Available at: www.researchpublications.org

Finite Automata Based on GUI

Ambadas Ashok Madake#1, Raajdeep Avinash Kohale #2

Niraj Dhobale#3, Ankur Akhaury#4

#1,#2,#3,#4 Information Technology, RTMNU, Nagpur

YCCE, Nagpur

[1]143aamadake05@gmail.com

[2]raajdeepkohale1@hotmail.com

[3]niraj22_dhoble@yahoo.com

[4]ankur_akhaury@yahoo.com

***Abstract*— We have implemented module of finite automata which is based on GUI and implemented with Java.GUI stands for Graphical User Interface.GUI is used to make the module user friendly so that we can easily understand and work on the module. It is a tool in which we can take number of states for designing finite automata and language L. This tool checks the language L is acceptable or not.**

***Keyword*— Finite Automata, Deterministic finite automata, Non-Deterministic finite automata.**

I. INTRODUCTION

Many students find it difficult to grasp the concepts within formal languages and automata theory because the abstract formal notation overwhelms them. In other areas of computer science (such as computer architecture),software simulators provide an excellent teaching tool for enabling active learning of specialized knowledge through abstraction, interactivity, and visualization [3].Theoretical computer simulators have some advantages with we distinguish two categories of FLAT educational software, namely: 1) Generic, multi-purpose software packages for teaching and integrating several related concepts of FLAT[5]. 2) Software tools oriented towards simulating a specific class of automata with educational purposes. In theoretical computer science, automata theory is the study of mathematical____objects called *abstract machines* or *automata* and the computational problems that can be solved using them[5]. *Automata* comes from the Greek word α□τόματα meaning "self-acting".From programming point of view GUI base on two techniques: 1) Object Oriented Programming For organizing program parts with common interfaces and common actions.2)Events: For connecting an event (like a mouse click) with a program action[6].

II. Basics related for finite automata theory:

A. *String :*

A string is a finite sequence of symbols from some alphabet[1].
Ex.Symbols= a,b,c
∑= {a,b,c}
String: abc.ab,ac,bc,a,b,c

B. *The length of the string :*

It is denoted as | | and defined as total number of symbols in a string[2].
For ex., if w= xyz then , |w| = 3
Where w is length of string
If |w|=0 then string is called as empty string and we use a notation
" ϵ" to denote empty string.

C. *Prefix:*

A prefix of a string is the string formed by taking any number of leading symbols of string[2].
For ex., if w= abc then , a , ab, abc are the prefixes of the w.
If |w|=n then there exist n+1 prefixes of w.

D. *Proper prefix:*

Any prefix of a string other than the string itself is called as a proper prefix of that string[1].
For ex., if w= abc then , a , ab are the proper prefixes of the w.

E. *Suffix :*

A suffix of a string is a string formed by taking any number of trailing symbols of a string[2].
For ex., if w= abc then , c , bc, abc are the suffixes of the w.
If |w|= n then there exists n+1 suffixes of w.

F. *Proper suffix:*

Any suffix of a string other than string itself is called as a proper suffix of that string[2].

For ex., w= abc then , c , bc, are the proper suffixes of the w.

G. *Alphabet :*

It is a finite set of symbols denoted by letter ∑[2].

H. *Language:*

It is a set of string formed by using the symbols belonging to some particular chosen alphabet[2].

For ex., if ∑ ={0,1 } then one of the language that can be defined over this ∑ will be L= { ϵ , 0, 00, 000, 1, 11, 111, ... }

E. *Set :*

A set is a well defined collection of objects and it is denoted by[2] :

1. Enumerating the member within " {"and"} "
   For ex., a= {0,1 , 2}
2. Using a set former or set builder notation in which the set is denoted as[2] :
   A={x|p(x)}
   This means A is a set of all those x for which the predicate p(x) is true.
   For example , a set of all integers divisible by 3 will be denoted as
   A={x|x is an integer and x mod 3=0}

III. Finite Automata:

The class of languages accepted by a FSA is called a finite state language[1].

A finite automata consists of finite number of states and finite number of transitions.

These transitions are defined on certain specific symbols called as input symbols.

Finite automata consist of 5 tuple form[2]:

M= {Q, ∑ , ∂, Q0 , F}

Where,

Q = Finite set of states

∑ = Input symbol

∂ = Transition function mapping from Q *∑→Q

i.e ∂: Q *∑→Q

qo = Initial state

F = Set of final state

∂ specifies the transition in the automata. If from state p there exist a transition going to state q on an input symbol a, then we write ∂(p,a)=q , hence ∂ is a function whose domain is a set of ordered pairs (p,a), where p is a state and a is an input symbol and a range is set of states[2].

IV. Types of finite automata**:**

A. *Non-Deterministic Finite Automata***:**

If the basic finite automata model is modified in such a way that from a state on an input symbol zero, one or more transitions are permitted then the corresponding finite automata is called Non-Deterministic Finite Automata[1]. Therefore, NFA is a finite automata in which there may exist more than one path corresponding to x in ∑*, (because zero, one or more transitions are permitted from a state on an input symbol), whereas in a DFA there exists exactly one path corresponding to x in ∑*. Hence , NFA is nothing but finite automata[2]:

It consists of 5 tuples:

M= (Q, ∑, ∂, qo, F)

Where,

Q = Finite set of states

∑ = Input symbol

∂ = Transition function mapping from Q *{ ∑ U ϵ }→2^Q

qo = Initial state

F = Set of final state

B. *Deterministic Finite Automata(DFA):*If there is a unique transition from one state on a single input symbol[1],the finite automata is called as "Deterministic Finite Automata"(DFA). It consists of 5 tuples[2]:

M= (Q, ∑, ∂, qo, F)

Where ,

Q = Finite set of states

∑ = Input symbol

∂ = Transition function mapping from Q *∑→Q

qo = Initial state , F = Set of final state

V. Experimental Result and Analysis:

The algorithm for this tools is being made in java language.[1]This tools can accept or reject given string.

It provides feedback to the user regarding their solutions of acceptance or rejection of the string. The program can simply state that an answer is correct or incorrect. This module is written in java, so it is cross-platform. Here, in Fig. 1 canvas is shown, with buttons to operate this too[5]l.

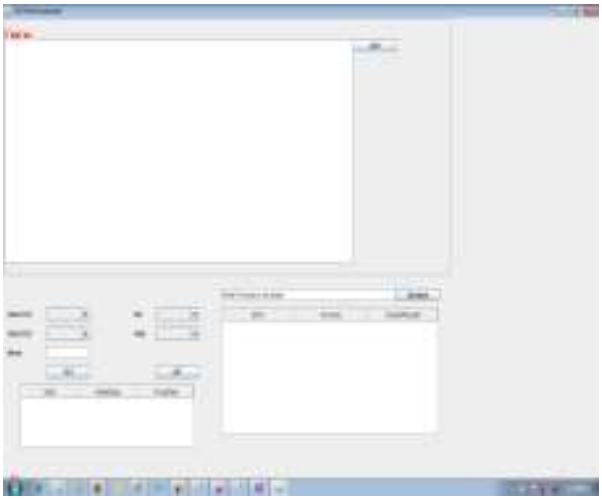

Fig. 1 Create Buttons on Canvas

Then in Fig. 2 transition states are being given as shown below.

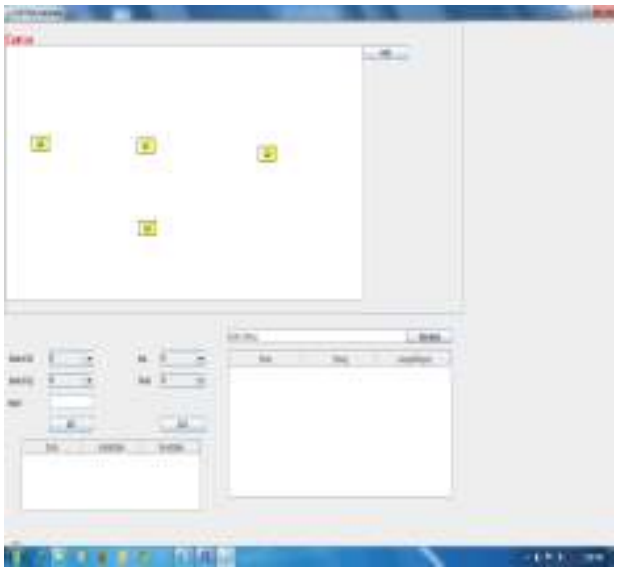

Fig. 2 Creation of Nodes in Canvas

After taking states we give transition arrows with input symbol. Then we give an input string, which is then shown in table of Fig. 3 whether accepted or not.

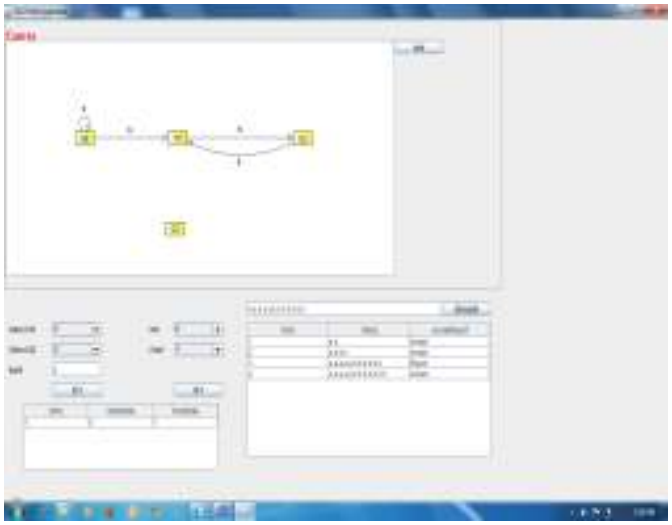

Fig. 3 Creating finite automata String Accept or Reject

REFERENCES

[1] *Theory of computation 2nd ed*. by Vivek Kulkarni
[2] *Theory of Computation 1st ed.* by O.G.Kakde
[3] *Regular Languages and Finite Automata or Part IA of the Computer Science* Tripos L, Marcelo Fiore ,Cambridge University Computer Laboratory
[4] Ling 409 *Lecture Notes*, Partee, Lecture 25
[5] *Finite Automata (DFA)* Gautam Kumar B.Tech student, Computer Science Engineering, BEC Bhubaneswar
[6] *Java2:The Complete Reference,* Herbert Schildt