FINITE AUTOMATA

Prof.PRIYANGA K.K, Computer Science Department

AUTOMATON

An automaton is defined as a system where energy, materials and information are transformed. transmitted and used for performing some functions without direct participation of man. Examples are automatic machine tools, automatic packing machines, and automatic photo printing machines.

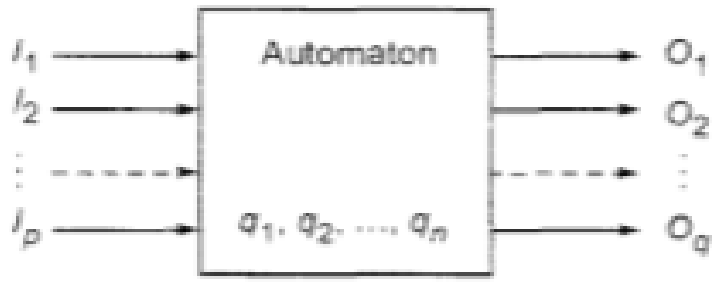


Fig. 3.1 Model of a discrete automaton.

- **<u>Input</u>: At each of the discrete instants of time t1, t2, ... ,tm the input values I1,I2..... ,Ip,each of which can take a finite number of fixed values from the input alphabet ∑, are applied to the input side of the model shown in figure above.**
- **<u>Output</u>: 01,O2, ..., Oq are the outputs of the model, each of which can take a finite number of fixed values from an output O.**
- **<u>States:</u> At any instant of time the automaton can be in one of the states q1, q2,q3, ..., qn.**
- **<u>State relation:</u> The next state of an automaton at any instant of time is determined by the present state and the present input.**
- **<u>Output relation:</u> The output is related to either state only or to both the input and the state. It should be noted that at any instant of time the automaton is in some state. On 'reading' an input symbol, the automaton moves to a next state which is given by the state relation.**

NOTE

- **An automaton in which the output depends only on the input is called an automaton without a memory.**
- **An automaton in which the output depends on the states as well is called automaton with a finite memory**

FINITE AUTOMATON(AUTOMATA)/DFA(DETERMINISTIC FINITE AUTOMATA

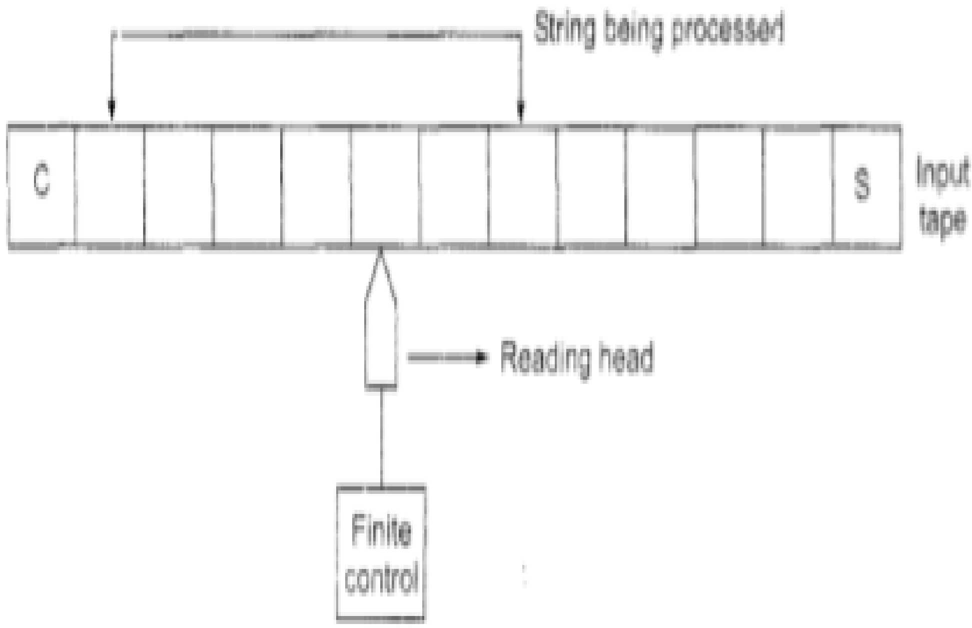
A finite automaton can be represented by a 5-tuple (Q, Σ,δ,qo,F). where

- **Q is a finite nonempty set of states.**
- **Σ is a finite nonempty set of inputs called the input alphabet.**
- **δ is a function which maps Q x Σ into Q and is usually called the direct transition function. This is the function which describes the change of states during the transition. This mapping is usually represented by a transition table or a transition diagram.**
- **qo ϵ Q is the initial state.**
- **F C Q is the set of final states. It is assumed here that there may be more than one final state.**

NOTE

- **The transition function which maps Q x Σ * into Q (i.e. maps a state and a string of input symbols including the empty string into a state) is called the indirect transition function.**
- **δ is also called the next state function**

SHEMATIC REPRESENTATION OF FINITE AUTOMATON



The various components are explained as follows:

(i) Input tape. The input tape is divided into squares, each square containing a single symbol from the input alphabet ∑. The end squares of the tape contain the endmarker ¢ at the left end and the endmarker $ at the right end. The absence of endmarkers indicates at the tape is of infinite length. The left-to-right sequence of symbols between the two endmarkers is the input string to be processed.

(ii) Reading head. The head examines only one square at a time and can move one square either to the left or to the right. we restrict the movement of the R-head only to the right side.

(iii) Finite control. The input to the finite control will usually be the symbol under the R-head, say a, and the present state of the machine, say q, to give the following outputs: (a) A motion of R-head along the tape to the next square (in some a null move, i.e. the R-head remaining to the same square is permitted); (b) the next state of the finite state machine given by δ(q, a).

TRANSITION SYSTEMS

A transition graph or a transition system is a finite directed labelled graph in which each vertex (or node) represents a state and the directed edges indicate the transition of a state and the edges are labelled with input/output.

NONDETERMINISTIC AUTOMATA

A nondeterministic finite automaton (NDFA) is a 5-tuple (Q, ∑, c, qo, F), where

(i) **Q is a finite nonempty set of states;**

(ii) **∑ is a finite nonempty set of inputs;**

(iii) **δ is the transition function mapping from Q x ∑ into 2^Q which is the power set of Q, the set of all subsets of Q;**

(iv) **qo ϵ Q is the initial state; and**

(v) **F C Q is the set of final states.**

NOTE

Any NDFA is a more general machine without being more powerful

MEALY MACHINE

A Mealy machine is a six-tuple (Q,Σ, Δ, δ, λ ,qo)where

- **Q is a finite set of states**
- **Σ is the input alphabet**
- **Δ is the output alphabet**
- **δ is the transition function Σ x Q into Δ**
- **λ is the output function mapping Σ xQ into Δ**
- **and qo is the initial state.**

MOORE MACHINE

A Moore machine is a six-tuple (Q,Σ, Δ, δ, λ ,qo)where

- **Q is a finite set of states**
- **Σ is the input alphabet**
- **Δ is the output alphabet**

- **δ is the transition function Σ x Q into Q**
- **λ is the output function mapping Q into Δ**
- **and qo is the initial state.**

NOTE

- **An automaton in which the output depends only on the states of the machine is called a Moore machine.**
- **An automaton in which the output depends on the state as well as on the input at any instant of time is called a Mealy machine**

REGULAR EXPRESSION

A formal recursive definition of regular expressions over Σ as follows:

1. Any terminal symbol (i.e. an element of Σ), $\Lambda$and Ø are regular expressions. When we view a in Σ as a regular expression, we denote it by a.

2. The union of two regular expressions R_1 and R_2 written as $R_1 + R_2$, is also a regular expression.

3. The concatenation of two regular expressions R_1 and R_2, written as R_1 R_2, is also a regular expression.

4. The iteration (or closure) of a regular expression R written as R*, is also a regular expression.

5. If R is a regular expression, then (R) is also a regular expression.

6. The regular expressions over Σ are precisely those obtained recursively by the application of the rules 1-5 once or several times.

REGULAR SET

Any set represented by a regular expression is called a regular set.

IDENTITIES FOR REGULAR EXPRESSIONS

I_1 : Ø+ R =R

I_2 : Ø R = R Ø = Ø

1_3 : ΛR =RΛ =R

1_4: Λ* = Λ and Ø * = Λ

I_5: R+R=R

Prof.PRIYANGA K.K, Computer Science Department

1_6 : R*R* =R*

1_7: RR* = R*R

I_8 : (R*)* =R*

1_9: A + RR* = R* = A + R*R

1_{10} :(PQ)*P = P(QP)*

I_{II} : (P + Q)* = (P*Q*)* = (p* + Q*)*

1_{12}: (P + Q)R =PR + QR and R(P + Q) =RP + RQ

ARDEN'S THEOROM

Let P and Q be two regular expressions over ∑. If P does not contain Λ, then the following equation in R, namely

R = Q + RP

has a unique solution (i.e. one and only one solution) given by

R = QP*.