

Definition

- Automata is the plural of automaton, and it means “something works automatically”.
- Automata simulates parts of computers.
- Automata helps in design software's
- Automata play a major role in theory of computation, compiler design, artificial intelligence.
- Automata theory is the study of computational problems that can be solved using abstract computational devices.
- Abstract device is a simplified model of real computation.

Automata theory

Mathematicians and Computer Scientists developed this theoretical computer science branch to simplify the logic of computation by using well defined abstract computational devices (models).

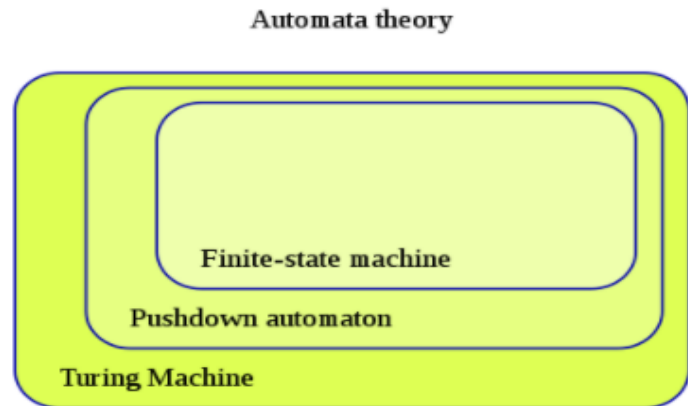
Automata Theory is the study of abstract computational devices. It forms a formal framework for designing and analyzing computing devices. These models are essential in several areas of computation (applied and theoretical fields).

An Automaton is a machine that operates singularly on input and follows a defined pattern or configuration to produce the desired output. Through automata, we learn how problems and compute functions are solved by the use of automata.

Automata theory is the study of abstract machines and automata, as well as the computational problems that can be solved using them. The abstract machine is a simplified model of real computation. The word automata (the plural of automaton) comes from the Greek word means "self-acting, self-moving", in other words, something works automatically. An automaton (Automata in plural) is an abstract self-propelled computing device which follows a predetermined sequence of operations automatically. An automaton with a finite number of states is called a Finite Automaton (FA) or Finite-State Machine (FSM).

Automata theory is related to formal language theory. In this context, automata are used as finite representations of formal languages that may be infinite. Automata are often classified by the class of formal languages they can recognize, as in the Chomsky hierarchy. Automata play a major role in theory of computation, compiler construction and artificial intelligence.

The finite-state machine belongs to a well-known type of automaton. This automaton consists of states (represented by circles) and transitions (represented by arrows). As the automaton sees a symbol of input, it makes a transition (or jump) to another state, according to its transition function, which takes the previous state and current input symbol as its arguments.



FINITE AUTOMATA

Finite Automata(FA) is the simplest machine to recognize patterns. It is used to characterize a Regular Language. Also it is used to analyze and recognize Natural Language Expressions. The finite automata or finite state machine is an abstract machine that has five elements or tuples. It has a set of states and rules for moving from one state to another but it depends upon the applied input symbol. Based on the states and the set of rules the input string can be either accepted or rejected. Basically, it is an abstract model of a digital computer which reads an input string and changes its internal state depending on the current input symbol. Every automaton defines a language i.e. set of strings it accepts. The following figure shows some essential features of general automation.

A finite automaton is a collection of three things:

1. A finite set of states, one of which is designated as the initial state, called the start state, and some (maybe none) of which are designated as final states.
2. An alphabet Σ of possible input letters, from which are formed strings, that are to be read one letter at a time.

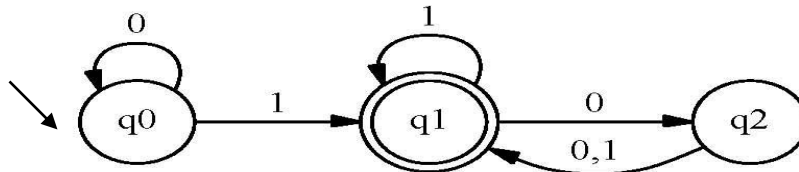
3. A finite set of transitions that tell for each state and for each letter of the input alphabet which state to go to next. The definition above is incomplete in the sense that it describes what a finite automaton is but not how it works. It works by being presented with an input string of letters that it reads letter by letter starting at the leftmost letter. Beginning at the start state the letters determine a sequence of states.

The sequence ends when the last input letter has been read. Instead of writing out the whole phrase "finite automaton" it is customary to refer to one by its initials, FA. The term FA is read by naming its letters, so we say "an FA" even though it stands for "a finite automaton" and we say "two FA's" even though it stands for "two finite automata".

Finite State Automata

How to Read State Diagrams

The following figure depicts a finite state automaton called M_1 :



- M_1 has three STATES, labeled q_0 , q_1 , q_2 .
- The START STATE is q_0 .
- The FINAL STATE (or ACCEPT STATE) is the one with a double circle, q_1 .
- The arrows going from one state to another are called TRANSITIONS.
- When this automaton receives an input string, it processes each symbol in the string from left to right, and produces an output. The output is either ACCEPT or REJECT. The processing begins in the machine's start state, and after reading each symbol, it moves from one state to another along the transition that has the symbol as its label. When the machine reads the last symbol, the output is ACCEPT if the machine is in a final (accept) state, and REJECT if it is not.

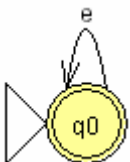
When we feed the input string 1101 to machine M_1 :

1. start in state q_0 ;
2. read 1, follow transition from q_0 to q_1 ;
3. read 1, follow transition from q_1 to q_1 ;
4. read 0, follow transition from q_1 to q_2 ;
5. read 1, follow transition from q_2 to q_1 ;
6. *accept* because M_1 is in an accept state q_1 at the end of the input.

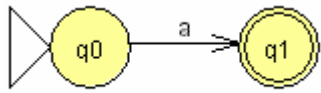
Language accepted by FA

String is accepted by a FA if and only if the FA starting at the initial state and ends in an accepting state after reading the string.

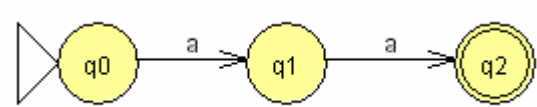
Examples of languages accepted by FA



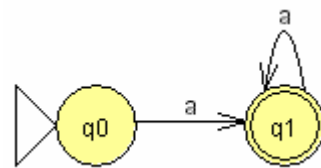
$$RE = \lambda, \epsilon$$



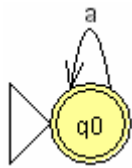
RE = a



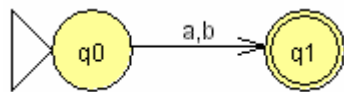
RE = aa



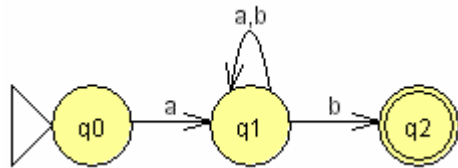
RE=a+ = aa*



RE=a*



RE=a+b



RE=a(a+b)*b