29/8/19

Module - II

Minimization of DFA by table - filling algorithm using Myhill - Nerode theorem.

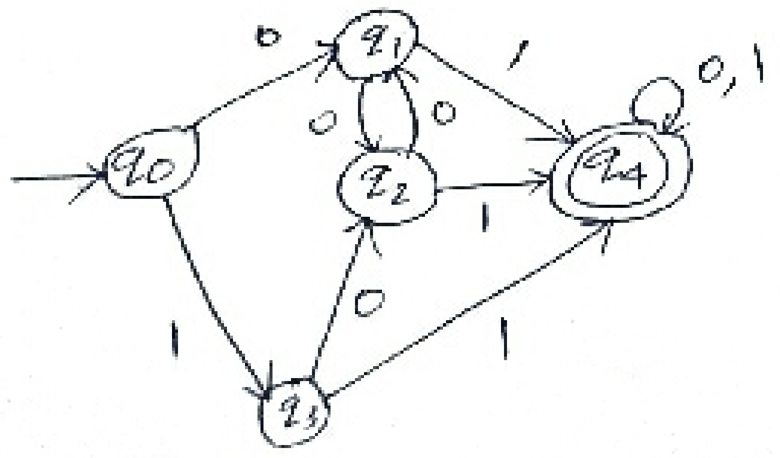
Given a DFA $M = (Q, \Sigma, \delta, q_0, F)$

Table - filling method steps.

1. Draw a table for all pairs of states in Q
2. Mark x in all pairs $(Q-F) \times F$
3. (i) If there are any unmarked pairs (R, S) such that $[\delta(R,a), \delta(S,a)]$ is marked, then mark the pair [R, S], where 'a' is an i/p symbol.

   (ii) Repeat this until no more ~~margin~~ markings can be made.
4. Combine all the unmarked pairs and make them a single state in the minimized DFA.

1.



δ	0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_2	q_4
q_2	q_1	q_4
q_3	q_2	q_4
$* q_4$	q_4	q_4

q_1	X			
q_2	X			
q_3	X			
q_4	X	X	X	X
	q_0	q_1	q_2	q_3

$(Q-F) \times F$

$\{q_0, q_1, q_2, q_3\} \times \{q_4\}$

$(q_0, q_1) \rightarrow \left.\begin{array}{l}\delta(q_0, 0) = q_1 \\ \delta(q_1, 0) = q_2\end{array}\right\}$ $\left.\begin{array}{l}\delta(q_0, 1) = q_3 \\ \delta(q_1, 1) = q_4\end{array}\right\}$ marked so mark (q_0, q_1)

$(q_0, q_2) \Rightarrow \begin{array}{l}\delta(q_0, 0) = q_1 \\ \delta(q_2, 0) = q_1\end{array}$ $\left.\begin{array}{l}\delta(q_0, 1) = q_3 \\ \delta(q_2, 1) = q_4\end{array}\right\}$

$(q_1, q_2) \Rightarrow \begin{array}{l}\delta(q_1, 0) = q_2 \\ \delta(q_2, 0) = q_1\end{array}$ $\begin{array}{l}\delta(q_1, 1) = q_4 \\ \delta(q_2, 1) = q_4\end{array}$

$(q_3, q_0) \rightarrow \delta(q_3, 0) = q_2 \qquad \delta(q_3, 1) = q_4$

$\qquad \delta(q_0, 0) = q_1 \qquad \delta(q_0, 1) = q_3$

$(q_3, q_1) \rightarrow \delta(q_3, 0) = q_2 \qquad \delta(q_3, 1) = q_4$

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

$(q_3, q_2) \rightarrow \delta(q_3, 0) = q_2 \qquad \delta(q_3, 1) = q_4$

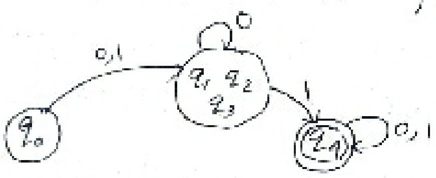
$\qquad \delta(q_2, 0) = q_1 \qquad \delta(q_2, 1) = q_4$

unmarked pairs

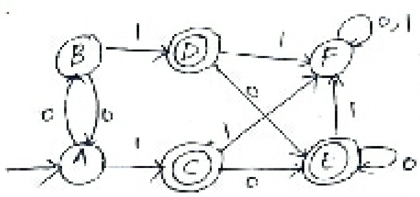
$(q_1, q_0)\ (q_1, q_3)\ (q_2 q_3)$

these act as transitive closure so consider as one state (q_1, q_2, q_3)

$\therefore Q = \{ q_0, (q_1, q_2, q_3), q_4 \}$



Q.



δ	0	1
→ A	B	C
B	A	D
* C	E	F
* D	E	F
* E	E	F
F	F	F

$(Q-F) \times F$

$\{A, B, F\} \times \{C, D, E\}$

	A	B	C	D	E
B					
C	X	X			
D	X	X			
E	X	X			
F	X	X	X	X	X

$\delta(A, F) \rightarrow \delta(A, 0) = B \qquad \delta(A, 1) = C$

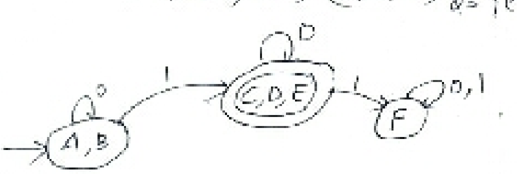
$\qquad \delta(F, 0) = F \qquad \delta(F, 1) = F$

unmarked pair

(A, B), (C, D), (C, E), (D, E) look in diagram

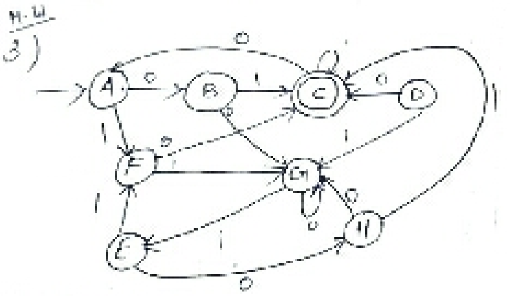
possibly

~~(A, B, C, D, E)~~ (C, D, E) Q = {(A, B), (C, D, E), F}



H.W

3)



δ	0	1
→ A	B	F
B	G	C
* C	A	C
D	C	G
E	H	F
F	C	G
G	G	E
H	G	C

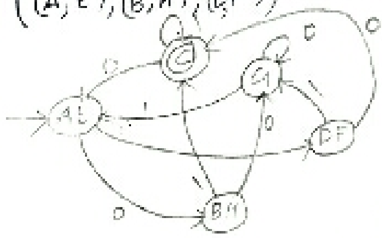
(Q - F) × F

{A, B, D, E, F, G, H} × {C}

	A	B	C	D	E	F	G
B	X						
C	X	X					
D	X	X	X				
E		X	X	X			
F	X	X	X		X		
G	X	X	X	X	X	X	
H	X		X	X	X	X	X

δ(A, B) → δ(A, 0) = B   δ(A, 1) = F
δ(B, 0) = G   δ(B, 1) = C

((A, E), (B, H), (D, F))



Finite Automata with Output.

Mealy Machine (Output associated with transitions)

$M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$

i/p string → length n
o/p string length → n

$Q \to$ finite no. of states

$\Sigma \to$ input alphabet.

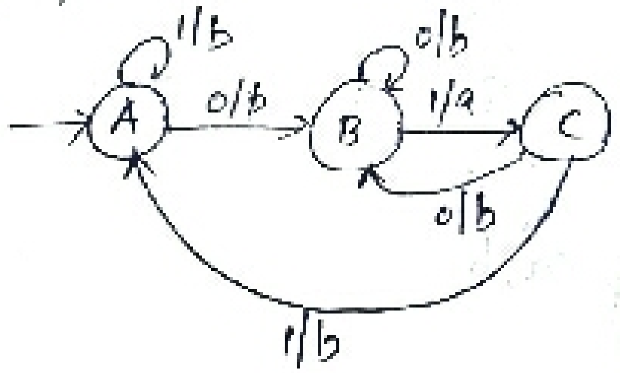
$\Delta \to$ output alphabet

$\delta \to$ transition function $\delta : Q \times \Sigma \to Q$

$\lambda \to$ output function $\lambda : Q \times \Sigma \to \Delta$

$q_0 \to$ start state

Eg:-



consider 1001

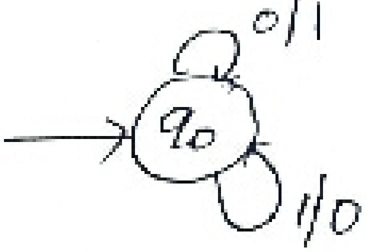
A → A → B → B → C

b b b a

1. Construct a mealy machine to find 1's complement of a binary integer.

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

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



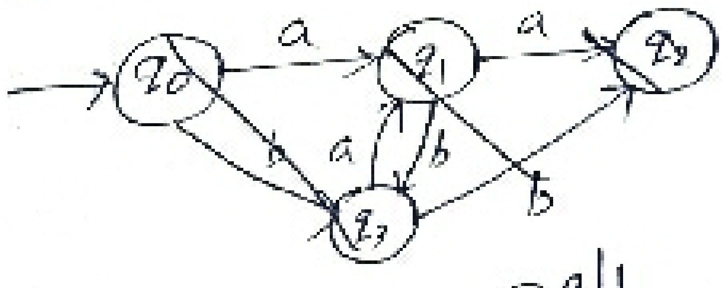
30/8/19

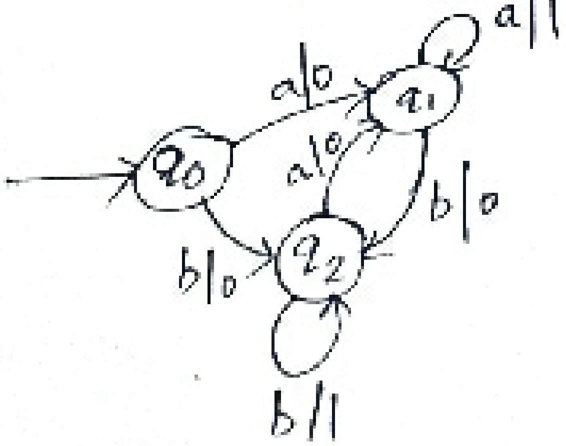
1. construct a mealy machine that accepts set of all strings end with either aa or bb over $\Sigma = \{a,b\}$.

Assume accept - 1
Reject - 0.

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

~~DFA~~





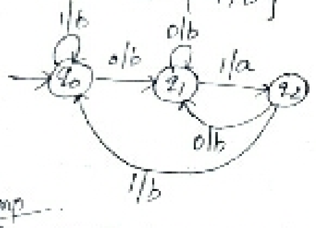
eg: abbaa
↓ ↓ ↓ ↓ ↓
0 0 1 0 1 accept

eg: aba
↓ ↓ ↓
0 0 0 reject.

2. Construct a mealy machine that prints 'a' when the sequence 01 is encountered in an i/p binary string.

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

$\Delta = \{a,b\}$



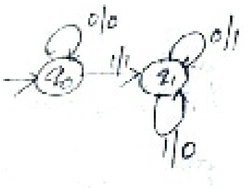
Imp....

3. Construct a mealy machine to compute a 2's complement of binary number.

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

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

Note:- Enter i/p string from LSB.



i/p dirn ←

01011

1's 10100 +

1

10101

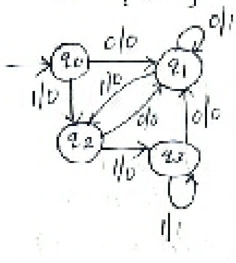
100100

2's 011100

4. Design a mealy machine to accept a set of all strings end with either '00' or '11'

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

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



5.

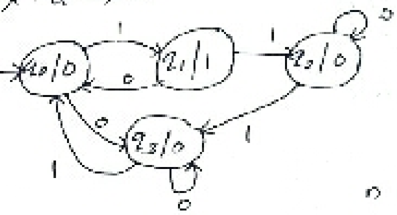
3/9/19

Moore machine (output associated with states)

$M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$

$\lambda : Q \rightarrow \Delta$

Eg:-



0 0 1 0

$\lambda(q_0)$ $\lambda(q_0)$ $\lambda(q_0)$ $\lambda(q_0)$

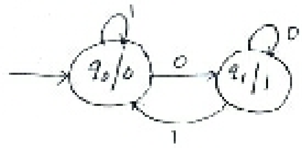
0 0 0 0

n length i/p string →

(n+1) length o/p string

1) Construct a moore machine to implement 1's complement of a binary number.

101
010

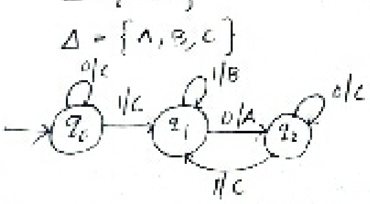


2. Construct mealy machine & moore machine that takes set of all strings over {0,1} as i/p and produce A as o/p if the i/p ends with 10 or produce B as o/p if the i/p ends with 11. Otherwise produce C as o/p.

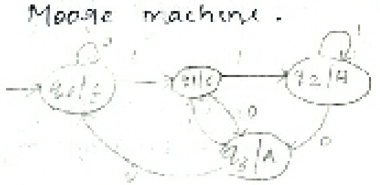
Mealy machine

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

$\Delta = \{A,B,C\}$



Moore machine.



Regular Expression.

Operations

1. union

$L_1 = \{001, 10, 111\}$

$L_2 = \{\epsilon, 001\}$

$L_1 \cup L_2 = \{\epsilon, 001, 10, 111\}$

2. Concatenation

$L_1 \cdot L_2 = \{001, 10, 111, 001001, 10001, 111001\}$

3. closure (star or kleene)

$L = \{0, 11\}$

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

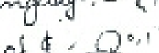
$L^1 = \{0, 11\}$

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

$L^* = L^0 \cup L^1 \cup L^2 \ldots$

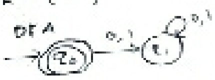
Empty language, $L = \{\phi\}$

DFA of ϕ



$L = \{\epsilon\}$

DFA



Let Σ be an alphabet then the regular expressions over Σ and the sets or languages that they denote are defined recursively as follows

1. ϕ is a regular expression denotes the empty language $L = \phi$.

2. ϵ is a regular expression denotes the set or language $L = \{\epsilon\}$

3. If a is any symbol in Σ then a is a regular expression denotes the language $L = \{a\}$.

4. If r and s are regular expressions denoting the languages R and S respectively. Then r+s, rs, r^* are regular expressions that denotes the $\{R \cup S\}$, $R.S$, R^* respectively.

4/9/19

1) Write a regular expression for the language of set of all strings that begin with 110.

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

$\Longrightarrow$ $110(0+1)^*$

2) end with 110

$(0+1)^* 110$

3) containing 110.

$(0+1)^* 110 (0+1)^*$

4) Any no of a's followed by any no of b's followed by any no. of c's

$a^* b^* c^*$

~~5) Se~~

5) Set of vowels in english language

$\Sigma = \{a, e, i, o, u\}$

$(a+e+i+o+u)^*$

6) Set of all strings containing exactly one 1.

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

$0^* 1 0^*$

7) Set of all strings of a's and b's of any length including empty string

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

8) Set of strings consisting of even no of a's followed by odd no. of b's.

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

9) Set of all strings of a's and b's of even length.

$\cancel{(aa)^* + (bb)^*}$

$\cancel{(ab)^*}$ $(aa+ab+ba+bb)^*$

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

H.W

10) all the strings of 0 and 1 with atleast 2 consecutive zeros.

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

11) Set of all strings begin with 1 and does not have 2 consecutive zeros.

12) all the strings begining with 0 and does not have subset 11

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

13) Set of all strings such that no. of zeros is odd

$$1^*01^*(1^*01^*01)^*$$

Identities of Regular Expression (Minimization Rule)

- Let r, p, q are regular expression and a, b are input symbols in Σ

1. $\phi + r = r$
2. $\phi r = r\phi = \phi$
3. $\epsilon r = r\epsilon = r$
4. $\epsilon^* = \epsilon$
5. $\phi^* = \epsilon$
6. $\phi + \epsilon = \epsilon$
7. $r + r = r$
8. $r^* r^* = r^*$
9. $r r^* = r^* r$
10. $(r^*)^* = r^*$
11. $\boxed{\epsilon + r r^* = \epsilon + r^* r = r^*}$
12. $(pq)^* p = p(qp)^*$
13. $(p+q)^* = p^* q^* = (p^* + q^*)^*$
14. $(p+q)r = pr + qr$

- $r(p+q) = rp + rq$

15. $r^* + \epsilon = r^*$

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

16. $r^* r + r = r^* r$
17. $(r+\epsilon) r^* = r^*(r+\epsilon) = r^*$
18. $(r+\epsilon)(r+\epsilon)^*(r+\epsilon) = r^*$

5/9/19

1. $\Sigma = \{a, b\}$

Start and ends with a $\{a, \ldots\}$

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

2) Start and ends with same symbol. $\{a, b, \ldots\}$

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

3) starts and ends with different symbol

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

4) strings of length exactly 3.

$(a+b).(a+b).(a+b)$

$(a+b)^3$.

5) strings of length atleast 3.

$(a+b)^3 \cdot (a+b)^*$

6) strings of length atmost 3.

$\epsilon + (a+b) + (a+b)^2 + (a+b)^3$

$(a+b+\epsilon)(a+b+\epsilon)(a+b+\epsilon)$

7) No. of a's = 2

$\cancel{aab^*} + b^* aa$

$b^* a b^* a b^*$

ba bb ab

baaba

8) No of a's atleast 2

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

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

9) No of a's atmost 2.

$\epsilon + (a+b^*) + (a+b^*)^2$

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

$\epsilon + (a+b^*) + (b^* a b^* a b^*)$

$b^* (a+\epsilon) b^* (a+\epsilon) b^*$

10) third symbol from left end is b → aabaa

$(a+b)(a+b)\, b\, (a+b)^*$

$(a+b)^2\, b\, (a+b)^*$

11) 28th symbol from right end is a.

$(a+b)^* a (a+b)^{27}$

12) $|w| = 0 \pmod 3$

$[(a+b)^3]^*$

13) $|w| = 2 \pmod 3$

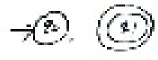
$(a+b)^2 [(a+b)^3]^*$

14) No of b's is divisible by 2

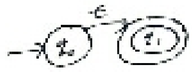
$a^* + (a^* b a^* b a^*)^*$

Conversion of Regular Expression to ϵ-NFA

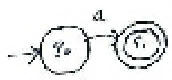
$r = \phi$



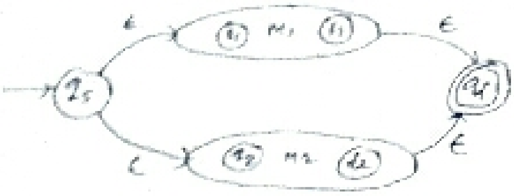
$r = \epsilon$



$r = a$

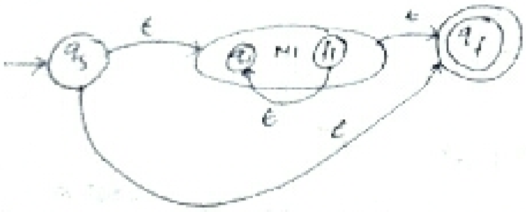


$M = M_1 + M_2$



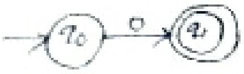
$M = M_1 M_2$

M^*



1) $(01)^* + 1$

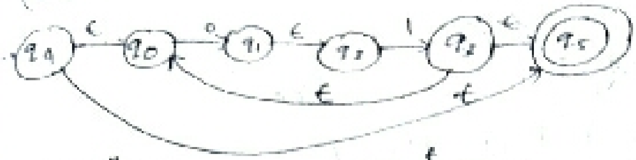
0



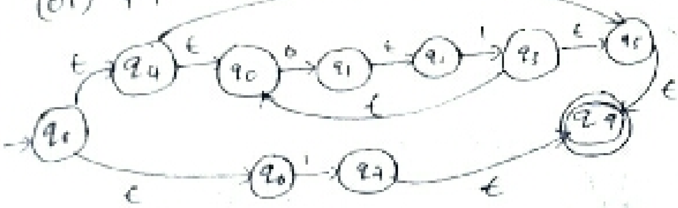
1

01

$(01)^*$



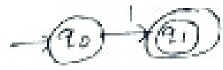
$(01)^* + 1$



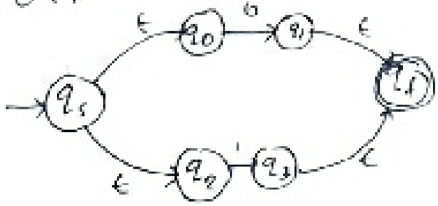
2) $(0+1)^* 10$

0

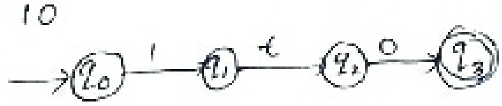
1



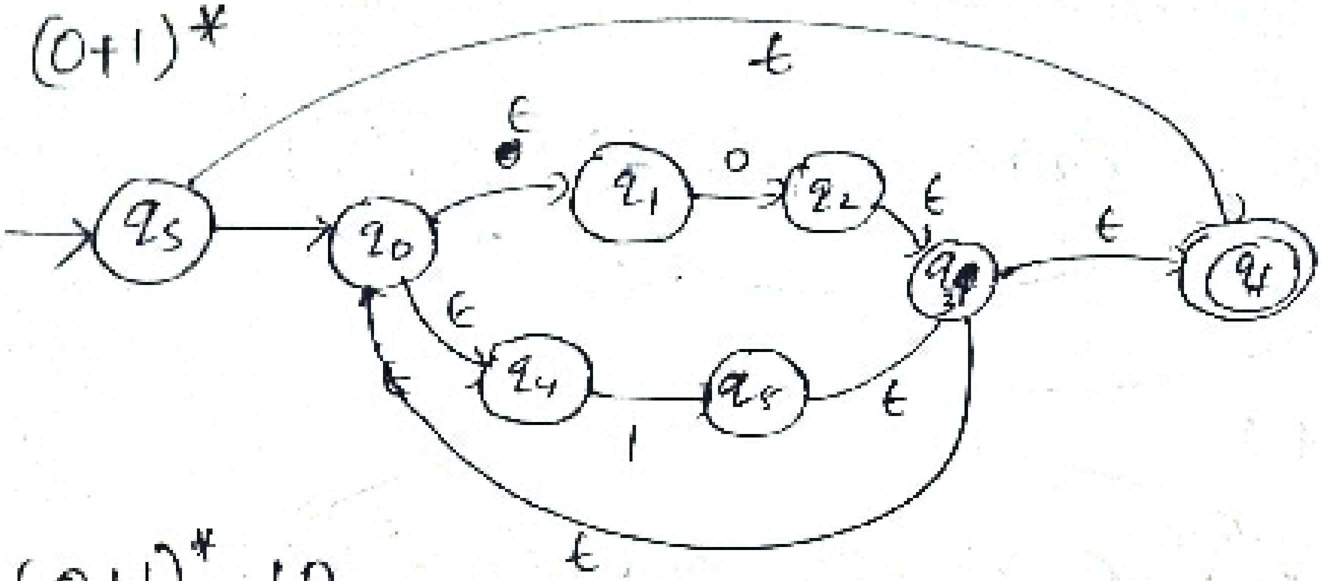
0+1



10

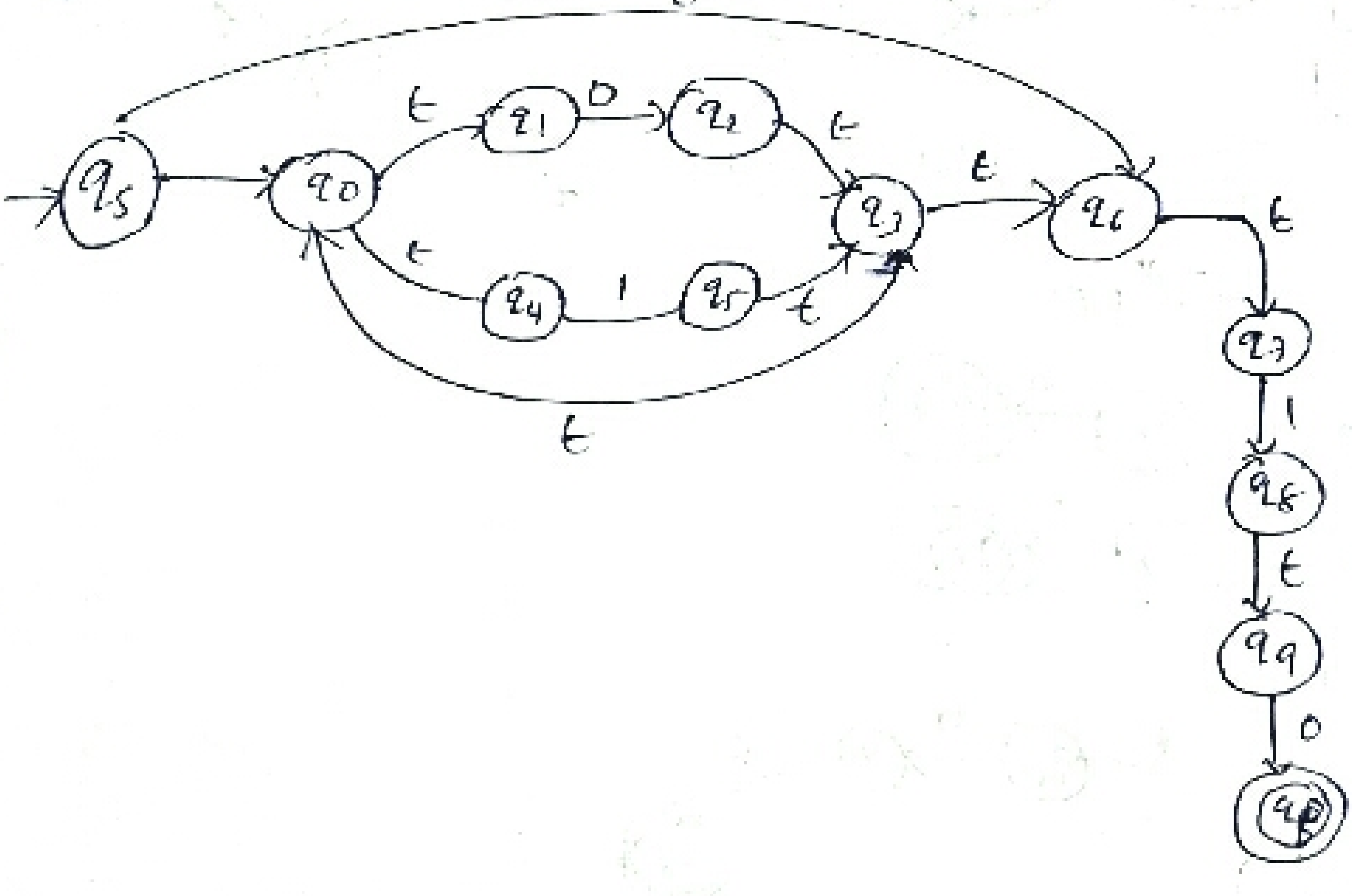


$(0+1)^*$



ϵ
ϵ
0
ϵ
ϵ
ϵ
1
ϵ
ϵ

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

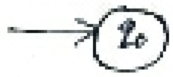


ϵ
ϵ
0
ϵ
ϵ
ϵ
1
ϵ
ϵ
ϵ
1
ϵ
0

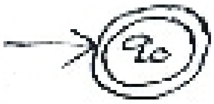
17/9/19

Conversion of regular expression to finite automata

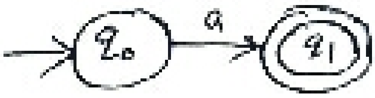
ϕ



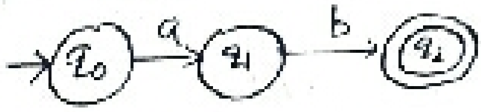
ϵ



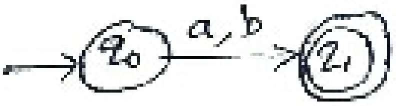
a



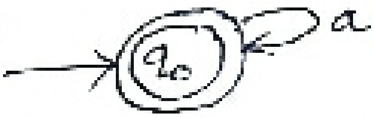
ab



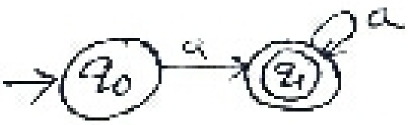
$a+b$



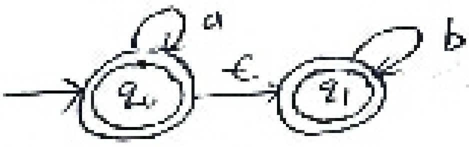
a^*



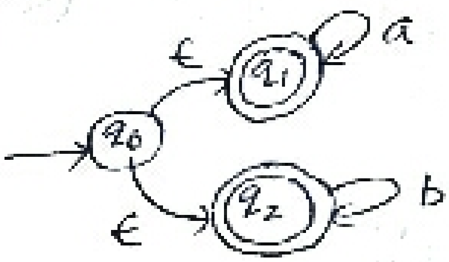
$a^+ (aa^*)$



$a^* b^*$



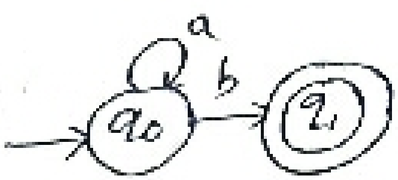
$a^* + b^*$



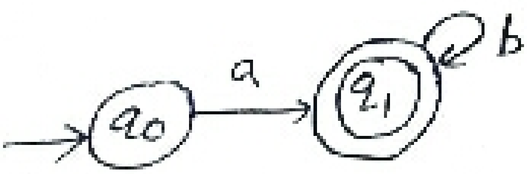
$(a+b)^*$

$(ab)^*$

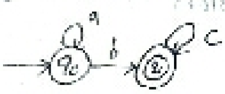
$a^* b$



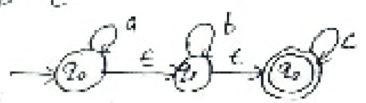
$a b^*$



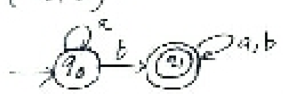
$a^* b c^*$



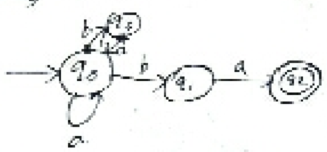
$a^* b^* c^*$



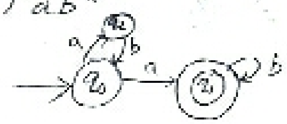
$a^* b (a+b)^*$



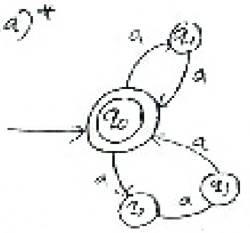
$(a+ba)^* ba$



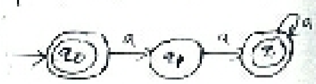
$(ab)^* ab^*$



$(aa+aaa)^*$



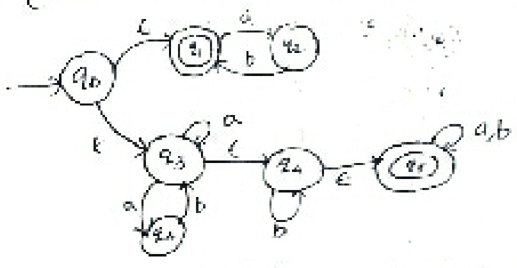
$L = \{ \epsilon, aa, aaa, \ldots \}$



$(a+aaaa)^*$

$L = \{ \epsilon, a, aa, \ldots \} = a^*$

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



1) $[a + ba(a+b)]^* a (ba)^* b^*$

2) $b(a+ba+abb)[ba(a+b)^*]$

3) $(a+b+ca)[bab+(a+b)]^* (ab)^*$

18/9/19

Conversion of finite automata to Regular expression

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

1.

$R_{13}^{(3)}$

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

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

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

$R_{22}^{(0)} = \phi \; \epsilon$

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

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

$R_{33}^{(0)} = \epsilon + 0$

$R_{31}^{(0)} = \phi$

$R_{32}^{(0)} = 1$

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

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

$R_{13}^{(1)} = \phi$

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

$R_{21}^{(1)} = \phi \; 0$

$R_{23}^{(1)} = \cancel{\epsilon+0} \; 1$

$R_{33}^{(1)} = \epsilon + 0$

$R_{31}^{(1)} = \phi$

$R_{32}^{(1)} = 1$

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

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

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

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

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

$$R_{12}^{(1)} = R_{12}^{(0)} + R_{11}^{(0)}(R_{11}^{(0)})^* R_{12}^{0}$$

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

$R_{13}^{(1)} = R_{13}^{0} + R_{11}^{0}(R_{11}^{0})^* R_{13}^{0} = \phi + (\epsilon+1)\cdot(\epsilon+1)^* \phi$

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

$R_{21}^{(1)} = R_{21}^{0} + R_{21}^{0}(R_{11}^{0})^* R_{12}^{0} = \phi + \phi(\epsilon+1)^* 0 = \phi \; 0$

$R_{23}^{(1)} = R_{23}^{0} + R_{21}^{0}(R_{11}^{0})^* R_{13}^{0} = (\cancel{\epsilon+0}\,1) + \phi(\epsilon+1)^* \phi$
$= (\cancel{\epsilon+0}\,1) + \phi$
$= (\cancel{\epsilon+0})\, 1$

$R_{33}^{(1)} = R_{33}^{0} + R_{31}^{0}(R_{11}^{0})^* R_{13}^{0} = (\epsilon+0) + \phi(\epsilon+1)^* \phi$
$= \epsilon + 0$

$R_{31}^{(1)} = R_{31}^{0} + R_{31}^{0}(R_{11}^{0})^* R_{11}^{0} = \phi + \phi(\epsilon+1)^*(\epsilon+1)$
$= \phi$

$R_{32}^{(1)} = R_{32}^{0} + R_{31}^{0}(R_{11}^{0})^* R_{12}^{0} = 1 + \phi(\epsilon+1)^* \cdot 0$
$= 1$

~~$R_{13}^{(3)} = R_{13}$~~

$R_{11}^{(2)} = R_{11}^{(1)} + R_{12}^{(1)}(R_{22}^{1})^* R_{21}^{1} = 1^* + 0 \cdot \epsilon^* 0 = 1^*$

$R_{12}^{(2)} = R_{12}^{1} + R_{12}^{1}(R_{22}^{1})^* R_{22}^{1} = 0 + 0 \cdot \epsilon \cdot \epsilon = 1^* 0$

$R_{13}^{(2)} = 1^* 0 1$

$R_{22}^{(2)} = \epsilon$

$R_{21}^{(2)} = \phi$

$R_{23}^{(2)} = 1$

$R_{33}^{(2)} = (\epsilon+0) + 11$

$R_{31}^{(2)} = \phi$

$R_{32}^{(1)} = 1$