2. FINITE AUTOMATA

CHAPTER - 2
FINITE AUTOMATA

2.1 INTRODUCTION

Theory of Computation (TOC) describes the basic ideas and models underlying computing.

Computation is executing an algorithm ie., it involves taking some inputs and performing required operation on it to produce an output.

Computation : input x $\longrightarrow$ $\boxed{\text{function } f}$ $\longrightarrow$ output $y=f(x)$

Broadly computation is simply a sequence of steps that can be performed by computer.

TOC suggests various abstract models of computation, represented mathematically.

The computer which performs computation are not actual computers; they are *abstract machines*. Our focus is on abstract machines that can be defined mathematically. Some examples of abstract machine are :

(i) Turing machines (powerful as real computers) - Universal model.

(ii) Finite automata (simple) - Restricted model.

Theory of computation have a wide range of *"applications"* in:

(i) Compiler Design

(ii) Robotics

(iii) Artificial Intelligence

(iv) Knowledge Engineering.

2.1.1 Finite Automata

It recognises *regular languages* only. It was developed by *"Scott Robin"* in 1950 as a model of a computer with limited memory.

It receives its input as a string, usually from an input tape. It delivers no output at all except an indication of whether the input is acceptable (or) not. Hence used for decision making problems.

Applications

(i) It is an useful tool in the design of Lexical analyser - a part of compiler that groups characters into tokens, indivisible units such as variable name and keyword.

(ii) Text editor

(iii) Pattern matching

(iv) File searching program

(v) Text processing (searching an occurrence of one string in a file)

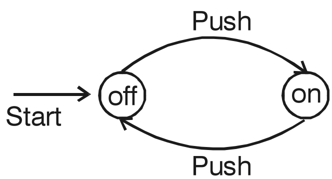
Limitations

(i) It can recognise only simple languages (regular)

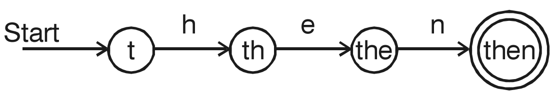
(ii) FA can be designed only for decision making problems.

Example 2.1

(i) Finite automaton for an On/Off switch - Digital systems



(ii) Lexical analysis - Recognising a string "then"



2.1.2 Context Free Languages

(i) It allows richer syntax than regular languages.

(ii) Context free grammars can be recognised by computing devices like *pushdown automata*.

(iii) Pushdown automata is a finite automata with an auxillary memory in the form of a stack.

(iv) It is immensely used in the design of parsers - another key portion of a compiler.

2.1.3 Turing Machines

(i) It was invented by British Mathematician *"Alan Turing"*

(ii) It is a most powerful abstract model.

(iii) It has infinite amount of tape memory accessible in both directions, that is left (or) right.

(iv) It can recognise recursively enumerable languages.

(v) It simulates digital computer in terms of power.

(vi) Some problems are theoritically solvable and are not practically solvable due to non polynomial time. If any function is not solvable by turing machine, it cannot be computed by digital computer.

2.2 FINITE STATE SYSTEMS

The *Finite Automaton* (FA) is a mathematical model of a system, with discrete *inputs* and *outputs* and a finite number of memory configuration called *states* and a set of *transitions* from state to state that occurs on input symbols from alphabet Σ.

Examples of finite automaton models are :

(i) Software for designing digital circuits-silicon compilers.

(ii) Lexical analyser of a compiler.

(iii) Searching for keywords in a file (or) on the web - Text editors.

The FA is classified as:

(i) Deterministic Finite Automata (DFA)

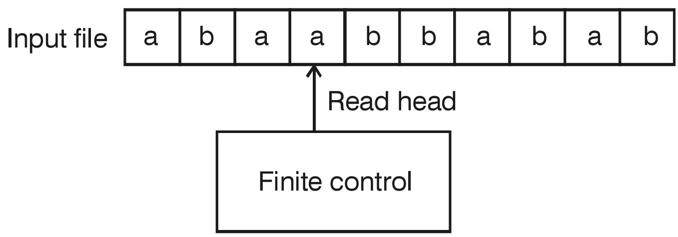
(ii) Non Deterministic Finite Automata (NFA)

The DFA and NFA are discussed in detail in the following sections.

2.3 DETERMINISTIC FINITE AUTOMATA (DFA)

A finite state automata is the simplest and most restricted model of a computer, we begin by looking at Deterministic Finite Automata (DFA).

2.3.1 Basic DFA



DFA is a language recogniser that has :

(i) An *input file* containing an input string.

(ii) A *finite control* - a device that can be in a finite number of states.

(iii) A *reader* - a sequential reading device

(iv) A *program*.

How DFA works ?

(i) *Initialisation :*

Reader (read head) should be over the leftmost symbol. Finite control is in start state.

(ii) *Single step :* Reader reads current symbol then, reader moves to the next symbol to the right.

Control enters a new state that (deterministically) depends on the current state and current symbol. There may be no desired next state, in which case the machine stops. The machine repeats this action.

(iii) *No current symbol :* All symbols have been read then, if control is in final state, the input string is accepted. Otherwise, the input string is not accepted.

2.3.2 DFA Specification

A DFA M is formally defined specified by a five tuple M = (Q, Σ, S, F, δ), where

(i) Q - a finite, non empty *set of states*.

(ii) Σ - a finite, non empty set of *input alphabets*.

(iii) S $\in$ Q - a *start state*.

(iv) F $\subseteq$ Q - a set of *final states* - *subset of Q*.

(v) δ - a *transition function*, defined as

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

current state	current symbol	next state
P	σ	δ (P, σ)

The transitions are represented in the form of transition table or transition diagram

Example 2.2

Transition table of DFA

DFA is specified by M = (Q, Σ, δ, S, F), where

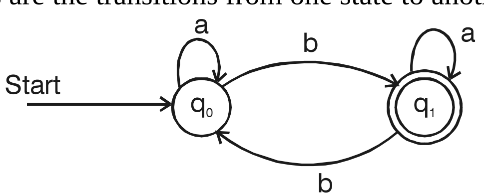
Q = $\{q_0, q_1\}$, S = q_0, F = $\{q_1\}$

$\Sigma = \{a, b\}$ and δ is given by

States	Inputs	
	a	b
$\rightarrow q_0$	q_0	q_1
q_1*	q_1	q_0

Transition diagram of DFA

"*Transition diagram*" associated with DFA is a directed graph whose vertices corresponds to states of DFA. The edges are the transitions from one state to another.



Note :

In the transition diagram, start state *s* is represented by $\rightarrow$ and the final states are represented by * or double circle.

2.3.3 Properties of Transition Function (δ)

(i) $\delta(q, \varepsilon) = q$

This means the state of the system can be changed only by an input symbol else remains in original state.

(ii) For all strings *w* and input symbol *a*

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

similarly $\delta(q, wa) = \delta(\delta(q, w), a)$

(iii) The transition function δ can be extended to (or) $\hat{\delta}$ that operates on states and strings (as opposed to states and symbols)

Basis : $\bar{\delta}(q, \varepsilon) = q$

Induction : $\bar{\delta}(q, xa) = \delta(\bar{\delta}(q, x), a)$

2.3.4 Language of a DFA

A string *x* is said to be accepted by DFA M = (Q, Σ, S, F, δ), if $\delta(q_0, x)$ = P for some P in F.

Method : A finite automata accepts a string $w = a_1 a_2 .... a_n$ if there is a path in the transition diagram which begins at a start state ends at an accepting state with the sequence of labels $a_1 a_2 ... a_n$

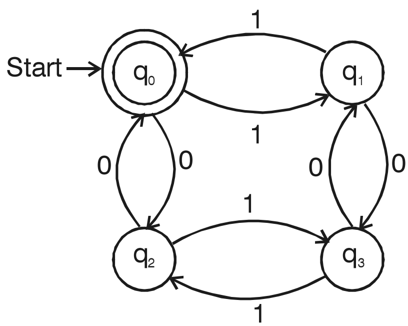
$\bar{\delta}$

* The Language accepted by finite automata (A) is

L (A) = $\{w : \bar{\delta}(q_0, w) \in F\}$ where F is a final state.

* The language accepted by finite automata's are called "*regular language*."

Example 2.3



Transition diagram

The DFA for the above transition is represented as

M = (Q, Σ, S, F, δ), where

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

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

S = q_0

F = $\{q_0\}$

δ : transition function is represented as a transition table

States	Inputs	
	0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

Suppose 110101 is input to M, check the validity of the input.

We note that finite automata is in start state and reads from left most.

$\therefore$ $\delta(q_0, 1) = q_1$

$\delta(q_1, 1) = q_0$ (Reader reads next symbols)

$\delta(q_0, 0) = q_2$ (Reader moves one position right)

$\delta(q_2, 1) = q_3$

$\delta(q_3, 0) = q_1$

$\delta(q_1, 1) = q_0$

since q_0 is a final state, the given string is accepted.

Example 2.4

Let M = (Q, Σ, S, F, δ) be a DFA with

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

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

S = q_0

F = $\{q_0\}$

δ is given by

States	Inputs	
	0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

Give the entire sequence for input string 110101.

Solution :

$$\begin{aligned}
\delta(q_0, 110101) &= \delta(\delta(q_0, 1), 10101) \\
&= \delta(q_1, 10101) \quad \text{(from transition table)} \\
&= \delta(\delta(q_1,1), 0101) \\
&= \delta(q_0, 0101) \\
&= \delta(\delta(q_0, 0), 101) \\
&= \delta(q_2, 101) \\
&= \delta(\delta(q_2, 1), 01) \\
&= \delta(q_3, 01) \\
&= \delta(\delta(q_3, 0), 1) \\
&= \delta(q_1, 1) \\
&= \delta(\delta(q_1,1), \varepsilon) \\
&= \delta(q_0, \varepsilon) \\
&= q_0 \text{ which is a final state. Hence accepted.}
\end{aligned}$$

The same operation $\delta(q_0, 110101)$ can be written directly from a transition table. That is:

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

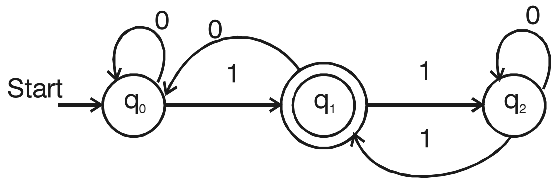
Example 2.5

Describe the language accepted by DFA M = ($\{q_0, q_1, q_2\}$, {0,1}, q_0, $\{q_1\}$, δ}), where δ is given by

States	Inputs	
	0	1
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_2	q_1

Solution :

The transition diagram is



$\delta(q_0, 0) = q_0$ non accepting state

$\delta(q_0, 01) = \delta(q_0, 1) = q_1$ accepting state

$\delta(q_0, 010) = \delta(q_0, 10) = \delta(q_1, 0) = q_0$ non accepting state.

Hence the above DFA accepts 01, 101, 0111. That is odd number of ones at the end of string.

Example 2.6

Design a DFA that accepts the strings defined by the language. L = $\{a^n\ b : n \geq 0\}$

Solution :

(i) The automata remains in initial state q_0 until the first b in encountered.

(ii) If b is the last symbol of the input, the string is accepted. It is called *trap state*.

(iii) If not, DFA goes to another state, from which it can never escape.

Example 2.7

Design a DFA that recognises the set of all strings on $\Sigma=\{a, b\}$ starting with the prefix *ab*.

Solution :

(i) Here the first two symbols are read and is found to be *ab*, no further decision need to be done. Hence it goes to *trap state*.

(ii) On the other hand, if the first symbol is not a and the second symbol is not b, automata goes to non final trap state.

Example 2.8

Construct a DFA that accepts input strings of 0's and 1's that ends with 11.

Solution :

The language accepted by this automata is

L={11, 011, 01011, 000111, ...}

(i) Machine will be in start state q_0

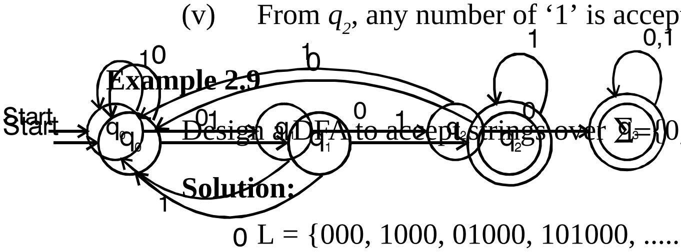
(ii) It will remain unchanged until it sees the input symbol '1'.

(iii) On getting '1', goes to q_1

(iv) From q_1, on input symbol '0' goes to q_0

From q_1, on input symbol '1', goes to q_2 (acceptance state)

(v) From q_2, any number of '1' is accepted and if any '0' comes, it goes to start state.



Example 2.9

Design a DFA to accept strings over Σ={0,1} with three consecutive 0's.

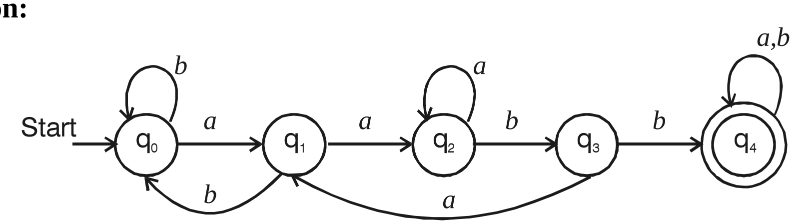
Solution:

L = {000, 1000, 01000, 101000, ...}

Example 2.10

Construct a DFA over Σ={a,b} containing a substring *aabb*.

Solution:



2.4 NON DETERMINISTIC FINITE AUTOMATA (NFA)

2.4.1 DFA Vs NFA

In a *DFA*,

(i) Each symbol causes a move (eventhough the state of the machine remains unchanged after the move)

(ii) The next state is completely determined by the current state and current symbol.

Where as in a *NFA*

(i) The machine can move without consuming any symbols

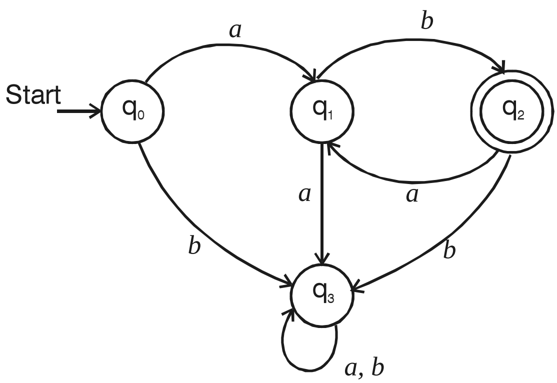
and sometimes there is no possible moves and sometimes there are more than one possible moves.

(ii) The state is only partially determined by the current state and input symbol.

Example 2.11

L = $\{(ab)^n \mid n > 0\}$. The corresponding DFA and NFA are :

DFA :



NFA :

2.4.2 Definition of NFA

The Non deterministic Finite Automata (NFA) is defined by a five tuple (Q, Σ, δ, S, F), where

Q - finite non empty set of states

Σ - Finite set of input alphabet

S∈Q - start state, belongs to Q

F⊆Q - set of final states, subset of Q.

δ - mapping function Q x Σ to 2^Q (2^Q is power set of Q).

2.4.3 Extended Transition Function ( )

Basis : $\overline{\delta}\ (q, \varepsilon) = \{q\}$

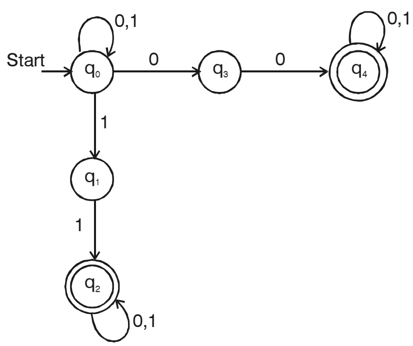
Induction : $\overline{\delta}\ (q, wa) = \bigcup_{P \in \overline{\delta}(q,w)} \delta(P, a)$ for each $w \in \Sigma^*$, $a \in \Sigma$ and P ∈ (q, w)

2.4.4 Language of a NFA

Language accepted by NFA is L (A) = $\{w : \overline{\delta}\ (q_0, w) \cap F \neq \phi\}$

Example 2.12

Consider the given NFA to check whether w=01001 is valid or not.



$\overline{\delta}$

Solution:

M = (Q, Σ, S, F, δ) is a NFA, where

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

Σ = {0,1}

S = q_0

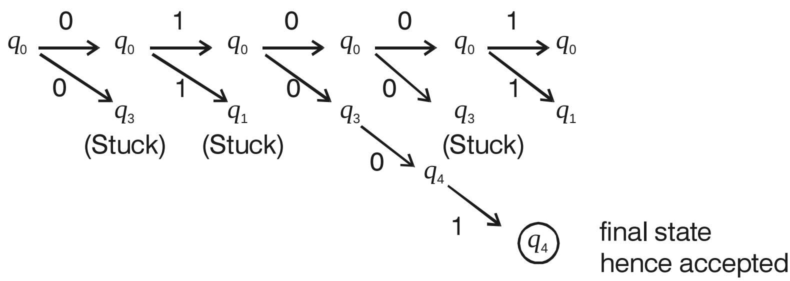
F = $\{q_2, q_4\}$

δ = Q x Σ to 2^Q

Transition table is

States	Inputs	
	0	1
q_0	$\{q_0,q_3\}$	$\{q_0,q_1\}$
q_1	ϕ	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$
q_3	$\{q_4\}$	ϕ
q_4	$\{q_4\}$	$\{q_4\}$

Method 1 :



The final transitions are q_0, q_1 and q_4, in which q_4 is a final state and hence the string is accepted.

Method 2 :

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

$$\begin{aligned}\delta(q_0, 01) &= \delta(\delta(q_0, 0), 1)\\ &= \delta(\{q_0, q_3\}, 1)\\ &= \delta(q_0, 1) \cup \delta(q_3, 1)\\ &= \{q_0, q_1\} \cup \phi\\ &= \{q_0, q_1\}\end{aligned}$$

$$\begin{aligned}\delta(q_0, 010) &= \delta(\delta(q_0, 01), 0)\\ &= \delta(\{q_0, q_1\}, 0)\\ &= \delta(q_0, 0) \cup \delta(q_1, 0)\\ &= \{q_0, q_3\} \cup \phi\\ &= \{q_0, q_3\}\end{aligned}$$

$$\begin{aligned}
\delta(q_0, 0100) &= \delta(\delta(q_0, 010), 0)\\
&= \delta(\{q_0, q_3\}, 0)\\
&= \delta(q_0, 0) \cup \delta(q_3, 0)\\
&= \{q_0, q_3\} \cup \{q_4\}\\
&= \{q_0, q_3, q_4\}
\end{aligned}$$

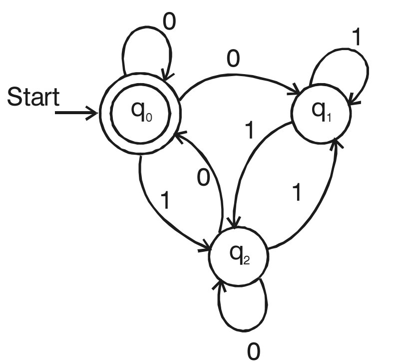
$$\begin{aligned}
\delta(q_0, 01001) &= \delta(\delta(q_0, 0100), 1)\\
&= \delta(\{q_0, q_3, q_4\}, 1)\\
&= \delta(q_0, 1) \cup \delta(q_3, 1) \cup \delta(q_4, 1)\\
&= \{q_0, q_1\} \cup \{q_4\} \cup \{q_4\}\\
&= \{q_0, q_1, q_4\}
\end{aligned}$$

$$\begin{aligned}
\delta(q_0, 01001) \cap F &= \{q_0, q_1, q_4\} \cap \{q_4\}\\
&= \{q_4\} = \text{final state}
\end{aligned}$$

Hence the given string is accepted.

Example 2.13

For the NFA shown, check whether the input string 0100 is accepted (or) not ?



Solution :

The transition table δ is

States	Inputs	
	0	1
q_0	$\{q_0,q_1\}$	$\{q_2\}$
q_1	ϕ	$\{q_1,q_2\}$
q_2	$\{q_0,q_2\}$	$\{q_1\}$

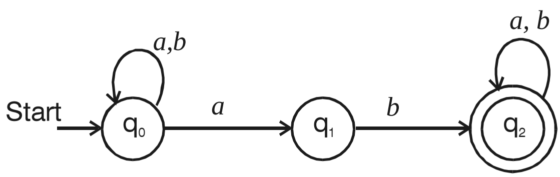
$$
\begin{aligned}
\text{Input string} &= 0100 \\
\delta(q_0, 0) &= \{q_0, q_1\} \\
\delta(q_0, 01) &= \delta(\delta(q_0, 0), 1) \\
&= \delta(\{q_0, q_1\}, 1) \\
&= \delta(q_0, 1) \cup \delta(q_1, 1) \\
&= \{q_2\} \cup \{q_1, q_2\} = \{q_1, q_2\} \\
\delta(q_0, 010) &= \delta(\delta(q_0, 01), 0) \\
&= \delta(\{q_1, q_2\}, 0) \\
&= \delta(q_1, 0) \cup \delta(q_2, 0) \\
&= \{q_0, q_1\} \cup \{q_0, q_2\} = \{q_0, q_1, q_2\} \\
\delta(q_0, 0100) &= \delta(\delta(q_0, 010), 0) \\
&= \delta(\{q_0, q_1, q_2\}, 0) \\
&= \delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0) \\
&= \{q_0, q_1\} \cup \{\phi\} \cup \{q_0, q_2\} \\
&= \{q_0, q_1, q_2\}
\end{aligned}
$$

Since it contains q_0 which is a final state, hence the string is accepted.

Example 2.14

Construct a NFA over alphabet $\Sigma = \{a, b\}$ that accepts string with substring ab.

Solution :



Example 2.15

Construct a NFA that accepts string which has 3rd symbol 'b' from right.

Solution :

Example 2.16

The NFA with states {1, 2, 3, 4, 5} and input alphabet $\Sigma=\{a, b\}$ has the following transition table.

States	Inputs	
	a	b
1	{1,2}	{1}
2	{3}	{3}
3	{4}	{4}
4	{5}	ϕ
5	ϕ	{5}

i) Calculate $\delta(1,ab)$

ii) Calculate $\delta(1,abab)$

Solution :

(i) $\delta(1,a) = \{1, 2\}$

$\delta(1, ab) = \delta(\{1, 2\}, b)$

$= \delta(1,b) \cup \delta(2,b)$

$= \{1\} \cup \{3\} = \{1, 3\}$

$\therefore \delta(1, ab) = \{1, 3\}$.

(ii) $\delta(1, a) = \{1, 2\}$

$\delta(1, ab) = \delta(\{1,2\}, b)$

$= \delta(1,b) \cup \delta(2, b)$

$= \{1\} \cup \{3\}$

$= \{1, 3\}$

$\delta(1, aba) = \delta(\{1,3\}, a)$

$= \delta(1,a) \cup \delta(3,a)$

$= \{1,2\} \cup \{4\}$

$= \{1,2,4\}$

$\delta(1,abab) = \delta(\{1,2,4\}, b)$

$= \delta(1,b) \cup \delta(2,b) \cup \delta(4,b)$

$= \{1\} \cup \{3\} \cup \{\phi\} = \{1,3\}$

$\therefore \delta(1,abab) = \{1,3\}$

2.5 EQUIVALENCE OF DFA AND NFA

(i) As every DFA is an NFA, the class of languages accepted by NFA's includes the class of languages accepted by DFA's.

(ii) DFA can simulate NFA.

(iii) For every NFA, there exist an equivalent DFA.

Theorem

For every NFA, there exists a DFA which simulates the behavior of NFA. If L is the set accepted by NFA, then there exists a DFA which also accepts L.

Proof

Let M= (Q, Σ, q_0, F, δ) be NFA accepting L

we construct DFA $M^1 = (Q^1, \Sigma, q_0^{\ 1}, F^1, \delta^1)$, where

(i) $Q^1 = 2^Q$ (power set of Q)

(any state in Q^1 is denoted by $[q_1, q_2, ...q_i]$ where $q_1, q_2, ... q_i \in$ Q)

(ii) $q_0^{\ 1} = [q_0]$

(iii) F^1 is set of final states.

Before defining δ^1, let us look at the construction of Q^1, $q_0^{\ 1}$ and F^1.

M is initially at q_0. On application of an input symbol say a, M can reach any of the states in $\delta(q_0, a)$. To describe M, just after application of the input symbol a, we require all the possible states that M can reach after the application of a. So, M^1, has to remember all these possible states at any instant of time.

As M (NFA) starts with inital state q_0. $q_0^{\ 1}$ is defined as $[q_0]$.

In M^1 (DFA) the final state (F^1) can be subset of Q containing all final states of F.

Now we define

$\delta^1 ([q_1, q_2, ... q_i], a) = \delta(q_1, a) \cup \delta(q_2, a) \cup .... \delta (q_i,a)$

equivalently,

$\delta^1([q_1, q_2, ...q_i], a) = [p_1, p_2,....p_j]$

if and only if

$\delta(\{q_1, q_2, ...q_i\}, a) = \{p_1, p_2,....p_j\}$

Proof by Induction

Input string x

$\delta^1(q_0^{\ 1}, x) = [q_1, q_2,....q_i]$

if and only if

$\delta(q_0, x) = \{q_1, q_2, ... q_i\}$

Basis

The result is trivial if string length is 0 i.e., $|x| = 0$

since $q_0^1 = [q_0]$. x must be ε

Induction

Suppose the hypothesis is true for inputs of length m.

Let xa be a string of length m +1 with a in Σ.

Then $\delta^1(q_0^1, xa) = \delta^1(\delta^1(q_0^1,x), a)$

By induction hypothesis

$\delta^1(q_0^1, x) = [p_1,p_2..............p_j]$

if and only if

$\delta(q_0, x) = \{p_1,p_2..............p_j\}$

By definition of δ^1

$\delta^1([p_1,p_2..............p_j], a) = [r_1,r_2..............r_k]$

if and only if

$\delta(\{p_1,p_2..............p_j\}, a) = \{r_1,r_2..............r_k\}$

Thus

$\delta^1(q_0^1, xa) = [r_1,r_2..............r_k]$

if and only if

$\delta(q_0, xa) = \{r_1,r_2..............r_k\}$

which establishes the inductive hypothesis.

Thus $L(M) = L(M^1)$

Example 2.17

Construct the DFA equivalent to the NFA $M=(\{q_0,q_1\}, \{0,1\}, \delta, q_0, \{q_1\})$ and δ is defined as:

States	Inputs	
	0	1
q_0	$\{q_0,q_1\}$	$\{q_1\}$
q_1	ϕ	$\{q_0,q_1\}$

Solution :

DFA $M^1 = (Q^1, \{0,1\}, \delta^1, [q_0], F^1)$ accepting L(M) as follows:

$Q^1 = 2^Q$ (all subsets of $Q=\{q_0,q_1\}$)

$= \{\phi,[q_0], [q_1], [q_0,q_1]\}$

$\delta^1([q_0],1) = [q_1]$, since $\delta(q_0,1) = \{q_1\}$

$\delta^1([q_0],0) = [q_0, q_1]$, since $\delta(q_0,0) = \{q_0,q_1\}$

$\delta^1([q_1],0) = \phi$ since $\delta(q_1,0) = \phi$

$\delta^1([q_1],1) = [q_0, q_1]$, since $\delta^1(q_1,1) = \{q_0,q_1\}$

$\delta^1([q_0,q_1],0) = [q_0, q_1]$, since $\delta (\{q_0,q_1\}, 0)$

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

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

$= \{q_0, q_1\}$

$\delta^1([q_0,q_1],1) = [q_0, q_1]$, since $\delta (\{q_0,q_1\}, 1)$

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

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

$= \{q_0, q_1\}$

DFA transition table is :

States	Inputs	
	0	1
$[q_0]$	$[q_0, q_1]$	$[q_1]$
$[q_1]$	ϕ	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_0,q_1]$

Set of final states of $M^1 = ([q_1], [q_0,q_1])$

Example 2.18

Construct a DFA equivalent to $M = (\{q_0,q_1\}, \{0,1\}\ \delta, q_0, \{q_0\})$ where δ is given as:

States	Inputs	
	0	1
q_0	$\{q_0\}$	$\{q_1\}$
q_1	$\{q_1\}$	$\{q_0, q_1\}$

Solution :

DFA $M^1 = Q^1, \Sigma, \delta', q_0', F')$, where

(i) the subset of the states $\{q_0, q_1\}$ is Q'

i.e. $(\phi, [q_0], [q_1], [q_0,q_1])$

(ii) $[q_0]$ is initial state

(iii) $[q_0]$ and $[q_0, q_1]$ are the final states, since these are the only states containing q_0(final state in NFA).

(iv) δ is defined as:

$\delta^1([q_0],0) = [q_0]$, since $\delta(q_0,0) = \{q_0\}$

$\delta^1([q_0],1) = [q_1]$, since $\delta(q_0,1) = \{q_1\}$

$\delta^1([q_1],0) = [q_1]$, since $\delta(q_0,1) = \{q_1\}$

$\delta^1([q_1],1) = [q_0,q_1]$, since $\delta(q_1,1) = \{q_0, q_1\}$

since $[q_0,q_1]$ is a new state, it has to be defined.

$$\begin{aligned}
\therefore\ \delta^1([q_0,q_1], 0) &= \delta(\{q_0,q_1\}, 0) \\
&= \delta(q_0,0) \cup \delta(q_1,0) \\
&= \{q_0\} \cup \{q_1\} = \{q_0,q_1\} = [q_0,q_1] \\
\delta^1([q_0,q_1], 1) &= \delta(\{q_0,q_1\}), 1) \\
&= \delta(q_0,1) \cup \delta(q_1,1) \\
&= \{q_1\} \cup \{q_0,q_1\} \\
&= \{q_0, q_1\} = [q_0, q_1]
\end{aligned}$$

DFA transition table is :

States	Inputs	
	0	1
$[q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1]$	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_0,q_1]$

Example 2.19

Construct a DFA for the given NFA $M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$ where δ is given as:

States	Inputs	
	0	1
q_0	$\{q_0, q_1\}$	$\{q_2\}$
q_1	$\{q_0\}$	$\{q_1\}$
q_2	ϕ	$\{q_0,q_1\}$

Solution :

We construct $M^1 = (Q^1, \{a,b\}, [q_0], \delta^1, F^1)$

Q^1 contains all subsets of $Q=\{q_0, q_1, q_2\}$

$= (\phi, [q_0], [q_1], [q_2], [q_0, q_1], [q_0, q_2], [q_1, q_2], [q_0, q_1, q_2])$

$\delta^1([q_0],a) = [q_0, q_1],$ since $\delta(q_0,a) = \{q_0, q_1\}$

$\delta^1([q_0],b) = [q_2],$ since $\delta(q_0,b) = \{q_2\}$

$\delta^1([q_2],a) = \phi,$ since $\delta(q_2,a) = \phi$

$\delta^1([q_2],b) = [q_0, q_1]$ since $\delta(q_2,b) = \{q_0, q_1\}$

since we have new state $[q_0, q_1]$, we have to find its transitions:

$$\begin{aligned}
\delta^1([q_0,q_1], a) &= \delta(\{q_0,q_1\}, a) \\
&= \delta(q_0,a) \cup \delta(q_1,a) \\
&= \{q_0,q_1\} \cup \{q_0\} \\
&= \{q_0,q_1\} \\
&= [q_0, q_1] \\
\delta^1([q_0,q_1], b) &= \delta(\{q_0,q_1\}, b) \\
&= \delta(q_0,b) \cup \delta(q_1,b) \\
&= \{q_2\} \cup \{q_1\} \\
&= \{q_1, q_2\} \\
&= [q_1, q_2]
\end{aligned}$$

Since $[q_1, q_2]$ is a new state, its transition has to be defined.

$$\begin{aligned}
\therefore\ \delta^1([q_1,q_2], a) &= \delta(\{q_1,q_2\}, a) \\
&= \delta(q_1,a) \cup \delta(q_2,a) \\
&= \{q_0\} \cup \{\phi\} \\
&= \{q_0\} \\
&= [q_0] \\
\delta^1([q_1,q_2], b) &= \delta(\{q_1,q_2\}, b) \\
&= \delta(q_1,b) \cup \delta(q_2,b) \\
&= \{q_1\} \cup \{q_0,q_1\} \\
&= \{q_0, q_1\} \\
&= [q_0, q_1]
\end{aligned}$$

DFA transition table is :

States	Inputs	
	a	b
$[q_0]$	$[q_0, q_1]$	$[q_2]$
$[q_2]$	ϕ	$[q_0,q_1]$
$[q_0,q_1]$	$[q_0,q_1]$	$[q_1,q_2]$
$[q_1,q_2]$	$[q_0]$	$[q_0,q_1]$

The final state in DFA is

$F^1 = ([q_2], [q_1,q_2])$ since it contains q_2 which is final state in NFA.

Example 2.20

Construct a DFA for the given NFA

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

States	Inputs	
	0	1
q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_1\}$
q_2	$\{q_3\}$	$\{q_3\}$
q_3	ϕ	$\{q_2\}$

Solution :

We construct $M^1 = (Q^1, \{0,1\}, \delta^1, [q_0], F^1)$ an equivalent DFA which accepts L(M)

Q^1 contains all subsets of Q

$= (\phi, [q_0], [q_1], [q_2], [q_3], [q_0, q_1], [q_0, q_2], [q_0, q_3], [q_1, q_2], [q_1, q_3], [q_2, q_3], [q_0, q_1, q_2], [q_1, q_2, q_3], [q_0, q_1, q_3], [q_0, q_2, q_3], [q_0, q_1, q_2, q_3])$

$\delta^1([q_0],0) = [q_0, q_1]$, since $\delta(q_0,0) = \{q_0, q_1\}$

$\delta^1([q_0],1) = [q_0]$, since $\delta(q_0,1) = \{q_0\}$

$$\begin{aligned}
\delta^1([q_0,q_1],0) &= \delta(\{q_0, q_1\}, 0) \\
&= \delta(q_0,0) \cup \delta(q_1,0) \\
&= \{q_0,q_1\} \cup \{q_2\} \\
&= \{q_0,q_1,q_2\}=[q_0,q_1,q_2]
\end{aligned}$$

$$\begin{aligned}
\delta^1([q_0,q_1], 1) &= \delta(\{q_0, q_1\}, 1) \\
&= \delta(q_0,1) \cup \delta(q_1,1) \\
&= \{q_0\} \cup \{q_1\}
\end{aligned}$$

$$
\begin{aligned}
&= \{q_0, q_1\} \\
&= [q_0, q_1]
\end{aligned}
$$

$$
\begin{aligned}
\delta^1([q_0,q_1,q_2], 0) &= \delta(\{q_0, q_1, q_2\}, 0) \\
&= \delta(q_0,0) \cup \delta(q_1,0) \cup \delta(q_2,0) \\
&= \{q_0,q_1\} \cup \{q_2\} \cup \{q_3\} \\
&= \{q_0, q_1, q_2, q_3\} \\
&= [q_0, q_1, q_2, q_3]
\end{aligned}
$$

$$
\begin{aligned}
\delta^1([q_0,q_1,q_2], 1) &= \delta(\{q_0, q_1, q_2\}, 1) \\
&= \delta(q_0,1) \cup \delta(q_1,1) \cup \delta(q_2,1) \\
&= \{q_0\} \cup \{q_1\} \cup \{q_3\} \\
&= \{q_0, q_1, q_3\} \\
&= [q_0, q_1, q_3]
\end{aligned}
$$

$$
\begin{aligned}
\delta^1([q_0,q_1,q_2, q_3], 0) &= \delta(\{q_0, q_1, q_2, q_3\}, 0) \\
&= \delta(q_0,0) \cup \delta(q_1,0) \cup \delta(q_2,0) \cup \delta(q_3,0) \\
&= \{q_0, q_1\} \cup \{q_2\} \cup \{q_3\} \cup \{\phi\} \\
&= \{q_0, q_1, q_2, q_3\} \\
&= [q_0, q_1, q_2, q_3]
\end{aligned}
$$

$$
\begin{aligned}
\delta^1([q_0,q_1,q_2, q_3], 1) &= \delta(\{q_0, q_1, q_2, q_3\}, 1) \\
&= \delta(q_0,1) \cup \delta(q_1,1) \cup \delta(q_2,1) \cup \delta(q_3,1) \\
&= \{q_0\} \cup \{q_1\} \cup \{q_2\} \cup \{q_3\} \\
&= \{q_0, q_1, q_2, q_3\} \\
&= [q_0, q_1, q_2, q_3]
\end{aligned}
$$

$$
\begin{aligned}
\delta^1([q_0,q_1,q_3], 0) &= \delta(\{q_0, q_1, q_3\}, 0) \\
&= \delta(q_0,0) \cup \delta(q_1,0) \cup \delta(q_3,0) \\
&= \{q_0, q_1\} \cup \{q_2\} \cup \{\phi\} \\
&= \{q_0, q_1, q_2\} \\
&= [q_0, q_1, q_2]
\end{aligned}
$$

$$
\begin{aligned}
\delta^1([q_0,q_1,q_3], 1) &= \delta(\{q_0, q_1, q_3\}, 1) \\
&= \delta(q_0,1) \cup \delta(q_1,1) \cup \delta(q_3,1) \\
&= \{q_0\} \cup \{q_1\} \cup \{q_2\} \\
&= \{q_0, q_1, q_2\} \\
&= [q_0, q_1, q_2]
\end{aligned}
$$

DFA transition table is :

States	Inputs	
	a	b
$[q_0]$	$[q_0, q_1]$	$[q_0]$
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0, q_1]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3]$
$[q_0, q_1, q_3]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$
$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$

The final states are

$F^1 = [q_0, q_1, q_3]$ and $[q_0, q_1, q_2, q_3]$

2.6 Finite Automata with ε-moves

The NFA can be extended to include transitions on empty input ε.

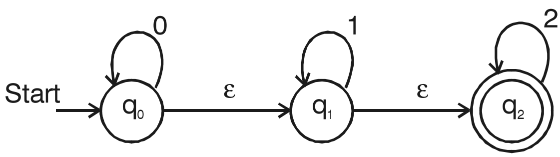
2.6.1 Definition of NFA with ε-moves

The NFA with ε moves is defined by 5 tuple (or) quintuple as similarly as NFA, except δ. $(Q, \Sigma, \delta, q_0, F)$ with all components as before and $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$

The intention is that $\delta(q, a)$ will consists of all states p such that there is a transition labeled 'a' from q to p, where a is either ε or any symbol in Σ.

Example 2.21

Finite automata with ε moves



The transition diagram of the NFA accepts the language consisting of any number of 0's followed by any number of 1's followed by any number of 2's.

For example, the string $w = 002$ is accepted by the NFA along the path – $q_0, q_0, q_0, q_1, q_2, q_2$, with arcs labeled 0, 0, ε, ε, 2.

2.6.2 Epsilon Closures

We extend the transition function δ to a function $\hat{\delta} : Q \times \Sigma^* \rightarrow 2^Q$, such that $\hat{\delta}(q,w)$ will be all states p such that we can go from q to p along a path labeled w, also including edges labeled ε. In constructing $\hat{\delta}$ it will be important to compute the set of states reachable from a given state q using ε - transitions only. We use ε- CLOSURE(q) to denote the set of all vertices p such that there is a path from q to p labeled ε.

(i) Let P be set of states.

(ii) Then ε- CLOSURE (P) =

Example 2.22

Construct the transition table and ε-CLOSURE for the Figure given in the previous page.

Solution :

States	Inputs			
	0	1	2	ε
q_0	q_0	ϕ	ϕ	q_1
q_1	ϕ	q_1	ϕ	q_2
q_2	ϕ	ϕ	q_2	ϕ

The ε– CLOSURE (q_0) = {q_0, q_1, q_2} i.e., the path consisting of q_0 alone is a path from q_0 to q_0 with all arcs labeled ε. The path from q_0 to q_1 labeled ε shows that q_1 is in ε- CLOSURE (q_0) and path q_0, q_1, q_2 shows q_2 is in ε - CLOSURE (q_0).

Similary

ε-CLOSURE (q_1) = {q_1, q_2}

ε-CLOSURE (q_2) = {q_2}

2.6.3 Transition function and language of ε-NFA

The transition function is defined as:

(i) $\hat{\delta}$ (q,ε) = ε - CLOSURE (q).

(ii) For w in Σ^*, and a in Σ

$\hat{\delta}(q, wa)$ = ε- CLOSURE (P),

where P={p | for some r in $\hat{\delta}(q, w)$, p in $\delta(r, a)$}

(iii) $\delta(R, a) = \bigcup_{q \text{ in } R} \delta(q, a)$

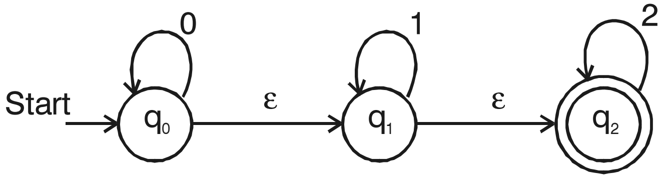
(iv) $(R, w) = \bigcup_{q \text{ in } R} \hat{\delta}(q, w)$

The language accepted by NFA with ε - move is defined as:

L(M) = {w | $\hat{\delta}(q_0, w)$ contains a state in F}

Example 2.23

Consider the NFA given below and find $\delta(q_0, 01)$



Solution :

$\delta(q_0,\varepsilon)$ = ε-CLOSURE $(q_0) = \{q_0, q_1, q_2\}$ → 1

Thus

$\delta(q_0,0)$ = ε-CLOSURE $(\delta(\delta(q_0, \varepsilon),0))$

= ε-CLOSURE $(\delta(\{q_0,q_1,q_2\}, 0))$

= ε-CLOSURE$(\delta(q_0,0)\cup\delta(q_1,0)\cup \delta(q_2,0))$

= ε-CLOSURE $(\{q_0\}\cup \{\phi\} \cup \{\phi\})$

= ε-CLOSURE $(\{q_0\})$

= $\{q_0, q_1, q_2\}$ from equation 1

Then

$\delta(q_0,01)$ = ε-CLOSURE $(\delta(\delta(q_0, 0),1))$

= ε-CLOSURE $(\delta(\{q_0,q_1,q_2\}, 1))$

= ε-CLOSURE$(\delta(q_0,1)\cup\delta(q_1,1)\cup \delta(q_2,1))$

= ε-CLOSURE $(\phi \cup \{q_1\} \cup \{\phi\})$

= ε-CLOSURE $(\{q_1\})$

= $\{q_1, q_2\}$ (because from q_1, on ε- transition we can reach q_2 and also remain in q_1 itself)

Since $\delta(q_0, 01) = \{q_1, q_2\}$ which contain final state $\{q_2\}$ it is an accepted string.

2.7 EQUIVALENCE OF NFA'S WITH AND WITHOUT ε - MOVES

Theorem

If L is accepted by NFA with ε-transitions, than L is accepted by an NFA without ε-transitions.

Proof

Let M = $(Q, \Sigma, \delta, q_0, F)$ be an NFA with ε - transitions. Construct M^1 which is NFA without ε - transition.

$M^1 = (Q, \Sigma, \delta^1, q_0, F^1)$ where

F^1 = $F \cup \{q_0\}$ if ε-CLOSURE (q_0) contains a state of F.

F otherwise

By induction : $\begin{cases} \delta^1 \text{ and } \quad \text{are same} \\ \delta \text{ and } \hat{\delta} \text{ are different} \end{cases}$

Let x be any string

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

This statement is not true if $x = \varepsilon$ because $\delta^1(q_0, \varepsilon) = \{q_0\}$ and

$\hat{\delta}(q_0, \varepsilon) = \varepsilon\text{- CLOSURE }(q_0)$

Basis step

$|x| = 1$ x is a symbol whose value is a

$\delta^1(q_0, a) = \hat{\delta}(q_0, a)$ (because by definition of $\hat{\delta}$)

Induction step

let $x = wa$ where a is in Σ.

$$\delta^1(q_0, wa) = \delta^1(\delta^1(q_0, w), a)$$

$$= \delta^1(\hat{\delta}(q_0, w), a)$$

$= \delta^1(p, a)$ [because by inductive hypothesis

$\delta(q_0, w) = \hat{\delta}(q_0, w) = p(\text{say})$]

Now we must show that

$$\delta^1(p, a) = \hat{\delta}(q_0, wa)$$

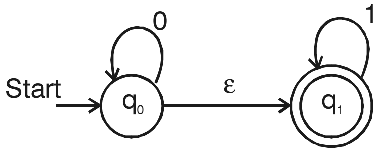
But

$$\begin{aligned} \delta^1(p, a) &= \bigcup_{q \text{ in } P} \delta^1(q, a) = \bigcup_{q \text{ in } P} \hat{\delta}(q, a) \\ &= \hat{\delta}(\hat{\delta}(q_0, w), a) \\ &= \hat{\delta}(q_0, wa) \\ &= \hat{\delta}(q_0, x) \end{aligned}$$

Hence $\delta^1(q_0, x) = \hat{\delta}(q_0, x)$

Example 2.24

Construct a NFA without ε-moves from NFA with ε - moves.



Solution :

ε - CLOSURE $(q_0) = \{q_0, q_1\}$

Let M $= (Q, \Sigma, q_0, \delta, F)$ be NFA with ε-transition

$M^1 = (Q, \Sigma, q_0, \delta^1, F^1)$ be NFA without ε-transition

$F^1 = \{q_0, q_1\}$

$\hat{\delta}(q_0,\varepsilon) = \varepsilon\text{-CLOSURE}(q_0) = \{q_0, q_1\}$............ 1

$\delta^1(q_0,0) = \hat{\delta}(q_0,0)$

$= \varepsilon\text{-CLOSURE}(\delta(\hat{\delta}(q_0, \varepsilon),0))$

$= \varepsilon\text{-CLOSURE}(\delta(\{q_0,q_1\}, 0))$

$= \varepsilon\text{-CLOSURE}(\delta(q_0,0)\cup\delta(q_1,0))$

$= \varepsilon\text{-CLOSURE}(\{q_0\}\cup\{\phi\})$

$= \varepsilon\text{-CLOSURE}(q_0)$

$= \{q_0, q_1\}$ (because of 1)

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

$= \varepsilon\text{-CLOSURE}(\delta(\hat{\delta}(q_0, \varepsilon),1))$

$= \varepsilon\text{-CLOSURE}(\delta(\{q_0,q_1\}, 1))$

$= \varepsilon\text{-CLOSURE}(\delta(q_0,1)\cup\delta(q_1,1))$

$\hat{\delta}$

$= \varepsilon\text{-CLOSURE}(\{\phi\} \cup \{q_1\})$

$= \varepsilon\text{-CLOSURE}(q_1)$

$= \{q_1\}$ ($\because$ from q_1 on ε it can go to q_1 only)

$\delta^1(q_1,0) = (q_1,0)$

$= \varepsilon\text{-CLOSURE}(\delta(\hat{\delta}(q_1, \varepsilon),0))$

$= \varepsilon\text{-CLOSURE}(\delta(\{q_1\}, 0))$

$= \varepsilon\text{-CLOSURE}(\delta(q_1,0))$

$= \varepsilon\text{-CLOSURE}(\phi)$

$= \phi$

$\delta^1(q_1,1) = \hat{\delta}(q_1,1)$

$= \varepsilon\text{-CLOSURE}(\delta(\hat{\delta}(q_1, \varepsilon),1))$

$= \varepsilon\text{-CLOSURE}(\delta(\{q_1\}, 1))$

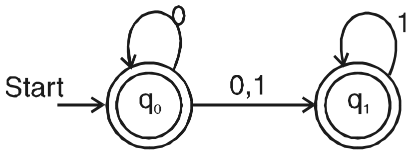
$= \varepsilon\text{-CLOSURE}(\delta(q_1,1))$

$= \varepsilon\text{-CLOSURE}(q_1)$

$= \{q_1\}$

Transition diagram and table is as follow:

States	Inputs	
	0	1
q_0	$\{q_0, q_1\}$	$\{q_1\}$
q_1	ϕ	$\{q_1\}$



2.8 SOLVED PROBLEMS

1. Construct a DFA equivalent to the given NFA M = $(\{p,q,r,s\}, \{0,1\}, \delta, p, \{s\})$ where δ is given as :

States	Inputs	
	0	1
p	$\{p, q\}$	$\{p\}$
q	$\{r\}$	$\{r\}$
r	$\{s\}$	$\{-\}$
s	$\{s\}$	$\{s\}$

Solution :

We construct DFA $M^1 = (Q^1, \Sigma, \delta^1, p, F^1)$, where $Q^1 = 2^Q$

$$\delta^1([p], 0) = [p,q], \text{ since } \delta(\{p\}, 0) = \{p,q\}$$

$$\delta^1([p], 1) = [p], \text{ since } \delta(\{p\}, 1) = \{p\}$$

Since $[p,q]$ is new state, we have to find its transitions on '0' and '1'

$$\delta^1([p,q], 0) = \delta(\{p,q\}, 0) = \delta(p,0) \cup \delta(q,0)$$

$$= \{p,q\} \cup \{r\} = \{p,q,r\} = [p,q,r]$$

$$\delta^1([p,q],1) = \delta(\{p,q\}, 1) = \delta(p,1) \cup \delta(q,1)$$

$$= \{p\}\cup\{r\} = \{p,r\} = [p,r]$$

Since $[p,q,r]$ & $[p,r]$ are new states, we have to find its transitions.

$$\delta^1([p,q,r],0) = \delta(\{p,q,r\},0) = \delta(p,0) \cup \delta(q,0) \cup \delta(r,0)$$

$$= \{p,q\}\cup\{r\}\cup\{s\}=\{p,q,r,s\}$$

$$
\begin{aligned}
&= [p,q,r,s] \\
\delta^1([p,q,r],1) &= \delta(\{p,q,r\},1) = \delta(p,1) \cup \delta(q,1) \cup \delta(r,1) \\
&= \{p\}\cup\{r\}\cup\{\phi\} \\
&= [p,r] \\
\delta^1([p,r],0) &= \delta(\{p,r\},0) = \delta(p,0) \cup \delta(r,0) = \{p,q\} \cup \{s\} \\
&= \{p,q,s\} = [p,q,s] \\
\delta^1([p,r],1) &= \delta(\{p,r\},1) = \delta(p,1) \cup \delta(r,1) = \{p\} \cup \{\phi\} \\
&= \{p,\phi\} = [p] \\
\delta^1([p,q,r,s],0) &= \delta(\{p,q,r,s\},0) \\
&= \delta(p,0) \cup \delta(q,0) \cup \delta(r,0) \cup \delta(s,0) \\
&= \{p,q\} \cup \{r\} \cup \{s\} \cup \{s\} = \{p,q,r,s\} \\
&= [p,q,r,s] \\
\delta^1([p,q,r,s],1) &= \delta(\{p,q,r,s\},1) \\
&= \delta(p,1) \cup \delta(q,1) \cup \delta(r,1) \cup \delta(s,1) \\
&= \{p\} \cup \{r\} \cup \{\phi\} \cup \{s\} = \{p,r,s\} \\
&= [p,r,s] \\
\delta^1([p,q,s],0) &= \delta(\{p,q,s\},0) = \delta(p,0) \cup \delta(q,0) \cup \delta(s,0) \\
&= \{p,q\} \cup \{r\} \cup \{s\} = \{p,q,r,s\} \\
&= [p,q,r,s] \\
\delta^1([p,q,s],1) &= \delta(\{p,q,s\},1) = \delta(p,1)\cup\delta(q,1)\cup\delta(s,1) \\
&= \{p\}\cup\{r\}\cup\{s\} = [p,r,s,] \\
\delta^1([p,s],0] &= \delta(\{p,s\},0) = \delta(p,0) \cup\delta(s,0) = \{p,q\}\cup\{s\} \\
&= \{p,q,s\} = [p,q,s] \\
\delta^1([p,s],1) &= \delta(\{p,s\},1) = \delta(p,1)\cup\delta(s,1) = \{p\}\cup\{s\} = \{p,s\} \\
&= [p,s] \\
\delta^1([p,r,s],0) &= \delta(\{p,r,s\},0) = \delta(p,0) \cup\delta(r,0) \cup\delta(s,0) \\
&= \{p,q\}\cup\{s\}\cup\{s\} \\
&= \{p,q,s\} = [p,q,s]
\end{aligned}
$$

$\delta^1([p,r,s],1) = \delta(\{p,r,s\},1) = \delta(p,1) \cup \delta(r,1) \cup \delta(s,1)$

$= \{p\} \cup \{\phi\} \cup \{s\}$

$= \{p,s\} = [p,s]$

States	Inputs	
	0	1
$[p]$	$[p,q]$	$[p]$
$[p,q]$	$[p,q,r]$	$[p,r]$
$[p,r]$	$[p,q,s]$	$[p]$
$[p,s]$	$[p,q,s]$	$[p,s]$
$[p,q,r]$	$[p,q,r,s]$	$[p,r]$
$[p,q,s]$	$[p,q,r,s]$	$[p,r,s]$
$[p,r,s]$	$[p,q,s]$	$[p,s]$
$[p,q,r,s]$	$[p,q,r,s]$	$[p,r,s]$

The final states are $[p,s]$, $[p,r,s]$, $[p,q,s]$ and $[p,q,r,s]$, since it contains $\{s\}$.

2. Construct a DFA for given NFA M = $(\{p,q,r,s\}, \{0,1\}, \delta, p, \{q,s\})$, where δ is

States	Inputs	
	0	1
p	$\{q,s\}$	$\{q\}$
q	$\{r\}$	$\{q,r\}$
r	$\{s\}$	$\{p\}$
s	–	$\{p\}$

Solution :

The equivalent DFA is as follows :

$Q^1 = 2^Q$

$= ([p], [q,s,], [q], [r], [q,r], [s], [r,s], [p,q,r]\ [q,r,s])$

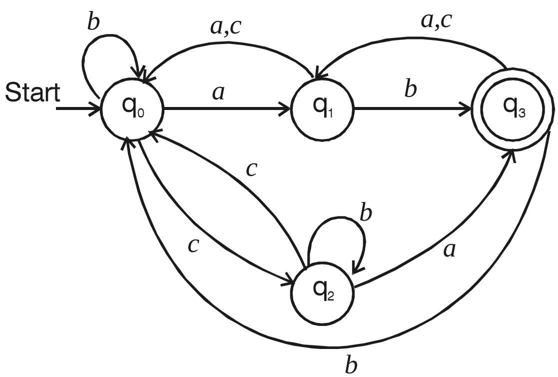
$\Sigma = \{0, 1\}$

p = Initial state

F^1 = Final states = $([q,s], [q,r,s]\ [q]\ [q,r]\ [s]\ [r,s]\ [q,q,r])$

States	Inputs	
	0	1
$[p]$	$[q,s]$	$[q]$
$[q]$	$[r]$	$[q,r]$
$[r]$	$[s]$	$[p]$
$[s]$	ϕ	$[p]$
$[q,s]$	$[r]$	$[p,q,r]$
$[q,r]$	$[r,s]$	$[p,q,r]$
$[r,s]$	$[s]$	$[p]$
$[p,q,r]$	$[q,r,s]$	$[p,q,r]$
$[q,r,s]$	$[r,s]$	$[p,q,r]$

3. For the given finite state automaton M



(a) Find its states

(b) Find its input symbols

(c) Find the initial state

(d) Find whether the following strings are accepted or not.

(i) *acabab* (ii) *bcbcca* (iii) *abba*

(e) Construct its equivalent transition table.

Solution :

(a) States of M = $\{q_0, q_1, q_2, q_3\}$

(b) Input symbols of M = $\{a, b, c\}$

(c) Initial state is : q_0

(d) (i) Transition of the string *acabab:*

$q_0 \xrightarrow{a} q_1 \xrightarrow{c} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_3 \xrightarrow{a} q_1 \xrightarrow{b} q_3$

- q_3 is a final state, hence it is *accepted*.

(ii) Transition of the string *bcbcca*:

$q_0 \xrightarrow{b} q_0 \xrightarrow{c} q_2 \xrightarrow{b} q_2 \xrightarrow{c} q_0 \xrightarrow{c} q_2 \xrightarrow{a} q_3$

- q_3 is a final state, hence it is *accepted*.

(iii) Transition of the string *abba*:

$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_3 \xrightarrow{b} q_0 \xrightarrow{a} q_1$

- q_1 is not a final state, hence it is *not accepted*.

(e) *Transition table :*

States	Inputs		
	a	b	c
q_0	q_1	q_0	q_2
q_1	q_0	q_3	q_0
q_2	q_3	q_2	q_0
q_3	q_1	q_0	q_1

4. Let $\Sigma=\{a,b\}$. Draw the transition diagram of a finite state automaton M that accepts the given set of strings having odd number of b's.

Solution :

Let M be a required DFA which contains two distinct states q_0,q_1. The required set of strings are *b, bbb, bbbbb,.......*

(i.e) $q_0 \rightarrow q_1 = b$

$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 = bbb$

$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 = bbbbb$

5. From the given automaton M, check whether the strings *ababba, baab* are accepted or not.

Solution :

(i) For the string *aba bba*

$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{b} q_2 \xrightarrow{a} q_2$

where q_2 is non-accepting state. Hence the string *ababba* is *not accepted* by M.

(ii) For the string *baab*

$q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1$

where q_1 is a final state, hence the string *baab* is *accepted* by M.

6. Construct a NFA transition diagram and its equivalent DFA to

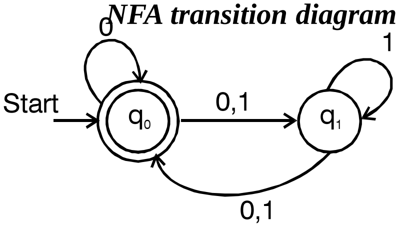
M = (Q, Σ, δ, q_0, F) where Q = $\{q_0, q_1\}$, $\Sigma = \{0,1\}$ F=$\{q_0\}$

and δ is given as:

States	Inputs	
	0	1
q_0	$\{q_0, q_1\}$	$\{q_1\}$
q_1	$\{q_0\}$	$\{q_0, q_1\}$

Solution :

NFA transition diagram :



DFA transition table:

We construct a DFA $M^1=(Q^1, \Sigma, \delta^1, q_0^{\,1}, F^1)$ from the given M, where

$Q^1 = \{[q_0], [q_1], [q_0,q_1]\}$

$q_0^{\,1} = \{[q_0]\}$ and δ^1 is given as:

States	Inputs	
	0	1
$[q_0]$	$[q_0,q_1]$	$[q_1]$
$[q_1]$	$[q_0]$	$[q_0,q_1]$
$[q_0,q_1]$	$[q_0,q_1]$	$[q_0,q_1]$

The set of accepting states $F^1 = \{[q_0], [q_0,q_1]\}$

DFA transition diagram :

7. Construct a NFA transition diagram and its equivalent DFA to: M = $(\{q_0,q_1,q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$

where δ is given as:

States	Inputs	
	a	b
q_0	$\{q_0, q_1\}$	$\{q_2\}$
q_1	$\{q_0\}$	$\{q_1\}$
q_2	–	$\{q_0,q_1\}$

Solution :

NFA transition diagram :

DFA transition table :

We construct a DFA $M^1 = (Q^1, \Sigma, \delta^1, [q_0], F^1)$ from the given M, where δ^1 is constructed as:

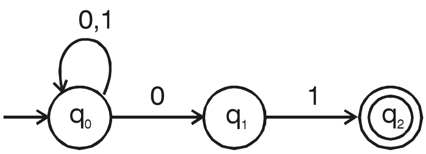
States	Inputs		Assumed New state
	a	b	
$[q_0]$	$[q_0,q_1]$	$[q_2]$	A
$[q_1]$	$[q_0]$	$[q_1]$	B
$[q_2]$	–	$[q_0,q_1]$	C
$[q_0,q_1]$	$[q_0,q_1]$	$[q_1,q_2]$	D
$[q_1,q_2]$	$[q_0]$	$[q_0,q_1]$	E

$Q^1 = \{[q_0], [q_1], [q_2], [q_0,q_1], [q_1, q_2]\}$

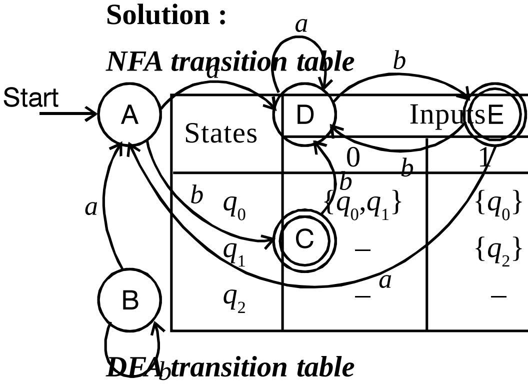
$F^1 = \{[q_2], [q_1,q_2]\}$

DFA transition diagram :

8. Obtain the DFA equivalent to the following NFA **(Nov/Dec 2003)**



Solution :



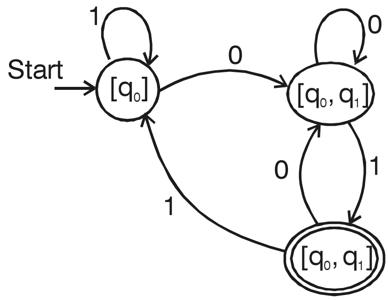
NFA transition table

States	Inputs	
	0	1
q_0	$\{q_0,q_1\}$	$\{q_0\}$
q_1	–	$\{q_2\}$
q_2	–	–

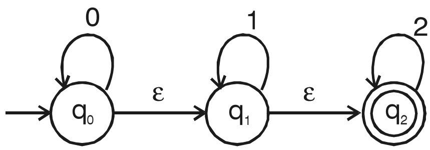
DFA transition table

States	Inputs	
	0	1
$[q_0]$	$[q_0,q_1]$	$[q_0]$
$[q_0,q_1]$	$[q_0,q_1]$	$[q_0,q_2]$
$[q_0,q_2]$	$[q_0,q_1]$	$[q_0]$

DFA transition diagram :



9. Obtain an NFA without ε-transition to the following NFA with ε transition **(Nov/Dec 2003)**



Solution :

NFA with ε transition table (M):

States	Inputs			
	0	1	2	ε
q_0	q_0	–	–	q_1
q_1	–	q_1	–	q_2
q_2	–	–	q_2	–

NFA without ε transition table (M^1) :

States	Inputs		
	0	1	2
q_0	$\{q_0,q_1,q_2\}$	$\{q_1,q_2\}$	$\{q_2\}$
q_1	–	$\{q_1,q_2\}$	$\{q_2\}$
q_2	–	–	$\{q_2\}$

The set of final states of $M^1 = F \cup \{q_0\} = \{q_0,q_2\}$ since ε-CLOSURE$\{q_0\}=\{q_0,q_1,q_2\}$

Transition diagram :