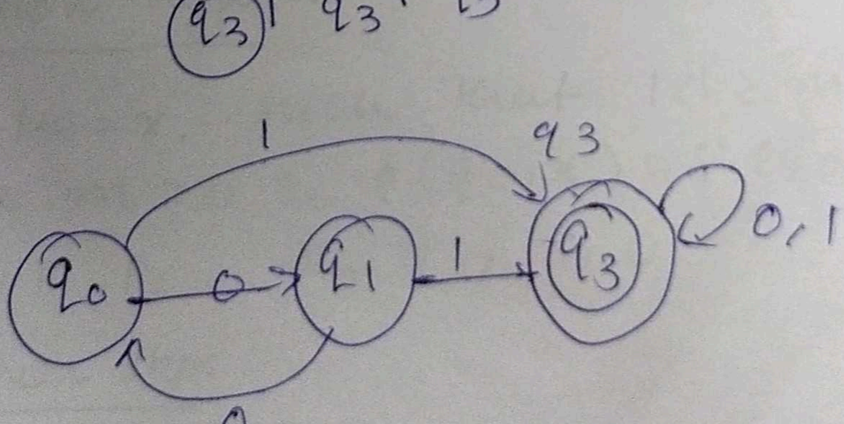
Step 4

ε/Q	0	1
→q0	q1	q3
q1	q0	q3

ε/Q	0	1
q3	q3	q3

Step 5

ε/Q	0	1
→q0	q1	q3
q1	q0	q3
q3	q3	q3



Q. Minimise the following DFA represented as Transition table.

Q/Σ	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_4	q_3
q_2	q_4	q_3
$* q_3$	q_5	q_6
$* q_4$	q_7	q_6
q_5	q_3	q_6
q_6	q_6	q_6
q_7	q_4	q_6

Set 1

Equivalance of NFA & DFA

Theorem: Let 'L' be the language accepted by NFA, then there exists a DFA that accepts L.

Proof (By Induction)

Let $N = (Q, \Sigma, \delta, q_0, F)$ be NFA and

$P = \{Q', \Sigma, \delta', q_0', F')$ be the equivalent DFA.

We have to prove that $L(N) = L(P)$.

$\delta(q_0, w) = \delta'(q_0', w)$

Basic Steps

Let $w = x$

$\delta'(q_0', x) = q_0$; since $q_0' = q_0$

$\delta(q_0, x) = q_0$

Induction Hypothesis.

Let $w = x$, such that $|x| \geq m$ for some $m > 0$ & $\delta(q_0, x) = \delta'(q_0, x) = \{p_0, p_1, p_2 \ldots p_n\}$ is true.

Induction steps

For all string of length $(m+1)$. We have to prove that.

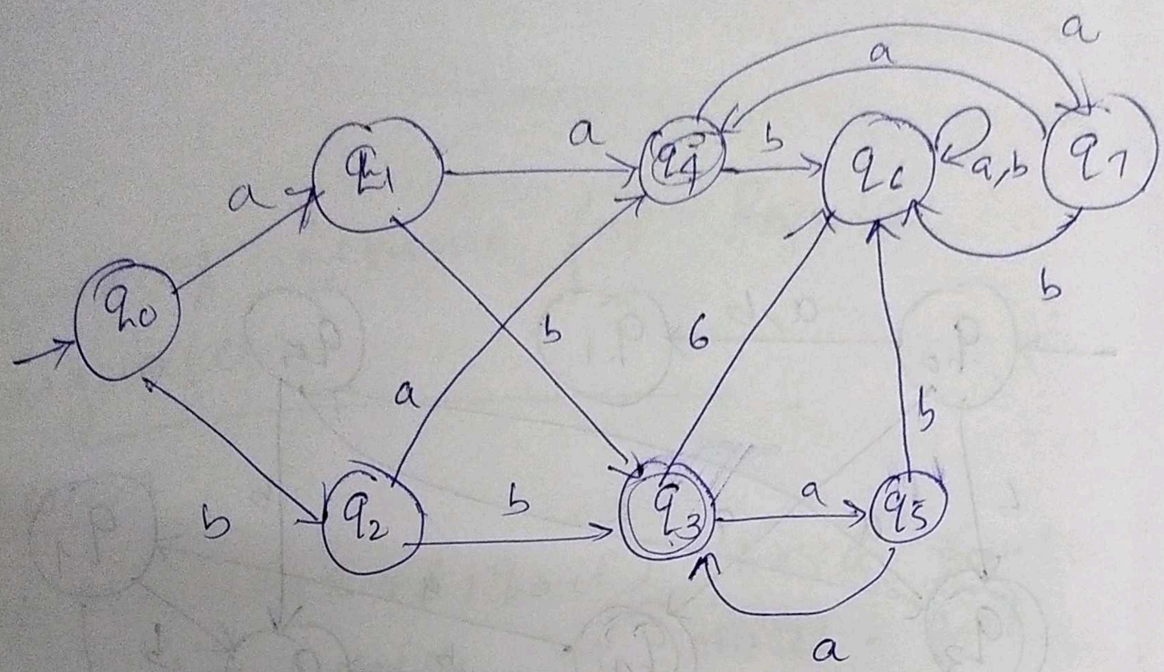
$\Rightarrow \delta(q_0, xa) = \delta'(q_0', xa)$

$$\delta(q_0, xa) = \delta(\delta(q_0, x), a)$$
$$= \delta(\{P_1, P_2, P_3, P_4 \ldots P_n\}, a)$$
$$= \delta(P_1, a) \cup \delta(P_2, a) \cup \delta(P_3, a) \ldots)$$
$$= \bigcup_{i=1}^{n} \delta(P_i, a)$$

$$\delta'(q_0', xa) = \delta(\delta'(q_0', x), a)$$
$$= \delta(\{P_1, P_2, \ldots P_n\}, a)$$
$$= \delta(P_1, a) \cup \delta(P_2, a) \cup \ldots \delta(P_n, a)$$
$$= \bigcup_{i=1}^{n} \delta(P_i, a)$$

i.e. $\delta(q_0, xa) = \delta'(q_0, xa)$

Hence proved.



Set 1

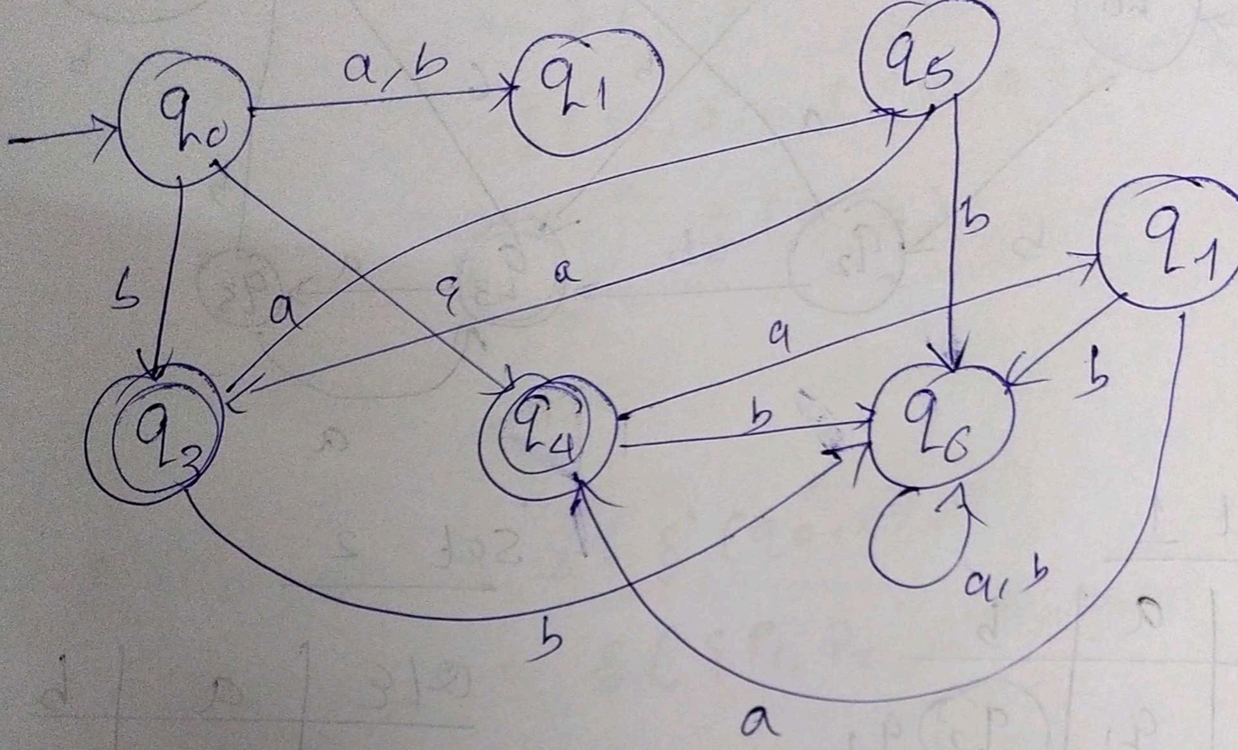
Q/E	a	b
q_0	q_1	q_2 q_1
q_1	q_4	q_3
q_2	q_4	q_3 ×
q_5	q_3	q_6
q_6	q_6	q_6
q_7	q_4	q_6.

Set 2

Q/E	a	b
* q_3	q_5	q_6
* q_4	q_7	q_6.

Q/E	a	b
→ q_0	q_1	q_1
q_1	q_4	q_3
q_5	q_3	q_6
q_6	q_6	q_6
q_7	~~q_6~~ q_4	q_6

Q/E	a	b
→ q_0	q_1	q_1
q_1	q_4	q_3
* q_3	q_5	q_6
* q_4	q_7	q_6.
q_5	q_3	q_6
q_6	q_6	q_6
q_7	q_4	q_6



Pumping Lemma for Regular Languages

Online

Pumping Lemma for Regular languages

Theorem

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a finite automaton with 'n' states. Let L be a regular language set accepted by M. For every string w in L ($w \in L$) such that $|w| \geq n$, w can be broken into 3 strings x, y & z. (ie $w = xyz$) such that

1. $y \neq \varepsilon$
2. $|xy| \leq n$
3. for all $i \geq 0$, the string xy^iz is also in L. (This means that we can always find a non-empty string 'y' not too far from the beginning as w that can be pumped ie repeating y any no. of times).

Proof :-

Suppose L is ~~rg~~ regular. Then $L = L(A)$ for some DFA 'A'. Suppose

A has 'n' states. Consider any string ω of length $\geq n$ i.e. $|\omega| \geq n$.

i.e., $\omega = a_1, a_2 \ldots a_m$ where $m \geq n$

and $a_1, a_2 \ldots$ are i/p symbols

Let $\delta(q_0, a_1 a_2 \ldots a_i) = q_i$ for

$i = 0, 1, \ldots m$

where δ is the transition function of A. q_0 is the start state. q_i is state of the automaton A that is reached after reading i symbols of ω. By the pigeonhole principle, it is not possible for all 'm' q_i's to be distinct, since there are only n states and $m \geq n$

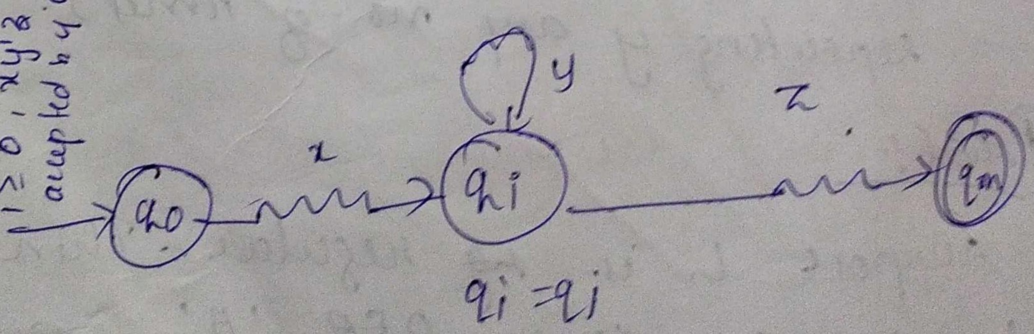
Thus we take 2 integers i & j with $0 \leq i < j \leq n$, such that $q_i = q_j$

Now $\omega = xyz$ is broken as follows

$x = a_1 a_2 \ldots a_i$

$y = a_{i+1}, a_{i+2} \ldots a_j$

$z = a_{j+1}, a_{j+2} \ldots a_m$



$q_i = q_j$

The automata start from the initial states $q_0 \ldots q_0$. applying the string x, it reaches $q_1 \ldots q_m$. Applying string 'xy', it comes to state q_i (i.e. q_i-q_j in effect comes to the same state). On applying 'z', it reaches q_n, which is the final state. Hence $xy^iz \in L$, as for any $i \geq 0$, xy^iz also accepted by automata A.

steps needed for proving that the given set is not ~~reg~~ regular.

Step 1 :-

Assume that L is regular.

Let n be the no. of states in corresponding finite Automata.

Step 2 :-

Choose a string ω such that $|\omega| \geq n$

Use pumping lemma to write $\omega = xyz$ with $|xy| \leq n$ and $|y| > 0$

Step 3 :-

Find a suitable integer i ~~xy~~, such that $xy^iz \notin L$, ~~this constraint~~

• Prove that the language $L = \{0^i 1^i / i >= 1\}$ is not regular.

→ Step 1 :-

Assume that L is regular.

Let n be the no. of states.

Step 2 :-

Let $\omega = 0^n 1^n$

$|\omega| = 2n > n$

By pumping lemma,

we can write $\omega = xyz$,

with $|xy| <= n$ ie $|y| \neq 0$. (or $|y| > 0$)

Step - 3 :-

We need to find i such that $xy^iz \notin L$

000000111111 (x, y, z)

~~0000001111~~

0000001[6] (x y z)

$0^n 1^n =$

$n = p + q + r$; $p \neq 0$, $q \neq 0$, $r \neq 0$

$p \neq$ ~~0~~ $0^n 1^n = \underset{x}{0^p} \underset{y}{0^q} \underset{z}{0^r 1^n}$

$|y| \neq 0$ $|xy| \leq n$. ie $(p+q)$

Take the integer as k ie let $k=2$.

then $xy^kz = xy^2z =$ ~~$0^p 0^q$~~ $0^p 0^{2q} 0^r 1^n$

$\therefore$ No. of 0's $= p + 2q + r$

$= (p+q+r) + q$

$= n + q$

$= xy^2z = 0^{n+q} 1^n \notin L$

$\therefore$ So, $0^i 1^i$ is not regular.

or

$w = 0^{n-k} 0^k 1^n$

$x = 0^{n-k}$ $y = 0^k$, $z = 1^n$.

step 3 :-

$xyz \notin L$

if $i = 0$

$xy^i z = xz$.

$xy^0 z = 0^{n-k} 1^n \notin L$

This is a ~~constru~~ contradiction.

Hence L is not regular.

Q. Show that $L = \{ ww \mid w \in (a,b)^+ \}$ is not regular.

→ ~~1)~~ Step 1:-

Assume L is reg^r.

Let n be the no of states in finite automaton accepting L.

step 2 :-

Let $w = a^n b$

$ww = a^n b a^n b$

$|ww| = |a^n b a^n b| = 2n + 2$

$= 2(n+1) > n$.

By pumping lemma,

we can write,

$$ww = xyz \quad \text{with } |xy| \le n \ \& \ |y| > 0$$

Step 3:-

find i, such that, $xy^i z \notin L$

case 1) y has no b's

$ww = a^n b a^n b$

$xyz = \underbrace{a^{n-i}}_{x} \ \underbrace{a^i}_{y} \underbrace{b \ a^n b}_{z}$

Let $k = \cancel{2}0$

$xy^k z = xz$.

$= a^{n-i} b a^{n b} \notin L$

This is a contradiction.

Hence L is not regular.

Q. Show that $L = \{ww^R \mid w \in (a,b)^*\}$ is not regular (reverse)

→ Step 1 :-

Assume L is regular

Let n be the no. of states in finite automata ~~ea~~ accepting L

step 2 :-

Let $w = a^n b$

$ww^r = a^n b \, \cancel{a^n b} \, b a^n \in L$

$|ww^R| = 2n+2 = 2(n+1) > n$

By pumping lemma,

~~w = x~~

$ww^r = xyz$, with $|xy| \le n$ & $|y| > 0$

step 3 :-

Find i, such that $xy^i z \notin L$

$xyz = \underbrace{a^{n-k}}_{x} \underbrace{a^k}_{y} \underbrace{bba^n}_{z}$

$i = 0$, $xy^i z = xz$

$= a^{n-k} bba^n \notin L$

This is a contradiction,

Hence $\underline{L}$ is not regular.

Q. Show that $L = \{a^p \mid p$ is a prime number $\}$ is not regular.

→ step 1 :-

Assume L is reg^r

Let n be the no of states in finite automata accepting L

Step 2 :-

let $w = a^p$ (p is a prime no $> n$)

By pumping lemma,

$w = xyz$ with $|xy| \leq n$ & $|y| \neq \varepsilon$

$w = a^p = xyz$

$|w| = |a^p| = p$, x, y & z

are simply strings

of a's.

So, $y = a^m$ where $m \geq 1$ & $m < n$

Step 3 :-

Let $k = p+1$

$w = xy^kz$

$|xy^kz| = |xyz| + |y^{k-1}|$

$= p + |a^{m(k-1)}|$

$= p + m(k-1)$

$= p + m(p)$

$= \underline{\underline{p(1+m)}} \notin L$

Product of 2 no.s is not a prime. no. so $xy^kz \notin L$.

This is a contradiction.

Hence L is not regular.

Tutorial

1) Show that $L = \{a^i b^j c^k \mid k > i+j\}$ is not regular. $w = a^n b^n c^{3n}$

2) S.T. is $L = \{a^{2n} \mid n \geq 1\}$ regular?

3) S.T. $L = \{0^n 1^m 2^n \mid n, m \geq 0\}$ is not regular.

Closure Properties of Regular sets or Regular Languages

1) Union of 2 regular set is regular $(L_1 \cup L_2)$

2) Intersection of 2 regular set is regular. $(L_1 \cap L_2)$

3) Concatenation of 2 regular set is regular $(L_1 L_2)$

4) Complement of regular set is regular $(\overline{L_1})$

5) Reversal of a regular set is regular (L_1^R)

6) Closure (L_1^*)

7) Difference of 2 reg set is reg