Definition: An automaton is a system where materials, energy, or information are transformed and transmitted for performing some operation without the direct participation of a human.

Any automated machine can be given as an example of automaton. A model of finite automata is given in Fig. 3.1.

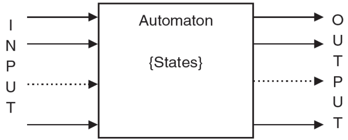


Fig. 3.1 *Block Diagram of Finite Automata*

3.3.1 Characteristics

- **Input (I/P):** The input is taken in each clock pulse. For every single instance of time $t_1, t_2, t_3 \ldots . t_n$, the inputs are taken as I_1, I_2, I_3 .... In. As there are a number of input lines, n number of inputs will be taken in each single time instance. The input for each input line is finite and taken from a set called the set of input alphabets Σ.

- **Output (O/P):** The output is generated in each clock pulse. For every single instance of time $t_1, t_2, t_3 \dots t_m$, the outputs are generated as $O_1, O_2, O_3 \dots O_m$. The output generated from each output line is finite and belongs to a set called the output alphabet set.
- **State:** At any discrete instance of time, the automaton can be in one of the states $q_1, q_2, q_3 \dots q_n$. The state belongs to a set called 'State' Q.
- **State transition:** At any instance of time, the automaton must be in one of the states that belong to the set Q. By getting an input in a clock pulse, the automaton must reside in a state. The state in which the automaton resides by getting that particular input is determined by state transition. The state transition is a function of the present state and the present input, which produces the next state. The function is represented as δ.
- **Output relation:** Similar to the state transition for state, there is a relation for output. The output depends either on the present state and present input or on the present state only depending on the type of machine.

3.4 Finite Automata

Definition: Finite automata (singular: automaton) are the machine formats of regular expression, which is the language format of type 3 grammar. An FA is defined as

$$M = \{Q, \Sigma, \delta, q_0, F\}$$

where
Q : finite non-empty set of states
Σ : finite non-empty set of input symbols
δ : transitional function
q_0 : beginning state
F : finite non-empty set of final states

Finite automata are one type of the finite state machine. It has a finite number of states. Finite automata can be thought of as a finite state machine without output.

Mechanically, finite automata can be described as an input tape containing the input symbols (Σ) with a reading head scanning the inputs from left to right. The inputs are fed to finite control which contains the transitional functions (δ). According to the transitional functions, the state (Q) change occurs.

The mechanical diagram of finite automata is given in Fig. 3.2.

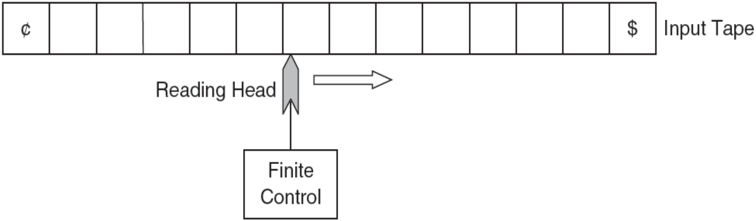


Fig. 3.2 *Mechanical Diagram of Finite Automata*

- **Input tape:** The input tape contains the input symbol. It is divided into several squares, which contain single characters of the input alphabet. Both the left and right ends of the input tape contain end markers. Between two end markers, the input string is placed. This string is needed to be processed from left to right.

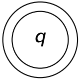
- **Reading head:** The head scans each square in the input tape and reads the input from the tape. The head can move from left to right or right to left. But, in most of the cases, the head moves from left to right. In two-way finite automata and the Turing machine, the head can move in both directions.
- **Finite control:** Finite control can be considered as the control unit of an FA. An automaton always resides in a state. The reading head scans the input from the input tape and sends it to finite control. In this finite control, it is decided that 'the machine is in this state and it is getting this input, so it will go to this state'. The state transition relations are written in this finite control.

3.5 Graphical and Tabular Representation of FA

Finite automata can be represented in two ways: (i) graphical and (ii) tabular.

1. In the graphical representation, a state is represented as q

   A beginning state is represented as $\rightarrow q$

A final state is represented as 

An FA is represented in graphical format in Fig. 3.3.

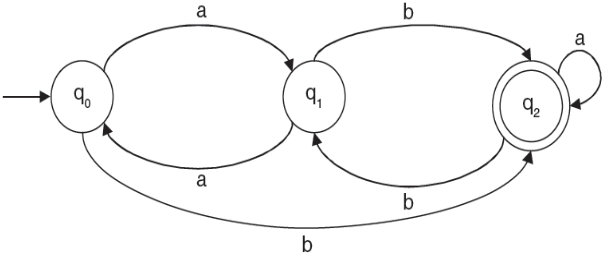


Fig. 3.3 *Graphical Representation of Finite Automata*

Here

$$
\begin{aligned}
& Q : \{q_0, q_1, q_2\} \\
& \Sigma : \{a, b\} \\
& \delta : \delta(q_0, a) \rightarrow q1 \\
& \quad \delta(q_0, b) \rightarrow q_2 \\
& \quad \delta(q_1, a) \rightarrow q_0 \\
& \quad \delta(q_1, b) \rightarrow q_2 \\
& \quad \delta(q_2, a) \rightarrow q_2 \\
& \quad \delta(q_2, b) \rightarrow q_1 \\
& q_0 : \{q_0\} \\
& F : \{q_2\}
\end{aligned}
$$

2. In the tabular format, a state is represented by the name of the state.
The beginning state is represented as $\rightarrow q_n$

The final state is represented as q_n

Present State	Next State a	Next State B
$\rightarrow q_0$	q_1	q_2
q_1	q_0	q_2
q_2	q_2	q_1

Here

$$
\begin{aligned}
&Q : \{q_0, q_1, q_2\}\\
&\Sigma : \{a, b\}\\
&\delta : \delta(q_0, a) \rightarrow q1\\
&\quad\quad \delta(q_0, b) \rightarrow q_2\\
&\quad\quad \delta(q_1, a) \rightarrow q_0\\
&\quad\quad \delta(q_1, b) \rightarrow q_2\\
&\quad\quad \delta(q_2, a) \rightarrow q_2\\
&\quad\quad \delta(q_2, b) \rightarrow q_1\\
&q_0 : \{q_0\}\\
&F : \{q_2\}
\end{aligned}
$$

3.6 Transitional System

A transitional system (sometimes called the transitional graph) is a finite-directed graph in which each node (or vertex) represents a state, and the directed arc indicates the transition of the state. The label of the arc indicates the input or output or both.

A transitional function has two properties.

1. **Property I:** $\delta(q, \Lambda) \rightarrow q$. It means if the input is given null for a state, the machine remains in the same state.

2 **Property II:** For all string X and input symbol $a \varepsilon \Sigma$,

$$\delta(q, Xa) \to \delta(\delta(q, X), a)$$
$$\delta(q, aX) \to \delta(\delta(q, a), X)$$

3.6.1 Acceptance of a String by Finite Automata

There are two conditions for declaring a string to be accepted by a finite automaton. The conditions are

Condition I: The string must be totally traversed.

Condition II: The machine must come to a final state.

In short, it can be said that if $\delta(q_0, W) = q_n$, where W is the string given as input to the FA, q_0 is the beginning state, and q_n belongs to the set of final states, then the string W can be said to be accepted by the FA.

If these two conditions are fulfilled, then we can declare a string to be accepted by an FA.

If any of the conditions are not fulfilled, then we can declare a string to be not accepted by an FA.

The following examples (Examples 3.1 and 3.2) describe this.

Example 3.1 Check whether the string 011001 is accepted or not by the following FA.

State	Input 0	Input 1
$\rightarrow q_0$	q_0	q_1
q_1	q_2	q_3
q_2	q_0	q_3
q_3	q_1	q_3

Solution: For the given FA, $Q = \{q_0, q_1, q_2, q_3\} \Sigma = \{0, 1\}$. The beginning state is q_0 and the final state is $_q$.

The transitional functions are given in the table.

For checking whether a string is accepted by an FA or not, we will assume that the input string is given input in the beginning state q0. But only a single input is given in each clock pulse. So, at first, the left most character is given the input to the beginning state. From the transitional function given in the table, the next state is determined. The next character of the input string is treated as the input to the state just achieved. And it will process like this till the string is fi nished or such a condition has arrived so that there is no transitional function mentioned in the table.

If the string is finished and the state achieved is the final state, then the string will be declared accepted by the machine.

If it does not happen, then the string will be declared not accepted.

$$\begin{aligned}\delta(q0, 011001) &\to \delta(q_0, 11001)\\ &\to \delta(q_1, 1001)\\ &\to \delta(q_3, 001)\\ &\to \delta(q_1, 01)\\ &\to \delta(q_2, 1)\\ &\to q_3\end{aligned}$$