31/10/22
Monday

UNIT-2

Regular Language and Regular Expression :-

Language is said to be a regular if there exist a finite acceptor for it.

Every regular language is described by a DFA and NFA.

The language accepted by a finite automata can be easily described by simple expressions called regular expression.

Regular expression is most efficient way to represent any language.

Language accepted by regular expression is known as regular language.

Regular Expression :-

One of the way to describe the regular language is regular Expression i.e an algebraic description or notation describes exactly same language as finite automata.

Regular Expression serves as input language for many systems that crosses the string.

Applications of Regular Expressions :-

search commands such as

1. Unix, GREP
2. Lexical ~~analysis~~ analyzer generators suchas lex or flex.

Formal definition of Regular Expression :-

Regular Expression defines any Regular language which is defined by finite automata.

Regular Expression involves combination of strings of symbols from some alphabets Σ, (,), and the operators + and * and •.

Where + is used to denote union operation.

• is used to denote concatnation operation.

* is used to denote * clauser | klean clourse

NOTE

1. ϕ is a regular expression denoting the empty set. i.e { }

2. ϵ / λ is regular expression denoting $\{\epsilon\}$ (epsilon or lambda)

3. $\forall a \in \Sigma$, a is regular expression denoting language {a}

Operations on Regular Languages :-

There are '3' basic ~~languages~~ operations on Regular languages.

1. Union
2. Concatnation
3. Klean Clourse

(1). Union :- Union of two languages L and M is $L \cup M$. It is the set of strings that are either L or M or both.

If $L = \{0, 01, 110\}$ and

$M = \{1, 11, 01\}$.

$L \cup M = \{0, 1, 01, 11, 110\}$

(2). Concatnetion :- Concationation of two languages L and M is set of strings that can be formed by taking any string in L and concatnating it with any string in M.

To denote the concatnation of languages L and M either with

- operator or no operator at all.

i.e $L.M$ or LM

Example :-

If $L = \{01, 11, 110\}$ and

$M = \{\epsilon, 1, 101\}$

then, $L \cdot M = \{01, 11, 110, 011, 111, 1101, 01101, 11101, 110101\}$

(3) Klean Clouse :- Klean clouse of Language is denoted by L^*. And it represents a set of strings that can be formed by taking any no. of strings from L.

$$L^* = \bigcup_{i>=0} L^i = L^0 \cup L^1 \cup L^2 \cup \cdots\cdots$$

Example :- If $L = \{0, 11\}$ then $L^* = L^0 \cup L^1 \cup L^2 \cup \cdots$

$L^0 = \{\epsilon\}$, $L^1 = \{0, 11\}$, $L^2 = \{00, 1111\}$

$L^3 = \{000, 01111, 1100, 111111\}$

$L = \{0, 11\}$

$L^0 = \{\epsilon\}$

$L^1 = \{0, 11\}$

$L^2 = L.L = \{0, 11\}\ \{0, 11\} = \{00, 011, 110, 1111\}$

$L^3 = L^2.L = \{00, 011, 110, 1111\} . \{0, 11\}$

$= \{000, 0011, 0110, 01111, 1100, 11011, 11110, 111111\}$

NOTE :-

Clouse of language ϕ is denoted by ~~any lanaguage~~

$\phi^* = \{\epsilon\}$

Clouse of null string & empty string is denoted by.

$\epsilon^* = \epsilon$

Identity Rules :-

Let Q, P, R be a regular expressions then the identity rules are as follows :

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

2. $\epsilon^* = \epsilon$

3. $\phi^* = \epsilon$

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

5. $\phi + R = R + \phi = R = R + R$

$\boxed{\epsilon + R = R + \epsilon = R + \epsilon}$

6. $R . R^* = R^* . R = R^+$

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

$= \epsilon + R^+$

$= R^*$

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

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

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

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

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

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

(1) prove the expression $\epsilon+1+(\epsilon+1)(\epsilon+1)^*(\epsilon+1) = 1^*$

Sol. Take L.H.S

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

$= \epsilon+1+(\epsilon+1).(\epsilon+1)^*(\epsilon+1)$ $\quad \therefore [\epsilon+R^*R = R^*]$

$= (\epsilon+1)[\epsilon+(\epsilon+1)^*(\epsilon+1)]$

$= (\epsilon+1)(\epsilon+1)^*$ $\quad \epsilon+R.R^* = R^*$ $\quad R.R^* = R^*R = R^+$ $\quad \therefore R.R^* = R^+$ $\quad \therefore (\epsilon+R)^+ = R^*$

$= (\epsilon+1)^+$

$= 1^*$ //

$R+R.R^* = R^*R$ $\quad (P^*Q^*)^* = (P+Q)^*$ $\quad (R^*)^* = R^*$ $\quad (\epsilon+R)^* = R^*$ $\quad R^*(\epsilon+R) = (\epsilon+R)R^* = R^*$ $\quad (\epsilon+R)^+ = 1^*$

$\epsilon.R = R.\epsilon = \epsilon$ $\quad \phi.R = R.\phi = \phi$ $\quad \epsilon^* = \epsilon$ $\quad \phi^* = \phi$ $\quad \epsilon+R = R+\epsilon = R$ $\quad R+R = R$ $\quad R^*.R^* = R^*$ $\quad (PQ)^*P = P(QP)^*$

2/11/22
Wednesday

Q. prove that $(1+00^*.1)+(1+00^*1)(0+10^*1)^*(0+10^*1) = 0^*1(0+10^*1)^*$

take L.H.S :-

$(1+00^*.1)+(1+00^*1)(0+10^*1)^*(0+10^*1)$

$(1+00^*1)[\epsilon+(0+10^*1)^*(0+10^*1)]$

$00^* = 0^+$

$\cancel{(1+0^+1)[\epsilon+(0+10^*1)[\epsilon+0+10^*1]}$

$(1+00^*1)[\epsilon+(0+10^*1)^*(0+10^*1)]$

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

$(1+00^*1)(0+10^*1)^*$

$(\epsilon+00^*)1(0+10^*1)^*$

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

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

Construct a regular expression from given step.

1. Write a regular expression for language accepting any combination of a's except null string over

$\downarrow$
ϵ

$\Sigma = a$.

2. write a regular expression for language accepting

all combination of a except null string

over $\Sigma = \{a\}$

<u>Ans</u>:- R.E $= a^+$

3. Write a regular expression for language accepting any combination of a and b over

4. COA

<u>Examples</u>:- 1. Construct R.E for language contains all strings having any no.of a's and b's except null string.

<u>Ans</u>:- R.E $= (a+b)^+$.

2. Construct R.E for language contains all strings any no.of a's followed by any no.of b's

<u>Ans</u>:- R.E $= a^* b^*$

3. Construct R.E language accepts all strings ending with 00 over $\Sigma = \{0,1\}$

<u>Ans</u>:- $(0+1)^* 00$

4. Construct R.E language accepts all strings that contains 00 as substring, over $\Sigma = \{0,1\}$

<u>Ans</u>:- $(0+1)^* . 00 (0+1)^*$

5. Construct R.E language accepts all strings over the alphabet starts with a and ends with b

Ans:- $a(a+b)^* b$

6. construct R.E language accepts all strings with atleast 2 b's and over $\Sigma = \{a, b\}$.

Ans:- $(a+b)^* \, b \, (a+b)^* \, b \, (a+b)^*$.

7. construct R.E language accepts all strings with exact 2 b's over $\Sigma = \{a, b\}$

Ans:- $a^* b a^* b a^*$

8. Construct R.E language accepts all strings, which contains 2 consecutive b's

Ans:- $(a+b)^* \, bb \, (a+b)^*$

9. construct R.E language which accepts All the strings which contains atleast one a followed by atleast one b followed by atleast one c.

Ans:- ~~[illegible]~~ $a^+ b^+ c^+$

10. construct R.E language which accepts all strings in which 3rd symbol from right end is b over string $\Sigma = \{a, b\}$.

Ans:- $(a+b)^* \, b \, (a+b)(a+b)$.

11. write R.E language, which accepts all the strings and which begin and end with either 00 or 11 over $\Sigma = \{0, 1\}$

Sol:- R.E can be divided into two subparts.

1. L_1 denotes strings which begin with 00 or 11.

2. L_1 denotes strings which ends with 00 or 11.

R.E for L_1 is $(00+11)(0+1)^*$

R.E for L_2 is $(0+1)^*(00+11)$

$\therefore (00+11)(0+1)^* + (0+1)^*(00+11)$

Conversion from DFA to R.E :-

Theorem:-

If $L = L(A)$ for some DFA, A. then there is some regular expression(R) such that $L = L(R)$.

$\therefore A = (Q, \Sigma, \delta, q_0, F)$

Proof:-

3/11/22 Thursday

Basics:-

R.E $R_{ij}^{(k)}$ is a R.E whose language is set of strings w such that, w is label of the path from state i to j in A.

And path has no intermediate node whose number greater than k.

To construct an expression $R_{ij}^{(k)}$ starting at $k=0$ and finally reaching at $k=n$.

Where n indicates no. of states.

For $k=0$ there are only two kinds of paths that meets the condition.

1. An arc from state i to state j.
2. Path of length 0 that consists of only some node 'i'. $\boxed{\text{If } i \neq j.}$ then

(a). If there is no symbol $R_{ij}^{(0)} = \phi$

(b) If There is a exactly one symbol i.e 'a' then

$R_{ij}^{(0)} = a$.

(c). If there are symbols $a_1, a_2, \ldots a_k$ then

$R_{ij}^{(0)} = a_1 + a_2 + \cdots + a_k$.

3. A path of length 0 that consists only some node 'i'. $\boxed{\text{if } i=j}$

(a). ~~Take~~ If there is no symbol then $R_{ij}^{(0)} = \phi + \epsilon = \epsilon$.

(b). If there is exactly one symbol i.e 'a' then

$R_{ij}^{(0)} = \epsilon + a$

(c). If there are symbols $a_1, a_2, \ldots a_k$ then

$$R_{ij}^{(0)} = \epsilon + a_1 + a_2 + \cdots + a_k.$$

Induction:-

Suppose there is a path from state 'i' to state 'j' that goes to no state higher than k.

There are two possible cases to consider.

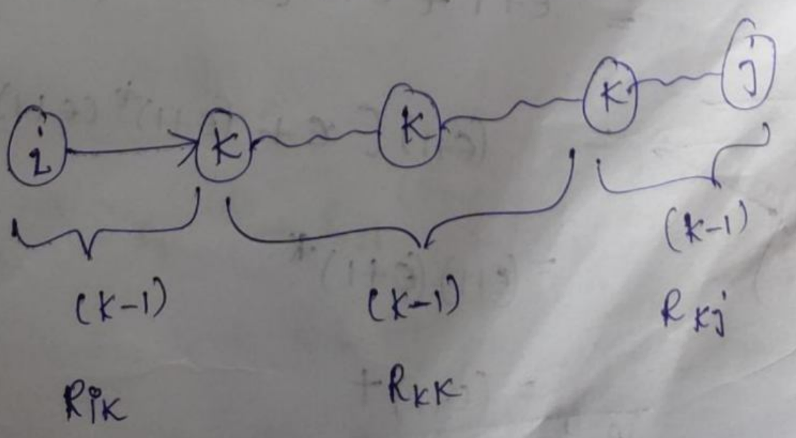
(1). The path that doesn't go through the state k at all. Then the label of the path is $R_{ij}^{(k-1)}$

(2). The path goes to the state 'k' atleast once then we can break the path into several pieces.

To combine expressions of path is as follows:

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)}$$

$$R_{ij}^{(k)} = R_{\underline{ij}}^{(k-1)} + R_{\underline{i}k}^{(k-1)} \cdot \left(R_{kk}^{(k-1)}\right)^{*} \cdot R_{kj}^{(k-1)}$$



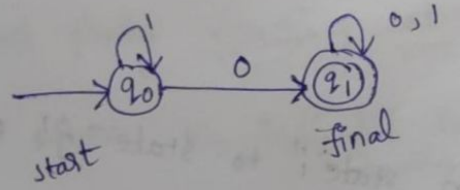
at k= [illegible]

[illegible]

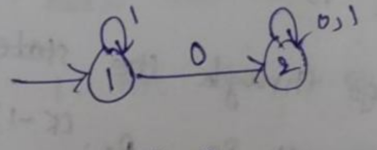
$= (\epsilon - 1)[\epsilon + (\epsilon-1)^{*}(\epsilon+1)]$

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

Convert the following DFA into R.E.



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



1 at $k=0$

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

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

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

$R_{22}^{(0)} = \epsilon + 0 + 1.$

at $k=1$

$R_{11}^{(1)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}$

$i=1, j=1, k=1$

$\cancel{R_{12}^{(1)} =}$ $= R_{11}^{(0)} + R_{11}^{(0)} \left(R_{11}^{(0)}\right)^* R_{11}^{(0)}$

$\cancel{R_{21}^{(1)} =}$ $= \epsilon + 1 + \epsilon + 1 (\epsilon + 1)^* (\epsilon + 1)$

$\cancel{R_{22}^{(1)} =}$ $= (\epsilon+1) [\epsilon + (\epsilon+1)^* (\epsilon+1)]$

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

$\left\{ \begin{array}{l} \epsilon + R.R^* = R^* \\ R.R^* = R^+ \end{array} \right.$

$= (\epsilon+1)^+$

$\boxed{R_{11}^{(1)} = 1^*}$

$R_{12}^{(1)}$

$i=1$
$j=2$
$k=1$

R_2

$i=2$
$j=1$
$k=$

$i=$
$j=$

~~$\not{k}$~~

$$R_{12}^{(1)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)}\left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}$$

$i=1$
$j=2$
$k=1$

$$= R_{12}^{(0)} + R_{11}^{(0)}\left(R_{11}^{(0)}\right)^* R_{12}^{(0)}$$

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

$\cancel{RR^* = R^+}$

$\cancel{(\epsilon+1)^+ \cdot 0}$

$\cancel{(\epsilon+R)^* = R^+}$

$\cancel{1^+ \cdot 0}$

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

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

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

$$\boxed{R_{12}^{(1)} = 1^* \cdot 0} \; //$$

$$R_{21}^{(1)} = R_{21}^{(0)} + R_{21}^{(0)}\left(R_{11}^{(0)}\right)^* R_{11}^{(0)}$$

$i=2$
$j=1$
$k=1$

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

$$= \phi\left[\epsilon + (\epsilon+1)^*(\epsilon+1)\right]$$

$$= \phi[\epsilon + R$$

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

$$= \cancel{\phi R^*}$$

$$= \phi 1^*$$

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

$i=2$
$j=2$
$k=1$

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

$=(0+\epsilon+1) \; //$

$= \epsilon+0+1$

$$= 0\left[\epsilon + \underbrace{\phi(\epsilon+1)^*}_{\phi})\right]$$

at $K=2$

$$R_{12}^{(2)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}$$

$i=1$
$j=2$
$k=2$

$$= R_{12}^{(1)} + R_{12}^{(1)} \left(R_{22}^{(1)}\right)^* R_{22}^{(1)}$$

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

$$= 1^*.0\,[\epsilon + (\epsilon+0+1)^*\,(\epsilon+0+1)]$$

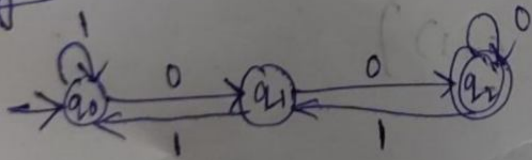
$$= 1^*.0\,[\epsilon + (0+1)^*\,(\epsilon+0+1)]$$

$$\therefore [(\epsilon+R)^* = R^*]$$

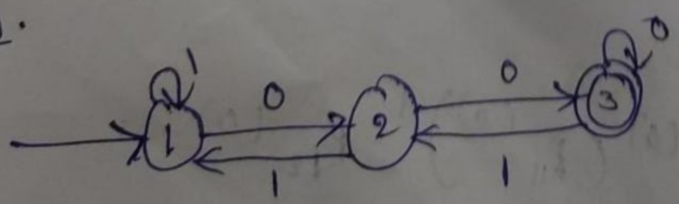
$$= 1^*0\,[\epsilon + (0+1)^{*+}]$$

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

4/11/22
Friday



eg.



At $K=0$

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

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

$R_{13}^{(0)} = \phi$

10/11/22
Thursday

Table filling Algorithm:-1

finding distinguishable pairs in DFA

Table filling algorithm is a recursive discovery of the distinguishable pair of

$$A = (Q, \Sigma, \delta, q_0, F)$$

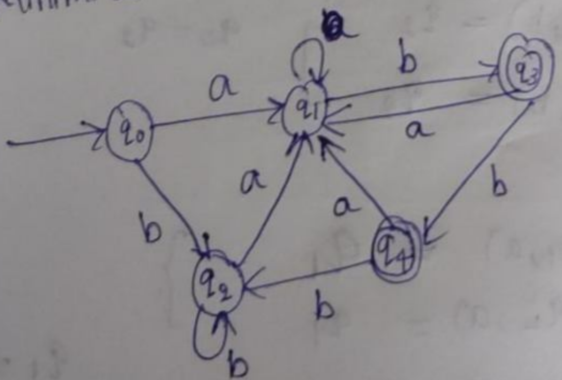
Basics:-

If 'p' is an accepting state and 'Q' is non-accepting state then the pair (P, Q) is distinguishable.

Induction:-

let 'p' and 'Q' be two states such that for some input symbol 'a'. If $\delta(P, a) = r$ and $\delta(Q, a) = s$ are two distinguishable pair (r, s) then P, Q are distinguishable states.

1. Minimize the following DFA.



Step 1: Mark final and non-final states as distinguishable (*)

	q_0	q_1	q_2	q_3 *
q_1	X			
q_2	✓	X		
* q_3	X	X	X	
* q_4	X	X	X	X

<u>Step 2:-</u> Identify the remaining states are distinguishable or not.

$$(q_0, q_1) = \left.\begin{array}{l} \delta(q_0, a) = q_1 \\ \delta(q_1, a) = q_1 \\ \delta(q_0, b) = q_2 \\ \delta(q_1, b) = q_3 \end{array}\right\} q_0 \neq q_1$$

$$(q_0, q_2) = \left.\begin{array}{l} \delta(q_0, a) = q_1 \\ \delta(q_2, a) = q_1 \end{array}\right\} q_0 = q_2$$

$$\left.\begin{array}{l} \delta(q_0, b) = q_2 \\ \delta(q_2, b) = q_2 \end{array}\right\} q_0 = q_2$$

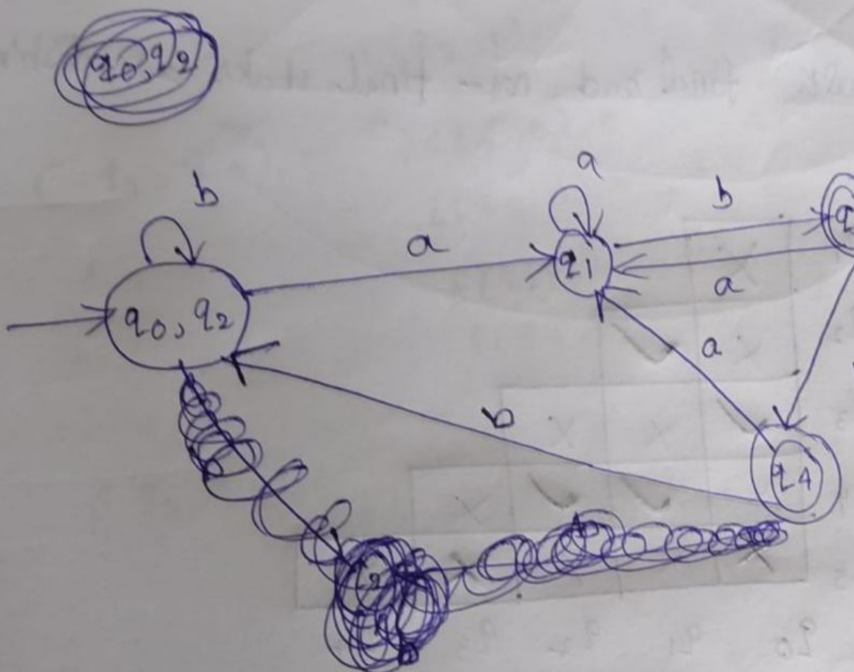
$$(q_1, q_2) \Rightarrow \left.\begin{array}{l} \delta(q_1, a) = q_1 \\ \delta(q_2, a) = q_1 \end{array}\right\} \quad q_1 \neq q_2$$

$$\left.\begin{array}{l} \delta(q_1, b) = q_3 \\ \delta(q_2, b) = q_2 \end{array}\right\}$$

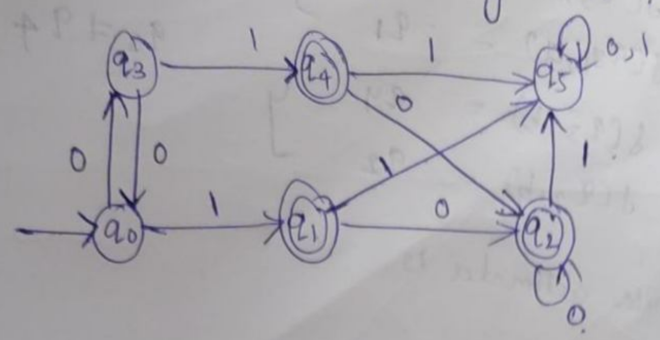
hable

$(q_3, q_4) \Rightarrow \delta(q_3, a) = q_1$
$\delta(q_4, a) = q_1$ } $q_3 \neq q_4$
$\delta(q_3, b) = q_4$
$\delta(q_4, b) = q_2$ }

∴ Minimized finite automata is



Q. Minimize the following DFA :-



Step1:- Mark final and non-final states as distinguishable

* q_1	X				
* q_2	X	✓			
q_3	✓	X	X		
* q_4	X	✓	✓	X	
q_5	X	X	X	X	X
	q_0	q_1 *	q_2 *	q_3	q_4 *

Step2:- Identify the remaining states are distinguishable.

$(q_0, q_5) = \left.\begin{array}{l} \delta(q_0, 0) = q_3 \\ \delta(q_5, 0) = q_5 \end{array}\right\}$ $\quad q_0 \neq q_5$

$\left.\begin{array}{l} \delta(q_0, 1) = q_1 \\ \delta(q_5, 1) = q_5 \end{array}\right\}$

$(q_0, q_3) = \left.\begin{array}{l} \delta(q_0, 0) = q_3 \\ \delta(q_3, 0) = q_0 \end{array}\right\}$ $\quad \therefore q_0 = q_3$ ✓

$\left.\begin{array}{l} \delta(q_0, 1) = q_1 \\ \delta(q_3, 1) = q_4 \end{array}\right\}$

From next step.

i.e $q_1 = q_4$

ngushable

$(q_1, q_4) = \delta(q_1, 0) = q_2$
$\delta(q_4, 0) = q_2$
$\delta(q_1, 1) = q_5$
$\delta(q_4, 1) = q_5$

$q_1 = q_4$ ✓

$(q_1, q_2) = \delta(q_1, 0) = q_2$
$\delta(q_2, 0) = q_2$
$\delta(q_1, 1) = q_5$
$\delta(q_2, 1) = q_5$

$q_1 = q_2$ ✓

$(q_2, q_4) = \delta(q_2, 0) = q_2$
$\delta(q_4, 0) = q_2$
$\delta(q_2, 1) = q_5$
$\delta(q_4, 1) = q_5$

$q_2 = q_4$. ✓

$(q_3, q_5) = \delta(q_3, 0) = q_0$
$\delta(q_5, 0) = q_5$
$\delta(q_3, 1) = q_4$
$\delta(q_5, 1) = q_5$

$q_3 \neq q_5$

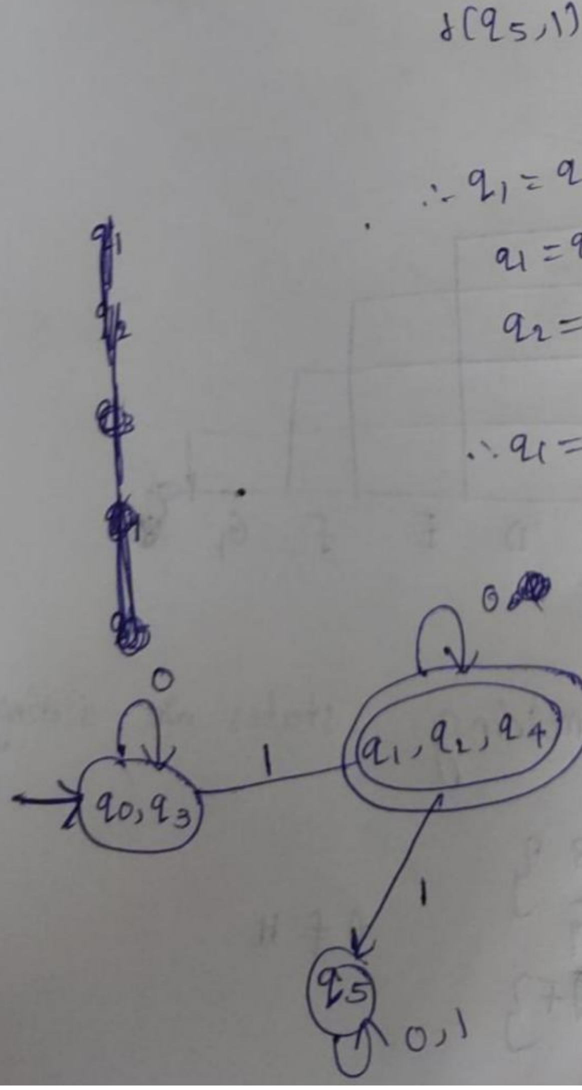
$\therefore q_1 = q_4 \quad , \quad q_0 = q_3$

$q_1 = q_2$

$q_2 = q_4$

$\therefore q_1 = q_2 = q_4$

q_1
q_2
q_3
q_4
q_5



t

$= q_4$