THEORY OF AUTOMATA AND FORMAL LANGUAGES

Use of Theory of automata

- Main application was sequential switching circuits, where the "state" was the settings of internal bits.
- Software for scanning large bodies of text (e.g.,web pages) for pattern finding
- Compiler design
- Several kinds of software can be modeled by FA.
- In C/C++, make a piece of code for each state. This code:

1. Reads the next input.
2. Decides on the next state.
3. Jumps to the beginning of the code for that state.

- An ***alphabet*** is any finite set of symbols. It is denoted by "Σ"
  Examples: {0,1} , {a,b,c}.
- *A string or word is a finite sequence of symbols chosen from Σ*
  Examples: 0,1,01,10,110,001,111,....
  - ***Empty string is ε***
- Length of a string *w*, denoted by "|*w*|", is equal to the *number of (non- ε) characters in the string*
  E.g., x = 010100 |x| = 6
  - *xy* = concatenation of two strings *x* and *y*

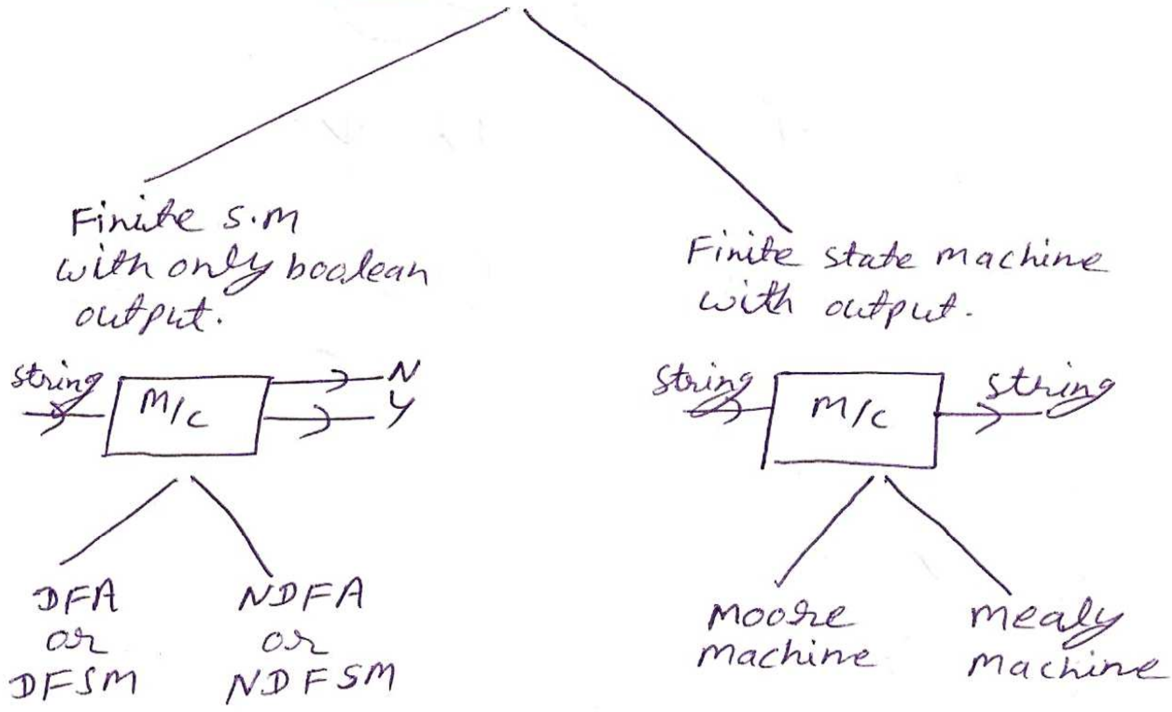
Let Σ be an alphabet.
- Σk = the set of all strings of length k
- Kleen closure $\Sigma^* = \Sigma 0 \cup \Sigma 1 \cup \Sigma 2 \cup \ldots$
- Positive closure $\Sigma^+ = \Sigma 1 \cup \Sigma 2 \cup \Sigma 3 \cup \ldots$

- *L is a said to be a language over alphabet Σ, only if L ε Σ**
  this is because Σ* is the set of all strings (of all possible length including 0) over the given alphabet Σ

Examples:
1. Let L be *the* language of all strings consisting of *n* 0's followed by *n* 1's:
L = { ε,01,0011,000111,...}
2. Let L be *the* language of all strings of with equal number of 0's and 1's:
L = { ε,01,10,0011,1100,0101,1010,1001,...}
Ø denotes the Empty language.
Let L = {ε}; then L is not empty language.

* A Grammar is a finite list of Rules defining a language.

Finite state machine
or
Finite Automata



Deterministic Finite Automata (DFA):

A DFA is a five tuple machine that can be represented as —

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

where (i) Q is a Finite nonempty set of states.

(ii) Σ is a finite nonempty set of input symbols or input alphabets.

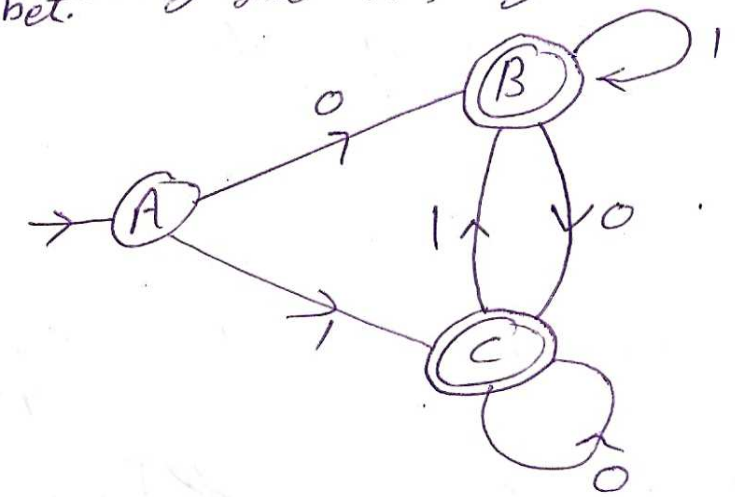
(iii) δ is a function which maps $Q \times \Sigma$ into Q and is usually called direct transection function. This function describes the changes of states during the transition. Usually, this mapping is represented by a transition table or a transition diagram.

(iv) q_0 is the initial state $q_0 \in Q$.

(v) F is the set of final state $F \in Q$.

Note: in DFA, ~~any m~~ every state has only one transection for a alphabet.

example:



here

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

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

$q_0 = \{A\}$

$F = \{B, C\}$

$\delta =$

$\delta(A, 0) = B$

$\delta(A, 1) = C$

$\delta(B, 0) = C$

$\delta(B, 1) = B$

$\delta(C, 0) = C$

$\delta(C, 1) = B$

or

Transition table

δ	0	1
A	B	C
B	C	B
C	C	B

Aceptability of String by DFA :

String X will be Accepted by D.F.A. if

$\delta(q_0, X) = q$ where $q \in F$ because Final state is accepting state.

ex. check that 1101 is accepted or not in Previous DFA

$$\delta(A, 1101) = \delta(\delta(A,1), 101)$$
$$= \delta(C, 101) = \delta(\delta(C,1), 01)$$
$$= \delta(B, 01) = \delta(\delta(B,0), 1)$$
$$= \delta(C, 1) = B$$

B is a Final state so string 1101 is accepted by this m/c.

Note :- This type of transection (string tranction) is called extended transection function.

Prob. 1. Design a DFA which accept all those string, ending with "11"

Language of a DFA :

Let DFA $A = (Q, \Sigma, \delta, q_0, F)$, then language of DFA A is $L(A) = \{ w \mid \hat{\delta}(q_0, w) \text{ is in } F \}$ means set of all those string which will accept by this DFA.

Prob. Give DFA's accepting the following strings over the alphabet $\{0,1\}$

(I) The set of all strings beginning with 101

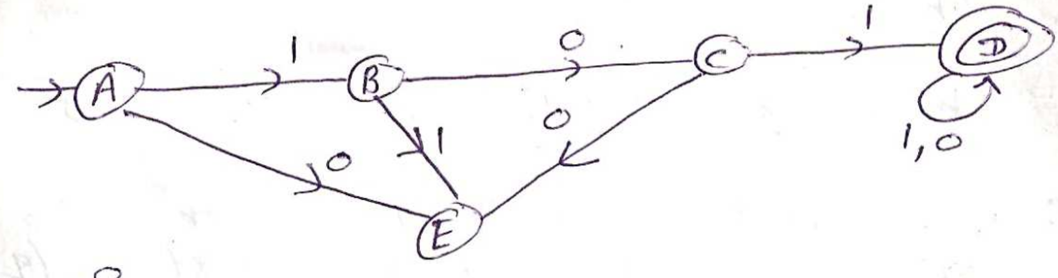
(II) The set of all string containing 1101 as a sub-string.

(III) The set of all strings not containg 110

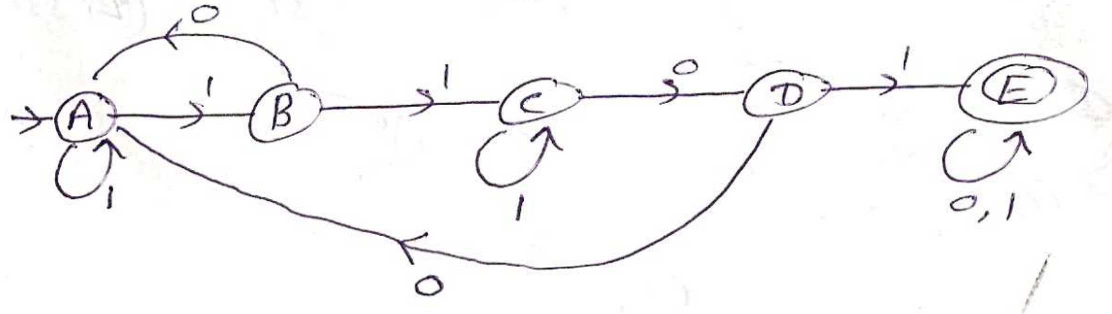
(IV) The set of all string that begin with 01 and end with 11.

Solution.

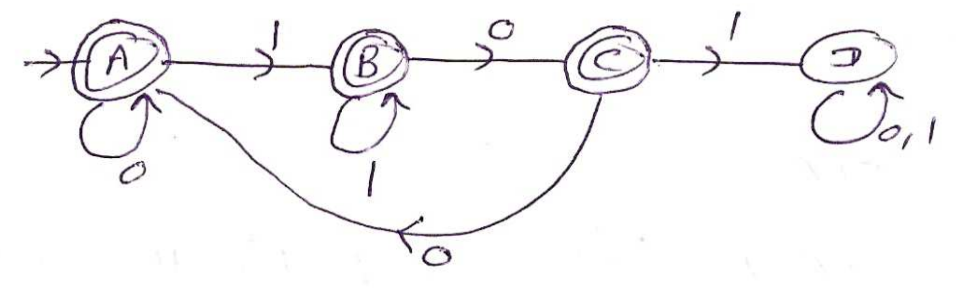
(I)



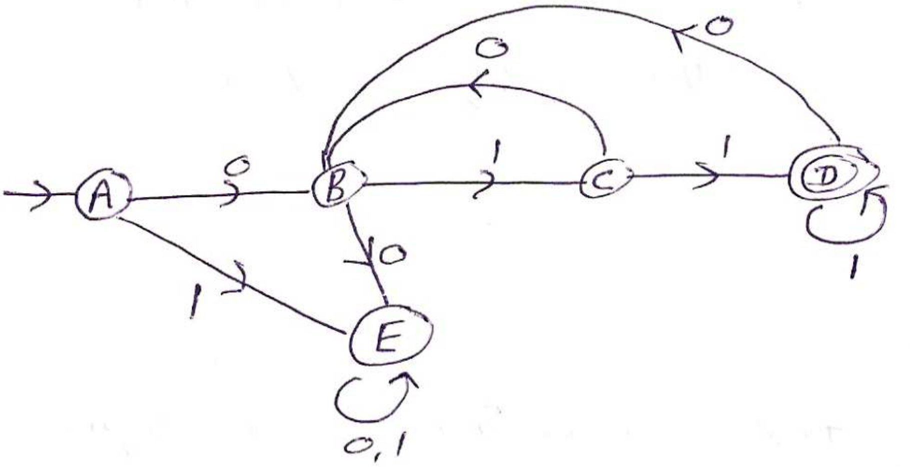
(II)



(III)



(IV)



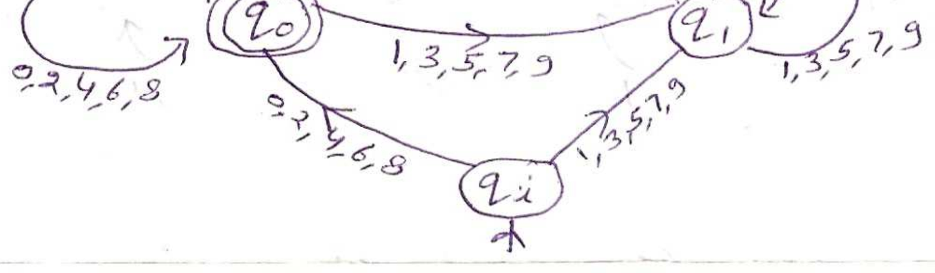
<u>Dummy state</u>: For any m/c all those transition which are not define for any state it will transit to dummy state. Dummy state will have only incoming edge from other state.

There will be only one dummy state for any m/c.

Q. Design a DFA for testing a number is divisible by 2 or not.

Sol. here $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

hint: remainder 0 means Final state (q_0 state)
remainder 1 means nonFinal state (q_1 state)

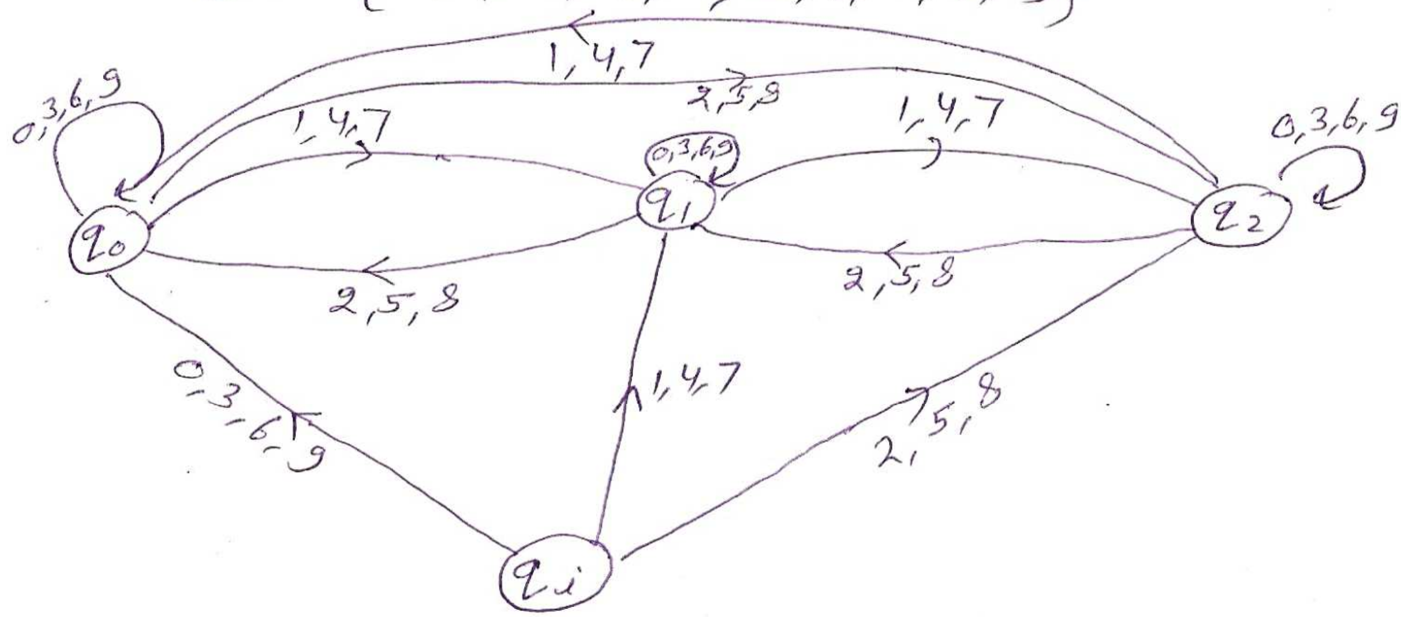


Prob. Design a DFA for ~~any no~~ testing a number is Divisible by 3 or not.

Solution:

Hint: remainder 0 (Final) (q_0 state)
remainder 1 nonFinal (q_1 state)
remainder 2 (nonFinal) (q_2 state)

$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

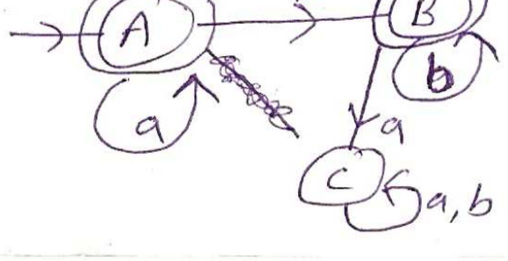


30 % 3 = 0
31 % 3 = 1
32 % 3 = 2
33 % 3 = 0
.
.
.
39 % 3 = 0

40 % 3 = 1
41 % 3 = 2
42 % 3 = 0
.
.
.
49 % 3 = 1

50 % 3 = 2
51 % 3 = 1
52 % 3 = 0
.
.
.
59 % 3 = 2

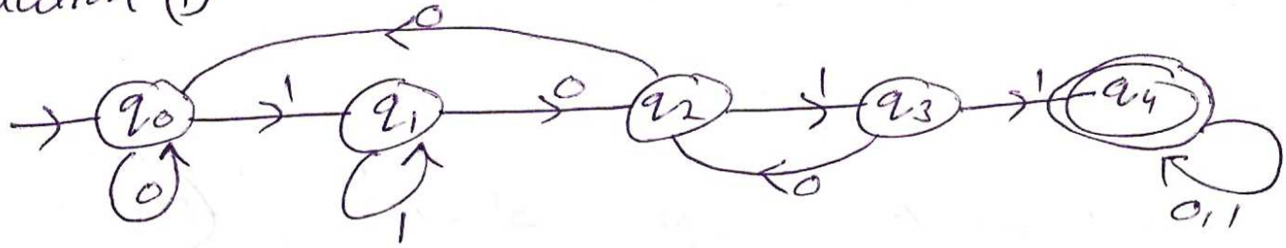
Prob. Design a DFA for $a^m b^n$ $\forall\, m, n \geq 0$



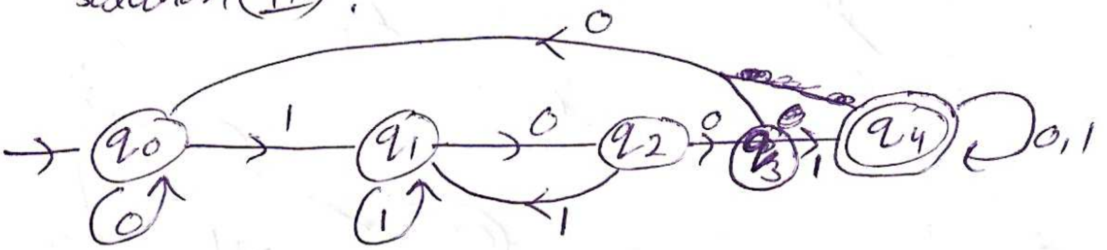
Assignment - 1

Q.(1) Design a DFA for set of strings which containing as substring (i) 1011 (ii) 1001 (iii) 10101 (iv) 11011

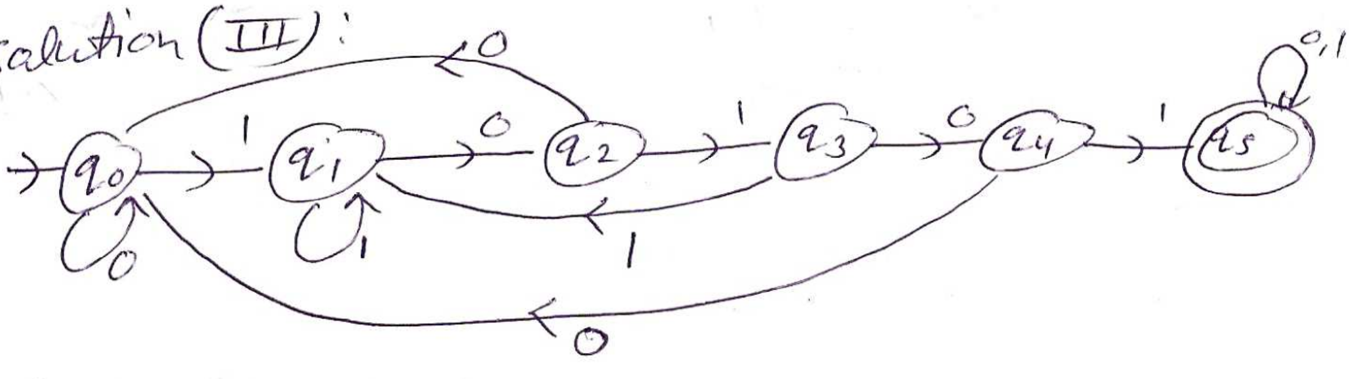
solution (i)



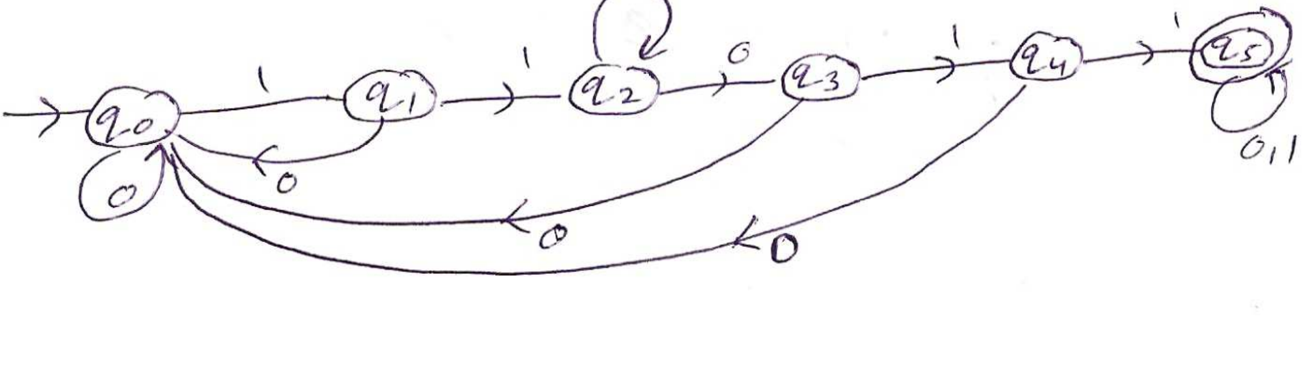
solution (II):



solution (III):



solution (iv):



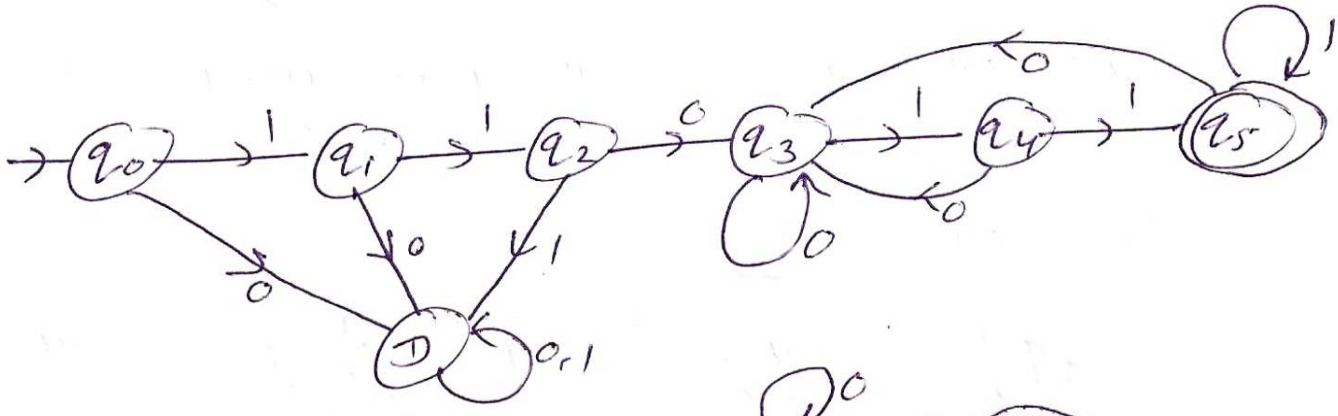
Q. (2) Desig a DFA for a set of string which

(i) start with 110 and end with 11

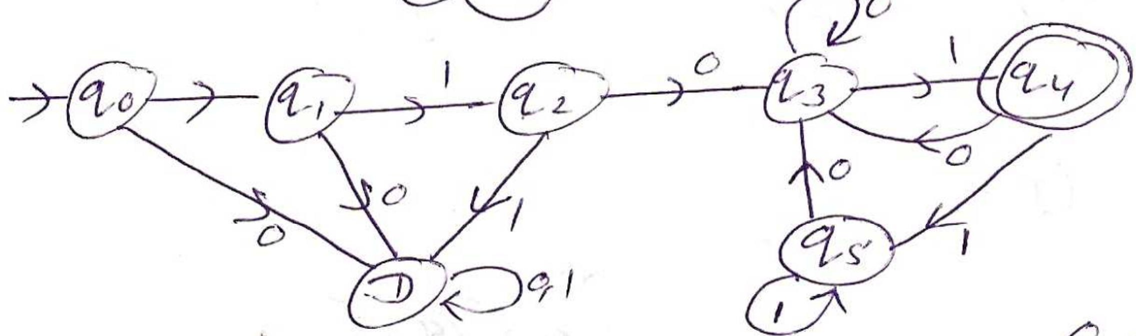
(ii) start with 110 and end with 01

(iii) start with 110 and end with 10

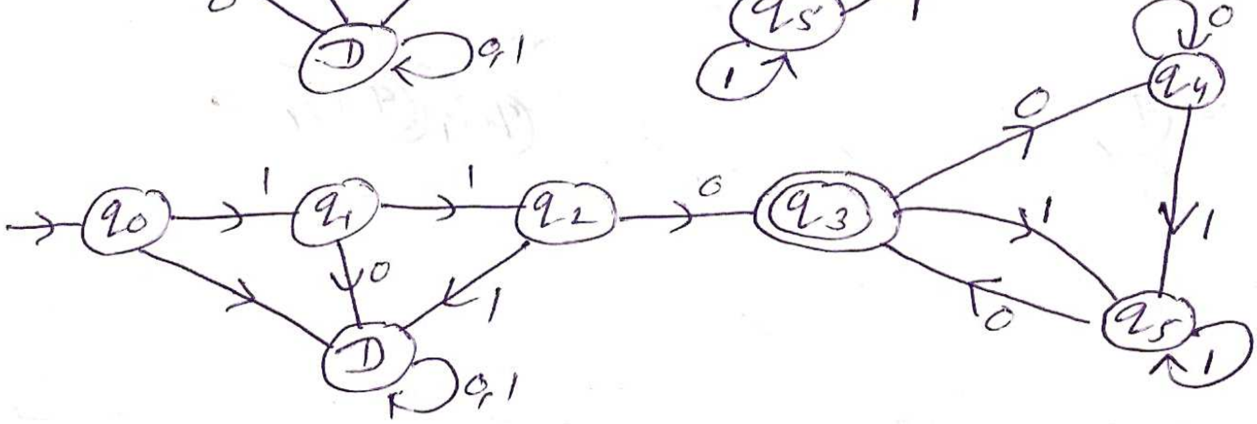
Sol. (i)



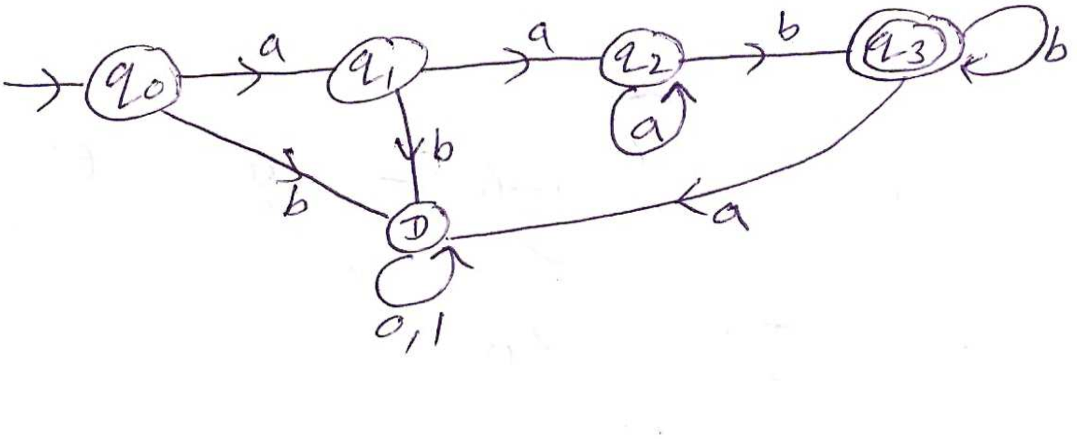
(ii)



(iii)



Q (3) $a^n b^m$ where $n \geq 2$ $m \geq 1$



Non Deterministic Finite Automata : (NDFA) / (NFA) :

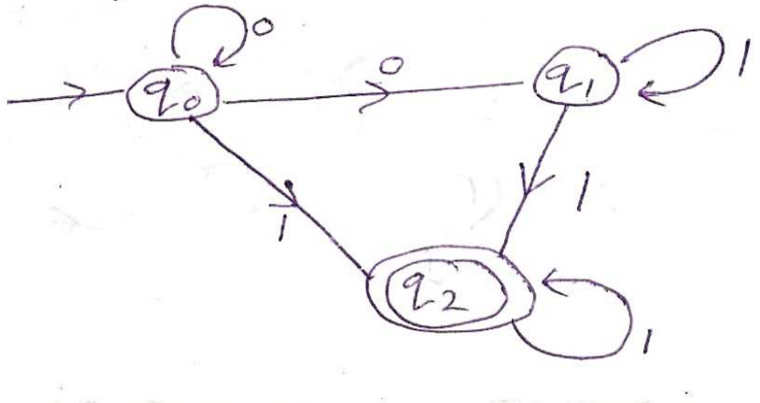
NDFA is a five tuples machine that can be represented as:-

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

where

* Q = Finite set of states.
* Σ = Finite nonempty set of input symbols or Alphabets.
* δ is the transition function which maps $Q \times \Sigma$ into any subset of Q.
* q_0 is the initial state.
* F is the set of Final state.

Note: The difference between a NFA and a DFA is only the type of value that δ returns. means in case of DFA δ return a single state and in case of NDFA δ return a set of state.



$Q = \{ q_0, q_1, q_2 \}$

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

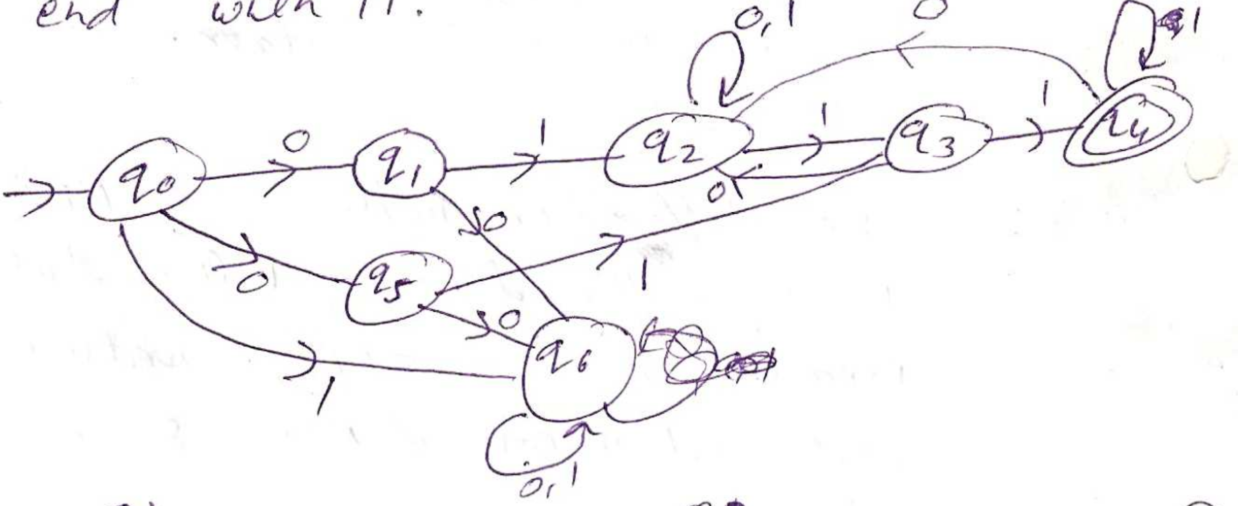
$q_0 = \{ q_0 \}$

$F = \{ q_2 \}$

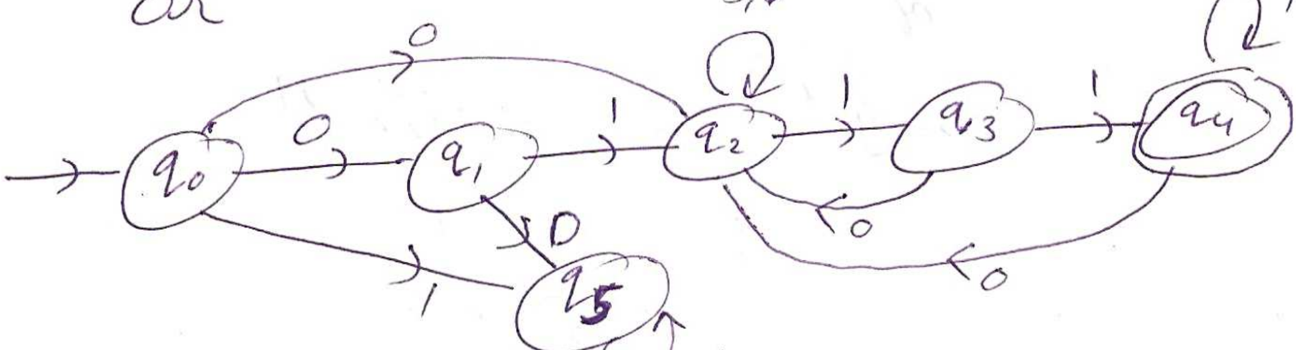
$\delta =$

δ	0	1
q_0	$\{q_0, q_1\}$	$\{q_2\}$
q_1	—	$\{q_1, q_2\}$
q_2	—	$\{q_2\}$

Q Design a Finite Automata m/c which accept the set of all string that begin with 01 and end with 11.



or



Enteded transition Function for NFA:

in Previous example check that the string 01111 is accept or not.

$$\hat{\delta}(\{q_0\}, 01111) = \hat{\delta}(\delta(q_0, 0), 1111)$$
$$= \hat{\delta}(\{q_1, q_2\}, 1111) = \hat{\delta}(\delta(q_1, 1) \cup \delta(q_2, 1), 111)$$
$$= \hat{\delta}(\{q_2, q_3\}, 111) = \hat{\delta}(\delta(q_2, 1) \cup \delta(q_3, 1), 11)$$
$$= \hat{\delta}(\{q_3, q_4\}, 11) = \hat{\delta}(\delta(q_3, 1) \cup \delta(q_4, 1), 1)$$
$$= \hat{\delta}(\{q_4\}, 1) = \{q_4\}$$

note: if atleast one Final state is in the Final output of extended transition F^n then string is accepted by NDFA.

Language of NFA:

Let NDFA $A = \{Q, \Sigma, \delta, q_0, F)$ then

the language of NDFA A is $L(A) = \{w \mid \hat{\delta}(q_0, w) \cap F \neq \phi$

means set of all those string w, for which $\hat{\delta}(q_0, w)$ have atleast one final state.

Note:

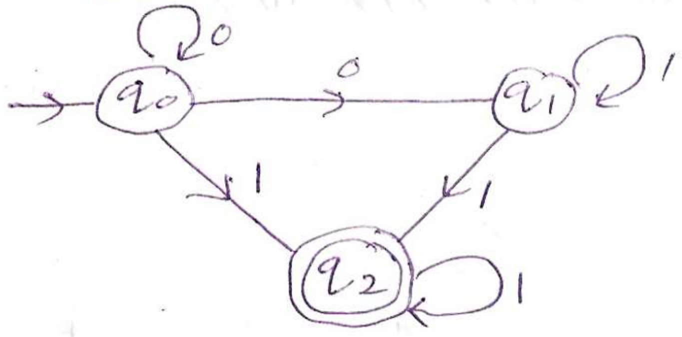
(i) Time complexity of NDFA is 2^n. Thus it is not prectically implemented.

(ii) DFA is a special case of NDFA.

(iii) For every NDFA there must be a unique DFA equivelent to it.

(iv) if initial state is ~~there~~ in final state set then ~~there~~ it must have ε string.

NFA to DFA conversion:



Given NFA

$m(Q_1, \Sigma_1, \delta_1, q_{01}, F_1)$

$Q_1 = \{q_0, q_1, q_2\}$

$\Sigma_1 = \{0, 1\}$

$q_{01} = \{q_0\}$

$F_1 = \{q_2\}$

δ_1	0	1
q_0	$\{q_0, q_1\}$	$\{q_2\}$
q_1	–	$\{q_1, q_2\}$
q_2	–	$\{q_2\}$

in NFA $|Q| = 3$

hence DFA have max $2^{|Q|} = 2^3 = 8$ states.

$\{ \phi, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0 q_1\}, \{q_1, q_2\}, \{q_0 q_2\}, \{q_0 q_1, q_2\} \}$

equivelent DFA $M_2 (Q_2, \Sigma_2, \delta_2, q_{02}, F_2)$

where

$Q_2 \le 2^{Q_1}$

$\Sigma_2 = \Sigma_1$

$q_{02} = q_{01}$

$F_2 = ?$

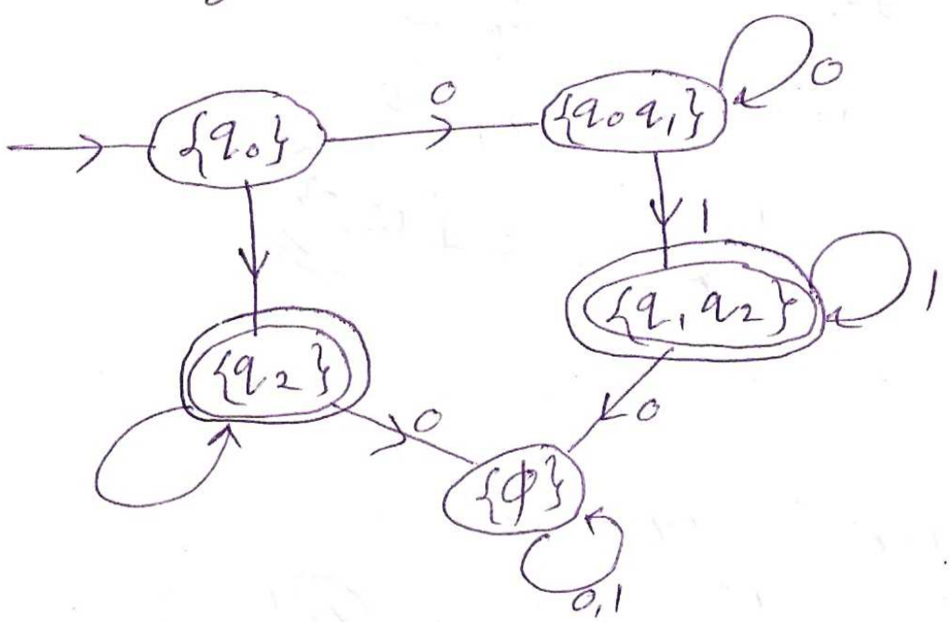
in DFA all those state is final state where Final state of given NFA appears

$\delta_2 (\{ q_0, q_1, q_2 \dots q_k \}, 0) = \delta_1 (q_0, 0) \cup \delta_1 (q_1, 0) \cup \dots \cup \delta_1 (q_k, 0)$

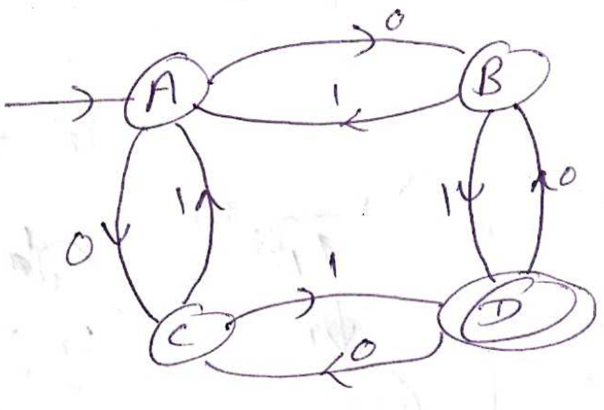
Transition table for DFA

δ_2	0	1
$\rightarrow \{q_0\}$	$\{q_0 q_1\}$	$\{q_2\}$
$\{q_0 q_1\}$	$\{q_0 q_1\}$	$\{q_1, q_2\}$
$\{q_2\}$	$\{\phi\}$	$\{q_2\}$
$\{q_1, q_2\}$	$\{\phi\}$	$\{q_1, q_2\}$
$\{\phi\}$	$\{\phi\}$	$\{\phi\}$

Transition diagram:



Q convert NFA to DFA.

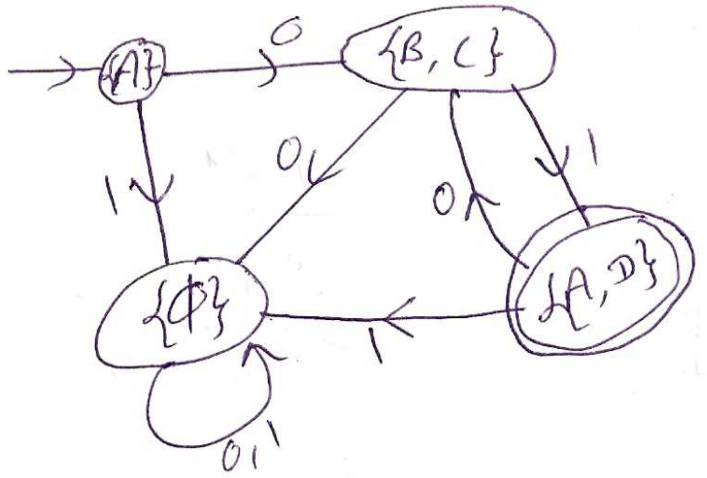


δ	0	1
A	{B, C}	ϕ
B	—	{D, A}
C	—	{A, D}
D	{B, C}	—

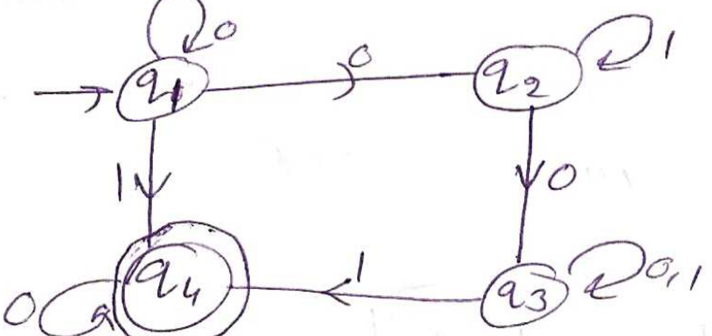
Transection table for DFA

δ	0	1
{A}	{B, C}	{ϕ}
{B, C}	{ϕ}	{A, D}
{ϕ}	{ϕ}	{ϕ}
{A, D}	{B, C}	{ϕ}

Transition diagram:

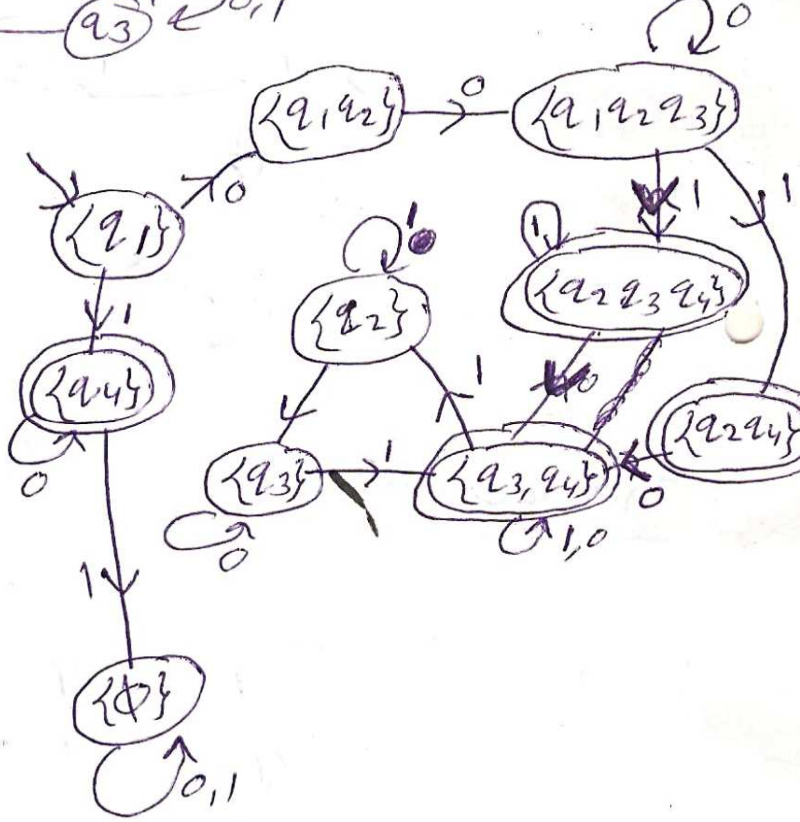


Q Convert NFA to DFA.



DFA Transition Table

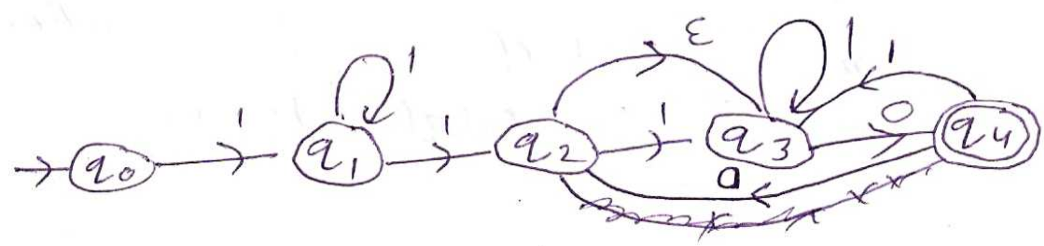
δ	0	1
$\{q_1\}$	$\{q_1, q_2\}$	$\{q_4\}$
$\{q_1, q_2\}$	$\{q_1, q_2 q_3\}$	$\{q_2, q_4\}$
$\{q_4\}$	$\{q_4\}$	ϕ
$\{q_1 q_2 q_3\}$	$\{q_1 q_2 q_3\}$	$\{q_2 q_3 q_4\}$
$\{q_2 q_4\}$	$\{q_3, q_4\}$	$\{q_2\}$
$\{q_2 q_3 q_4\}$	$\{q_3 q_4\}$	$\{q_2 q_3 q_4\}$
$\{q_3 q_4\}$	$\{q_3, q_4\}$	$\{q_3 q_4\}$
ϕ	ϕ	ϕ
$\{q_2\}$	$\{q_3\}$	$\{q_2\}$
$\{q_3\}$	$\{q_3\}$	$\{q_3 q_4\}$



NFA with ε-transitions

if a Finite Automata is modified to permit transitio without input symbols along with other transition ~~of~~ on input symbols then we get a NFA with ε transition, because the transition made without symbols are called as ε transition.

ex



This m/c accept all those string which ~~accept~~ start with 11 and end with 10

NFA with ε transitions will also be denoted as Five tuple.

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

where Q, Σ, q_0, F are having usual meaning and δ defines a mapping from

$$Q \times (\Sigma \cup \varepsilon) \text{ to } 2^{Q}.$$

ε - closure (q) = set of all those states of ~~FA~~

the NFA with ε transitions ~~which can be reached~~ which can be reached from q on a path labeled by ε. i.e. without consuming any input symbol.

a closure (q) = The union of ε-closure ~~set~~ of all those states of NFA with ε transitions which can be reached from q on a path labeled by "a".

<u>conversion of NFA with ε Transition to NFA without transition</u> :

Given NFA with ε

$$M_1 = \{ Q_1, \Sigma_1, \delta_1, q_{01}, F_1 \}$$

obtain NFA without ε transition

$$M_2 = \{ Q_2, \Sigma_2, \delta_2, q_{02}, F_2 \}$$

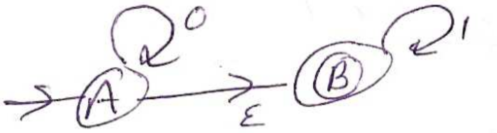
where $Q_2 = Q_1$, $\Sigma_2 = \Sigma_1$, $q_{02} = q_{01}$

$\delta_2 (q, 0)$ = 0 closure (q) in given m/c

$F_2 = F_1$ if ε closure (q_{01}) does not contain any final state

$F_2 = F_1 \cup q_{02}$ if ε closure (q_{01}) contain any Final state.

Q convert in NFA without ε transition



{ Language of this m/c is $0^m 1^n \mid m,n \geq 0$ }

ε closure of

ε closure of $(A) = \{A, B\}$
ε closure of $(B) = \{B\}$

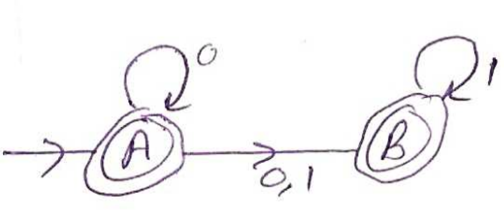
0 closure of $(A) = \{A, B\}$ → { 0 closure of A means: union of ε closure of all those states where we can reached after 0 transition From the states of ε closure of q. }
1 closure of $(A) = \{B\}$
0 closure of $(B) = \phi$
1 closure of $(B) = \{B\}$

NFA without ε

$Q_2 = Q_1 = \{A, B\}$
$\Sigma_2 = \Sigma_1 = \{0, 1\}$
$q_{02} = q_{01} = \{A\}$

$\delta_2(A, 0)$ = 0 closure of $(A) = \{A, B\}$
$\delta_2(A, 1)$ = 1 closure of $(A) = \{B\}$
$\delta_2(B, 0)$ = 0 closure of $(B) = \phi$
$\delta_2(B, 1)$ = 1 closure of $(B) = \{B\}$

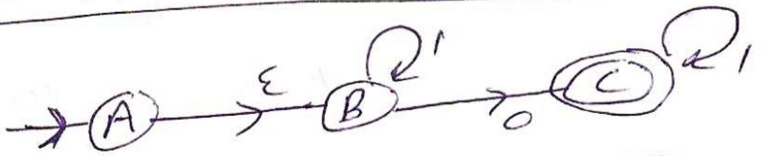
δ_2	0	1
A	{A, B}	{B}
B	–	{B}



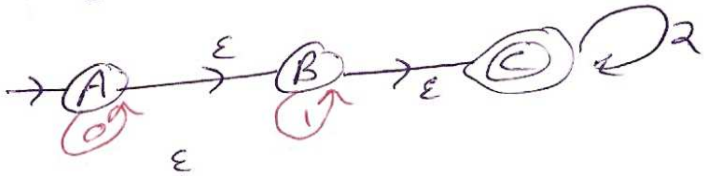
Here ε closure of (q_{01}) contain Final state B so
$F_2 = F_1 \cup q_{01} = B \cup A = \{A, B\}$

Q Convert in NFA without transition:

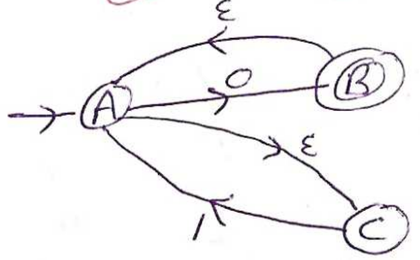
(i)



(ii)



(iii)



Solution:

(i) ε closure of (A) = {A, B}

ε closure of (B) = {B}

ε closure of (C) = {C}

0 closure of (A) = {C}

0 closure of (B) = {C}

0 closure of (C) = ϕ

1 closure of (A) = {B}

1 closure of B = {B}

1 closure of C = {C}

δ_2	0	1
A	{C}	{B}
B	{C}	{B}
C	—	{C}