TAFL complete notes - bvvbv

Web Designing (Dr. A.P.J. Abdul Kalam Technical University)

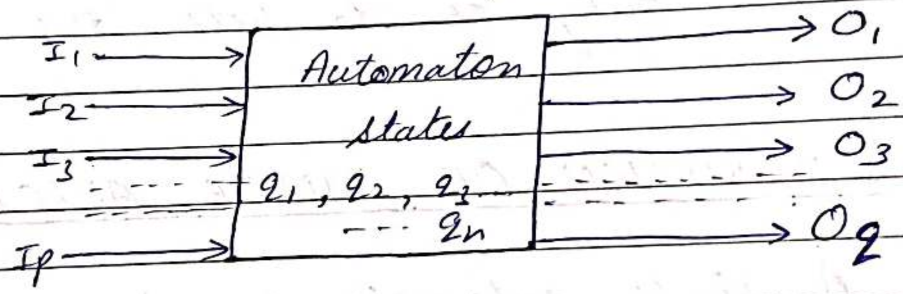
THEORY OF AUTOMATA & FORMAL LANGUAGES

Date
Page No.

AUTOMATA - An automata is defined as a system where energy, materials and information are transformed, transmitted and used for performing some functions without direct participation of human.

Ex:- Automatic machine tools, automatic packaging machine and automatic photo printing machines.

In computer science, this term means discrete automatak and is defined in more automatak way



Model of discrete automation

The characterstics of automaton are -

1. Input → At each of the discrete instants of time $T_1, T_2, T_3 \ldots T_m$, the input values $I_1, I_2, I_3, \ldots I_p$, each of which can take a finite number of fixed values from the i/p alphabet 'Σ' are applied to the i/p side of model as shown in model.

2. Output → $O_1, O_2, O_3 \cdots\cdots O_q$ are the o/p of the model, each of which can take a finite number of fixed values from an o/p 'O'.

3. States → At any instant of time, the automata can be in one of the states $q_1, q_2, q_3 \cdots\cdots q_n$.

4. State Relation → The next state of automata at any instant of time is determine by the present state and the present i/p.

5. Output Relation → The o/p is related to either state only or to both the i/p and state.

→ DESCRIPTION OF FINITE AUTOMATA -

A finite automata can be represented by a 5 tuple $(Q, \Sigma, \delta, q_0, F)$ where-

1. Q is a finite non-empty sets of states
2. Σ is a finite non-empty sets of i/p called i/p alphabets.
3. δ is a transition fxn which maps $Q \times \Sigma$ into Q i.e.

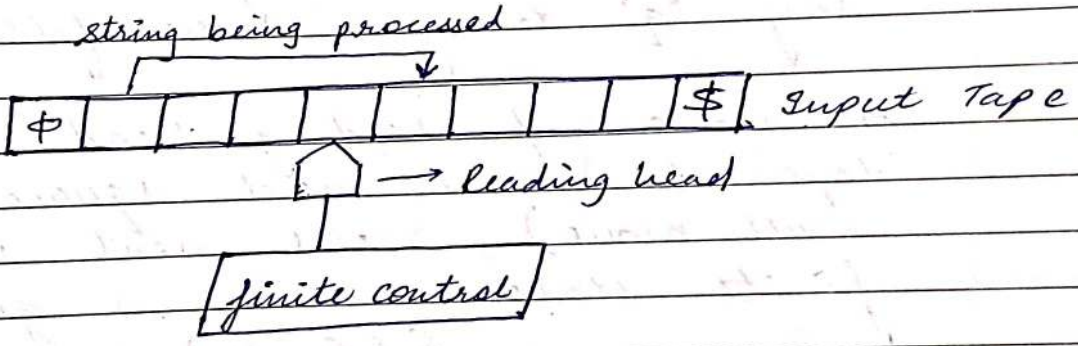
$$\delta : Q \times \Sigma \longrightarrow Q$$

and is usually called direct transition function.

This is the function which describes the change of state during transition. This mapping is usually represented by a transition table or a transition diagram.

4. $q_0 \in Q$ is the initial state.
5. $F \subseteq Q$ is the set of final states. It is assume that there may be more than one final state.

→ Block diagram of finite Automata -



The block diagram of finite automata is shown in above diagram.
The various components are explained as follows -

1. Input Tape → The input tape is divided into squares. Each square containing a single symbol from i/p alphabet. The end squares of the tape contains the end marker '¢' at left end and the end marker '$' at right end.

The absence of end markers indicates that the tape is of infinite length. The left to right sequence of symbols b/w the 2 end markers is the i/p string to be processed.

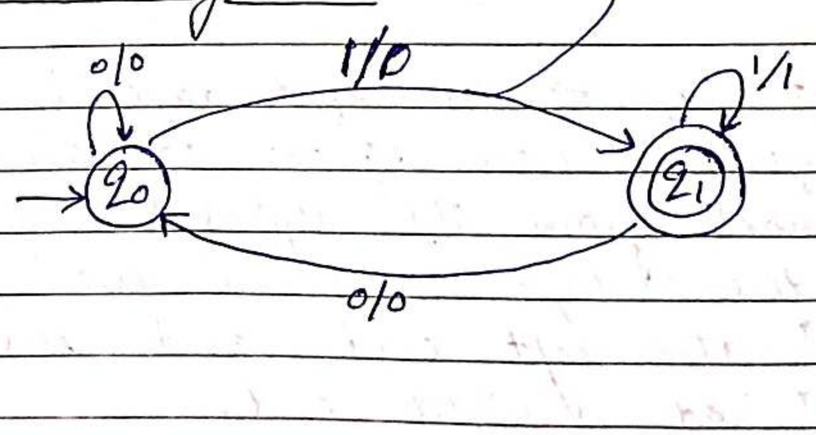
2. <u>Reading head</u> → The head examines only the one square at a time and can move one square.

3. <u>Finite Control</u> → The i/p to the finite control will usually be the symbol under the reading head say 'a' and the present state of the machine say 'q' to give the following o/p.

A motion of reading head along the tape to the next square (in some null move), the next state of the finite state machine is given by-

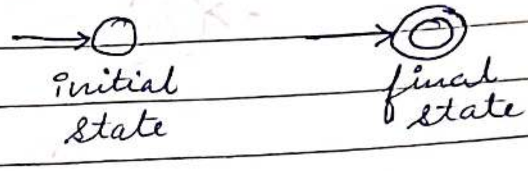
$$\delta(q, a)$$

→ <u>Transition System</u> -



$Q = \{q_0, q_1\}$

$\Sigma = \{0, 1\}$



$\delta : Q \times \Sigma \rightarrow Q$

	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_0	q_1

Ex:- 0011

$\delta(q_0, \underline{0}011)$ → initial state

$\delta(q_0, 0\underline{0}11)$

$\delta(q_0, 00\underline{1}1)$

$\delta(q_1, 001\underline{1})$

$\delta(q_1, 0011-)$ ↳ final state

i/p string

accept   reject

<u>Properties of Transition Function</u> -

<u>Property 1</u> - $\delta(q, \Lambda) = q$ (NULL)

is a finite automata.

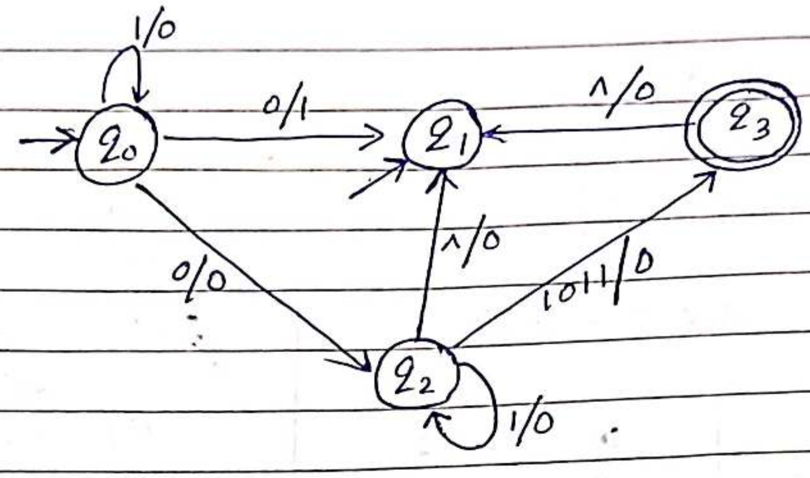
This means that, the state of system can be change only by an i/p symbol.

<u>Property 2</u> - For all strings W and i/p symbol 'a' -

$$\delta(q, aw) = \delta(\delta(q, a), w)$$

and $\delta(q, wa) = \delta(\delta(q, w), a)$.

This property gives the state after the automata consumes or read the first symbol of string 'aw' and state after the automata consumes a prefix of string 'wa'.



accept $\boxed{101011}$

$\boxed{111010}$ reject ✗

Q. Consider the finite state automata whose transition function 'δ' is given in table (transition table). Give the entire sequence of states for the i/p string 110101.

State	Input 0	Input 1
$\rightarrow q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

$\rightarrow$ $\delta(q_0, 110101) = \delta(q_1, 10101)$

$= \delta(q_0, 0101)$

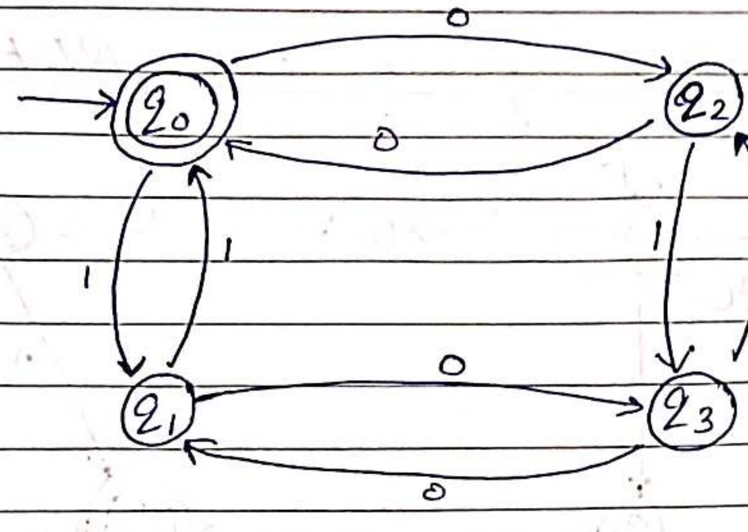
$= \delta(q_2, 101)$

$= \delta(q_3, 01)$

$= \delta(q_1, 1)$

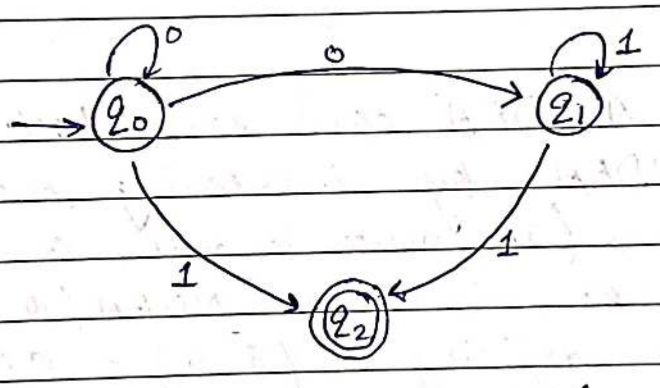
∴ string accepted $= \delta(q_0, \wedge)$

$= q_0$



The symbol ($\downarrow$) indicates that the current i/p symbol is being processed by the machine.

→ NON - DETERMINISTIC FINITE AUTOMATA -



	0	1
$\rightarrow q_0$	$[q_0, q_1]$	q_2
q_1	–	$[q_1, q_2]$
q_2	–	–

A NDFA is a 5 tuple $(Q, \Sigma, \delta, q_0, F)$

where Q = finite non-empty set of states

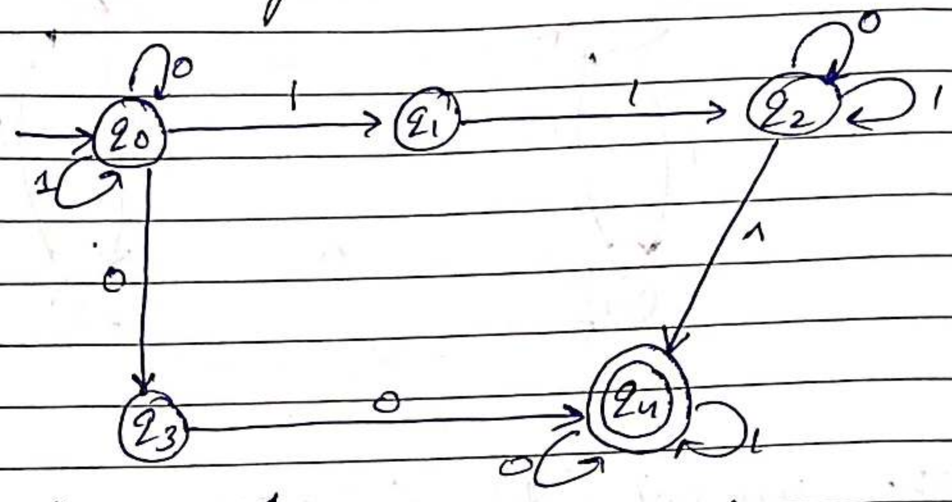
Σ = finite non-empty set of i/ps.

δ = transition fxn mapping from $Q \times \Sigma \rightarrow 2^Q$ which is the power set of Q, i.e. set of all subsets of Q.

q_0 = $q_0 \in Q$ is initial state

F = $F \subseteq Q$ is the set of final states

→ Transition System For NDFA -



$\boxed{0100}$

	0	1
q_0	$[q_0, q_3]$	$[q_0, q_1]$
q_1	-	q_2
q_2	q_2	q_2
q_3	q_4	-
q_4	q_4	q_4

$\delta(q_0, 0100) = \delta(q_0, 100$
$= \delta(q_0, 00$
$= \delta(q_3, 0)$
$= q_4$

NOTE - A DFA can stimulate the behaviour of NDFA by increasing the no. of states. (In other words, a DFA $(Q, \Sigma, \delta, q_0, F)$ can be viewed as NDFA $(Q, \Sigma, \delta', q_0, F)$ by defining $\delta'(q, a) = \delta(q, a)$)

NOTE - For every NDFA, there exist a DFA, which simulates the behaviour of NDFA. Alternatively, if 'L' is accepted by NDFA then there exist a DFA which also accepts 'L'.

Date | Page No.

Q. Construct a deterministic automata equivalent to
$M = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_0\})$
where δ is defined as –

states	I/P 0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_1	q_0, q_1

Sol^n – For deterministic automata M_1,

1. The states are subsets of $\{q_0, q_1\}$ –
   ϕ, $[q_0]$, $[q_1]$, $[q_0, q_1]$
2. $[q_0]$ is the initial state
3. $[q_0]$, $[q_0, q_1]$ are the final states as these are the only states containing q_0.
4. δ is defined by the state transition table given below –

states	Input 0	1
ϕ	ϕ	ϕ
$[q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1]$	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_0, q_1]$

Q. Find deterministic automata equivalent to
$M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$
where δ is defined as –

states	a	b
$\rightarrow q_0$	q_0, q_1	q_2
q_1	q_0	q_1
q_2	–	q_0, q_1

Sol^n- States for DFA = subset of $\{q_0, q_1, q_2\}$

ϕ, $[q_0]$, $[q_1]$, $[q_2]$, $[q_0, q_1]$, $[q_1, q_2]$, $[q_2, q_0]$, $[q_0, q_1, q_2]$

old method

states	a	b
ϕ	ϕ	ϕ
$[q_0]$	$[q_0, q_1]$	$[q_2]$
$[q_1]$	$[q_0]$	$[q_1]$
$[q_2]$		$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_2, q_1]$
$[q_1, q_2]$	$[q_0]$ ~~q~~	$[q_0, q_1]$
$[q_2, q_0]$	$[q_0, q_1]$	$[q_0, q_1, q_2]$
$[q_0, q_1, q_2]$	$[q_0, q_1]$	$[q_0, q_1, q_2]$

new method

initial state = $[q_0]$

final states = $[q_2]$, $[q_1, q_2]$, $[q_2, q_0]$, $[q_0, q_1, q_2]$

States	Input a	b
$\rightarrow [q_0]$	$[q_0, q_1]$	$[q_2]$
$[q_2]$	ϕ	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_0]$	$[q_0, q_1]$

Date | Page No.

Q. Construct a DFA equivalent to M –

$M = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, \{q_3\})$

where δ is defined as –

	a = 0	b = 1
q_0	q_0, q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	–	q_2

Sol^n –

initial state = q_0

final states = $[q_3]$, $[q_0, q_3]$, $[q_1, q_3]$, $[q_2, q_3]$, $[q_0, q_1, q_3]$, $[q_0, q_2, q_3]$, $[q_0, q_1, q_2, q_3]$

	a = 0	b = 1
$\rightarrow q_0$	$[q_0, q_1]$	q_0
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0, q_1]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3]$
$[q_0, q_1, q_3]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$
$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3, q_2]$

Q. Construct a DFA equivalent to $M = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, \{q_3\})$

where δ is defined as-

	a = 0	b = 1
q_0	q_0, q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	–	q_2

Solⁿ- initial state = q_0

final states = $[q_3]$, $[q_0, q_3]$, $[q_1, q_3]$, $[q_2, q_3]$, $[q_0, q_1, q_3]$, $[q_0, q_2, q_3]$, $[q_0, q_1, q_2, q_3]$

	a = 0	b = 1
$\rightarrow q_0$	$[q_0, q_1]$	q_0
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0, q_1]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3]$
$[q_0, q_1, q_3]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$
$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3, q_2]$

→ Mealy and Moore Machine -

Moore Machine - A Moore machine is a 6 tuple $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ where -

Q is a finite non-empty set of state

Σ is a finite non-empty set of i/p alphabets.

Δ is a finite non-empty set of o/p alphabets.

δ is a transition fxn from $Q \times \Sigma \rightarrow Q$

λ is the o/p fxn mapping from $\lambda : Q \rightarrow \Delta$

q_0 is the initial state.

Example of Moore Machine

Present State	Next state $a=0$	Next state $a=1$	λ
$\rightarrow q_0$	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

i/p string 0111

$$q_0 \xrightarrow{0} q_3 \xrightarrow{1} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_2$$

$$\begin{array}{ccccc} q_0 & q_3 & q_0 & q_1 & q_2 \\ \downarrow o/p & \downarrow o/p & \downarrow o/p & \downarrow o/p & \downarrow o/p \\ 0 & 0 & 0 & 1 & 0 \end{array}$$

o/p string ⇒ 00010

For the i/p string 0111, the transition of states is given by.

and the o/p string is — 00010.

For the i/p string NULL, the o/p is the o/p of ~~pre~~ initial state.

Mealy Machine :- A mealy machine is a 6 tuple $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ where
Q is a finite non-empty set of states.
$\lambda : Q \times \Sigma \longrightarrow \Delta$
rest all are same

Present State	Next State $a=0$ State	o/p	$a=1$ state	o/p
$\rightarrow q_1$	q_3	0	q_2	0
q_2	q_1	1	q_4	0
q_3	q_2	1	q_1	1
q_4	q_4	1	q_3	0

I/p string : 0011 $\qquad \lambda : Q \times \Sigma \rightarrow \Delta$

$$q_1 \xrightarrow{0} q_3 \xrightarrow{0} q_2 \xrightarrow{1} q_4 \xrightarrow{1} q_3$$
$$0 \quad 1 \quad 0 \quad 0$$

$\boxed{0100} \Rightarrow$ o/p string

→ **Procedure for transforming a Mealy Machine to Moore Machine -**

Q. Convert given mealy machine to moore machine

Present State	Next state $a=0$ state	o/p	$a=1$ state	o/p
q_1	q_3	0	q_2	0
q_2	q_1	1	q_4	0
q_3	q_2	1	q_1	1
q_4	q_4	1	q_3	0

I/p string : 0011

Solⁿ- In this problem, q_1 is associated with 1 o/p and q_2 is associated with 2 diff o/ps : 0 & 1. Similarly q_3 is associated with 1 o/p and q_4 is associated with 2 diff o/ps : 0 & 1.

Now, we will split q_2 into q_{20} and q_{21}.
Similarly, q_4 is split into q_{40} and q_{41}.

Now the given table can be reconstructed for the new states as follows:

Present state	Next state a=0	o/p	a=1	o/p
$\rightarrow q_1$	q_3	0	q_{20}	0
q_{20}	q_1	1	q_{40}	0
q_{21}	q_1	1	q_{40}	0
q_3	q_{21}	1	q_1	1
q_{40}	q_{41}	1	q_3	0
q_{41}	q_{41}	1	q_3	0

The Pair of states and o/p in the next state problem can be rearranged as given below :-

Present state	Next state a=0	a=1	o/p
$\rightarrow q_1$	q_3	q_{20}	1
q_{20}	q_1	q_{40}	0
q_{21}	q_1	q_{40}	1
q_3	q_{21}	q_1	0
q_{40}	q_{41}	q_3	0
q_{41}	q_{41}	q_3	1

Date
Page No.

The above table gives the moore machine, here we observe that, the o/p of initial state is 1, it means that if i/p is null (^) then we get o/p 1. To overcome this situation, we must add a new starting state q_0 whose state transition are identical with those of q_1 but the o/p will be zero.

Present state	Next state a=0	a=1	o/p
→ q_0	q_3	q_{20}	0
q_1	q_3	q_{20}	1
q_{20}	q_1	q_{40}	0
q_{21}	q_1	q_{40}	1
q_3	q_{21}	q_1	0
q_{40}	q_{41}	q_3	0
q_{41}	q_{41}	q_3	1

→ Procedure for transforming a Moore machine into mealy machine -

Q. Construct a mealy machine which is equivalent to the moore machine given in table.

Present state	Next state a=0	a=1	o/p
q_0	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

Date | Page No.

Sol^n-

Present state	Next state a=0	o/p	a=1	o/p
q_0	q_3	0	q_1	1
q_1	q_1	1	q_2	0
q_2	q_2	0	q_3	0
q_3	q_3	0	q_0	0

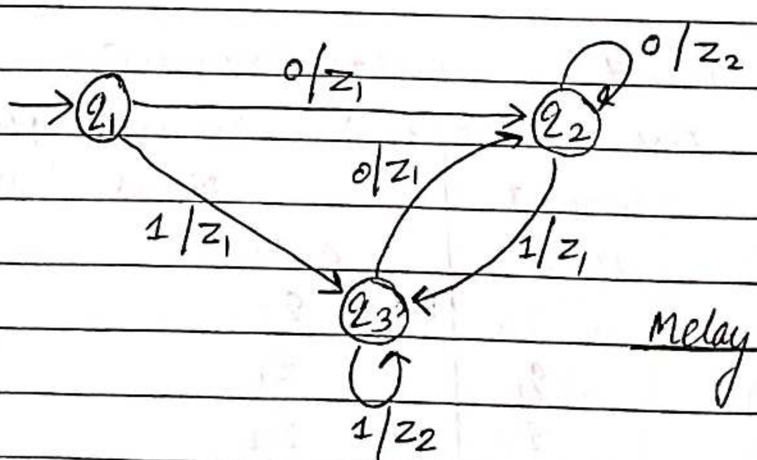
Q. Moore Machine.

	a=0	a=1	o/p
q_1	q_1	q_2	0
q_2	q_1	q_3	0
q_3	q_1	q_3	1

→ Melay Machine

	a=0	o/p	a=1	o/p
q_1	q_1	0	q_2	0
q_2	q_1	0	q_3	1
X. q_3	q_1	0	q_3	1

Q. Consider a melay machine as shown in diagram. Construct a moore machine to this melay machine.



Melay Machine

Date | Page No.

Sol^n-

Present State	Next State a=0 State	o/p	a=1 State	o/p
$\rightarrow q_1$	q_2	z_1	q_3	z_1
q_2	q_2	z_2	q_3	z_1
q_3	q_2	z_1	q_3	z_2

Check next state and split those state which have diff o/p

split q_2 into q_{21} and q_{22}

split q_3 into q_{31} and q_{32}

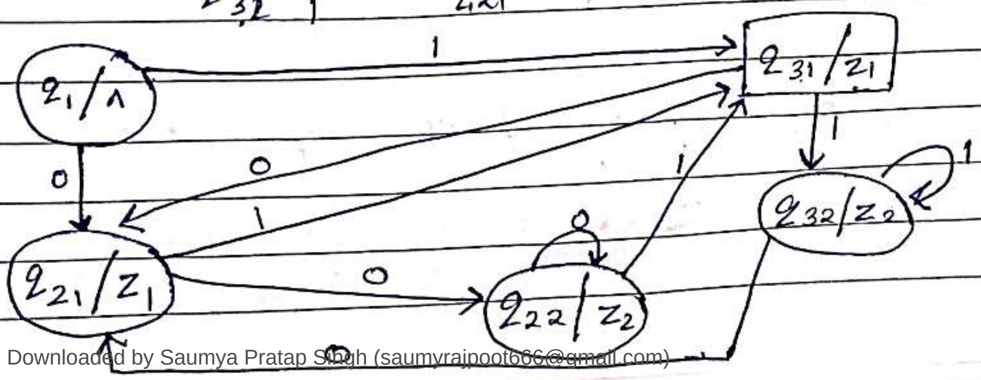
$q_1 \longrightarrow \wedge$

Present state	Next State a=0	o/p	a=1	o/p
$\rightarrow q_1$	q_{21}	z_1	q_{31}	z_1
q_{21}	q_{22}	z_2	q_{31}	z_1
q_{22}	q_{22}	z_2	q_{31}	z_1
q_{31}	q_{21}	z_1	q_{32}	z_2
q_{32}	q_{21}	z_1	q_{32}	z_2

Now, all internal states have same o/p.

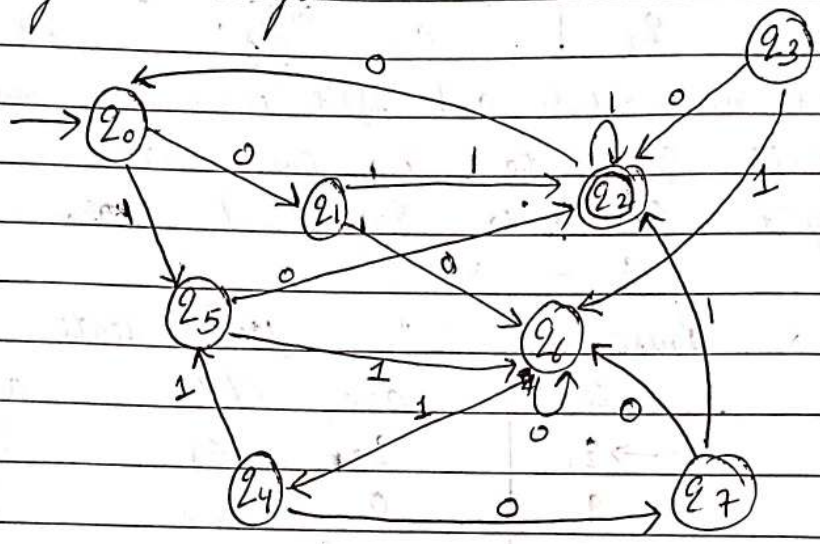
Moore Machine

Present state	Next State a=0	a=1	o/p
$\rightarrow q_1$	q_{21}	q_{31}	$\wedge$
q_{21}	q_{22}	q_{31}	z_1
q_{22}	q_{22}	q_{31}	z_2
q_{31}	q_{21}	q_{32}	z_1
q_{32}	q_{21}	q_{32}	z_2



→ Minimization of automata -

Q. Construct a minimum state automata equivalent to the finite automata in given diagram.



Sol^n -

Present state	$a=0$	$a=1$
$\rightarrow q_0$	q_1	q_5
q_1	q_6	q_2
q_2	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

By appling first step, we get -

$Q_1^0 = F = \{q_2\}$

$Q_2^0 = Q - Q_1^0 = \{q_1, q_3, q_4, q_5, q_6, q_7\}$

Date

Page No

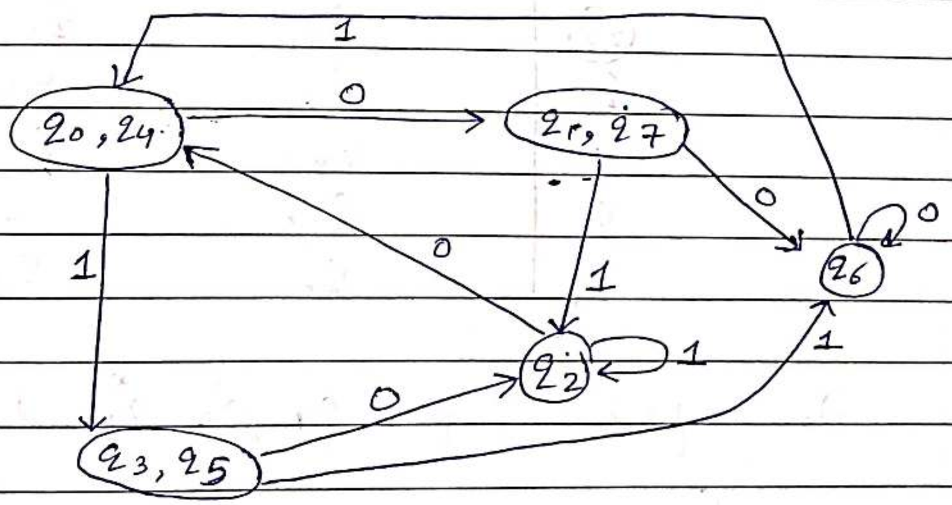
$\pi_0 = \{\{q_2\}, \{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}\}$

$\pi_1 = \{\{q_2\}, \{q_0, q_4, q_6\}, \{q_1, q_7\}, \{q_3, q_5\}\}$

$\pi_2 = \{\{q_2\}, \{q_0, q_4\}, \{q_6\}, \{q_1, q_7\}, \{q_3, q_5\}\}$

Minimum state -

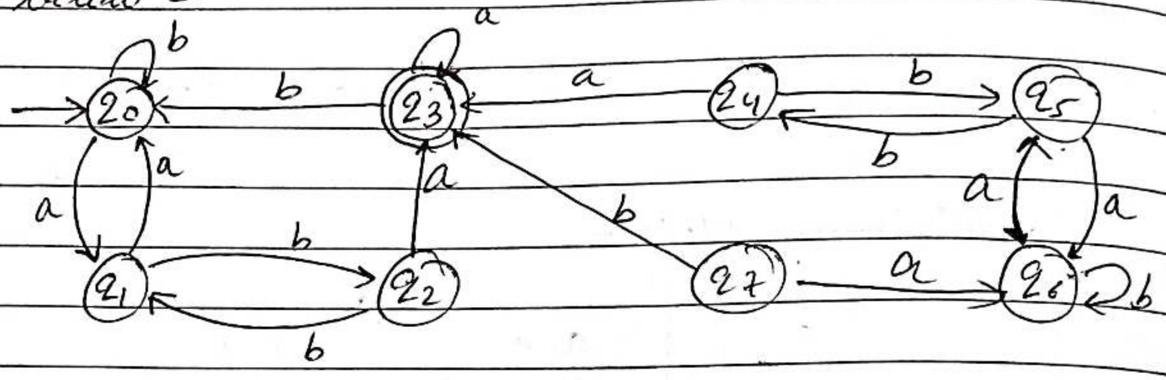
	a=0	a=1
→ [q_0, q_4]	[q_1, q_7]	[q_3, q_5]
[q_1, q_7]	[q_6]	[q_2]
[q_2]	[q_0, q_4]	[q_2]
[q_3, q_5]	[q_2]	[q_6]
[q_6]	[q_6]	[q_0, q_4]



This is the required minimum state automata.

Date | Page No.

Q. Construct a minimum state automata equivalent to the transition diagram given below -



Sol^n -

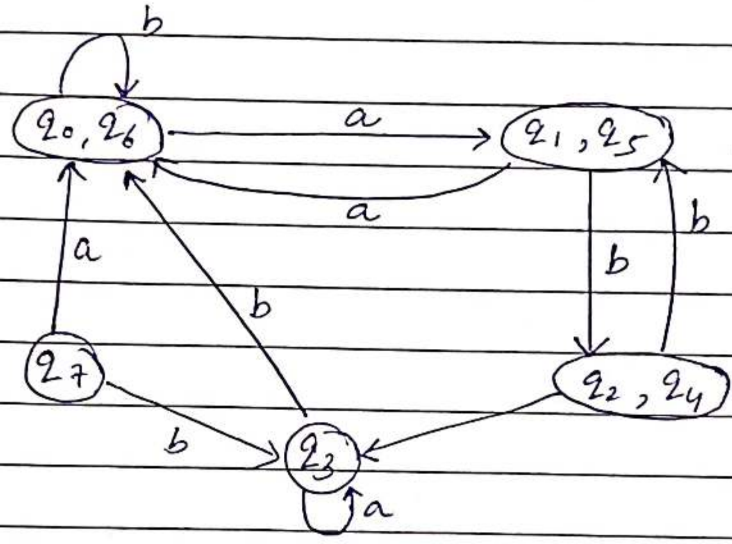
Present state	Next state A = a	B = b
$\rightarrow q_0$	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_3	q_0
q_4	q_3	q_5
q_5	q_6	q_4
q_6	q_5	q_6
q_7	q_6	q_3

$\pi_0 = \{\{q_3\}, \{q_0, q_1, \overset{*}{q_2}, \overset{*}{q_4}, q_5, q_6, q_7\}\}$

$\pi_1 = \{\{q_3\}, \{q_2, q_4\}, \{q_7\}, \{q_0, q_1, q_5, q_6\}\}$

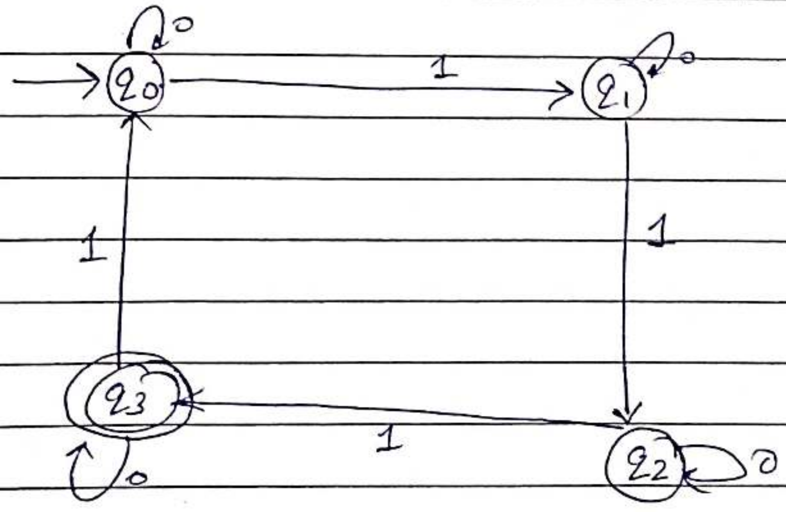
$\pi_2 = \{\{q_3\}, \{q_2, q_4\}, \{q_7\}, \{q_0, q_6\}, \{q_1, q_5\}\}$

	A = a	B = b
→ $[q_0, q_6]$	$[q_1, q_5]$	$[q_0, q_6]$
$[q_1, q_5]$	$[q_0, q_6]$	$[q_2, q_4]$
$[q_2, q_4]$	$[q_3]$	$[q_1, q_5]$
$[q_3]$	$[q_3]$	$[q_0, q_6]$
$[q_7]$	$[q_0, q_6]$	$[q_3]$



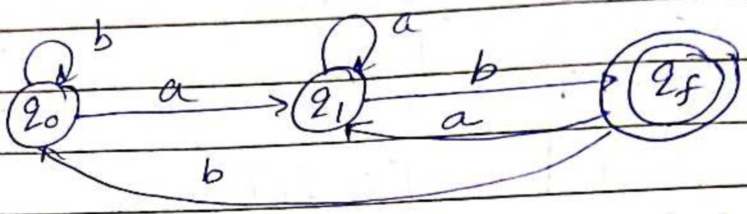
Q. Construct a DFA accepting all strings W over {0,1} such that the no. of 1's in W is 3 mod 4.

Soln.



Q. Construct a DFA accepting all string over {a, b} ending in ab

→



- The regular expressions are useful for representing certain sets of string in an algebric fashion
- These describes the languages accepted by finite state automata.
- Any terminal symbol (i.e., an element of Σ) null (∧) and ϕ are regular expression. When we view 'a' in Σ as a regular expression, we denote it by a.
- The union of 2 regular expression R_1 and R_2 written as $(R_1 + R_2)$ is also a regular expression.
- The concatination of 2 regular expression R_1 and R_2 written as R_1R_2 is also a regular expression
- The iteration (or closure) of a regular expression 'R' written as R^* is also a regular expression
- If capital R is a regular expression then (R) is also a regular expression.

Date

Page No.

Ex:- Describe the following sets by regular expression.

(a) $\{101\}$ (e) $\{abb, a, b, bba\}$

(b) $\{abba\}$ (f) $\{\wedge, 0, 00, 000, \ldots\}$

(c) $\{01, 10\}$ (g) $\{1, 11, 111, \ldots\}$

(d) $\{\wedge, ab\}$

Solⁿ:- (a) 101

(b) $abba$

(c) $01 + 10$

(d) $\wedge + ab$

(e) $abb + a + b + bba$

(f) 0^*

(g) $1(1^*)$

Ex:- Describe the following sets by regular expressions -

(i) L_1 = set of all strings of $\{0, 1\}$ ending in 00.

(ii) L_2 = set of all strings of $\{0, 1\}$ beginning with 0 and ending with 1.

(iii) $L_3 = \{\wedge, 11, 1111, 111111, \ldots\}$

Solⁿ:- (i) $(0+1)^* 00$

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

(iii) $(11)^*$

Date | Page No.

→ Identities for Regular Expression –

I_1 $\quad \Phi + R = R$

I_2 $\quad \Phi R = R\Phi = \Phi$

I_3 $\quad \Lambda R = R\Lambda = R$

I_4 $\quad \Lambda^* = \Lambda$ and $\Phi^* = \Lambda$

I_5 $\quad R + R = R$

I_6 $\quad R^* R^* = R^*$

I_7 $\quad RR^* = R^* R$

I_8 $\quad (R^*)^* = R^*$

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

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

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

I_{12} $\quad (P+Q)R = PR + QR$

$\quad\quad R(P+Q) = RP + RQ$

→ Transition System containing null moves –

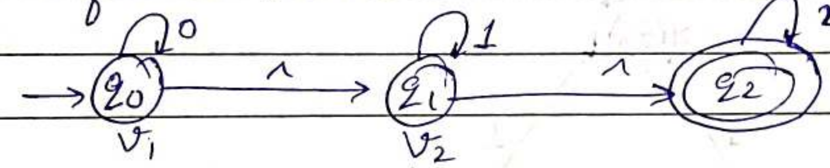
Suppose we want to replace a null move from a vertex V_1 to vertex V_2 then we proceed as follows –

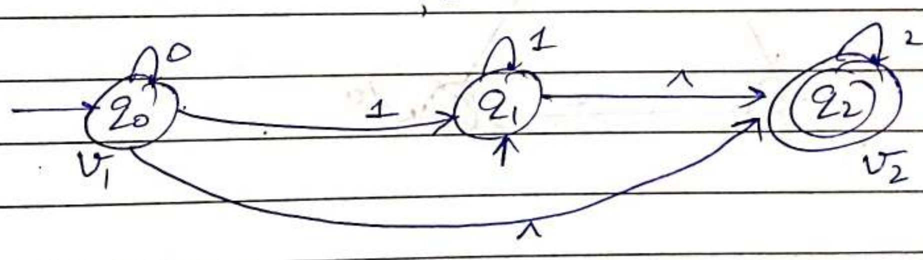
Step-1 Find all edges starting from vertex V_2.

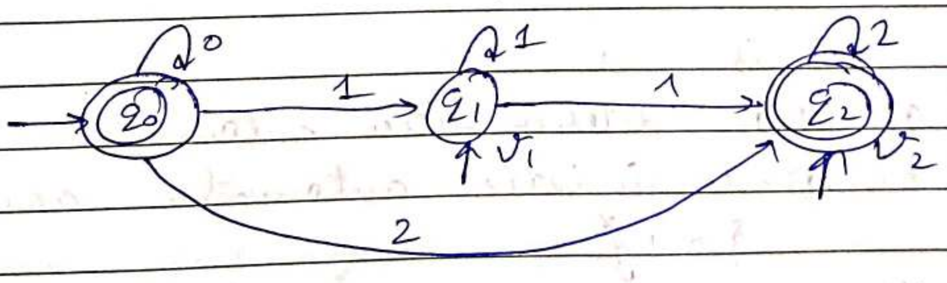
Step-2 Duplicate all these edges starting from vertex V_1 without changing the edge labels.

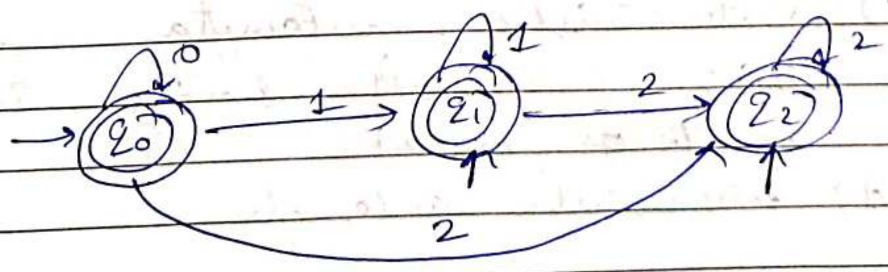
Step-3 If V_1 is an initial state, make V_2 also as initial state.

Step-4 If V_2 is final state, make V_1 also as final state.

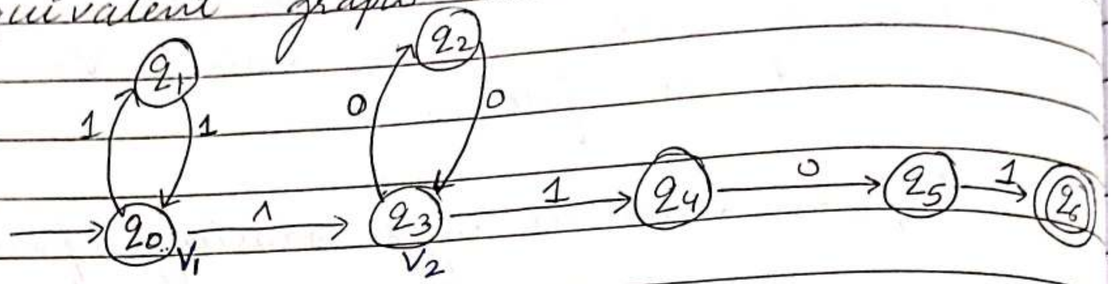


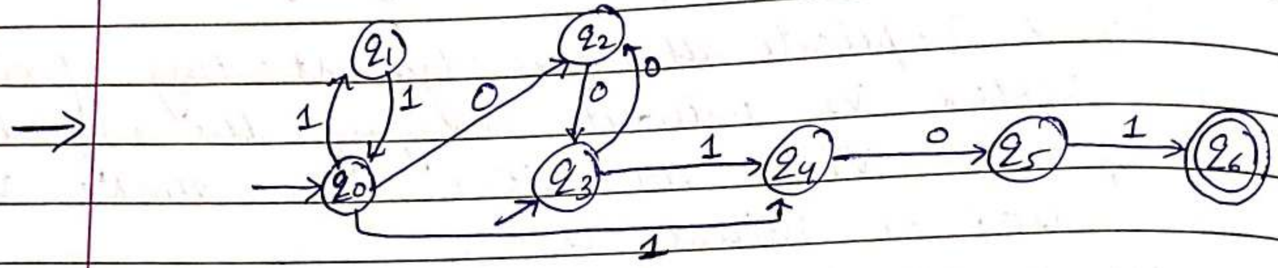




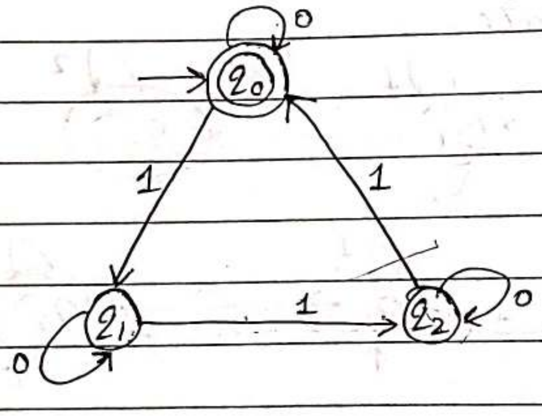


Q. Consider a graph containing a null move as shown in diagram. Obtain an equivalent graph without null (^) move





Q.



① M is a

✓ a) Non deterministic automata

✗ b) Deterministic automata accepting $\{0,1\}^*$

✗ c) Deterministic automata accepting all string over $\{0,1\}$ having $3m$ 0's and $3n$ 1's, $m, n \geq 1$

✓ d) deterministic automata

② M accepts

~~a)~~ 01110 b) 10001 c) 01010 d) 11111

③ T(M) =

a) $\{0^{3m} 1^{3n} : m, n \geq 0\}$

b) $\{0^{3m} 1^{3n} : m, n \geq 1\}$

✓c) $\{W \mid W$ has 111 as a sub string$\}$

(d) $\{W \mid W$ has $3n$ 1's, $n \geq 1\}$

④ If q_2 is also made a final state then M accepts

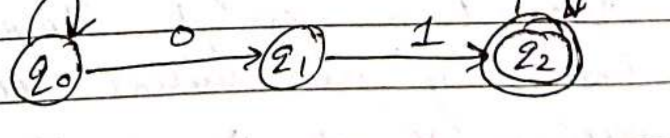
✓(a) 01110 and 01100

(b) 10001 and 10000

(c) 0110 but not 011101

(d) $0^{3n}, n \geq 1$ but not $1^{3n}, n \geq 1$

Q. Study the automata given below and state whether the given statements are true or false.



Date
Page No.

Sol^n- M is a non-deterministic automata.

$\delta(q_1, 1)$ is defined - Yes it is defined

0100111 is accepted by M → True

010101010 is not accepted by M → False

→ ## ARDEEN'S THEOREM

Let P and Q are 2 regular expression over summation. If P doesn't contain NULL, then the following eq^n in R →
$R = Q + RP$ has a unique solution
$R = QP^*$

Algebric Method using ARDEEN'S Theorem-

The following method is an extension of Arden's theorem. This is used to find out the regular expression recognised by a transition system.

The following assumptions are made regarding the transition system-

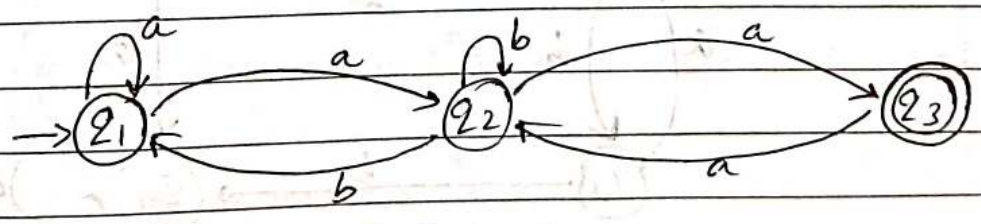
1. The transition graph doesn't have NULL move.
2. It has only one initial state say V_1.
3. It's vertices $V_1, V_2, V_3, \dots V_n$
4. V_i, the regular expression, represent the set of strings accepted by the system even though V_i is a final state.

5. α_{ij}, denotes the regular expression representing the set of labels of edge from v_i to v_j. When there is no such edge, $\alpha_{ij} = \phi$. Consequently, we get following set of equations:

$$V_1 = V_1\alpha_{11} + V_2\alpha_{21} + \cdots\cdots + V_n\alpha_{n1} + \Lambda$$
$$V_2 = V_1\alpha_{12} + V_2\alpha_{22} + \cdots + V_n\alpha_{n2}$$
$$\vdots$$
$$V_n = V_1\alpha_{1n} + V_2\alpha_{2n} + \cdots\cdots + V_n\alpha_{nn}$$

By repeatedly applying substitution and the arden's theorem, we can express v_i in terms of α_{ij}, we have to take the unions of all v_i's corresponding to the final state.

Ex:- Consider the transition system as shown in diagram. Prove that the strings recognised are
$$(a + a(b+aa)^*b)^*a(b+aa)^*a$$



Sol^n-
$$q_1 = q_1a + q_2b + \Lambda$$
$$q_2 = q_1a + q_2b + q_3a$$
$$q_3 = q_2a$$

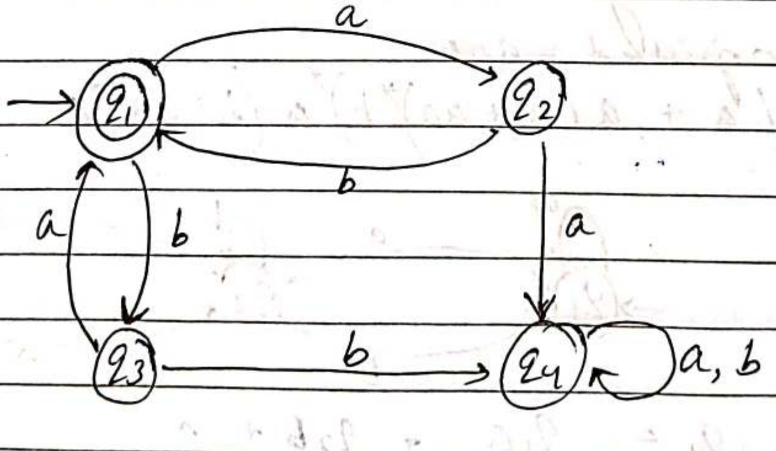
$$q_2 = q_1 a + q_2 b + q_3 a$$
$$q_2 = q_1 a + q_2 b + q_2 aa$$
$$q_2 = q_1 a + q_2 (b + aa)$$

$$R = Q + RP$$
Sol^n $\longrightarrow$ $R = QP^*$
$$q_2 = q_1 a (b+aa)^*$$
$$q_1 = q_1 a + q_1 a (b+aa)^* b + \Lambda$$
$$q_1 = q_1 (a + a(b+aa)^* b) + \Lambda$$
$$R = RP + Q$$

$$q_1 = \Lambda (a + a(b+aa)^* b)^*$$
$$q_1 = (a + a(b+aa)^* b)^*$$

$$q_2 = (a + a(b+aa)^* b)^* a (b+aa)^*$$
$$q_3 = (a + a(b+aa)^* b)^* a (b+aa)^* a$$

Ex: Find the regular expression for a given finite automata.



Sol^n →

$$q_1 = q_2 b + q_3 a + \Lambda$$
$$q_2 = q_1 a$$
$$q_3 = q_1 b$$
$$q_4 = q_3 b + q_4 a + q_4 b + q_2 a$$

$$q_1 = q_1 ab + q_1 ba + \Lambda$$
$$\underbrace{q_1}_{R} = \underbrace{q_1}_{R}(\underbrace{2ab}_{P}) + \underbrace{\Lambda}_{Q}$$
$$R = RP + Q$$

Sol^n → $R = QP^*$

$$= \Lambda(2ab)^*$$
$$\boxed{q_1 = (2ab)^*}$$

$$\boxed{q_2 = (2ab)^* a} \qquad \boxed{q_3 = (2ab)^* b}$$

$$q_1 = q_1(ab + ba) + \Lambda$$
$$R = RP + Q$$
$$R = QP^*$$
$$q_1 = \Lambda(ab + ba)^*$$
$$\boxed{q_1 = (ab+ba)^*}$$

$$\boxed{q_2 = (ab + ba)^* a} \qquad \boxed{q_3 = (ab+ba)^* b}$$

$$q_4 = q_4(a+b) + (ab+ba)^* bb + (ab+ba)^* aa$$

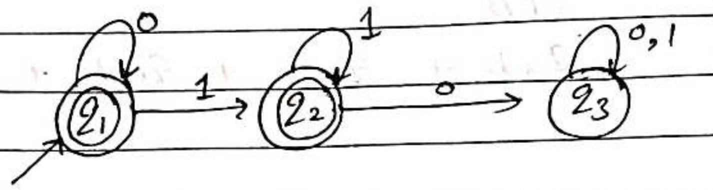
$$\boxed{q_4 = (ab+ba)^*(bb+aa)(a+b)^*}$$

no need

$1^* = (\wedge, 1, 11$

Date
Page No.

Q. Find regular expression for given transition diagram.



Sol^n →

$$q_1 = q_1 0 + \wedge$$
$$q_2 = q_1 1 + q_2 1$$
$$q_3 = q_2 0 + q_3 0 + q_3 1$$

$$\boxed{q_1 = \wedge 0^* = 0^*}$$

$$q_2 = q_2 1 + 0^* 1$$

$$\boxed{q_2 = (0^* 1) 1^*}$$

$$q_3 = q_3 (0+1) + 0^* 1 1^* 0$$

$$\boxed{q_3 = 0^* 1 1^* 0 (0+1)^*}$$ no need

Answer will be the union of final state.

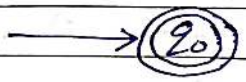
$$q_1 + q_2 = 0^* + (0^* 1) 1^*$$
$$= 0^* (\wedge + 1 1^*)$$
$$= 0^* 1^*$$

$$\boxed{q_1 + q_2 = 0^* 1^*}$$

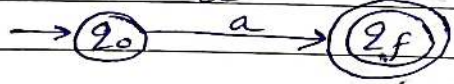
→ Kleene's Theorem -

If R is a regular expression over summation representing $L \subseteq \Sigma^*$ then there exist an N.DFA M with null moves such that

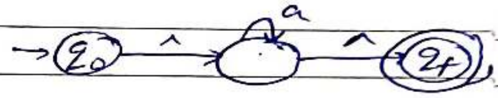
$\boxed{L = T(M)}$



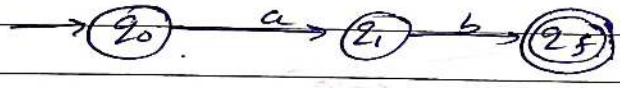
a



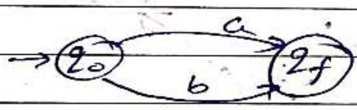
a^*



ab

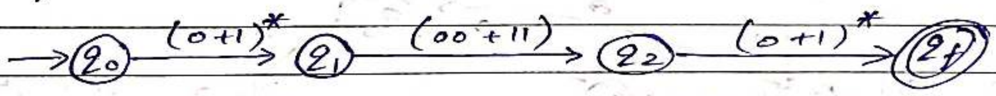


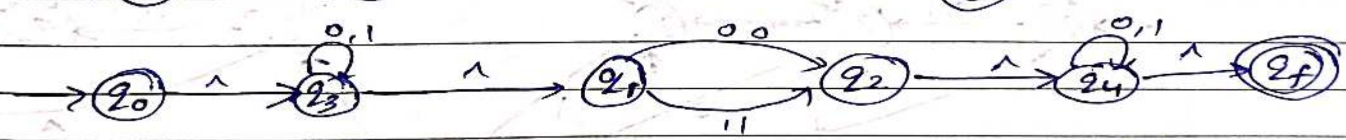
a+b

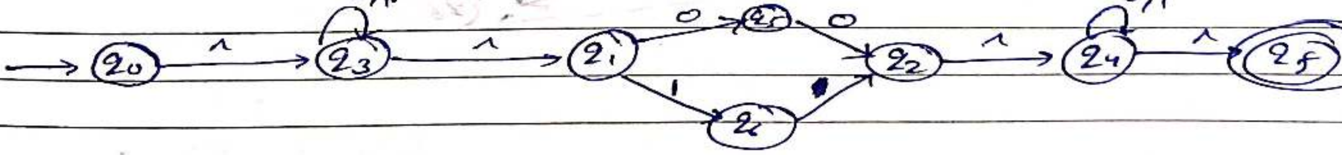


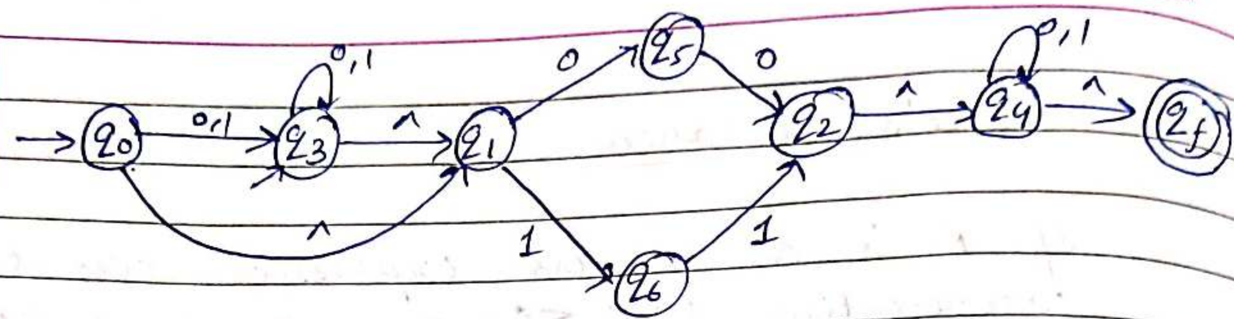
Q. Construct a DFA equivalent to regular expression $(0+1)^*(00+11)(0+1)^*$

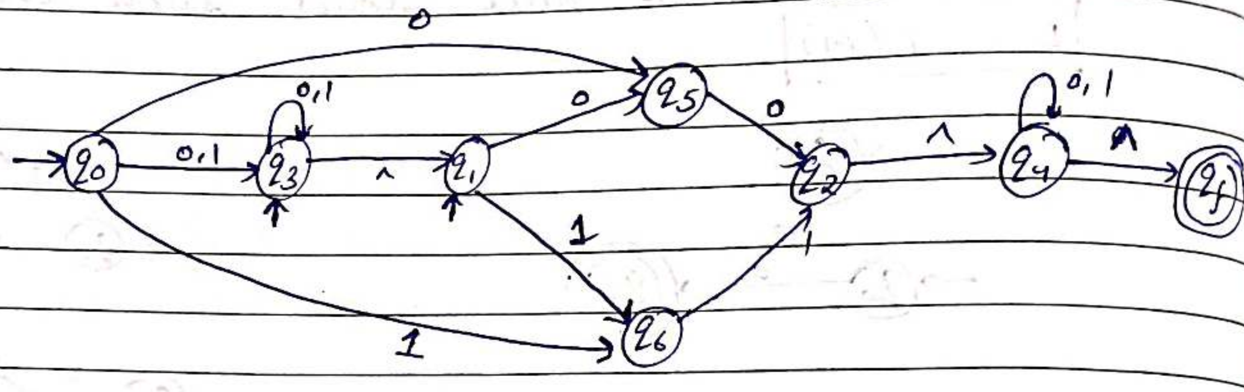
Sol^n-

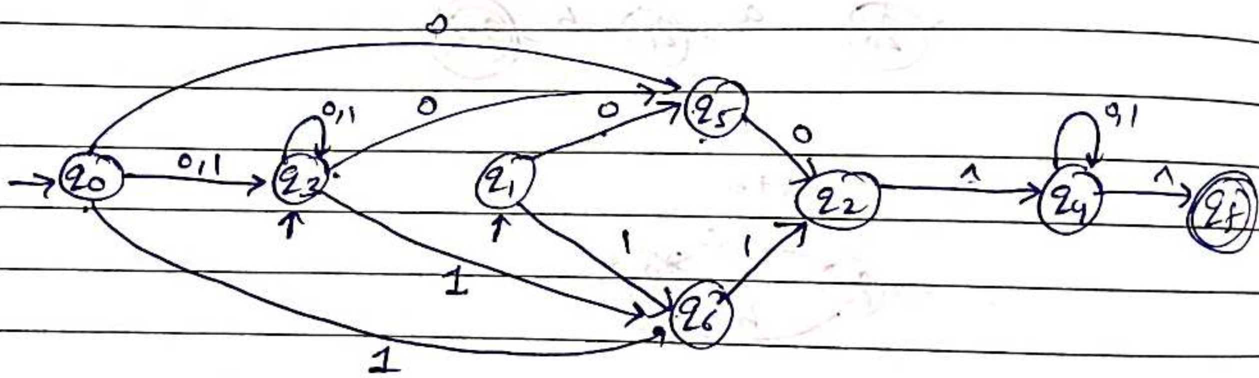


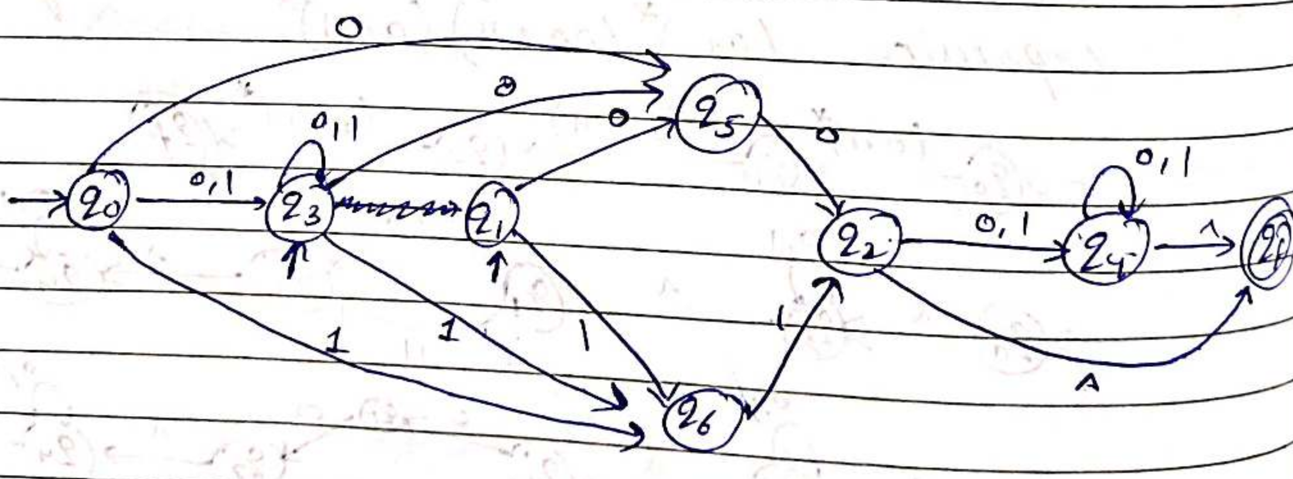


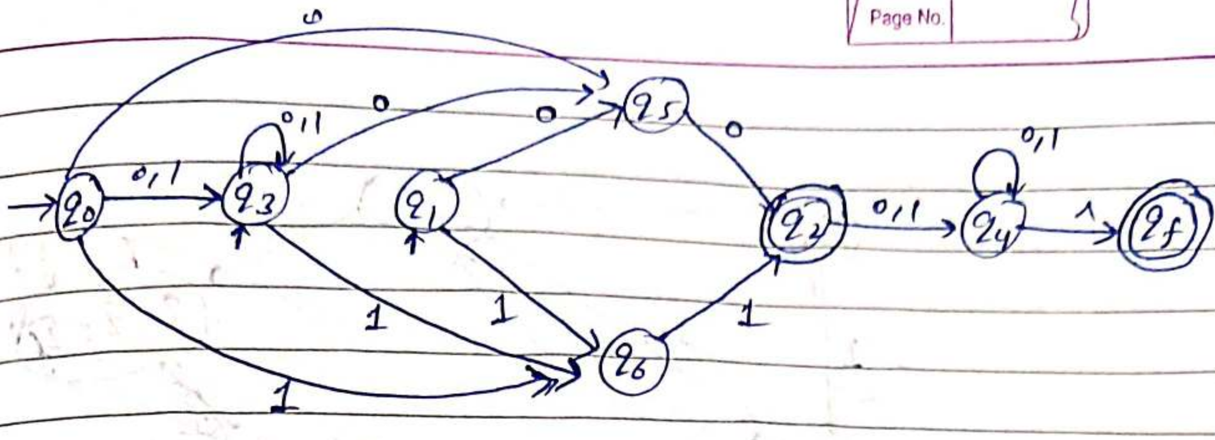


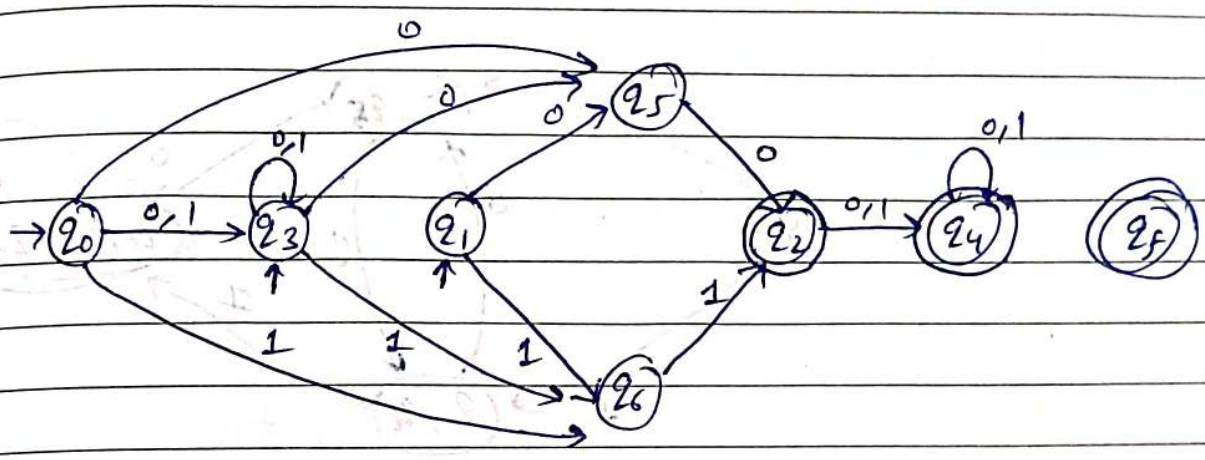

q0
0,1
q3
0,1
^
q1
0
q5
0
q2
^
q4
0,1
^
qf
^
1
q6
1

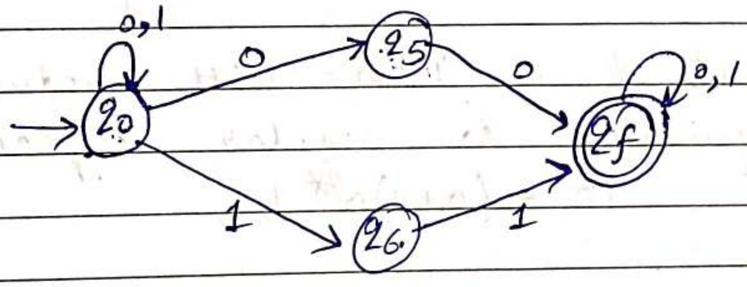

0
q0
0,1
q3
0,1
^
q1
0
q5
0
q2
^
q4
0,1
^
qf
1
q6
1
1


0
q0
0,1
q3
0,1
0
q1
0
q5
0
q2
^
q4
0,1
^
qf
1
1
1
q6
1


0
0
q0
0,1
q3
0,1
q1
0
q5
0
q2
0,1
q4
0,1
^
qf
^
1
1
1
1
q6



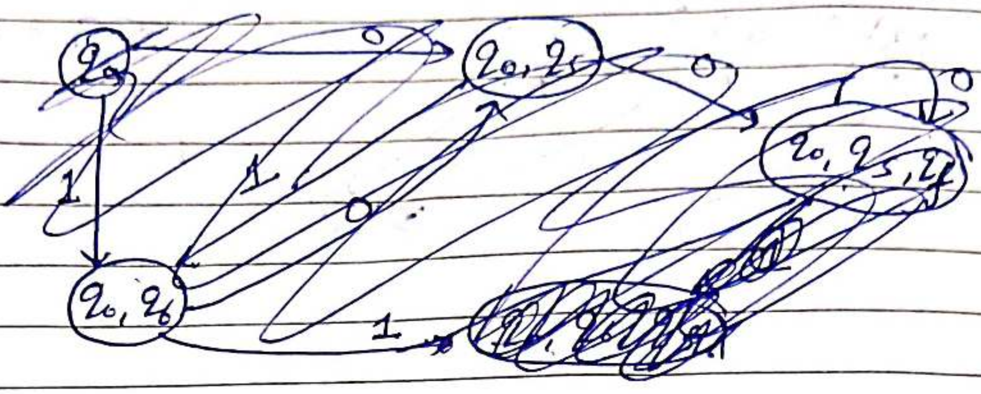


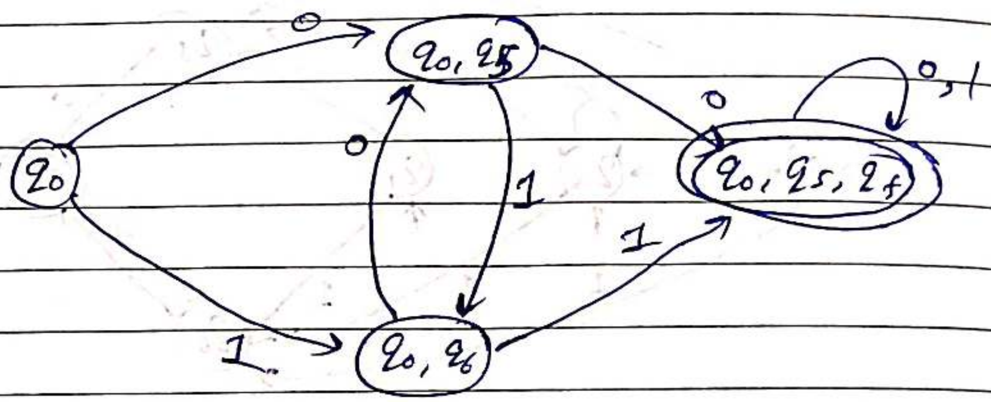


initial state $= [q_0]$

final state $= [q_f], [q_0, q_f], [q_5, q_f], [q_6, q_f],$
$[q_0, q_5, q_f], [q_0, q_6, q_f], [q_5, q_6, q_f]$
$[q_0, q_5, q_6; q_f]$

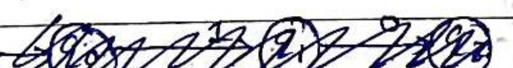
	0	1
$\rightarrow [q_0]$	$[q_0, q_5]$	$[q_0, q_6]$
$[q_0, q_5]$	$[q_0, q_5, q_f]$	$[q_0, q_6]$
$[q_0, q_6]$	$[q_0, q_5]$	$[q_0, q_6, q_f]$
same{ $[q_0, q_5, q_f]$	$[q_0, q_5, q_f]$	$[q_0, q_6, q_f]$
$[q_0, q_6, q_f]$	$[q_0, q_5, q_f]$	$[q_0, q_6, q_f]$

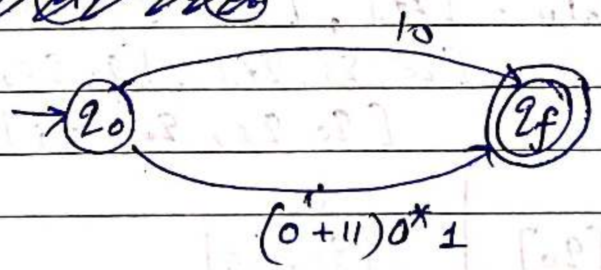


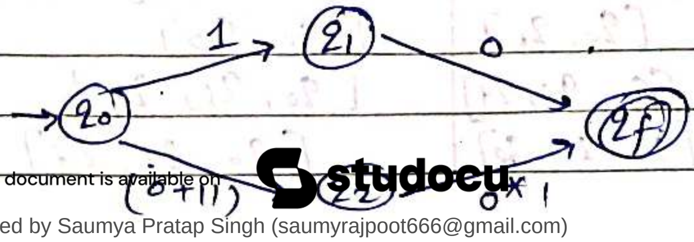


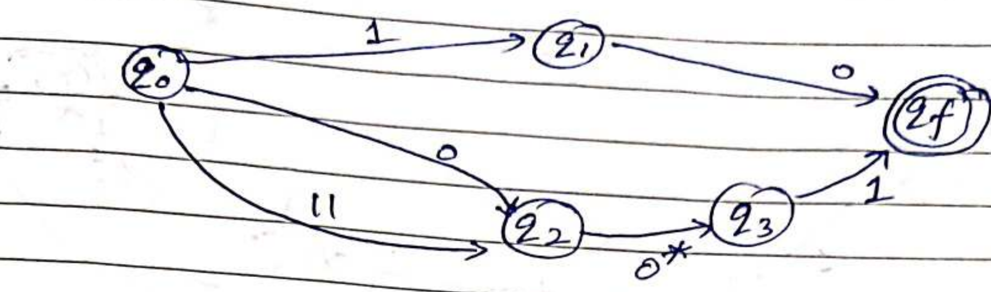
Q. Construct a DFA with reduced state equivalent to regular expression
$$10 + (0+11)0^* 1$$

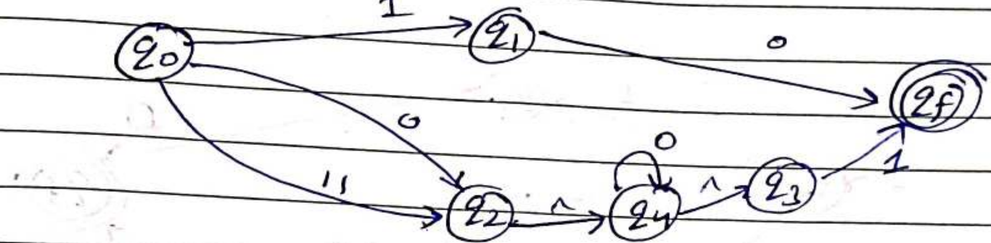
Sol^n →

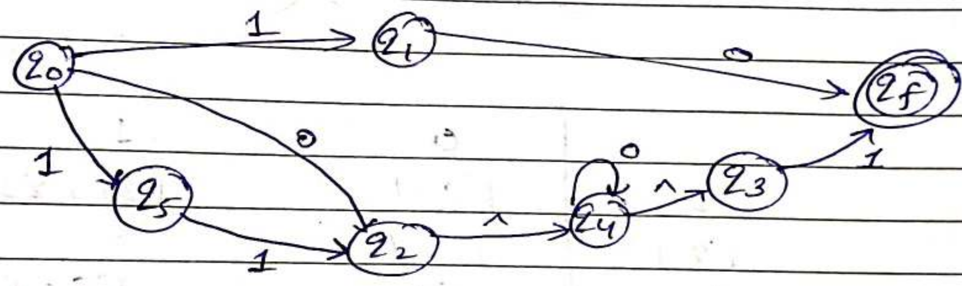


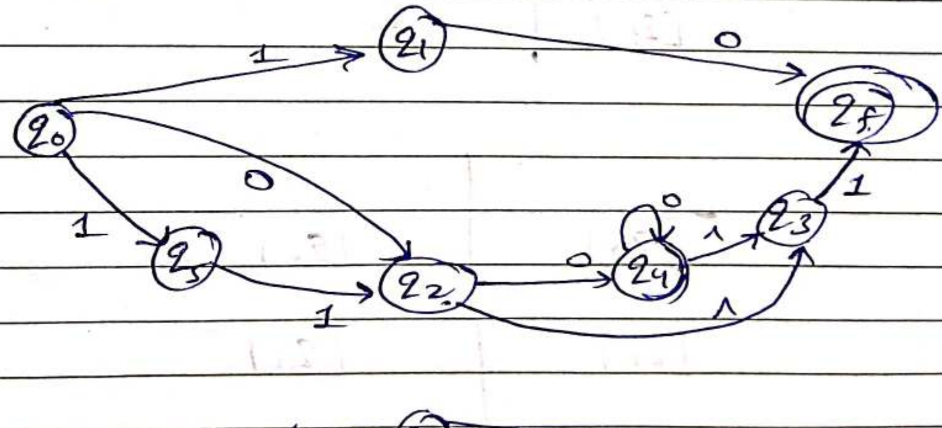


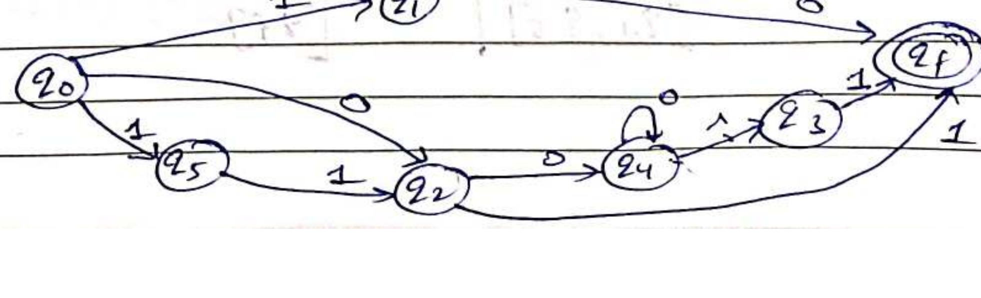


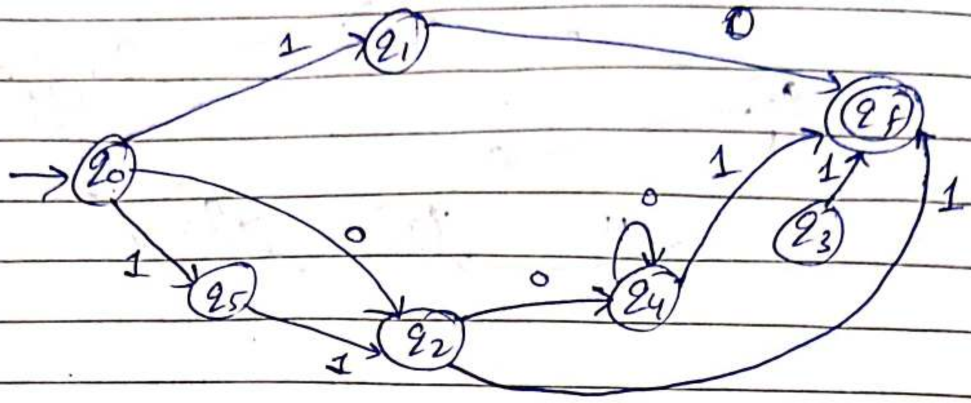

q0
q1
q2
q3
qf
1
0
11
0*
1

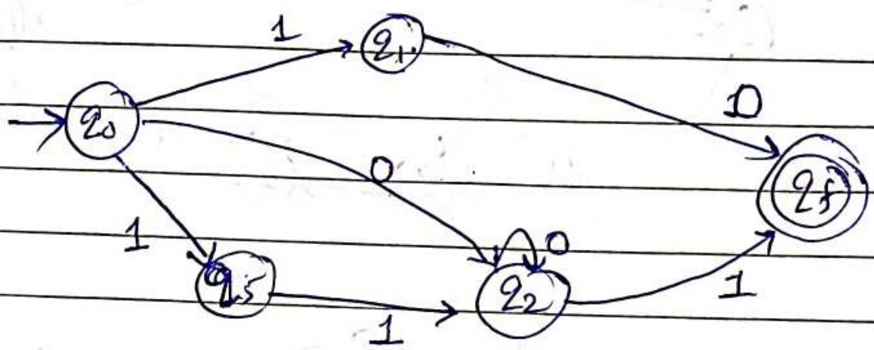

q0
q1
q2
q4
q3
qf
1
0
11
^
0
1


q0
q1
q5
q2
q4
q3
qf
1
0
^
1


q0
q1
q5
q2
q4
q3
qf
1
0
^
1

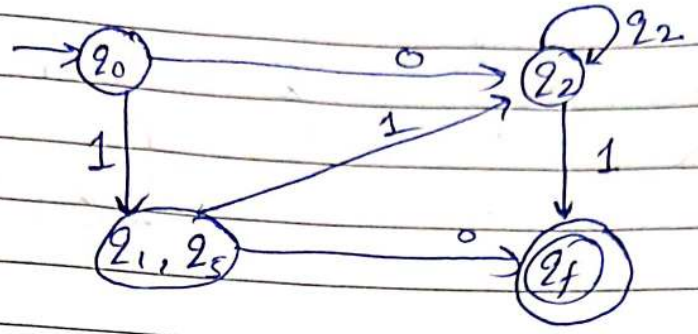

q0
q1
q5
q2
q4
q3
qf
1
0
^
1





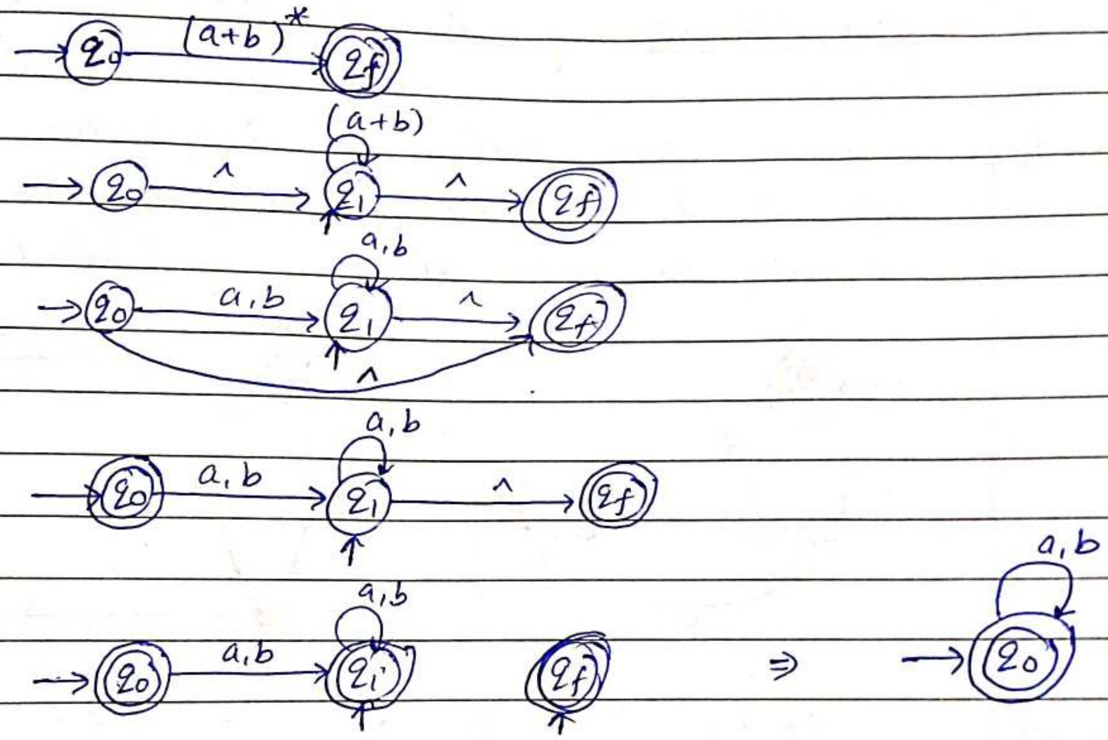
	0	1
[q_0]	[q_2]	[q_1, q_5]
[q_1]	[q_f]	–
[q_2]	[q_2]	[q_f]
[q_5]	–	[q_2]
[q_f]	–	–

	0	1
[q_0]	[q_2]	[q_1, q_5]
[q_2]	[q_2]	[q_f]
[q_1, q_5]	[q_f]	[q_2]

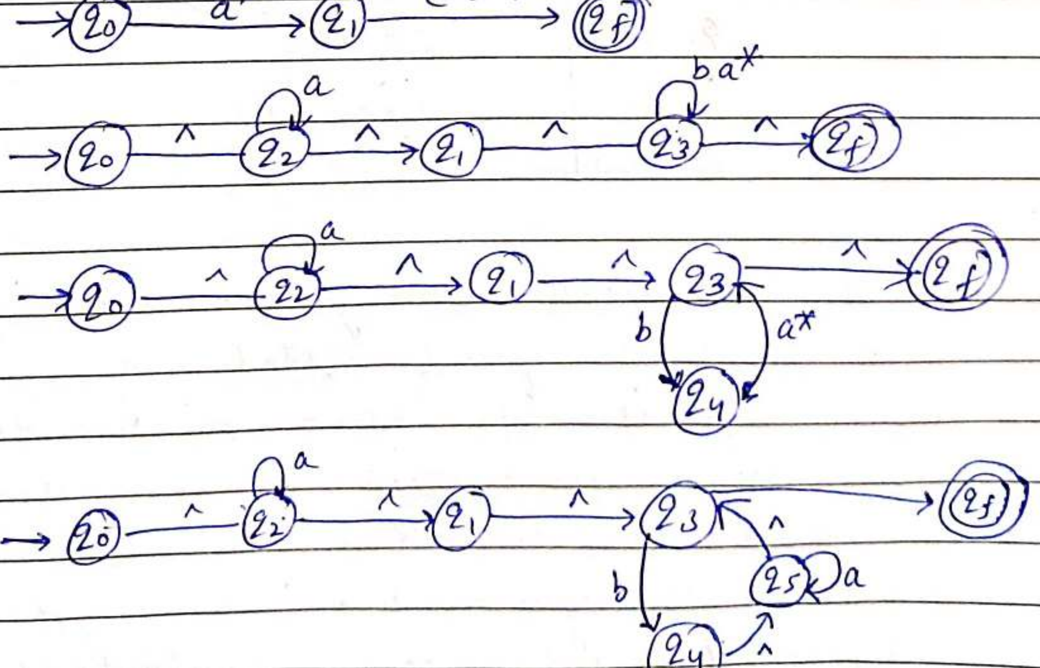


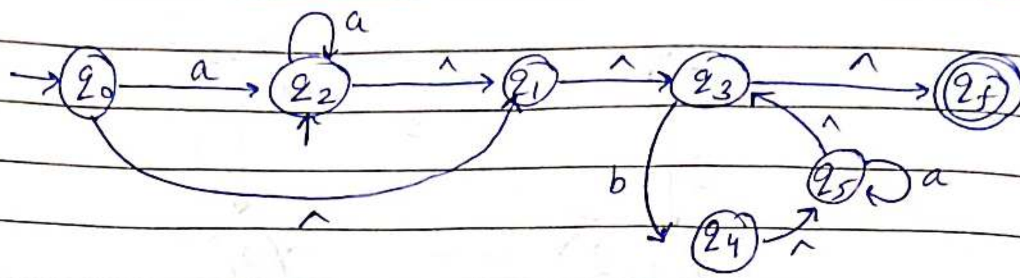
Q. Prove that $(a+b)^* = a^*(ba^*)^*$

Sol^n- LHS



RHS

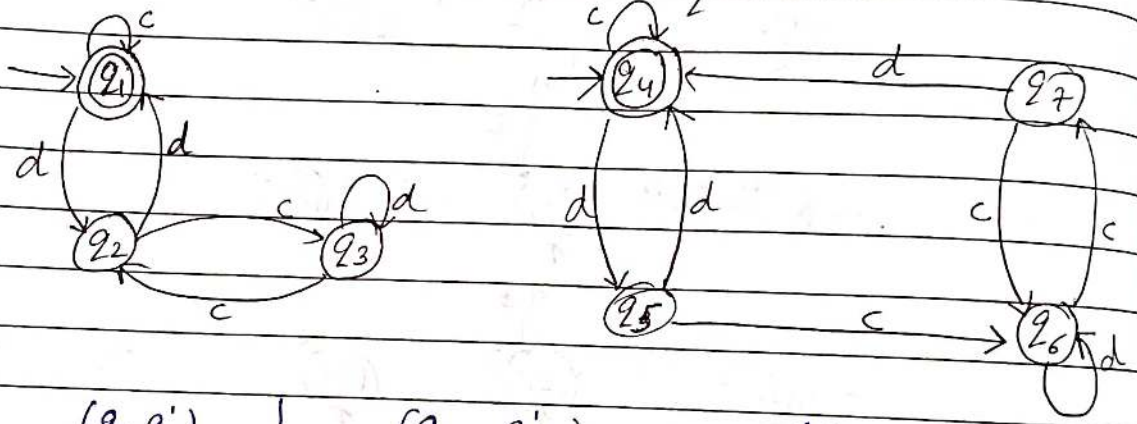




continue...

→ **Equivalence of 2 finite numbers-**

Q. Consider the following 2 DFA's M and M' over (0, 1) as shown in diagram. Determine whether M and M' are equivalent.



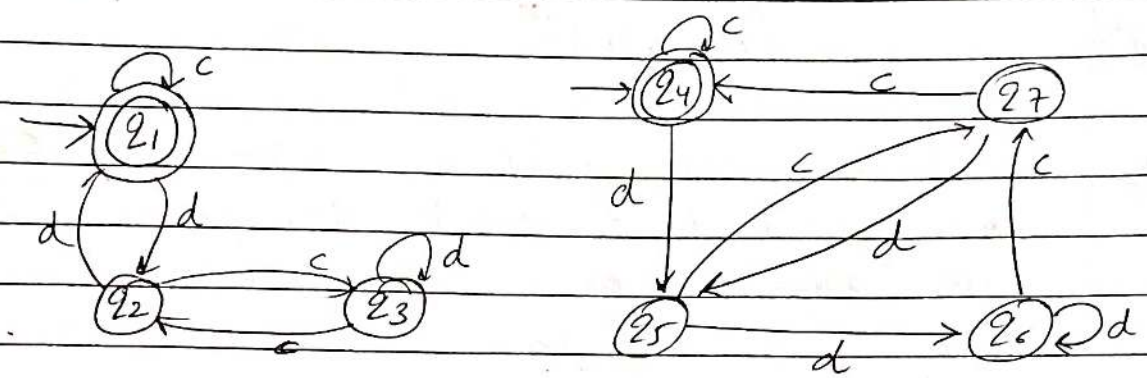
Sol^n-

(q, q')	(q_c, q'_c)	(q_d, q'_d)
(q_1, q_4)	(q_1, q_4)	(q_2, q_5)
(q_2, q_5)	(q_3, q_6)	(q_1, q_4)
(q_3, q_6)	(q_2, q_7)	(q_3, q_6)
(q_2, q_7)	(q_3, q_6)	(q_1, q_4)

If we reach a pair (q, q') such that q is a final state of 'M' and q' is a non-final state of 'M'' or vice-versa, we terminate the construction and conclude that M and M' are not equivalent. Since we do not get a pair (q, q') where q is final state and q' is a non-final state or vice versa at every row

We proceed until all the elements in the second and third columns are also in the first column. Therefore M and M' are equivalent.

Q:



Sol^n^

(q, q')	(q_c, q'_c)	(q_d, q'_d)
(q_1, q_4)	(q_1, q_4)	(q_2, q_5)
(q_2, q_5)	(q_3, q_7)	$\boxed{(q_1, q_6)}$
(q_3, q_7)	~~(q_3, q_4)~~	~~(q_3, q_7)~~
~~(q_1, q_6)~~	~~(q_1, q_7)~~	~~(q_2, q_6)~~

Since, we got a pair (q_1, q_6) in which q_1 is ~~non~~ final state and q_6 is non-final state.

$\therefore$ M and M' are not equivalent

Q. Prove that the regular expression

$$R = \Lambda + 1^*(011)^*(1^*(011)^*)^*$$
$$= (1 + 011)^*$$

$\rightarrow$ LHS $\Rightarrow (1^*(011)^*)^*$ $\{ \Lambda + RR^* = R^* \}$

$\Rightarrow (1 + 011)^*$ $\{ (P+Q)^* = (P^*Q^*)^* \}$

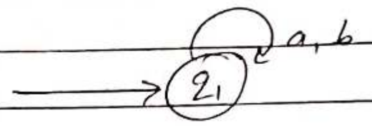
Q. Give the regular expression for representing the set L of strings in which every 0 is immediately followed by at least 2 1's.

1111...
011...
1011... $\Rightarrow (1+011)^*$

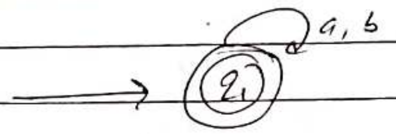
Sol^n - $(1+011)^*$

Q. Find the set of string over $\Sigma = \{a, b\}$ recognised by the transition system as shown in diagram

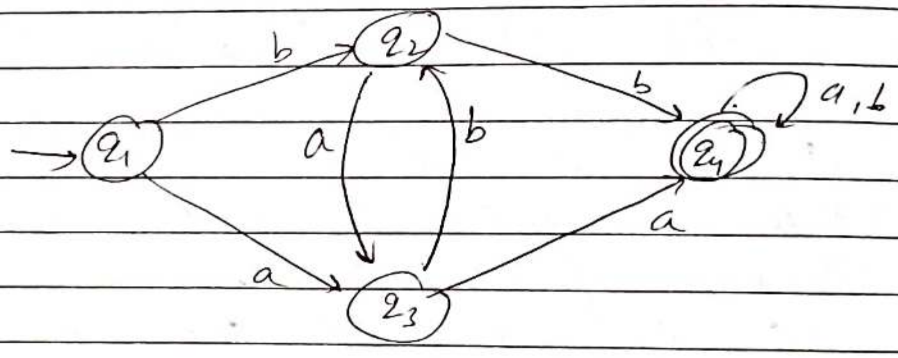
(a)



(b)



(c)



(a) ϕ because, q_1 which is initial state is not the final state due to Arden's theo.

(b)
$$q_1 = q_1 a + q_1 b + \wedge$$
$$q_1 = q_1(a+b) + \wedge$$
$$R = RP + Q$$
$$q_1 = \wedge (a+b)^*$$
$$q_1 = (a+b)^*$$

(c) $q_1 = \wedge$

$q_2 = q_1 b + q_3 b$

$q_3 = q_1 a + q_2 a$

$q_4 = q_4 a + q_4 b + q_3 a + q_2 b$

→ The set of all string ab containing two successive a's or two successive b's.

⇒ aa, bb, baa, abb, bba, bbb, aaa, aab, - - - -

$q_2 = b + q_3 b$

$q_3 = a + (b + q_3 b)a$

$q_3 = a + ba + q_3 ba$

$q_3 = (a+ba) + q_3 ba$

$R = Q + RP$

$R = QP^*$

$q_3 = (a+ba)(ba^*)$

$q_4 = q_4 a + q_4 b + ((a+ba)(ba)^*)a + (b + (a+ba)(ba)^*)b$

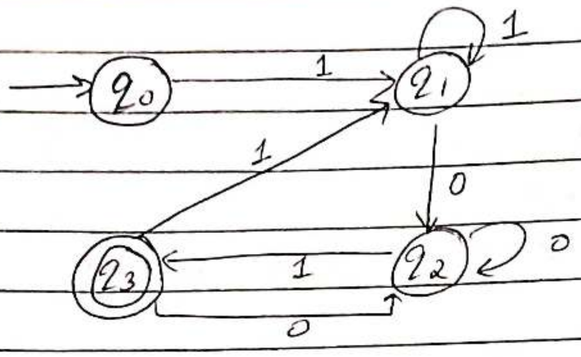
$q_4 = q_4 (a+b) + ((a+ba)(ba)^*)a + (b+(a+ba)(ba)^*)b$

$R = RP + Q$

$R = QP^*$

$q_4 = \left(((a+ba)(ba)^*)a + ((b + (a+ba)(ba)^*)b)\right)(a+b)^*$

Q.



Sol^n

$q_0 = \wedge$

$q_1 = q_0 1 + q_3 1 + q_1 1$

$q_2 = q_1 0 + q_2 0 + q_3 0$

$q_3 = q_2 1$

$q_2 = q_1 0 (0 + 10)^*$

$q_1 = 1 (1 + 0 (0 + 10)^* 11)^*$

$q_2 = q_1 0 + q_2 0 + q_3 0$

$q_2 = q_1 0 + q_2 0 + q_2 10$

$q_2 = q_1 0 + q_2 (0 + 10)$

$R = Q + RP \Rightarrow R = QP^*$

$q_2 = q_1 0 (0 + 10)^*$

$q_1 = 1 + (q_1 0 (0+10)^* 11) + q_1 1$

$q_1 = 1 + (q_1 (1 + 0 (0 + 10)^* 11))$

$R = Q + RP \Rightarrow R = QP^*$

$q_1 = 1 (1 + 0 (0 + 10)^* 11)^*$

$q_2 = 1 (1 + 0 (0 + 10)^* 11)^* 0 (0 + 10)^*$

$q_3 = 1 (1 + 0 (0 + 10)^* 11)^* 0 (0 + 10)^* 1$

$xyz \cdot a^i$

Date | Page No.

Appliation of - Pumping Lemma -

This can be used to prove that, certain sets are not regular. Following steps are used to prove that given set is not regular.

Steps -

1. Assume that, L is regular. Let 'n' be the no. of states in the corresponding finite automata.
2. Choose a string 'W' such that length of W $|W| \geq n$. Use pumping lemma to write $W = xyz$ with $|xy| \leq n$ and $|y| \geq 0$
3. Find a suitable integer 'i' such that $xy^iz \notin L$. This contradicts our assumption, hence L is not regular.

Q. Show that the set $L = \{ a^{i^2} \mid i \geq 1 \}$ is not regular

Solⁿ Steps

① Suppose, L is regular. Let 'n' be the no. of states in finite automata, accepting L

② Let $w = a^{n^2}$, $|W| = n^2 > n$.

By pumping lemma, we can write $w = xyz$ with $|xy| \leq n$ and $|y| > 0$

Here ~~xyz~~ x, y, z are simply strings of 'a'

$y = a^{1}$ $w = a^5$ $L = \{a, a^?, a^?, a^?, a^? \ldots$

$xy^2z \cdot y$ a^5 y^4 $xyz = a^5$

$xy^2z =$

Date

Page No.

③ Consider xy^2z and $|y| = n$ or $y = a^n$

$$|xy^2z| = |x| + 2|y| + |z|$$

and $|xyz| = n^2$

$$|x| + |y| + |z| = n^2$$

So, $|xyz| < |xy^2z|$ as $|xy| \leq n$ & $|y| \leq n$

Also $|xy^2z| = |x| + 2|y| + |z|$
$= |x| + |y| + |z| + |y|$
$= |xyz| + |y|$
$= n^2 + n$

We can say that $n^2 < n^2 + n < n^2 + n + (n+1)$
This shows that, $|xy^2z|$ lies b/w n^2 and $(n+1)^2$ but not equal to any one of them.
Thus, $|xy^2z|$ is not a perfect square.
So, $xy^2z \notin L$. This is contradiction.
So, L is not regular.

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

Solⁿ ① We suppose that L is regular. Let n be the no. of states in finite automata accepting L.

$w = a^n$

$xyz = aaa \ldots a$

$x = a^4 \quad y = a^6 \quad z = a^1$

$xy^5z \to \notin L$

(2) Let $w = a^p$ where p is prime number greater than n. $(p \geq n)$

By pumping lemma, w can be written as $w = xyz$ with $|xy| \leq n$ and $|y| > 0$

Here xyz are simply strings of a's.

So, let $y = a^m$ for $m \geq 1$ and $\leq n$

(3) Let $i = p+1$ then $|xy^iz| = |xyz| + |y^{i-1}|$

$$= p + (i-1)m$$
$$= p + pm$$
$$= p(1+m)$$

$p(1+m)$ is not prime. So, $xy^iz \notin L$

This is contradiction. So L is not regular.

Q Show that $L = \{0^i 1^i \mid i \geq 1\}$ is not regular

Sol^n (1) We suppose that L is regular. Let n be the no. of states in finite automata accepting L.

(2) Let $w = 0^n 1^n$ then $|w| = n + n = 2n > n$

By pumping lemma, we write

$w = xyz$ with $|xy| \leq n$ & $|y| > 0$

(3) We want to find i, so that $xy^iz \notin L$ for getting a contradiction. The string y can be in any of following forms

Case - I y has 0's i.e. $y = 0^k$ for some $k \geq 1$

If we can take $i = 0$ as $xyz = 0^n 1^n$,

& $xz = 0^{n-k} 1^n$ as $k \geq 1$, $n-k \neq n$. So $xz \notin L$

Date
Page No

Case - II y has only 1's i.e. $y = 1^L$ for some $L \geq 1$.

We can take $i = 0$

$xyz = 0^n 1^n$, $xz = 0^n 1^{n-L}$,

$n - L \neq n$. So $xz \notin L$

Case - III y has both 0's & 1's

i.e. $y = 0^k 1^j$

for some $k \geq 1$ and $j \geq 1$

We can take $i = 2$

$xy^2z = 0^{n-k} 0^k 1^j 0^k 1^j 1^{n-j}$

So $xy^2z \notin L$

not in the form of $0^n 1^n$

$\therefore$ We get contradiction.

We got contradiction in all the 3 cases.

So L is not regular.