Unit: 2

Page No. 27
Date

Regular Expression

finite Automata with Output

The finite automata is a collection of 5 tuples. $\{Q, \Sigma, \delta, q_0, f\}$. If we do not get final state then it is said that the string is rejected. that means there is no need of the output for the finite automata but sometimes there is a need for the specified output then in such a case we require finite automata along with output. There are two types of machines of such a type: 1) Mealy machine
2) Moore machine

1) Moore machine

is a final state machine in which next state is decided by current state and current input symbol. The output symbol at a given time depends only on the present state of the machine of the machine. Moore machine has six tuples. $\{Q, \Sigma, \delta, \Delta, \lambda, q_0\}$

$\Delta \rightarrow$ output symbols $(Q \times \Sigma \rightarrow \Delta)$

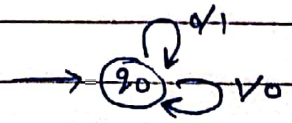
$\lambda \rightarrow$ output mapping function

2) Mealy machine

is a machine in which output symbol depends upon the present input symbol and present state of the machine. Mealy machine is a six tuple machine $\{Q, \Sigma, \delta, \Delta, \lambda, q_0\}$ $(Q \times \Sigma \rightarrow \Delta)$

Q. Design a Mealy machine, which prints 1's complement of input bit string over alphabet $\Sigma = \{0, 1\}$.

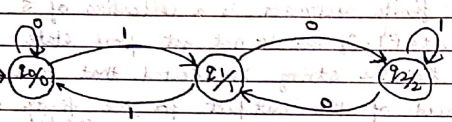
Sol.



Page No.
Date

Conversion of Moore to Mealy Machine

Q

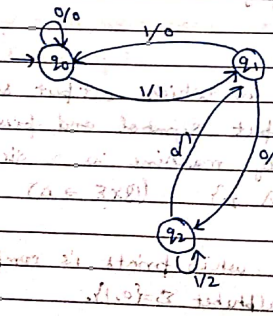


Moore machine table

	Input		
Q	0	1	output
→q0	q0	q1	0
q1	q2	q0	1
q2	q1	q2	2

Mealy machine table

	Input 0		Input 1	
Q	MS	O/P	MS	O/P
→q0	q0	0	q1	1
q1	q2	2	q0	0
q2	q1	1	q2	2



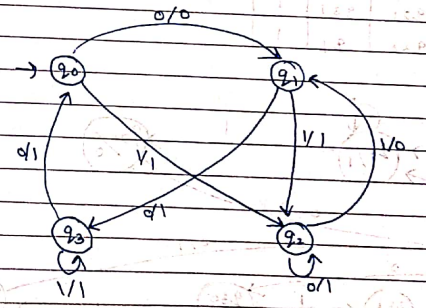
Page No.
Date

Q

	Input		
Q	0	1	O/P
→q0	q1	q2	1
q1	q3	q2	0
q2	q2	q1	1
q3	q0	q3	1

Sol. Mealy machine table

	Input 0		Input 1	
Q	MS	O/P	MS	O/P
→q0	q1	0	q2	1
q1	q3	1	q2	1
q2	q2	1	q1	0
q3	q0	1	q3	1



Page No.
Date

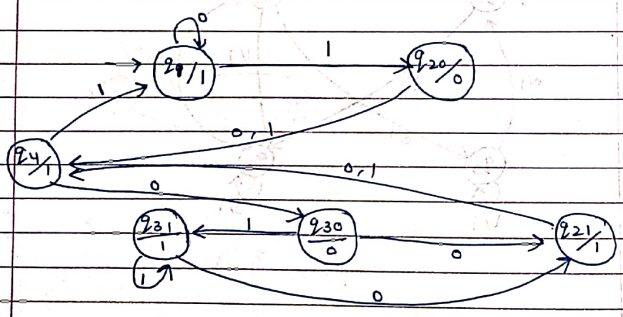
Conversion of Mealy to Moore Machine

Q

Q	Input 0		Input 1	
	NS	O/P	NS	O/P
→ q_1	q_1	1	q_2	0
q_2	q_4	1	q_4	1
q_3	q_2	1	q_3	1
q_4	q_3	0	q_1	1

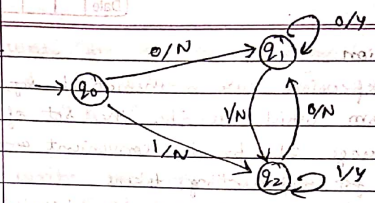
Moore Machine Table

Q	Next State 0	Next State 1	O/P
→ q_1	q_1	q_{20}	1
q_{20}	q_4	q_4	0
q_{21}	q_4	q_4	1
q_{30}	q_{21}	q_{31}	0
q_{31}	q_{21}	q_{31}	1
q_4	q_{30}	q_1	1



Page No.
Date

Q

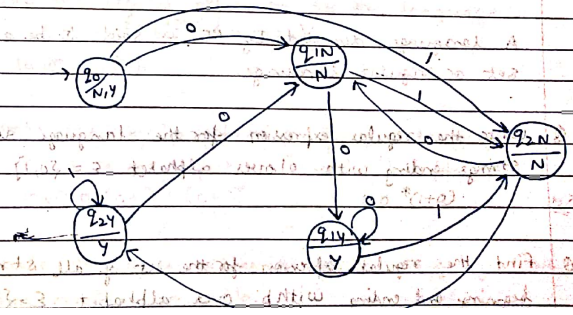


Sol Mealy machine table

Q	Input 0		Input 1	
	NS	OP	NS	OP
→ q_0	q_1	N	q_2	N
q_1	q_1	Y	q_2	N
q_2	q_1	N	q_2	Y

Moore Machine table

Q	Next state 0	Next state 1	O/P
→ q_0	q_{1N}	q_{2N}	N, Y
q_{1N}	q_{1Y}	q_{2N}	N
q_{2N}	q_{1N}	q_{2Y}	N
q_{1Y}	q_{1Y}	q_{2N}	Y
q_{2Y}	q_{1N}	q_{2Y}	Y



Page No.
Date

Regular Expression

Regular expressions are mathematical ~~symbolism~~ symbolism which describe the set of strings of specific language. It provide convenient and useful notation for representing tokens.

There are some rules that describe definition of the regular expression over the input set denoted by Σ.

Step 1: Rule 1: ϵ is a regular expression that denotes the set containing empty strings.

Rule 2: If R_1 and R_2 are regular expressions then $R = R_1 + R_2$ is also regular expression which represent union operation.

Rule 3: If R_1 and R_2 are regular expression ~~which represents concatenation~~ then $R = R_1 R_2$ is also a regular expression which represents concatenation operation.

Rule 4: If R_1 is a regular expression than $R = R_1^*$ is also a regular expression which represents kleen closure.

A language denoted by RE is said to be a regular set or regular language.

Q. Write the regular expression for the language where all strings ending with 01 over alphabet $\Sigma = \{0,1\}$.

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

Q. Find the regular expression for the set of all strings beginning and ending with b over alphabet $\Sigma = \{a,b\}$

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

Page No.
Date

Q. Write the regular expression for the language
$L = \{ a^n b^m / n \geq 4, m \leq 3 \}$

Sol. $a^4 a^* (\epsilon + b + b^2 + b^3)$

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

Sol. $a b^3 b^* (a+b)^+$

Q. Construct the regular expression where all strings that do not end with double letter over alphabet $\Sigma = \{a,b\}$

Sol. $(a+b)^* (ab + ba)$

Q. Write the regular expression for which left most symbol is differ from right most symbol.

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

Q. Write the regular expression for the language
$L = \{ w \in (a,b)^* / n_a(w) \bmod 3 = 0 \}$

Sol. $[b^* a b^* a b^* a b^*]^*$

Q. Write the regular expression for the language
$L = \{ a^{2n} b^{2m+1} / n, m \geq 0 \}$

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

Q. Write the regular expression for the language
$L = \{$ second input is 0 and 4th input is 1 $\}$

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

Q. Write the regular expression for the language which has exactly four 1's.

Sol. $0^* 1 0^* 1 0^* 1 0^* 1 0^*$

Page No.
Date

Q. Write the regular expression for the language
$L = \{ (01)^i 1^{2j} / i,j \geq 0 \}$

Sol. $(01)^* (11)^*$

Q. Write the regular expression for the language
$L = \{ a^n b^m$, where $n+m$ is even$\}$

Sol. Case 1: when both are even
$R_1 = (aa)^* (bb)^*$

Case 2: when both are odd
$R_2 = a(aa)^* b(bb)^*$

$R = R_1 + R_2$
$= (aa)^* (bb)^* + a(aa)^* b(bb)^*$

Q. Write the regular expression for the language
$L = \{ a^n b^m$, where $n+m$ is odd$\}$

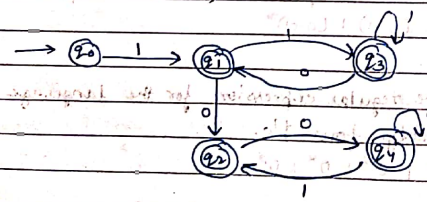
Sol: Case 1: when a is odd & b is even
$R_1 = a(aa)^* (bb)^*$

Case 2: when a is even & b is odd
$R_2 = (aa)^* b(bb)^*$

$R = R_1 + R_2$
$= a(aa)^* (bb)^* + (aa)^* b(bb)^*$

Q. Draw finite automata recognizing the following language:
$1(1+10)^* + 10(0+01)^*$

Sol.



Page No.
Date

Basic Operations in Regular Expression

1) Union:
If R_1 and R_2 are two regular expressions over Σ then $L(R_1 \cup R_2)$ is denoted by $L(R_1 + R_2)$.
$L(R_1 \cup R_2)$ is a string from R_1 or a string from R_2.
$L(R_1 + R_2) = L(R_1) \cup L(R_2)$.

2) Concatenation:
If R_1 and R_2 are two regular expressions over Σ then $L(R_1 \cap R_2)$ is denoted by $L(R_1 R_2)$.
$L(R_1 R_2)$ is a string from R_1 followed by a string from R_2.
$L(R_1 R_2) = L(R_1) L(R_2)$.

3) Closure:
If R_1 and R_2 are two regular expressions over Σ the $L(R^*)$ is a string obtained by concatenating n elements for $n \geq 0$.

Identities for Regular Expression

1) $\phi + R = R$
2) $\phi R = R\phi = \phi$
3) $\varepsilon R = R\varepsilon = R$
4) $\varepsilon^* = \varepsilon$ and $\phi^* = \varepsilon$
5) $R + R = R$
6) $RR^* = R^*$
7) $RR^* = R^* R$
8) $(R^*)^* = R^*$
9) $\varepsilon + RR^* = R^* = \varepsilon + R^* R$
10) $(PQ)^* P = P(QP)^*$
11) $(P+Q)^* = (P^* Q^*)^* = (P^* + Q^*)^*$
12) $(P+Q)R = PR + QR$ & $R(P+Q) = RP + RQ$

Page No.
Date

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

Sol: L.H.S = $(1+00^*1)+(1+00^*1)(0+10^*1)^*(0+10^*1)$

$a^*a = a^*$

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

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

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

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

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

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

$aa^* = a^*$

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

Arden's Theorem

Let P and Q are two regular expression over alphabet Σ. P does not contain null string ϵ. Then equation $R = Q+RP$ has a unique solution given by

$R = QP^*$

for ~~correctness~~ unique solution:

$R = Q+RP$

$= Q+(Q+RP)P$

$= Q+QP+RP^2$

$= Q+QP+(Q+RP)P^2$

$= Q+QP+QP^2+RP^3$

$= Q+QP+QP^2+(Q+RP)P^3$

$= Q+QP+QP^2+QP^3+RP^4$

$= Q(\epsilon+P+P^2+P^3+\cdots)$

$= QP^*$

for Correctness:

$P = Q+RP \quad (R = QP^*)$

$= Q+QP^*P$

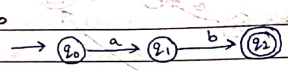
$= Q(\epsilon+P^*P)$

$= QP^*$

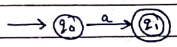
Page No.
Date

Thomson's Construction Algorithm (or) Algorithm for converting Regular Expression to NFA with ϵ.

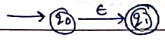
(1) ab



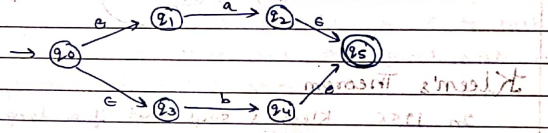
(2) a



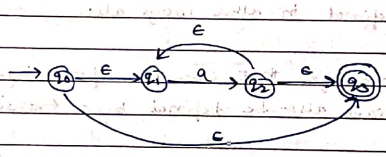
(3) ϵ



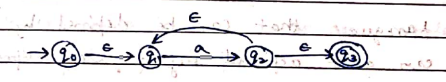
(4) a+b / a/b



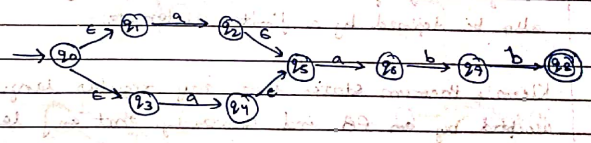
(5) a^*



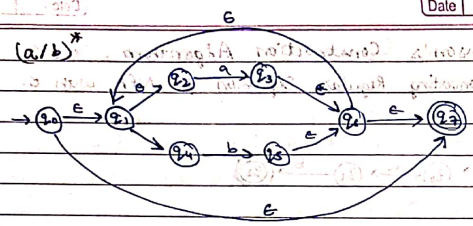
(6) a^+



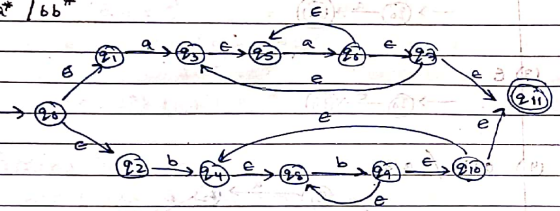
Q.1 : (a/b) a bb



Q.2. $(a/b)^*$



Q.3. aa^* / bb^*



Kleen's Theorem

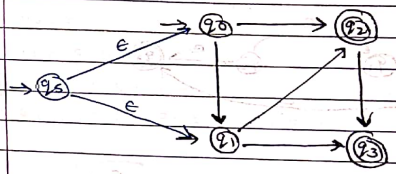
In 1956, Kleen's says that if a language is defined by any of these three ways than it can also be defined in other ways also.

1. Every language that can be defined by finite automata can also be defined by a transition graph.
2. Every language that can be defined by transition graph can also defined by regular expression.
3. Every language that can be defined by R.E can also be defined by a finite automata.

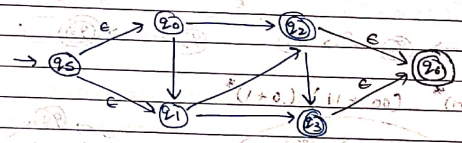
Kleen's theorem states that any regular language is accepted by an FA and conversely, that any language accepted by an FA is regular.

Transition Graph to Regular Expression

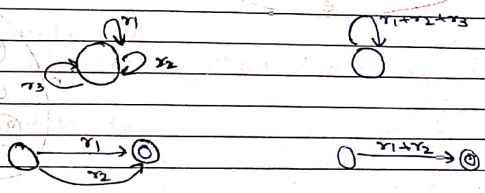
1. If transition graph has more than one initial state, add new state and labelled the arcs by ϵ to each of the initial states.

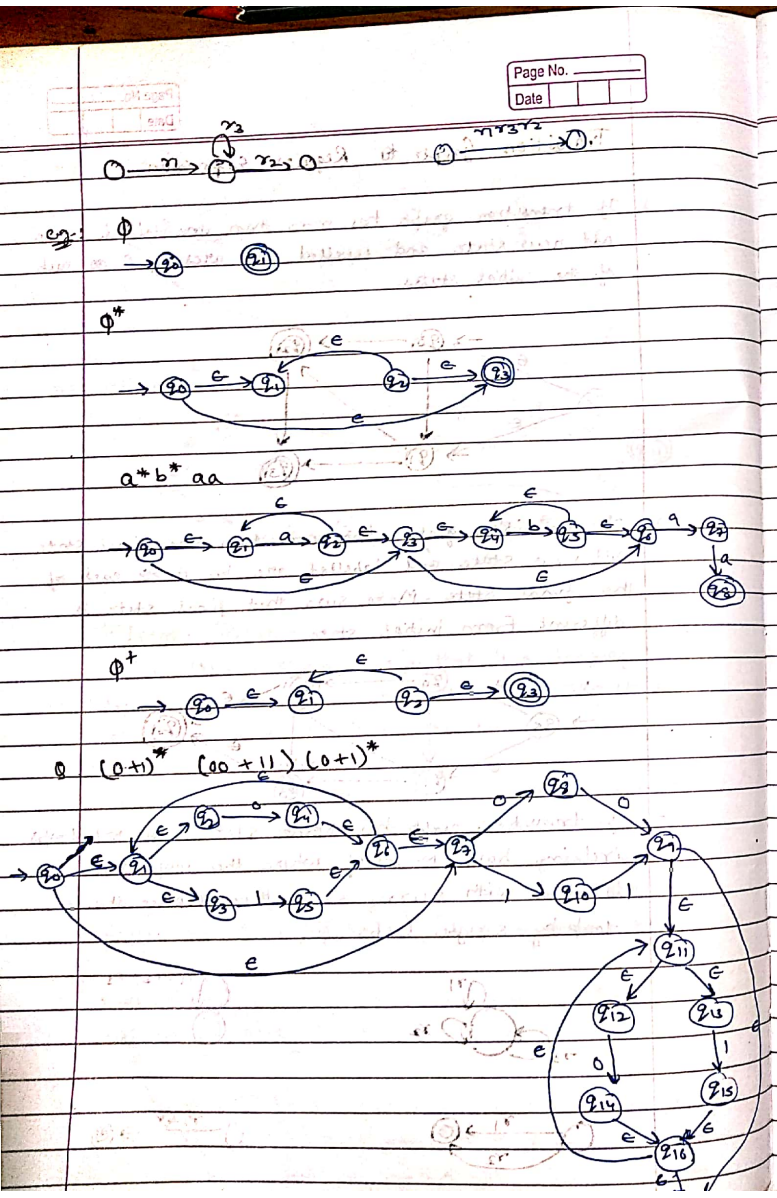


2. If transition graph has more than one final state, add new state and labelled arc by ϵ to each of the final state. Make sure that final state is different from initial state.



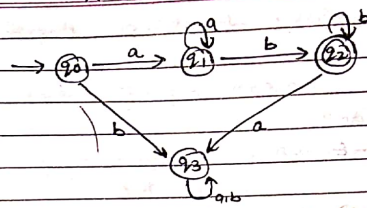
3. If transition graph has some state with $n>1$ loops circling back to itself where the loops are labelled with $r_1, r_2, \ldots, r_n$ then replace the n loop by single loop.





Arden's Theorem:

1.



$q_0 = \epsilon$

$q_1 = q_0 a + q_1 a$

$q_2 = q_1 b + q_2 b$

$q_3 = q_0 b + q_2 a + q_3 (a+b)$

$\boxed{q_0 = \epsilon}$

$q_1 = \epsilon . a + q_1 a$

$q_1 = a + q_1 a$ $\quad [R = Q + RP]$

$\boxed{q_1 = aa^*}$

$q_2 = aa^* b + q_2 b$ $\quad [R = Q + RP]$

$\boxed{q_2 = aa^* bb^*}$

Algebraic Laws of Regular Expressions:

1. Commutative Law

$$P + Q = Q + P$$

2. Associative Law

a) for union : $A + (B + C) = (A + B) + C$

b) for concatenation: $(P \cdot Q) \cdot R = P \cdot (Q \cdot R)$

3. Distributive Law

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

4. Idempotent Law

$$L + L = L$$

$$LL = L$$

5. Closure Laws

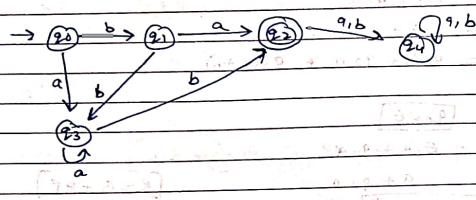
$\phi^* = \epsilon$

$\epsilon^* = \epsilon$

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

$L^+ = L^* - \epsilon$

Q. $R = ba + (a+bb)\,a^*b$ Without Thomson Construction



Pumping Lemma Theorem

If L is a regular language than L has corresponding finite automata with n states such that any string w, whose length of w is greater than equal to n ($|w| \geq n$), may be divided into three parts $w = xyz$, such that following condition must be true,

1. $xy^iz \in L$ for $i \geq 0$
2. $|y| > 0$
3. $|xy| \leq n$

1. $L = \{ a^p \mid p \text{ is prime} \}$

Sol Step 1: Generate language

$L = \{ aa, aaa, aaaaa, aaaaaaa, \ldots \}$

Step 2: $L \rightarrow RL$ & FA $\Rightarrow n$ states $= 5$.

Step 3: $w = xyz$

$|y| > 0$ & $|xy| \leq 5$

a a a a a

x y z

Step 4: $xy^iz \in L$ for $i \geq 0$

at $i=1$, $xyz = aaaaa \in L$

Satisfy

at $i=2$, $xyyz = aaaaaa \notin L$

not Satisfy

So, this is non-regular language

Q. $L = \{ (0^n)^i \mid i \geq 1 \}$

Sol Step 1:

$L = \{ 01, 0101, 010101, \ldots \}$

Step 2:

$L \rightarrow RL$ & FA $\Rightarrow n$ states $= 6$.

Step 3:

$w = xyz$

$|y| > 0$ & $|xy| \leq 6$

0 1 0 1 0 1

x y z

Step 4: $xy^iz \in L$ for $i \geq 0$

at $i=1$ $xyz = 010101 \in L$

Satisfy

at $i=2$ $xyyz = 01010101 \in L$

Satisfy

So, it is a regular language.

Page No.
Date

Closure properties of non-Regular language

1) Union of two regular languages are regular.
2) Intersection of two regular languages are regular.
3) Complement of regular language is regular.
4) Difference of regular language is regular.
5) Concatenation of regular language are regular.
6) Reversal of RL is R egular.
7) Closure of RL is regular.
8) Hommorphism of RL is regular.
9) Inverse Hommorphism of RL is regular.

Decision properties of finite Automata.

1) Emptiness property
2) finiteness property.
3) Equivalence property
4) Membership function.

Q. Let $\Sigma_1 = \{0,1\}$, define by $h(0) = 01$, $h(1) = 112$, find $h(010)$.

Sol. $h(0) = 01$

$h(1) = 112$

Hommorphism, $h(010) = 01112001$

for inverse

$h(0)$

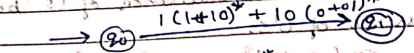
The term hommorphism means substitution of string by some other symbols.

Page No.
Date

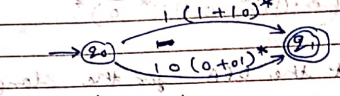
Q. Draw finite automata recognizing the following language: $1(1+10)^* + 10(0+01)^*$

Sol.

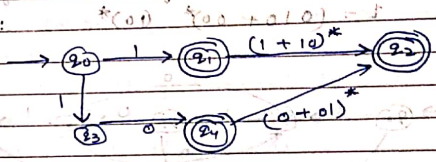
Step 1:



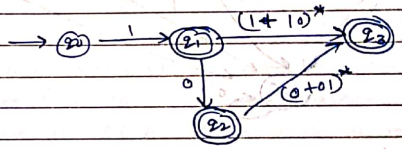
Step 2:



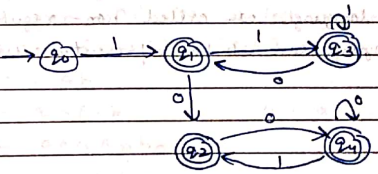
Step 3:



Step 4:



Step 5:



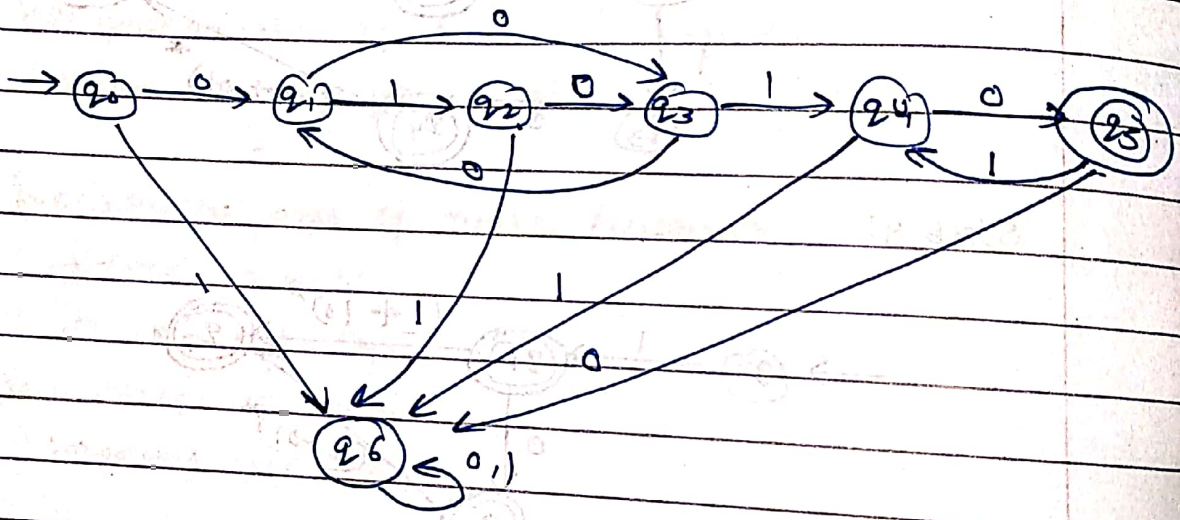
Page No.
Date

Q. Write the regular expression for the language containing the strings over {0,1} in which there are atleast two occurence of 1's between any two occurence of 0's.

Sol. $r = (0\,(1)^*\,0\,(1)^*\,0)^*$

Q. Construct the DFA for the language

$L = (010 + 00)^*\ (10)^*$



Non-Regular language:

Language which can not be represented using regular expressions. Such language are called non-regular languages. Non-regular language can not be represented using finite automata.