Learning automata-a review of theory of automata, formal languages and computation

Meenu

Abstract

This paper is concerned with learning of theory of automata, formal languages and computation. The automata can provide us new aspect of technology which can help us for the automation of every machine. Help to build software for designing and checking the behavior of digital circuits that have finite number of distinct states. The aim of this paper is to discover things involved in automata. The paper will also cover how automata works and changes its state form one to another.

Keywords- automata, finite state machine, DFA, NFA, machines, and languages.

I.INTRODUCTION

Automata theory is branch of computer science which means a machine that performs specific range of functions according to a predetermined or predefined set of coded instruction.

The concept of automata theory

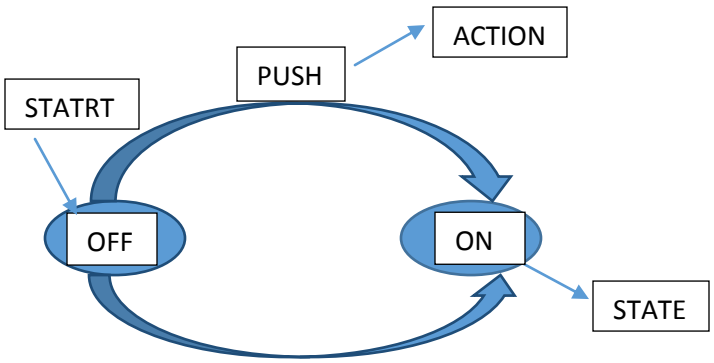
There are some central concepts of automata theory that is language and grammar.

A language is a set of string of symbols from some alphabets which make sentences.

A grammar defines a set of rules. With the help of these rules valid sentences in the language are constructed.

There is a term 'finite state machine' is used because in this machine the number of states and input elements both are finite. Just because of finite states and elements the finite state machine composed with limited amount of memory.

The best example of finite state machine is automatic door. The door open when it sense that some person is coming toward it. Door closewhen it sense that no one is coming. So it has only two states ON and OFF depend upon condition. [1]

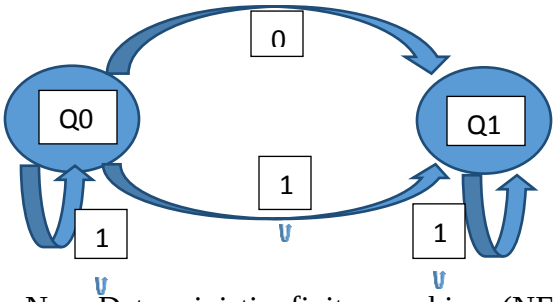


The finite state machine always consists five elements-

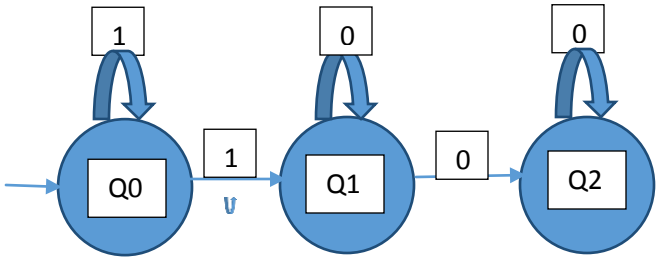
1. Input- automata take some input and produce output.
2. Output- automata take some input and produce output.
3. State of automata- automata have finite number of internal state.
4. State relation- one state at any instant of time.
5. Output relation- the output of automata related to state only or the both the input and the state.

Types of finite state machines-

a) Deterministic finite machine (DFA)- In DFA for each input only and only one move from one state to next state.



b) Non Deterministic finite machine (NFA) – NFA means choice of moves for an automata. In NFA there may be more than one move or no move at all for given input. In this it is possible to move in several states at once.



II. CONVERSION OF NFA TO DFA

The conversion involve three steps-

1. Convert the given transition system into state transition table where each state corresponds to a row and each input symbol corresponds to a column.
2. Construct the successor table that lists subsets of states reachable from the set of initial states.

3. The transition graph given by the successor table is the required deterministic systems. The final states contain some final states of NDFA.[2]

III. REGULAR EXPRESSION AND REGULAR LANGUAGES FOR AUTOMATA

Regular Language- Regular language is a formal language that can be expressed using regular expression.

Regular Expression- regular expression is set of rules. For each rule there is correspondence language. The language accepted by finite automata are easily described by simple expressions called regular expressions. [1] In other words, Algebraic description of language is called regular expression. [3] Regular expression also used symbol and operators like other expressions. Regular expression basically consist three parts-

1. The constant € and ∅ are regular expressions, denoting the language {€} and ∅, that is L(€=) {€} and L(∅)=∅.
2. If *a*is a symbol, then *a* is a regular expression. This regular expression denotes the language {a} that is L(a) = {a}.
3. The variable must be capitalized and italic representing any language.

Operators used in regular expression

There are three permitted operators for regular expressions-

1. Union - If E and F are regular expressions, then E+F is a regular expression which representing the union of L(E) andL(F)That is L(E+F)= L(E)U L(F)
2. Concatenation - If E and F are regular expressions, then E+F is a regular expression which representing the L (E+F) = L (E) UL (F).
3. Closure- if E is a regular expression, then E* is a regular expression which is closure of L (E).That is L(E*) = (L(E))*.

IV.CONTEXT FREE LANGUAGE AND CONTEXT FREE GRAMMAR

Context free language- A context free language is a language generated by some context free grammar.

Context free grammar- A context free grammar consisting set of rules which are finite for guard tuple (N, T, P, S) where

N is a set of non-terminal symbols.

T is a set of terminals where N∩T=NULL.

P is a set of rules

S is the start symbol.

Derivation trees are used to show how a string can derived from context free grammar.[4]

Derivation Tree- the derivation tree also called ordered rooted tree that graphically represents the semantic information a string derived from a context free grammar.

- Root Vertex- must be labeled by the start symbol.
- Vertex- labeled by a non-terminal symbol.
- Leaves – labeled by a terminal symbol or €.

There are different two approaches to draw a derivation tree-

1. Top-down approach- start with starting symbol S. Goes down to tree leaves using productions.
2. Bottom-up approach- starts from tree leaves. Proceeds upward to the root which is the starting symbol S. [5][6]

V.DIFFERENT TYPES OF DERIVATION TREE

Yield of a tree- the yield of a tree is the final string obtained by concatenating the labels of the leaves of the tree from left to right, ignoring the nulls.

Partial derivation tree- it is sub tree of derivation tree such that either all of its children are in the sub tree or none of them are in the sub tree.

Leftmost derivation- a leftmost derivation is obtained by applying production to the leftmost variable in each step.

Rightmost derivation- a rightmost derivation is obtained by applying production to the rightmost variable in each step. [5]

VI. APPLICATIONS OF FINITE STATE MACHINE

a) Word processor programs
b) Digital logical design
c) Lexical analyzer
d) Switching circuit design
e) Text editors

VII. CONCLUSION

In this paper, we present and analyze theory of automata, languages for automata and its computation. The aim is to discover how automata works.

As for future work we the planning to evaluate different models which are working on automata theory.

REFERENCES

1. Surinder singh chahal, guljeet kaur chahal, theory of automata, formal languages and computation.
2. http://www.ggu.ac.in
3. http://www.info.univ-tours.fr
4. http://www.tutorialpoint.com
5. B.khoussainov, A. Nerode, Automata Theory and Its Applications, Springer.
6. W. Thomas, languages, automata and logics, handbook of formal languages, vol. 3, pages 389-455.