AUTO GENERATION OF DFA WITH STARTING AND ENDING CONSTRAINTS

[1]TULASHIRAM B. PISAL, [2]ARCHANA A. GHATULE

[1]Department of Computer Science, Sinhgad Institute of Computer Sciences, Pandharpur(MS), India
[2]Principal, AEF's, Ashoka Center for Business and Computer Studies, Nashik
E-mail: [1]pisaltbresearch@gmail.com, [2]archuaag2009@gmail.com

Abstract— Theory of Computation is a mathematical based subject. We observed that many computer science learnersface difficulty in designing Deterministic Finite Automat (DFA) in Theory of Computation. It is always being an issue for the learners to understand the examples because, if an example has been changed then the step are also changes. JFLAP (Java Formal Languages and Automata Package) is a software tool used for designing finite automata, pushdown automata, and Turing machines. We can design any kind of DFA in JFLAP by manually. Drawing a DFA manually is a time consuming process and it also cannot handle the validations for acceptance or rejection of strings. It does not represent the transition table and definition of DFA, which is helpful for understanding the design of DFA. In this paper we are implementing the DFA, which include design of transition graph, transition table and definition of DFA using tuples. We have constructed a DFA with different conditions for starting and ending of string.

Keywords— DFA, Transition Table, Transition Graph (TG), Input Symbol, Minimum String.

I. INTRODUCTION

The learners face difficulty in designing the DFA in Theory of Computation. Theory of Computation is a mathematical based subject. In mathematics, if the example has been changed then the steps remains as it is. But in Theory of Computation, if the example has been changed then the steps are also changes. Example changes then logic are also changes.
To handle this difficulty in designing of DFA, researcher uses JFLAP tool[10,11]. But in JFLAP tool designing is done manually. It is difficult to design a DFA as well as it does not handle validation of input alphabets and strings. In JFLAP it is not possible to draw a transition table which is helpful for better understanding for learners. The researcher uses JFLAP tools for demonstration of acceptance or rejection of a particular example with few strings[1,2].So it is difficult for learners and it will not give the clear idea about the DFA.
The objective of this paper is to give an easy way of learning and designing finite automata that accept a DFA which having different conditions for starting and ending of the string. Here in the implementation phase, examples of different strings are discussed also gives the acceptance or rejection of particular string. This is implemented by using java applet [12, 13].

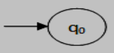
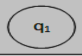
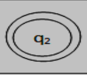
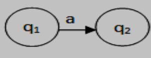
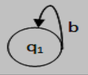
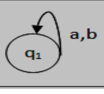
II. METHODOLOGY

Automata theory is a branch of theoretical computer science. It includes Finite Automata (FA), Push Down Automata (PDA), and Turing Machine(TM).
FA is a mathematical model which represents the process done inside the machine. FA includes DFA, NFA and FA with output. A DFA is used to check whether the string will accept or reject. A DFA string reading start from starting state and it reaches to final state after reading last input symbol then string is accepted otherwise string is rejected.
A DFA is defined as an abstract mathematical model. Due to the deterministic nature, a DFA is implementable in hardware and software for solving various problems such as string matching, design digital circuit, in communication protocols like Simple Mail Transfer Protocol (SMTP) and lexical analyzer of compiler phase [7-9].
Finite Automata (M) is denoted by five tuples i.e.

M= (Q, $\sum$, δ, q_0, F)
Where
Q= a finite set of states
$\sum$= a finite set of input symbols
δ= is the state-transition function: $\delta: Qx\sum \rightarrow Q$
q_0= is the initial state i.e.$q0 \in Q$
F = is the set of final states i.e. F is subset of Q.

DFA is represented by using transition graph. The vertexes are labeled by q_0, q_1,…,q_n. Before going towards the design of DFA, we have a need to understand some basic concepts of DFA. These Basic concepts are given in the following **Table 1**.

Table1: Basic Concepts used for designing of a DFA

Sr.No.	Name	Representation	Description
1	Start State or Initial State		q_0 is a initial or starting state. A DFA has only one starting state.
2	Intermediate State		q_1 is a intermediate state. DFA has any number of (Zero or more than zero) intermediate states.
3	Final State		q_2 is a final state. DFA has one or more final states.
4	Arrow		Arrow is used to indicate transition between any two states with input symbol. Here q_1 and q_2 are two states and 'a' is the input symbol.
5	Loop		Loop indicates any number of transitions.
6	Exception State or Rejection State		q_1 is exception state. All input symbols of exception state have transitions to itself to exception state. If more than one input symbols then these symbols are separated by comma (,).

We are represented δ from transition graph.
If w is strings, q_0 is initial state and q_2 is final state then

$$\delta\,(q_0, w) \overset{*}{\Longrightarrow} q_2$$

Transaction rule for DFA: "Each and every input symbol has only one transition from each and every state" [3-6].

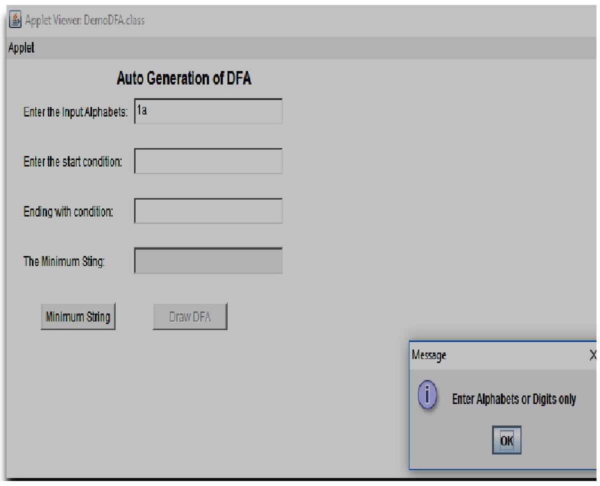
III. ALGORITHM TO DRAW TRANSITION GRAPH

The automatic implementation of DFA by using following algorithm:

Step1: Start
Step2: Accept the input alphabets, start and end condition from user.
Step3: Validate the entered input alphabets, the start and end condition.
Step4: Find out the minimum string using start and end condition.
Step5: Draw a TG for minimum string with applying DFA rule.
Step6: Construct a transition table for above TG.
Step7: Represent M using TG and transition table.
Step8: Check the acceptance or rejection of given strings using M.
Step9: Stop

IV. IMPLEMENTATION

The designing of DFA is done by above algorithm. These steps are as follows:
Step-1: Enter the valid input alphabets: The input alphabets can include only the collection of characters or numbers. It cannot accept combinations of both characters and numbers, it does not accept the special characters and symbols.
Enter the valid start and/or end condition: We can accept at least one condition for both start and end for generation of DFA. The valid start and end condition is every character or number from the input alphabets. It is shown in following figure Fig.1.


Applet Viewer: DemoDFA.class
Applet
Auto Generation of DFA
Enter the Input Alphabets: 1a
Enter the start condition:
Ending with condition:
The Minimum Sting:

Message
Enter Alphabets or Digits only

Fig.1. Validation of string for input alphabets

Step 2: After entering valid string for input alphabets and checking validation for start and end condition, we have need to find minimum string for drawing a DFA. In Fig.2.we have found the minimum string.

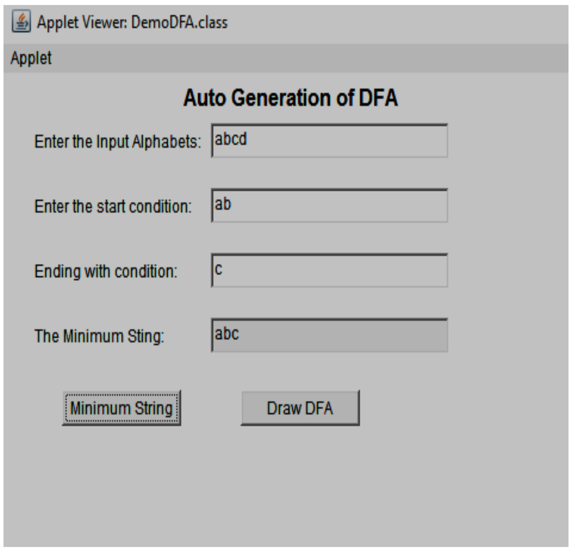
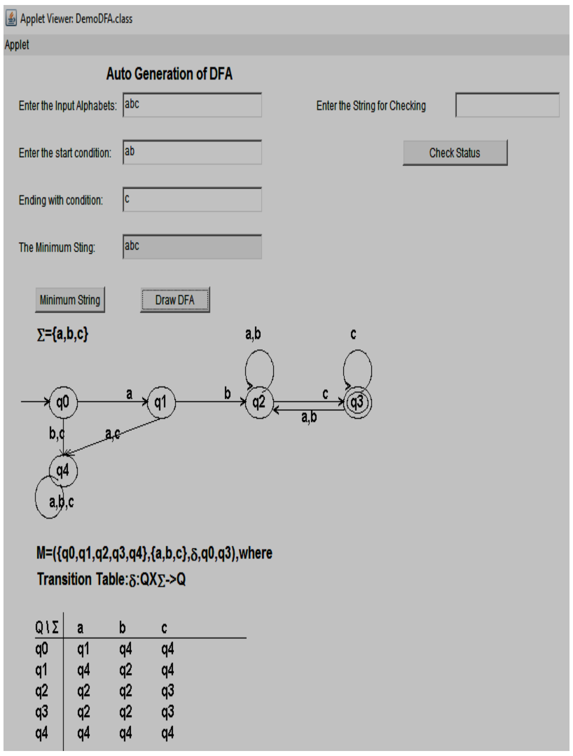

Applet Viewer: DemoDFA.class
Applet
Auto Generation of DFA
Enter the Input Alphabets: abcd
Enter the start condition: ab
Ending with condition: c
The Minimum Sting: abc

Fig.2. Finding a minimum string

Step 3: We applies rules of DFA for drawing complete DFA as well as representing transition table and definition. It is shown in following Fig.3.

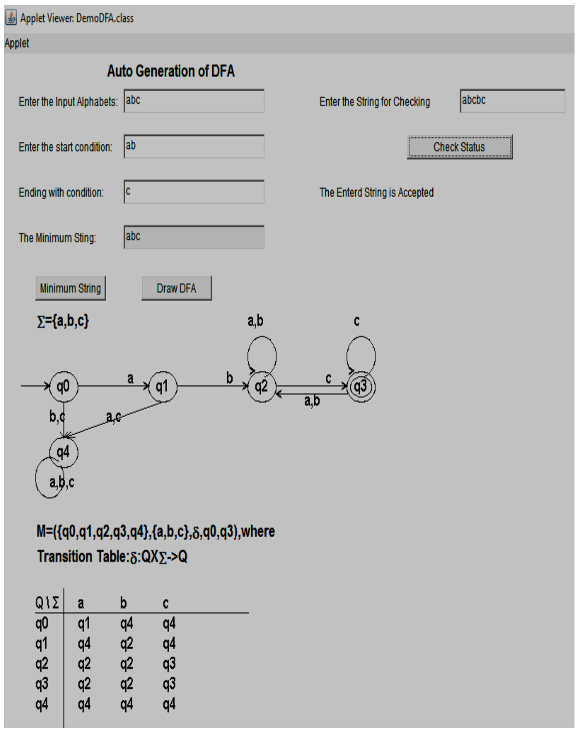

Applet Viewer: DemoDFA.class
Applet
Auto Generation of DFA
Enter the Input Alphabets: abc
Enter the start condition: ab
Ending with condition: c
The Minimum Sting: abc
Enter the String for Checking

M=({q0,q1,q2,q3,q4},{a,b,c},δ,q0,q3),where
Transition Table:δ:QXΣ->Q

Q \ Σ	a	b	c
q0	q1	q4	q4
q1	q4	q2	q4
q2	q2	q2	q3
q3	q2	q2	q3
q4	q4	q4	q4

Fig.3. Complete DFA for string start with ab and end with c

Step 4: After drawing a complete DFA, we can draw a transition table and represent the definition, Here we check the acceptance or rejections of Strings. The following Fig.4. Shows the acceptance of string and Fig.5. Shows rejection of a string for DFA start with ab and end with c.



Auto Generation of DFA

Enter the Input Alphabets: abc

Enter the String for Checking: abcbc

Enter the start condition: ab

Ending with condition: c

The Enterd String is Accepted

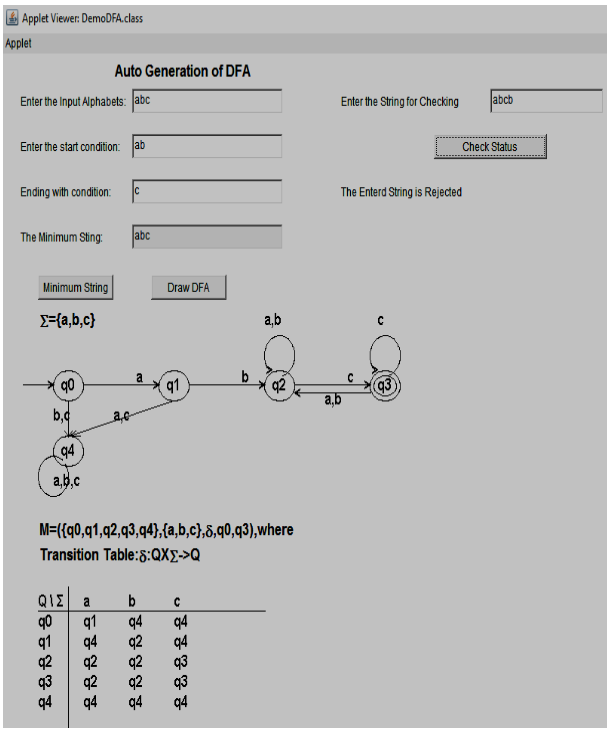
The Minimum Sting: abc

M=({q0,q1,q2,q3,q4},{a,b,c},δ,q0,q3),where

Transition Table:δ:QXΣ->Q

Q \ Σ	a	b	c
q0	q1	q4	q4
q1	q4	q2	q4
q2	q2	q2	q3
q3	q2	q2	q3
q4	q4	q4	q4

Fig.4. The acceptance of string abcbc



Auto Generation of DFA

Enter the Input Alphabets: abc

Enter the String for Checking: abcb

Enter the start condition: ab

Ending with condition: c

The Enterd String is Rejected

The Minimum Sting: abc

M=({q0,q1,q2,q3,q4},{a,b,c},δ,q0,q3),where

Transition Table:δ:QXΣ->Q

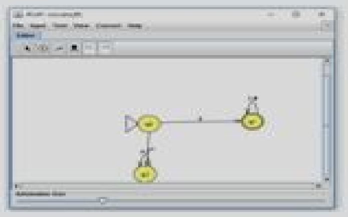
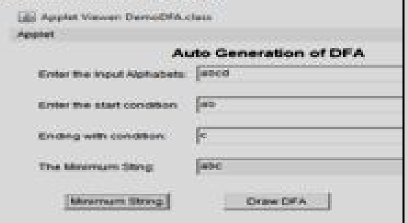
Q \ Σ	a	b	c
q0	q1	q4	q4
q1	q4	q2	q4
q2	q2	q2	q3
q3	q2	q2	q3
q4	q4	q4	q4

Fig.4. The acceptance of string abcb

V. RESULT

The result of this algorithm as compared to the JFLAP tools result are given in following tables**Table 2**.

Table2: The result of algorithm compare with JFLAP tools result

Sr.No.	Characteristic	Representation	
		JFLAP	Auto Generation of DFA
1	Validation for input symbol	Validation is not handeled in JFLAP tool. 	Validation is handeled in auto generation of DFA.
2	Finding Minimum String	Such facility is not provided in JFLAP.	It finds minimum string.  Applet Viewer: DemoDFA.class Applet Auto Generation of DFA Enter the Input Alphabets: abcd Enter the start condition: ab Ending with condition: c The Minimum Sting: abc Minimum String Draw DFA
3	Drawing TG	TG is drawing manually.	TG is drawing automatically.
4	Drawing Transition Table	Drawing transition table facilitie is not provided in JFLAP.	Transition table is drawan in auto generation of DFA.
5	Definition	Defination can not be represented in JFLAP.	Defination is represented auto generation of DFA.
6	String acceptance or Rejection	It has complicated process.	It is very easy process.

CONCLUSIONS

Result of this implementation is compared with the JFLAP tool. Results were revels that Performance of auto generation of DFA implementation gives the better performancethan JFLAP tool. Design of DFA automatically handles all types of validations and follows the rules of DFA. Design of DFA is done using transition graph, transition table and definition of DFA which is helpful tofor better understanding. Also it gives the acceptance or rejections of the user entered strings according to DFA.

ACKNOWLEDGMENTS

This research is a part of my research entitled “Sequential Pattern Mining using Turing Machine”. We thank Dr. Arjun P. Ghatule for his help and for the discussions on the topics in this paper. I also thanks to Mr. Umate P.R. for his help and discussions on topics of this paper. The paper is partially supported by Sinhgad Institute of Computer Sciences, Pandharpur of Solapur University, Solapur(MS), India.

REFERENCES

[1] Danish Ather, Raghuraj Singh and Vinodani Katiyar, "Simplifying Designing Techniques: To Design DFA that Accept Strings over ∑= {a, b} having at least x Number of a and y Number of b", International Journal of Computer Applications (0975 – 8887) Volume 91 – No.7, pp. 12-17, April 2014.

[2] Danish Ather, Raghuraj Singh and Vinodani Katiyar, "An Efficient Algorithm to Design DFA That Accept Strings Over Input Symbol a, b Having Atmost x Number of a & y Number of b", Journal of Nature Inspired Computing (JNIC) 30 Vol. 1, No. 2, pp. 30-33, 2013.

[3] Deterministic Finite Automaton [Online] Available at http://www.wikipedia.org/wiki/Deterministic_finite_automation.

[4] John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, “Automata Theory, Language, and Computation”, Delhi: Pearson, pp.35-50, 2008.

[5] Vivek Kulkarni, “Theory of Computation”, Pune: Tech-Max, pp.1.1-2.19, 2007.

[6] Dilip Kumar Sultania, “Theory of Computation”, Pune: Tech-Max, pp.1.1-2.47, 2010.

[7] Nazir Ahmad Zafar and Fawaz Alsaade, "Syntax-Tree Regular Expression Based DFA Formal Construction", Intelligent Information Management, pp. 138-146, 2012.

[8] Jan Holub and Stanislav Stekr, "Implementation of Deterministic Finite Automata on Parallel Computers", This research has been partially supported by the Ministry of Education, Youth and Sports under research program MSM 6840770014 and the Czech Science Foundation as project No. 201/06/1039.

[9] Pratima Sarkar and Chinmoy Kar, "Adaptive E-learning using Deterministic Finite Automata", International Journal of Computer Applications (0975 – 8887) Volume 97 – No.21, PP. 14-17, July 2014.

[10] JFAP Tool for Simulating Results and Validation. [Online]. Available: http://www.jflap.org.

[11] Ankur Singh and Jainendra Singh, "Implementation of Recursively Enumerable Languages using Universal Turing Machine in JFLAP", International Journal of Information and Computation Technology, ISSN 0974-2239 Volume 4, Number 1, pp. 79-84, 2014.

[12] Java Applet [Online] Available at https://en.wikipedia.org/wiki/Java_applet

[13] Java Applet [Online] Available at http://www.tutorialspoint.com/java/java_applet_basics.htm

★★★