AUTOMATA THEORY

UNIT- IV

Compiled by Prof K Maheswaran

K. MAHESWARAN

HISTORY

Automata theory is the study of abstract computing devices, or "machines." Before there were computers, in the 1930's, A. Turing studied an abstract machine that had all the capabilities of today's computer

In the 1940's and 1950's, simpler kinds of machines, which we today call "finite automata," These automata, originally proposed to model brain function, turned out to be extremely useful for a variety of other purposes,

HISTORY…

1950's, the linguist N. Chomsky began the study of formal "grammars." While not strictly machines, these grammars have close relationships to abstract automata and

In 1969, S. Cook extended Turing's study

- All of these theoretical developments bear directly on what computer scientists do today.

Finite Automata

Finite automata are a useful model for many important kinds of hardware and software.

1. Software for designing and checking the behavior of digital circuits.
2. The "lexical analyzer" of a typical compiler, that is, the compiler component that breaks the input text into logical units, such as identifiers, keywords, and punctuation.
3. Software for scanning large bodies of text, such as collections of Web pages, to find occurrences of words, phrases, or other patterns.
4. Software for verifying systems of all types that have a finite number of distinct states, such as communications protocols or protocols for secure exchange of information.

Applications of Finite Automata

- Text search

AUTOMATA

AUTOMATA

- DFA- Deterministic Finite Automation
- NFA- Non-deterministic Finite Automation

DFA(definition)

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

where,

Q - is a set of finite States

$\sum$- set of input symbols.(input alphabets)

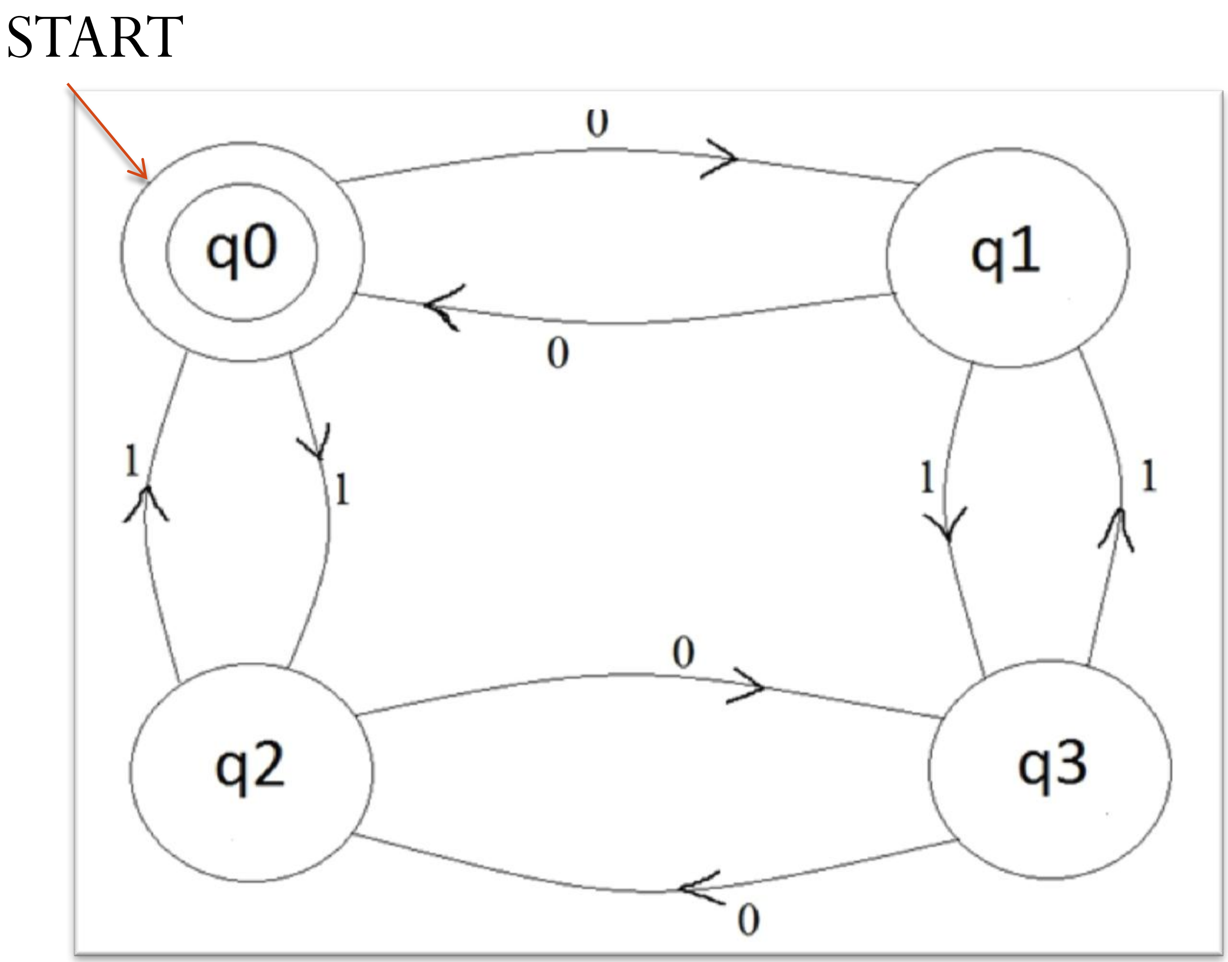
q_0- initial states.

F - set of final states.

δ - $Q * \Sigma \longrightarrow Q$

This is called "5 –tuple" form

Example: Fig. 1: machine



Convert the following FA into diagrammatic and tabular format.

Given 5 –tuple form:

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

where $Q=\{q_0, q_1, q_2, q_3\}$

$\Sigma=[0,1]$

q_0=initial state

$F=\{q_0\}$

Transient notation:

$\delta(q_0,0)=q_1$ $\delta(q_0,1)=q_2$

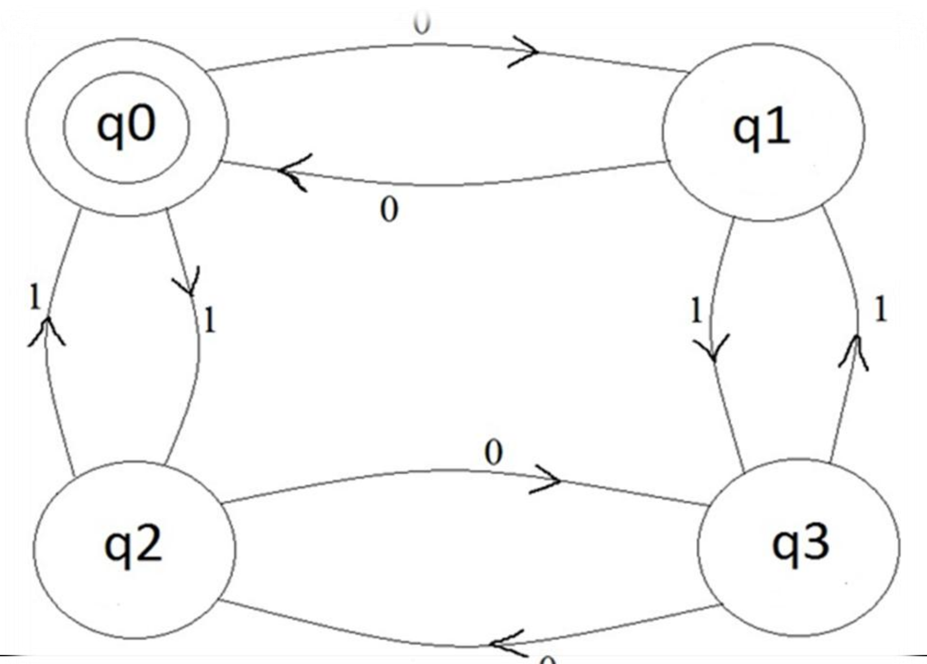
$\delta(q_1,0)=q_0$ $\delta(q_1,1)=q_3$

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

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

Solution:

diagrammatic form



Example(contd..,)

State Table:

Inputs

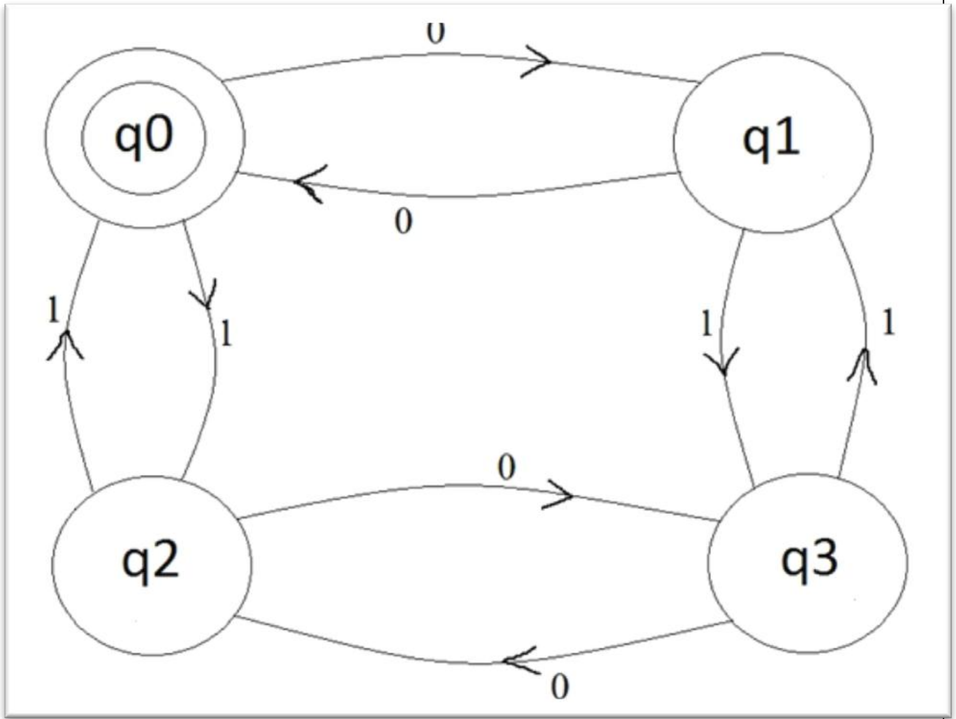
Symbols

δ	0	1
q_0	q_1	q_2
q_1	q_0	q_3
q_2	q_3	q_0
q_3	q_2	q_1

Problem:

1. **Check Whether 0110 is accepted or rejected:**

Solution:



$$\delta(q_0,0)=q_1$$

$$\delta(q_1,1)=q_3$$

$$\delta(q_3,1)= q_1$$

$$\delta(q_1,0)= q_0 \in F$$

Therefore, 0110 is accepted.

2 .Check Whether 101 is accepted or rejected:

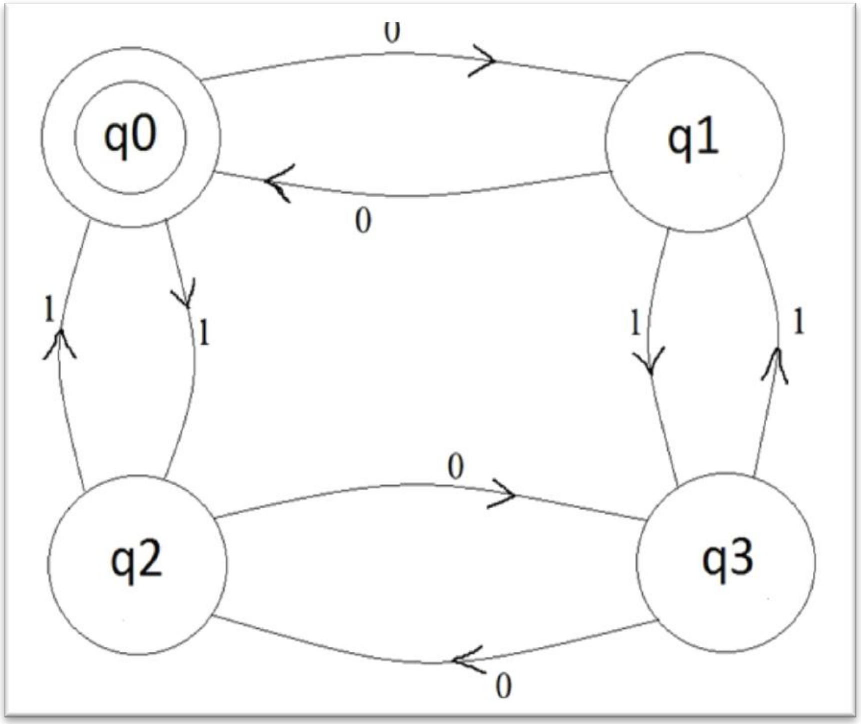
Solution:

$$\delta(q_0,1)=q_2$$

$$\delta(q_2,0)=q_3$$

$$\delta(q_3,1)= q_1 \notin F$$

Therefore, 101 is rejected.

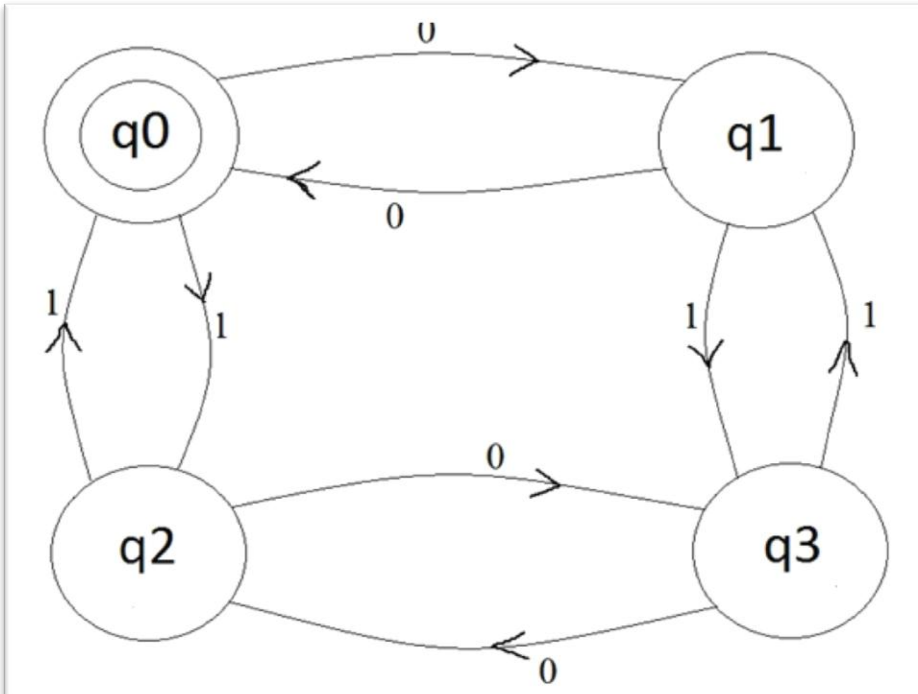


Language accepted by machine M (L(M))

Given:

$M=(Q, \sum, \delta, q_0, F)$ is the machine, x is the string(word) then the language accepted by the machine, $L(M)=\{ x \mid \delta (q_0 , x) = q \in F\}$

Example 3. Derive the language accepted by the machine, and check whether 1001 is accepted or not.



Solution:

Language accepted by the machine,

L(M)= {x | x consist of even no. of 0's and even no. of 1's}.

Non-deterministic Finite Automation (NFA)

<u>**Definition:**</u>

$$\mathbf{M=(Q, \sum, \delta, q_0, F)}$$

where,

Q - finite no. of states

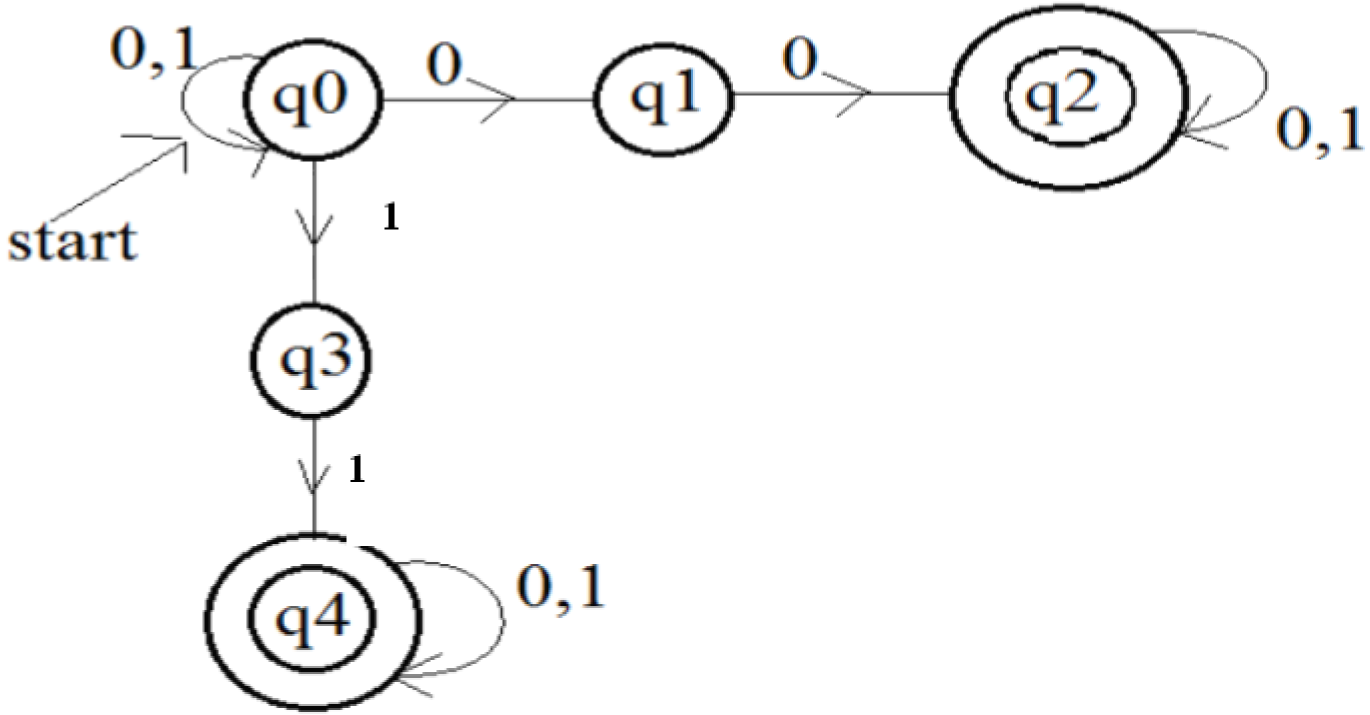
∑ - finite no. of Inputs

δ - is a function- (Q X ∑) $\longrightarrow$ $\mathbf{2^Q}$

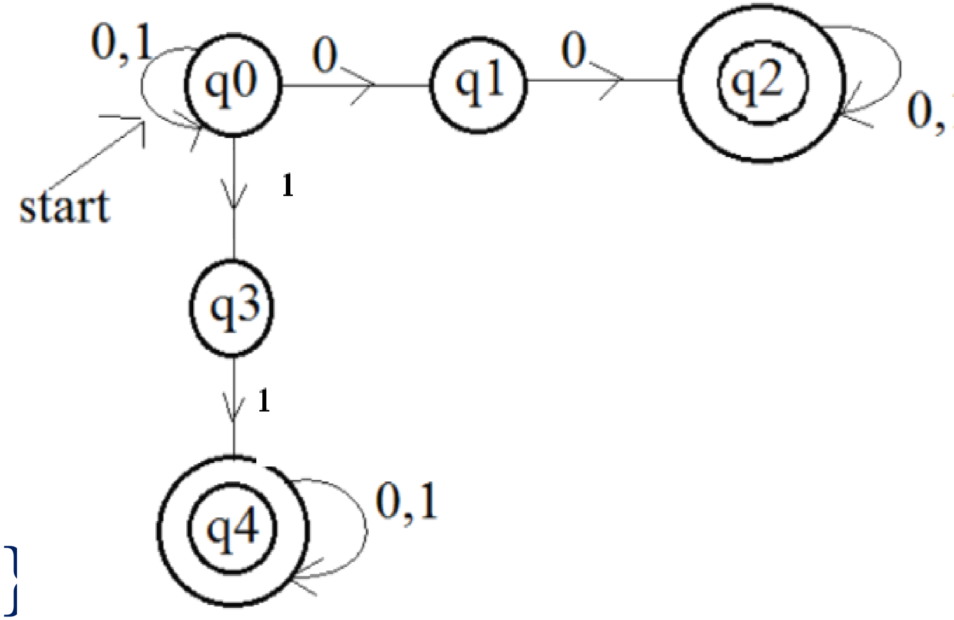
q_0 - Starting State

F – final state.

Fig.2:



Ex: 1 check whether 1010 accept or not



Solution:

$\delta\,(q_0,1) = \{(q_0,q_3)\}$

$\delta\,(\{q_0,q_3\},0) = \delta\,((q_0,0)\cup\delta(q_3,0))$

$= \{q_0, q_1\} \cup \phi = \{q_0,q_1\}$

$\delta\,(\{q_0,q_1\},1) = \delta((q_0,1)\cup(q_1,1))$

$= \{q_0,q_3\} \cup \phi = \{q_0,q_3\}$

Contd..,

$\delta(\{q_0,q_3\},0) = \delta(q_0,0) \cup \delta(q_3,0)$

$= \{q_0,q_1\} \cup \phi = \{q_0,q_1\}$

$\delta(q_0,1010) = \{q_0,q_1\} \notin F$ i.e q_2,q_4

Therefore, $L(M)=\{x \mid \delta(q_0,x) \cap F \neq \phi\}$

So 1010 is **rejected**.

Note:

$\delta : Q * \Sigma \rightarrow Q$	---	DFA
$\delta : Q * \Sigma \rightarrow 2^Q$	---	NFA

Regular Expression

- ϕ is a Regular Expression denoting an Empty Set { }.
- Є is a Regular Expression denoting a Set {Є}.
- 'a' is a Regular Expression denoting a Set {a}.
- If r and s is a Regular Expression denoting a Set R and S then (r+s), (rs), r* are Regular Expression denoting RUS, R.S, R*S respectively.

Example:

$0 = \{0\}$

$0^* = \{Є , 0 , 00 , 000 , ...\}$

$0^+ = \{0 , 00 , 000 , ...\}$

$0^+ = 00^* = 0^*0$

Definition:

$$L^* = \bigcup_{i=0}^{\infty} L^i$$

$$L^+ = \bigcup_{i=1}^{\infty} L^i$$

Example:

1. L1 = { 10 , 1 }
L2 = { 011 , 11} Find L1 . L2 = ?

Solution:

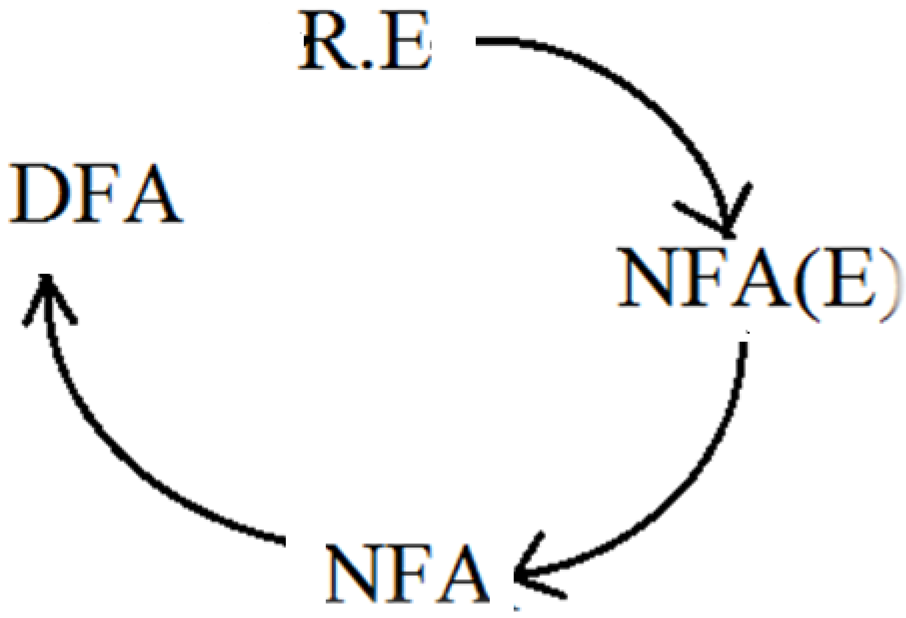
L1 . L2 ={10 011, 1011, 111}

2. **(0+1)* = {Є, 0, 1, 00, 10,01,……}**

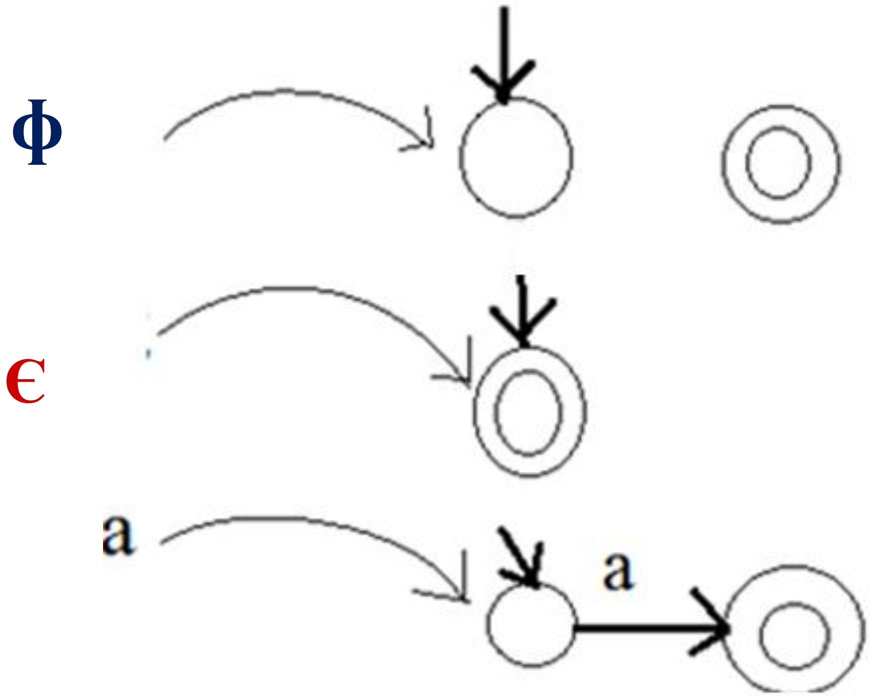
Home work:

3. **(0+1)* 00 (0+1)* -- It contains 2 consecutive zero.**
4. **(1+0)***
5. **(0+ Є) (1+10)***
6. **1* 2* 3* …..**

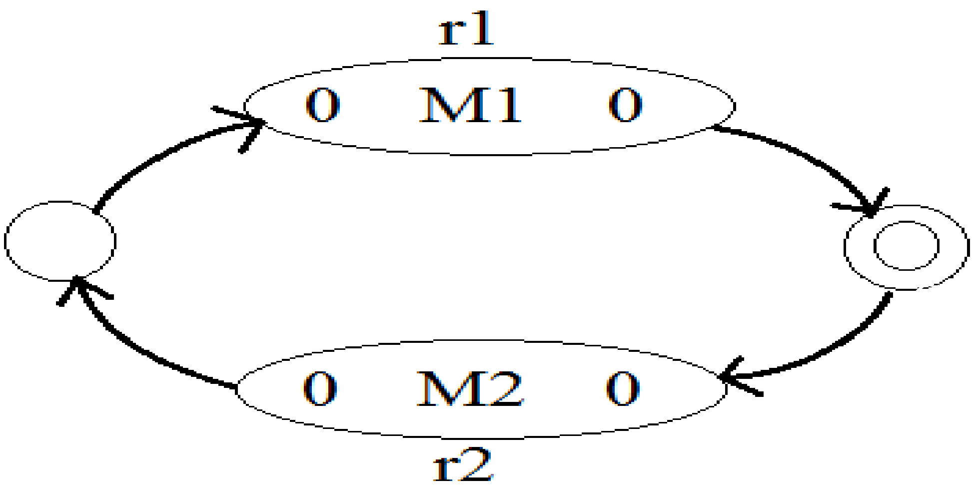
1. R . E => NFA with Є.
2. NFA with Є => NFA without Є.
3. NFA without Є => DFA.
4. DFA => R . E



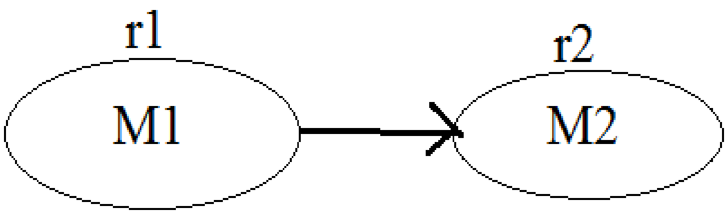
Regular Expression



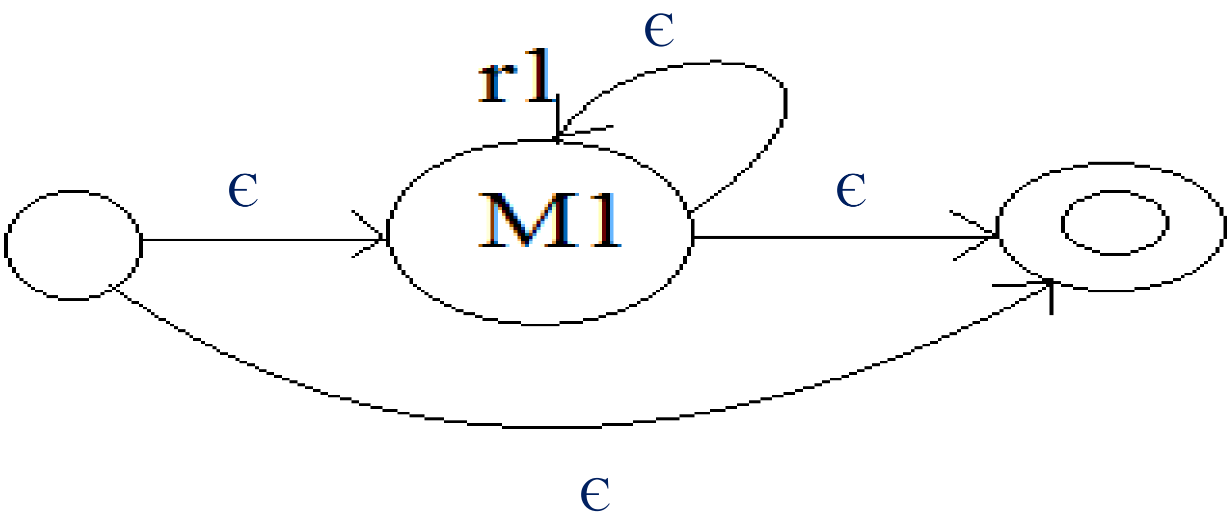
$r_1 + r_2$



$r_1 \cdot r_2$

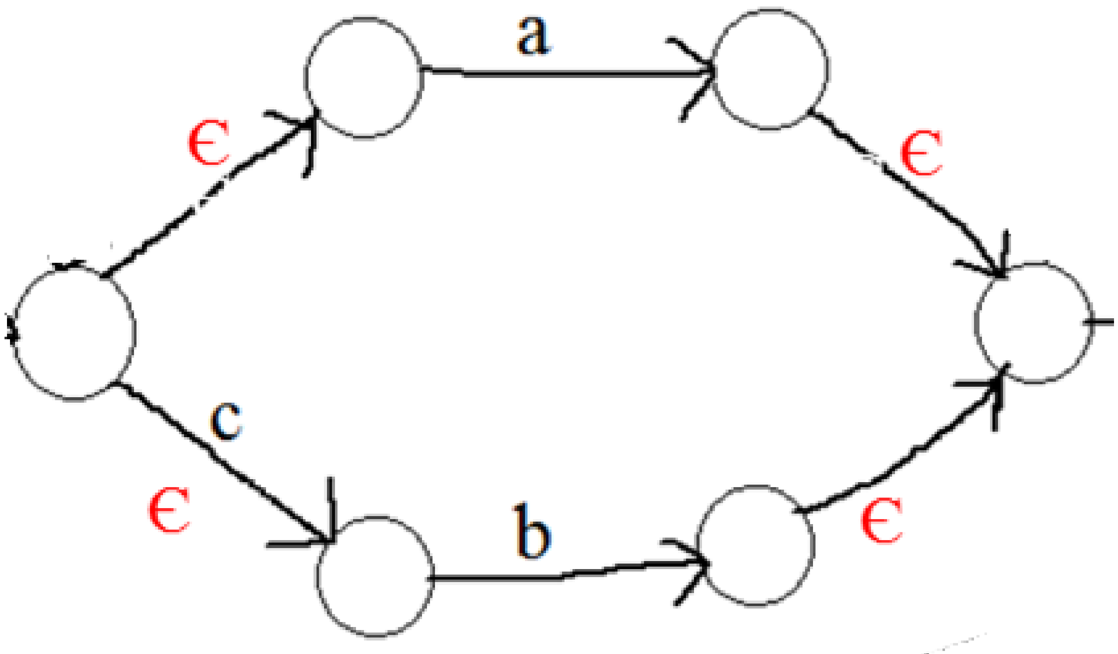


$r_1^* = \{\epsilon, r, rr, \ldots\ldots\}$

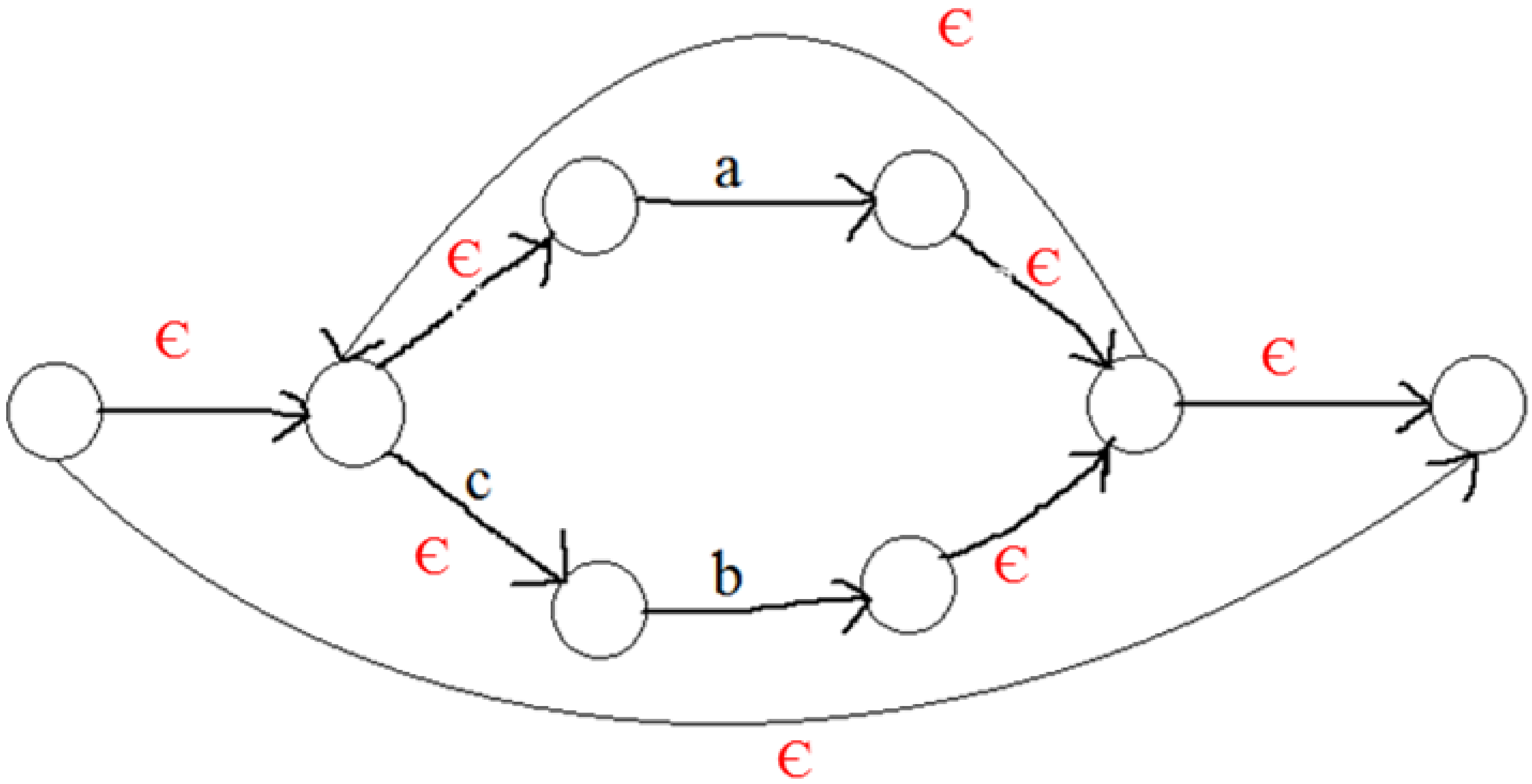


Construct a NFA for a given R.E. :(a+b)*abb

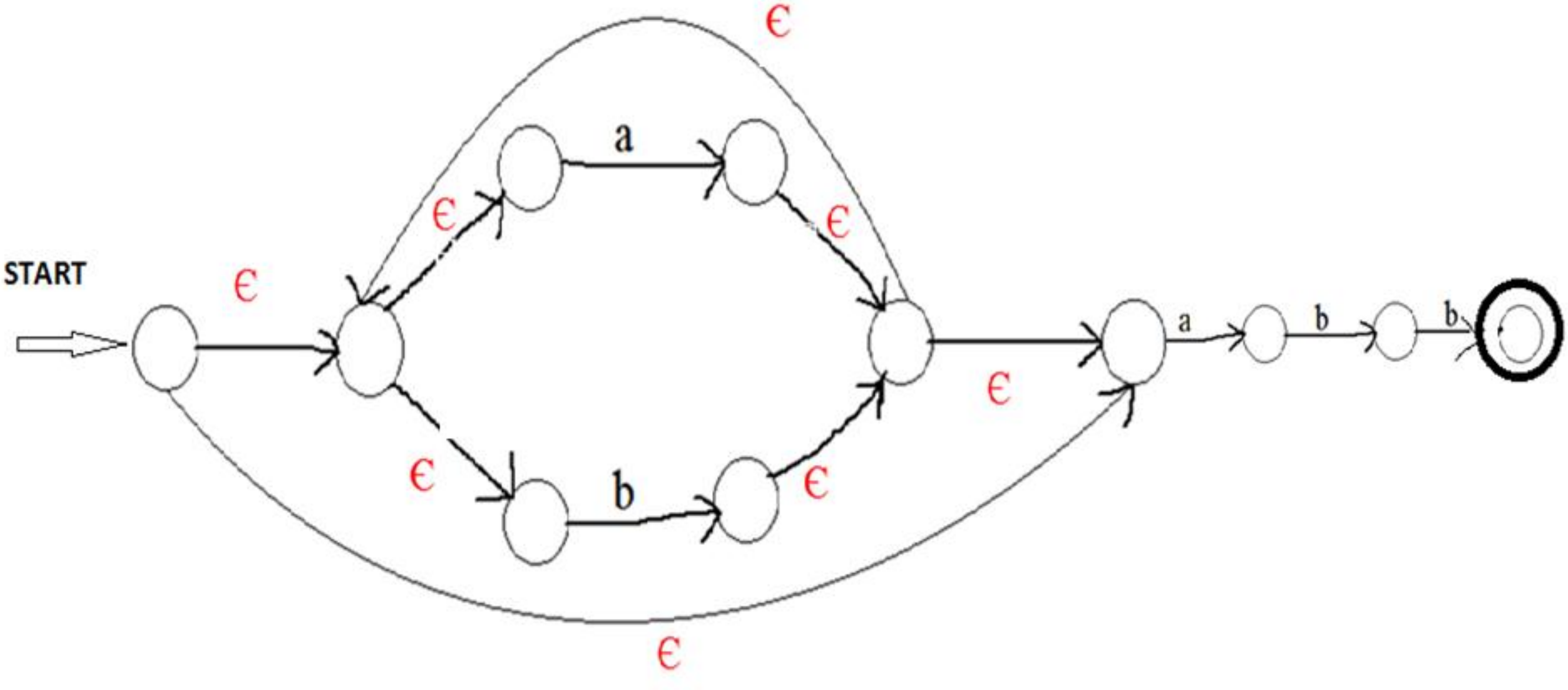
- **a+b :**



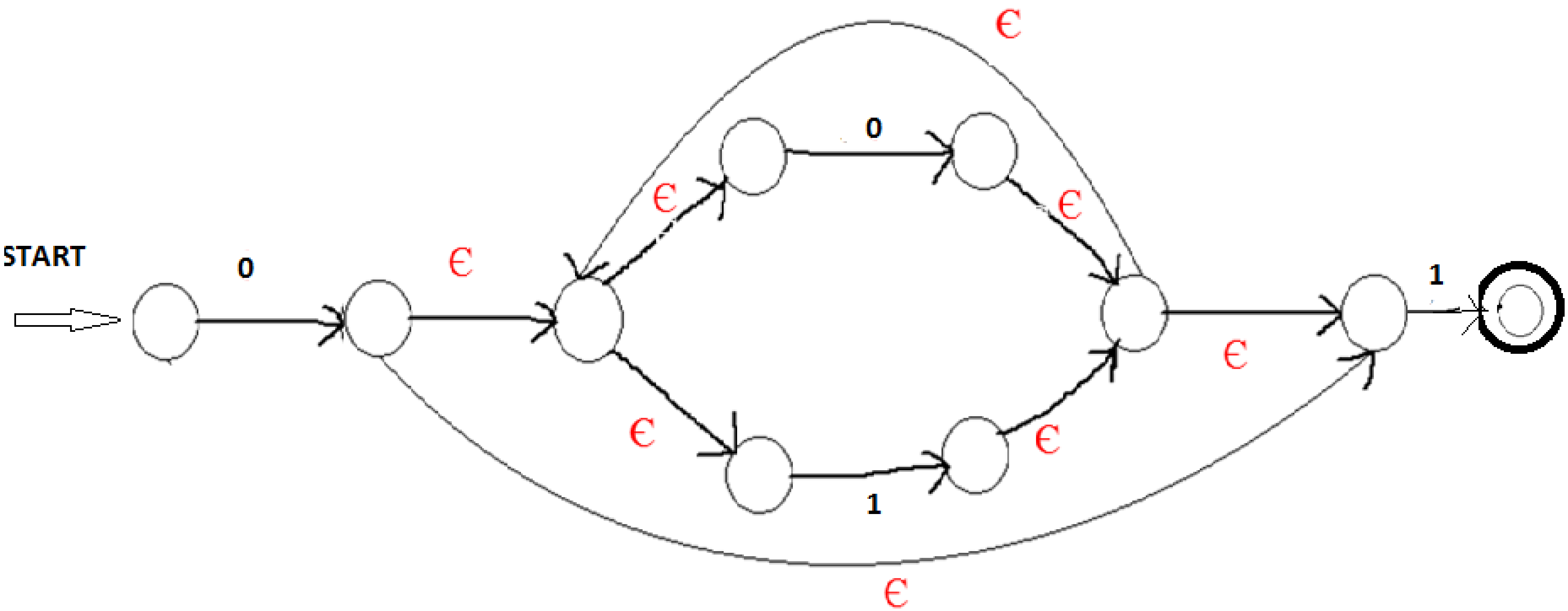
(a+b)*



(a+b)*abb =>



Home Work: 0 (0 + 1) * 1



How to convert NFA with ∈ to NFA without ∈

GIVEN: **$M = (Q, \Sigma, \delta, q_0, F)$ NFA with ∈**

Construct: $M' = (Q', \Sigma', \delta', q_0', F')$ NFA without ∈

where $Q' = Q$,

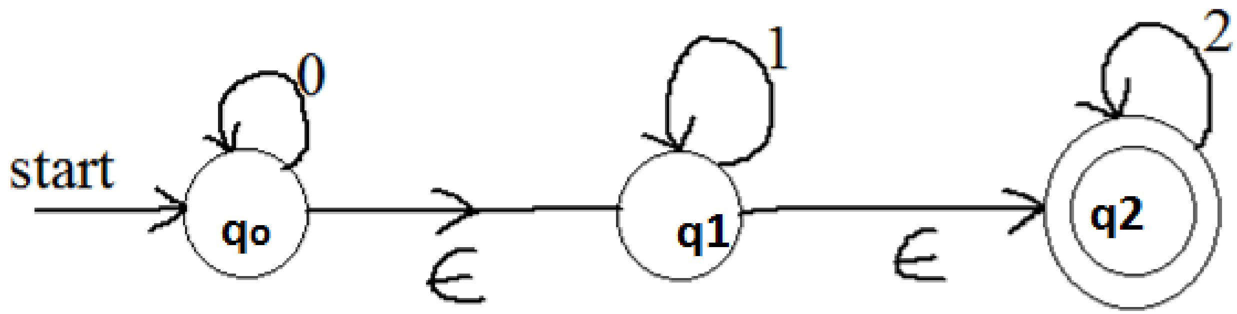
$\Sigma' = \Sigma$,

$\delta'(q_1, a) = \hat{\delta}(q_0, a)$,

$q_0' = q_0$,

$$F' = \begin{cases} F \cup \{q_0\} & \text{if } \in\text{-closure}(q_0) \text{ contains a state of } F. \\ F & \text{otherwise.} \end{cases}$$

Convert the following NFA with Є to NFA without Є



Solution:

Tabular format:

δ	0	1	2	Є
q_0	$\{q_0\}$	Φ	Φ	$\{q_1\}$
q_1	Φ	$\{q_1\}$	Φ	$\{q_2\}$
q_2	Φ	Φ	$\{q_2\}$	Φ

$$\hat{\delta}(q_0,0) = \delta(q_0, \in 0 \in)$$

$\in$

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

0

$= \{q_0\}$

$\in$

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

$$\hat{\delta}(q_0,1) = \delta(q_0, \in 1 \in)$$

$\in$

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

1

$= \{q_1\}$

$\in$

$= \{q_1, q_2\}$

Similarly We can derive for all other.

$\hat{\delta}(q_0,0) = \delta(q_0, \epsilon\ 0\ \epsilon)$

$\downarrow \epsilon$

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

$\downarrow 0$

$= \{q_1\}$

$\downarrow \epsilon$

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

$\hat{\delta}(q_0,1) = \delta(q_0, \epsilon\ 1\ \epsilon)$

$\downarrow \epsilon$

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

$\downarrow 0$

$= \{q_1\}$

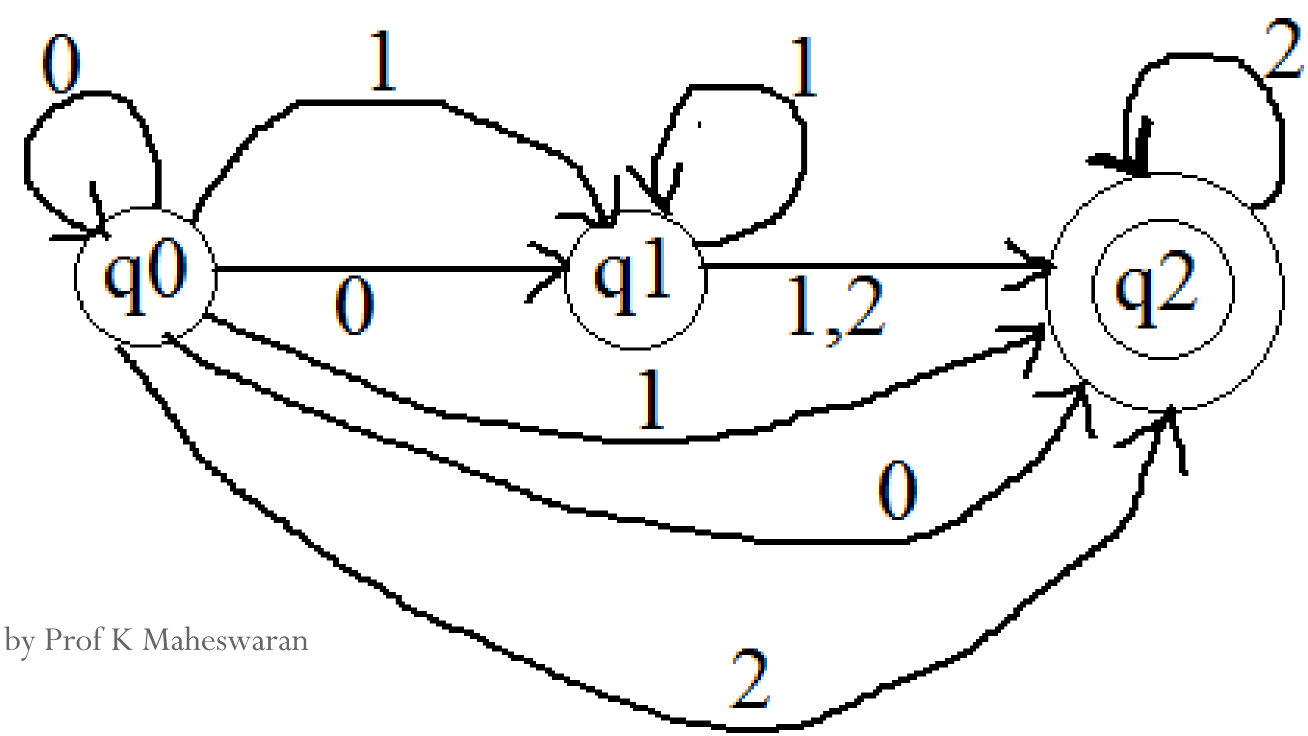
$\downarrow \epsilon$

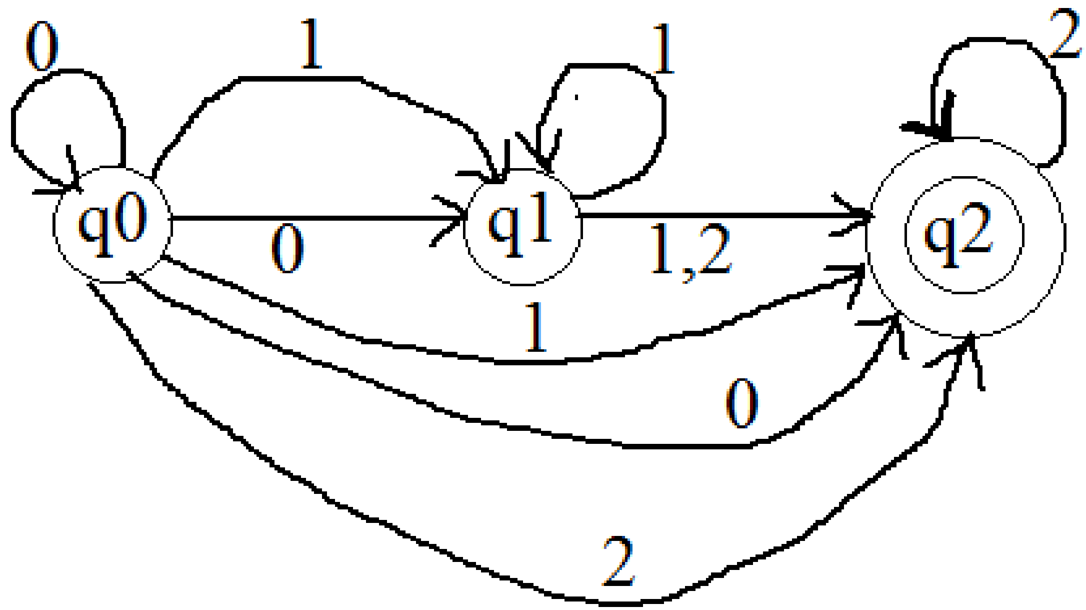
$= \{q_1,q_2\}$

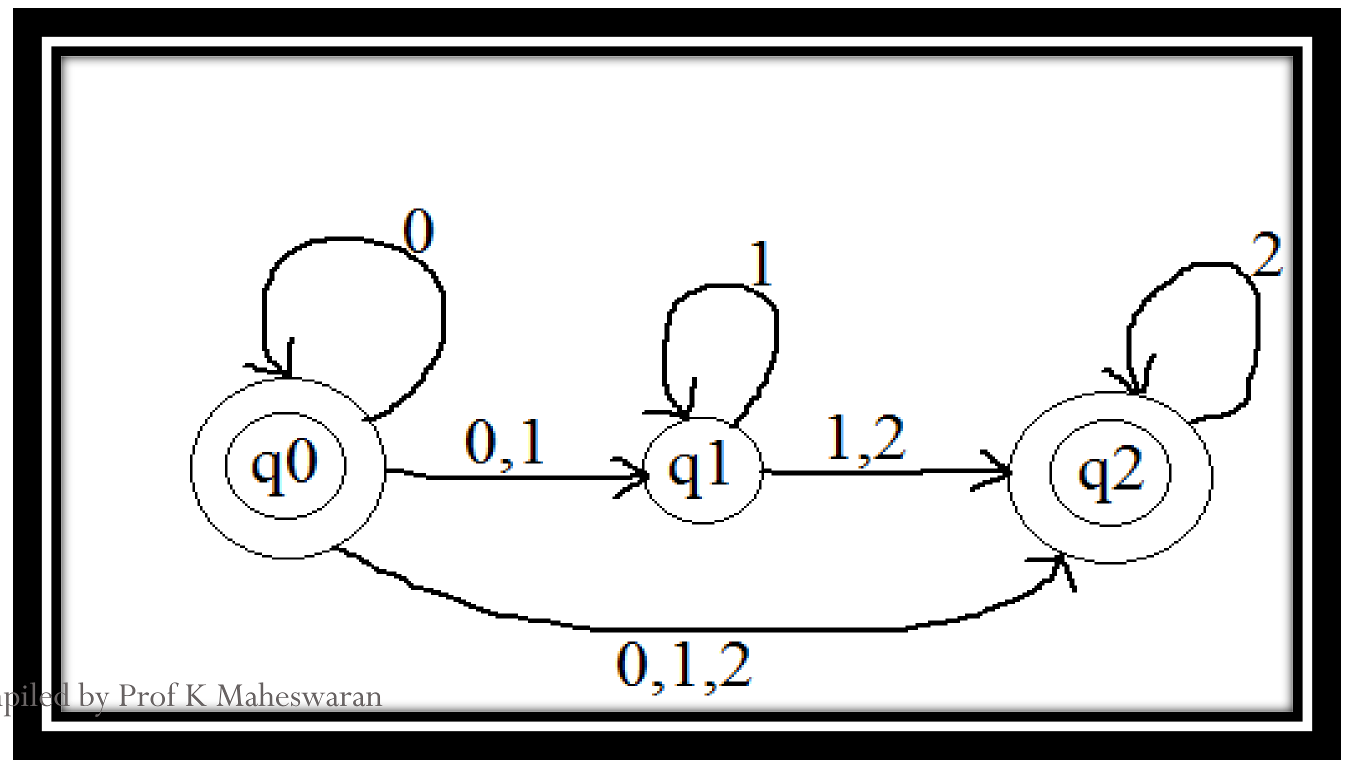
Similarly We can derive for all other.

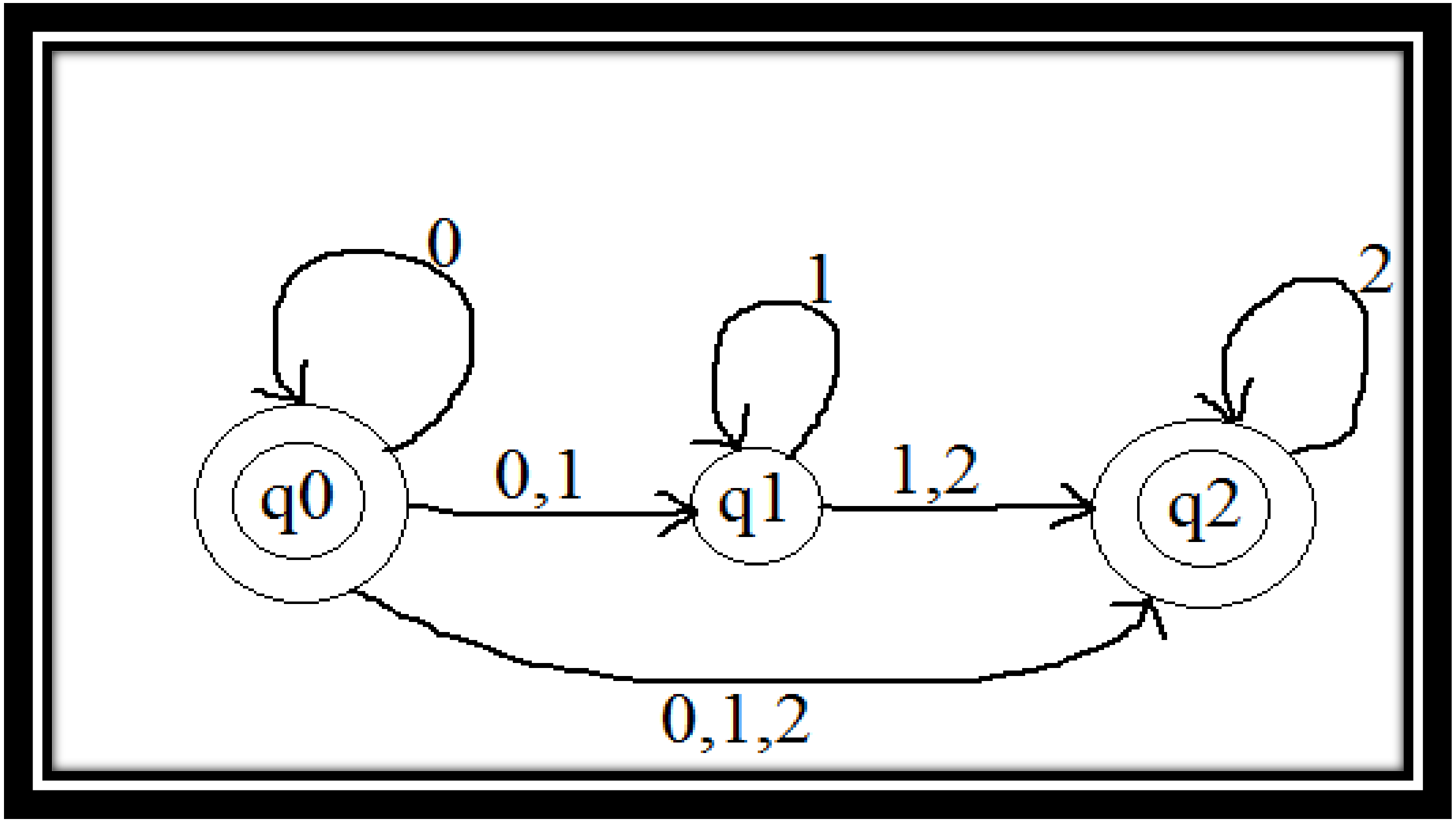
δ'	0	1	2
q_0	$\{q_0,q_1,q_2\}$	$\{q_1,q_2\}$	$\{q_2\}$
q_1	Φ	$\{q_1,q_2\}$	$\{q_2\}$
q_2	Φ	Φ	$\{q_2\}$

δ'	0	1	2
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	Φ	$\{q_1, q_2\}$	$\{q_2\}$
q_2	Φ	Φ	$\{q_2\}$




0
1
1
2
q0
0
q1
1,2
q2
1
0
2


0
1
2
q0
0,1
q1
1,2
q2
0,1,2



0
1
2
q0
0,1
q1
1,2
q2
0,1,2

Construct DFA(M') from NFA(M):

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

where

$Q' = 2^{Q}$

$\Sigma' = \Sigma$

$q_0 = [q_0]$

F' = set of all states in Q' containing a final statement of M.

$\delta' ([q_1, q_2, \ldots , qi] , a) = [P_1, P_2, \ldots , P_j]$

$\delta' (\{q_1, q_2, \ldots , qi\} , a) = \{P_1, P_2, \ldots , P_j\}$

Problem:

Construct DFA:

Given: NFA

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

where

$Q=\{ q_0, q_1 \}$

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

$F=\{ q_1 \}$

δ	0	1
q_0	$\{q_0,q_1\}$	$\{q_1\}$
q_1	Φ	$\{q_0,q_1\}$

Required DFA:

$M' = (Q', \sum', \delta', q_0', F')$

where

$Q' = 2^Q$ i.e., $=\{ \Phi, [q_0], [q_1], [q_0,q_1] \}$

$\sum' = \sum = \{ 0, 1 \}$

$q_0' = [q_0]$

$F' = \{[q_1], [q_0,q_1]\}$

$\delta' = Q \times \sum$

δ'	0	1
Φ	Φ	Φ
$[q_0]$	$[q_0,q_1]$	$[q_0]$
$[q_1]$	Φ	$[q_0,q_1]$
$[q_0,q_1]$	$[q_0,q_1]$	$[q_0,q_1]$

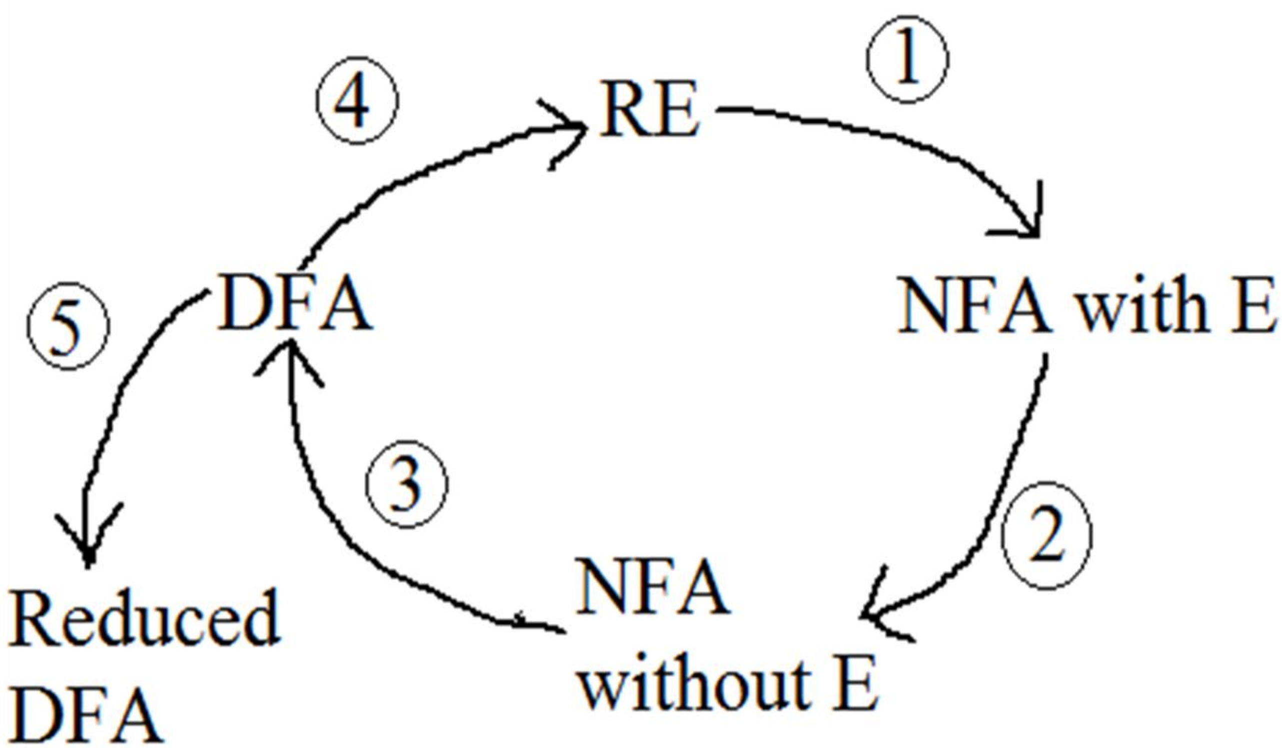
Prove: L (M) = L (M')

$L(M) \quad L(M')$

$L(M') \subset L(M)$

$X \in L(M) \Rightarrow x \in L(M')$

$\{x \mid \delta(q_0, x) \cap F = \Phi\} \Leftrightarrow \{x \mid \delta(q_0', x) \in F'\}$



RE
1
NFA with E
2
NFA
without E
3
DFA
4
5
Reduced
DFA

END