Applications of Automata in Electronic Machines and Android Games (Finite Automata)

E. Anitha
Department of Mathematics, Loyola College,Chennai-600 034, India

Abstract
A series of studies had been undergone by various resource people about, how the Automata have been playing the crucial roles in real world requirement. This paper describes the basic concepts and structure of Automata. We discuss how the electronic machine and Android games are formulated by using Automata Theory. Here we define finite Automata and provide elementary applications of Finite Automata. We discuss with few examples how this process is done by using Finite Automata.

Key words: Finite automata, Definitions, Types,Constructions, Applications.

Introduction

Automata Theory is the study of abstract computing devices or machine [1]. Automata are the prime examples of general computational systems over discrete spaces and have a long history both in theory and application. It is also related to Fuzzy. We introduced the concept of Fuzzy automata in 1995. Fuzzy automata are devices which accept Fuzzy languages and are able to create capabilities which are not easily achievable by other mathematical tools. [2]. In 1930's Alen Turing presented an abstract machine that has all the capabilities of today's computers. In 1940's and 1950's simpler kinds of machine which we today called finite Automata were studied by numbers of Researches. In Computer Science we find many examples of finite state systems and the theory of finite automata is a useful design tool for these systems. Automation is also tempting to view the human brain as a finite state system. The number of brain cell or neutrons are limited probably 2^{35} at most. Although there is evidence to the contrary, that the state of each neuron can be described by a small number of bits, if so then finite state theory applies to the brain [1]. The theory of finite automata is almost half a century old. It started with a seminal paper of kleene and within a few years developed into a rich mathematical research topic. This is much due to several influential papers. From the very beginning research on finite automata was very intense, so that fundamental results and the beauty of the theory was revealed in a short period. However, some challenging problems were solved only much later or are still open. From the beginning finite automata constituted a core of computer science part of the reason is that they capture something very fundamental as is witnessed by a numerous different characterizations of the family of rational languages that is languages defined by finite Automata very important reason is the usefulness of finite automata in many applications of Computer Science. In fact, the interrelation of finite automata and their applications in computer science is a splendid example of a really fruitful connection of theory and practice finite automata played a main role in the theory of programming languages [3]. Finite Automata is a mathematical model of machine. We discuss the mathematical model of the computers and algorithms. Here we are discussing more sophisticated machines for accepting and generating languages, which are restricted model of the actual computers called finite automata or finite state machine. These machines are very similar to the central processing unit of a computer. Absence of the memory makes these Machines more restricted model computer is also deterministic, by which we mean that, on reading one particular input instruction the machine converts itself from the state it was in any particular other state, where the resultant state is completely determined by the priory state and the input instruction. Finite automaton is called 'finite' because the number of possible states and a number of letters in the alphabet are both finite and 'automation'. Because the change of the state is totally governed by the input [1].

Uses of Automata

There are several reasons why the study of automata and complexity is an important and is part of computer science. Let us list some important aspects of the automata theory [1].

1. Automata theory plays an important role when we are making software for designing and checking the behavior of digital circuit [1].
2. The "Lexical analyses" of the typical compiler that is the compiler component that breaks the input text into logical units, such as identifiers Keywords and Punctuation [1].
3. Software for scanning large bodies of text, such as collections of web pages to find occurrences of words, phrases or other patterns [1].
4. Automata theory is key to software for verifying systems of all types that have a finite number of distinct states, such as communication protocols or protocol for secure exchange of information [1].
5. Automata theory is most useful concept of software for Natural language processing [1].
6. Direct application to formulate traffic light, making play toys, It plays a major role in making electronic machine (e.g. Vending machine, Toll Machine,…) and Android games (Pac man, Treasure Hunt, monkey and Banana, ...).

Definition and structure of finite Automata

The finite Automata will consist of the input output relation at every state and also the changes of the states that will occur in receiving the input at a particular state [4].

A finite Automata can be represented by a 5-tuple (Q,Σ,δ,q_0,F)

Where (i) Q is a finite non-empty set of states.

(ii) Σ is a finite non-empty set of inputs called an input alphabet.

(iii) δ is a function which Maps Q x Σ to Q is called transition function.

(iv) $q_0 \in Q$ is the initial state.

(v) $F \subseteq Q$ is the set of final states. It is assumed that there may be more than one final state [7].

L=any string with 'a' and 'b'

L= {ab, aab, aba, ba, bba, aaab…}

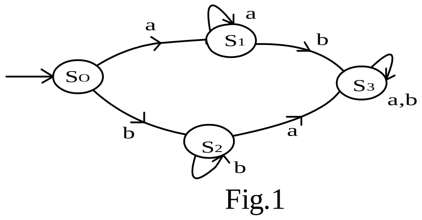

Fig.1

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

Q - $\{S_O,S_1,S_2,S_3\}$

Σ- {a, b}

q_0 – Starting state { S_O}

F - Final state {S_2}

δ - Table

δ	A	B
S_O	S_1	S_2
S_1	S_1	S_3
S_2	S_3	S_2
S_3	S_3	S_3

Table.1

Languages

A set of all strings of which are chosen from Σ. [5], [6].

Strings

A string (or word) is a finite sequence of symbols chosen from some alphabet. [5],[6].

Alphabets

An Alphabet is a finite non-empty set of symbols. [5], [6].

Symbols

A symbol is an abstract entity that cannot be defined formally just like point and line in geometry. The symbols are may be letters and digits. [5],[6].

Example

A primary example is an electric switching circuit, only two situation are possible that is 'on' and 'off'.

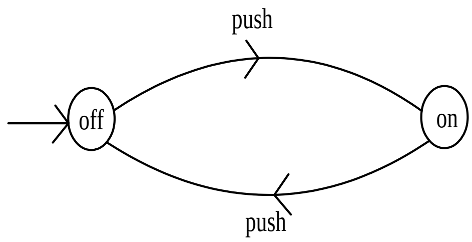

Fig.2

Let consider the symbol '1' and '0'. Let us say '1' for ON and '0' for OFF. We can interchange the situation by pushing the switch [1].

Finite state machines (or) finite automata

A finite state machine (FSM) a mathematical model of a system with certain input the model finally gives an output. The input when given to a machine it is processed by various states, these are called intermediate states. The states which define behavior and may produce actions and transition which are moving from one state to another. The rules or conditions must be met to allow a state transition input events are either externally or internally generated, which may possibly trigger rules and lead to state transitions [7]. There are two categories in FSM namely (i) Finite state machine with output (ii) Finite state machine without output .[4].

(i)Finite State Machine With Output

The primary difference is that there is no set of final states and that the transition function not only puts you in a new state, but also generates an output symbol.

(a)Moore Machine

If the output function depends only on the present state and is independent of the current input, the model is called a Moore machine. [8], [4].

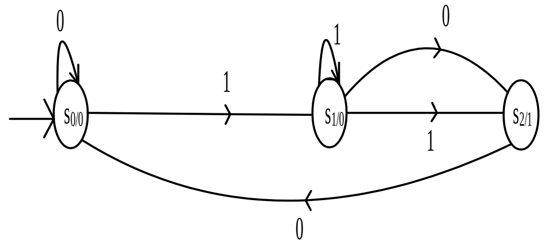

Fig.3

Meaning of state: S_0 - No elements of the sequence observed, S_1 - “1” observed, S_2 - “10” observed.

(b) Mealy Machine

If the output function is a function of transition (a function of present state and the present input the model is called a mealy machine. [8], [4].

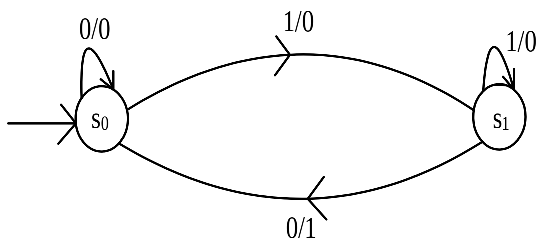

Fig.4

Meaning of state: S_0 - No elements of the sequence observed, S_1 - “1” observed.

Moore and Mealy machines are two methods of implementing a Finite State Machine. These are widely used in integrated circuits for implementing simple functions to complex algorithms. Any function that is inherently state-dependent will require the FSM implementation in hardware. (E.g. Serial Bus Transfer protocols, like USB and power management applications).

Moore machines are generally preferred in RTL (Register-transfer level**).** The Moore machines have been used in the logic circuits. It is applied in clocked synchronize circuits and cooking. The microwave oven is working in the base of Moore machine. Most digital electronic systems are designed as clocked sequential systems. Clocked sequential systems are a restricted form of Moore machine. The application of Moore and Mealy is clocked simultaneous machines where state transitions take place on clock edges and the best example of a mealy machine is a door.

(ii) Finite State Machine Without Output

(a) Deterministic Finite Automata

When the finite control is in particular state, reading an input symbol, it can have only one possible move. [9], [10]

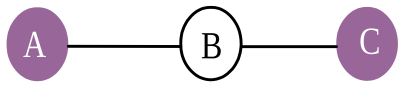

Fig.5

(b) Non- Deterministic Finite Automata

If several moves are possible, then the automaton is non deterministic. In a non-deterministic finite automaton, when the control is in a particular state, reading an input symbol, then several ‘next moves’ are possible.[9],[10].

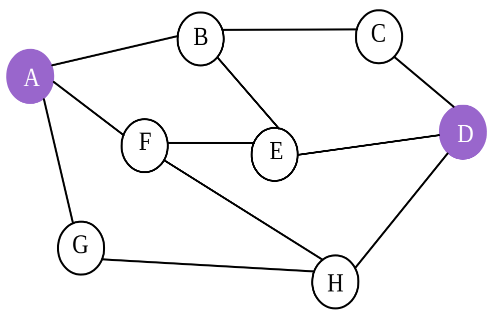

Fig.6

Applications of Finite Automata

Let we see the few constructions for applications of Automata.

(1) Construct a finite state machine to collect the fees from the students in different departments namely Mathematics ,Statistics, Physics. The machine accepts Rs.100, Rs. 500, Rs 1000. The machine accepts money until Rs.1500 has been put in. It gives change back for any amount greater than Rs.1500. Then the students can push buttons to receive the receipts according to their department.

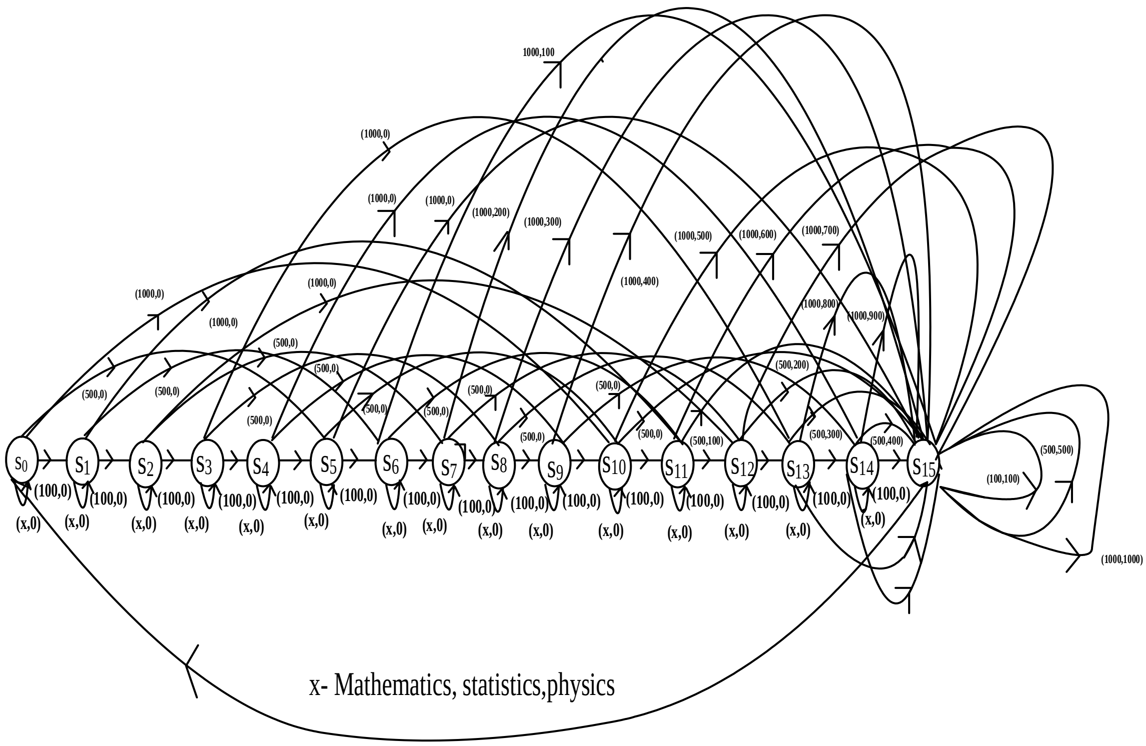

Fig.7

A= {Q, Σ, δ, q_0, F}

Q-{S_O,S_1,S_2,S_3,S_4,S_5,S_6,S_7,S_8,S_9,S_{10},S_{11}, S_{12},S_{13},S_{14},S_{15}}

Σ- {100, 500, 1000}

q_0-Starting state { S_O}

F-Final state {S_{15}}

δ table

	INPUT			OUTPUT		
δ	**100**	**500**	**1000**	**100**	**500**	**1000**
S_O	S_1	S_5	S_{10}	0	0	0
S_1	S_2	S_6	S_{11}	0	0	0
S_2	S_3	S_7	S_{12}	0	0	0
S_3	S_4	S_8	S_{13}	0	0	0
S_4	S_5	S_9	S_{14}	0	0	0
S_5	S_6	S_{10}	S_{15}	0	0	0
S_6	S_7	S_{11}	S_{15}	0	0	100
S_7	S_8	S_{12}	S_{15}	0	0	200
S_8	S_9	S_{13}	S_{15}	0	0	300
S_9	S_{10}	S_{14}	S_{15}	0	0	400
S_{10}	S_{11}	S_{15}	S_{15}	0	0	500
S_{11}	S_{12}	S_{15}	S_{15}	0	100	600
S_{12}	S_{13}	S_{15}	S_{15}	0	200	700
S_{13}	S_{14}	S_{15}	S_{15}	0	300	800
S_{14}	S_{15}	S_{15}	S_{15}	0	400	900
S_{15}	S_{15}	S_{15}	S_{15}	100	500	1000

Table.2

(2) Construct a finite state machine for the log on procedure for a computer, where the user logs in by entering a user identification number, a password and a mobile number which are considered to be a single input. If the password is incorrect, and the mobile number is incorrect, the user is asked for the user identification number again.

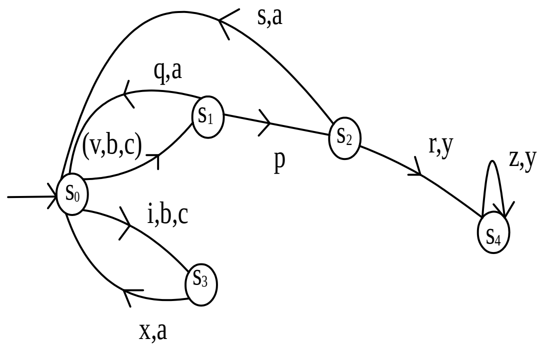

Fig.8

v - Valid id,
i - Invalid id,
p - Valid password,
q - Invalid password,
r - Valid mobile number,
s - Invalid mobile number,
a - “Enter user ID”,
b - “Enter password”,
c - “Enter mobile number”,
y - Prompt,
z - Any input.

(3) Construct a finite automaton to accept by choosing the right answer from the given options to the given questions.

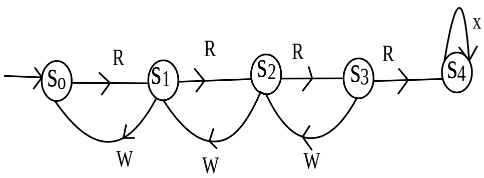

Fig.9

R - Choosing right option
W - Choosing wrong option
X - Final result

Conclusion

The present work provides a deeper understanding of some foundational aspects of usage of automata provide more interesting problem for further study.Finite automata without any external accepting power.As we said earlier finite state machine plays a crucial role.It applied in many fields.By the further research we can product and formulate infinite systems.

References

[1] Rakesh Dube, Adesh Pandey and Ritu Gupta "Discrete Structures and Automata Theory" Ch.12, PP.314-319 Ghaziabad, India.

.

[2]S.Moghari, M.M. Zahedi and R.Ameri, "New Direction in Fuzzy Tree Automata" . Iran http://journals.usb.ac.ir/Fuzzy/en-us/Articles/Article_391/ .

[3] Juhani Kashumaki "Application of Finite Automata" University of Turku, Finland under the Reference. K.Diks et.al.(Eds): MFSC 2002, LNCS 2420, PP.40-58, 2002.

[4] Anand Sharma, "Theory of Automata and Formal Languages". Second edition, Ch-2, PP.17,87-114, Bulandshahr (U.P), India.

[5] Dr. K. Anuradha, Dr. Y. Vijayalata, "Formal Languages and Automata Theory" Ch-1. PP 1-7. Hyderabad (A.P), India.

[6] John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, "Introduction to Automata Theory Languages and Computation". Ch-1, 27-29, 3rd Edition Cornell University, Stanford University, California (U.S)

[7] T. Veerarajan, "Discrete Mathematics with Graph Theory and combinatorics" Ch-8, PP 461-464. India.

[8] Babu Ram , " Discrete Mathematics'' , Ch.9 ,PP.515-518, Maharshi Dayanand University, Rohtak.

[9] Dr. Rani siromoney , " Formal Languages and Automata'' , Ch.5, PP.61-65.India.

[10] Stephen A. Wiitala , " Discrete Mathematics (A Unified Approach) , Ch.7, PP.304-312,Norwich University