Journal of Nature Inspired Computing (JNIC)
Vol. 1, No. 2, 2013

www.worldsciencepublisher.org

An Efficient Algorithm to Design DFA That Accept Strings Over Input Symbol a, b Having Atmost x Number of a & y Number of b.

[1]Danish Ather, [2]Raghuraj Singh, [3]Vinodani Katiyar

[1]Teerthanker Mahaveer Univerisity, Moradabad, India
[2]Computer Science & Engineering Department, H.B.T.I., Kanpur,India.
[3]SRCET Computer Science & Engineering Department, Lucknow, India

email: danishather@gmail.com

ABSTRACT: Theory of computation is always been an issue for the students to understand. So there is a research gap which will ease the process of teaching learning. Our research objective is to develop method to make teaching learning process of theory of computation easier, simpler and understandable. In this paper we develop an algorithm which will generalize the design of finite automata that Accept Strings Over Input Symbol a, b Having Atmost x Number of a & y Number of b.

KEYWORD DFA, Transition Table,FA

1 INTRODUCTION

Objective of this paper is to develop an efficient algorithm that generalize the method of design of finite automata that accept strings over input symbol a, b having atmost x number of a & y number of b.

2 MYTHOLOGY

In automata theory, a branch of theoretical computer science, a deterministic finite automaton (DFA)—also known as deterministic finite state machine—is a finite state machine that accepts/rejects finite strings of symbols and only produces a unique computation (or run) of the automaton for each input string.[1] 'Deterministic' refers to the uniqueness of the computation. In search of simplest models to capture the finite state machines, McCulloch and Pitts were among the first researchers to introduce a concept similar to finite automaton in 1943.[2][3]

A DFA is defined as an abstract mathematical concept, but due to the deterministic nature of a DFA, it is implementable in hardware and software for solving various specific problems. For example, a DFA can model software that decides whether or not online user-input such as email addresses are valid.[4]

Finite Automata (M) is defined as a set of five tuples (Q, ∑, δ, Q0, F)

Where

Q= a finite, non-empty set of states

∑= a finite, non-empty set of inputs

δ is the state-transition function:

$\delta: Q X \sum \rightarrow Q$

Q0 is the initial state

F is the set of final states, a (possibly empty) subset of Q.

δ can be represents using either of three approach given below

Transition Graph.

Transition Function.

Transition Table.

We had used the transition table as the approach to represent δ.

Algorithm

By applying this Algorithm we can design Deterministic Finite Automata that accept strings over input symbol a, b having atmost x number of a & y number of b

Algorithm to draw TG

Deterministic Finite Automata M= (Q, ∑, δ, Q0, F)

Where

Q= {q11,q12,q13,….q21,q22,……qij}
∑= {a,b}
δ:QX∑→Q{Represented by Transition Graph }
Q0 = qij when i=j and i=j=1. i.e. q11
F={q11,q12,q13,….q21,q22,……qij}
Let Q be the set of states in Deterministic Finite Automata such that Q={q11,q12,q13,….q21,q22,……qij}
Where i = 1 to x+1
j= 1 to y+1
Input Symbol ∑={a,b}
q11 is the initial state.
Design a directed transition graph having (x+1)*(y+1) states and mark all states as final states.
Label each node as q11,q12,q13,….q21,q22,……qij
Where i = 1 to x+1; j= 1 to y+1; x= na & y=nb
FOR i= 1 to x
do
FOR j= 1 to y
do
if i=j=1 then qij ε Q0 (Initial State)
else there exist a edge E such that δ(qij,a)→qij+1
done inner loop
done outer loop
FOR i= 1 to x
do
FOR j= 1 to y
do
if i=j=1 then qij ε Q0 (Initial State)
else there exist a edge E such that δ(qji,b)→qj+1,i
if i=x+1 and j=y+1 then there exist δ(qij, a)→qij and δ(qij, a)→qij.
done inner loop
done outer loop
q11 being the initial state
DFA "M" will strings over input symbol a, b having atmost x number of a & y number of b if all the input is consumed and halting state is the final state.

3 IMPLEMENTATION

3.1 Design a DFA that accept Strings Over Input Symbol a, b having atmost three a's & three of b.

Let the resultant DFA is M= (Q, ∑, δ, Q0 , F)
Q = {q11, q12, q13, q14, q21, q22, q23, q24, q31, q32, q33, q34, q41, q42, q43, q44}
∑ = {a,b}
δ is given by transaction graph in Figure 1
Q0 = { q11}
F = {q11, q12, q13, q14, q21, q22, q23, q24, q31, q32, q33, q34, q41, q42, q43, q44}

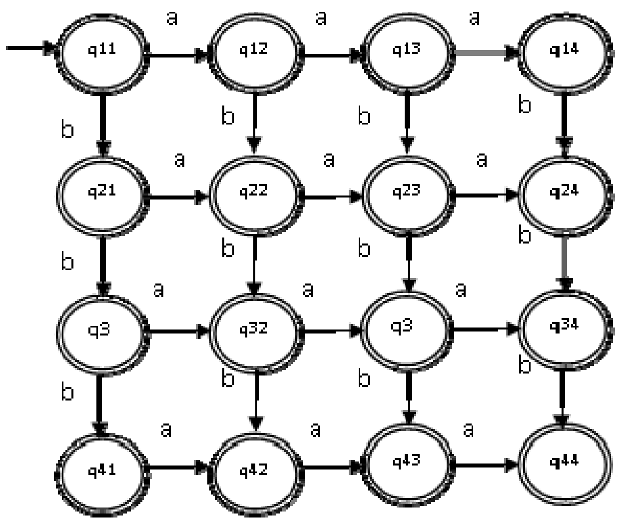

Figure 1 TG of DFA that accept Strings Over Input Symbol a, b having Exactly three a's & three of b.

Design a DFA that accept Strings over Input Symbol a, b having atmost two a's & only one b.

Let the resultant DFA is M= (Q, ∑, δ, Q0 , F)
Q = {q11, q12, q13,q21, q22, q23}
∑ = {a, b}
δ is given by transaction graph in Figure 2
Q0 = { q11}
F ={q11, q12, q13,q21, q22, q23}

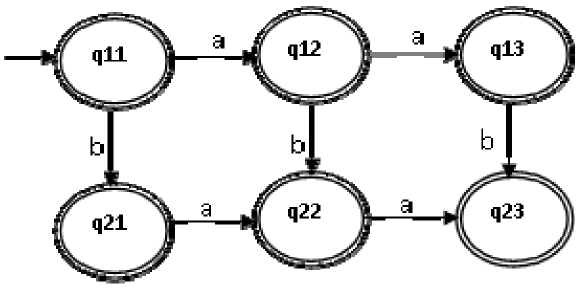

Figure 2 TG of DFA that accept Strings Over Input Symbol a, b having Exactly two a's & only one b

3.2 Design a DFA that accept Strings over Input Symbol a, b having atmost only one a & two b.

Let the resultant DFA is M= (Q, ∑, δ, Q0 , F)
Q = {q11, q12, q21, q22, q31, q32}
∑ = {a,b}
δ is given by transaction graph in Figure 3
Q0 = { q11}
F = {q11, q12, q21, q22, q31, q32}

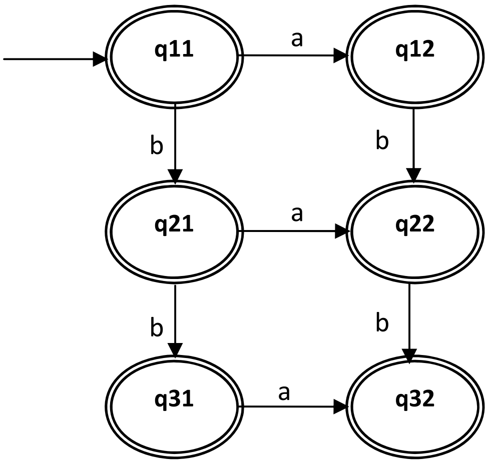

Figure 3 TG of DFA that accept Strings over Input Symbol a, b having exactly only one a & two b.

4 RESULT ANALYSIS AND DISCUSSION

The simulation results presented in this section were run with Java Formal Languages and Automata (jFLAP).

4.1 Simulation of DFA that accept Strings Over Input Symbol a, b having atmost three a's & three of b.

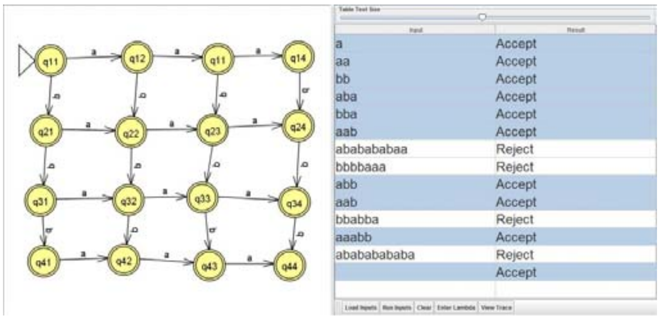

Figure 4 Multiple string test result based on 5.4.1

4.2 Simulation of DFA that accept Strings Over Input Symbol a, b having atmost two a's & only one b.

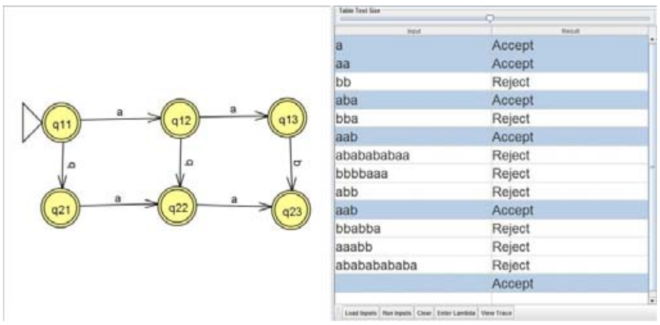

Figure 5 Multiple string test result based on 5.4.2

4.3 Simulation of DFA that accept Strings over Input Symbol a, b having atmost only one a & two b.

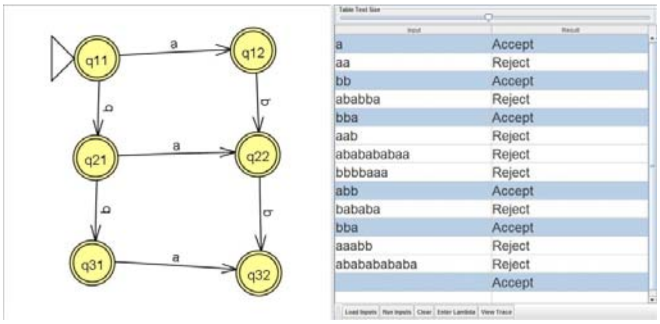

Figure 6 Multiple string test result based on 5.4.3

4.4 Test for existence of NFA in DFA that accept Strings Over Input Symbol a, b having atmost three a's & three of b.

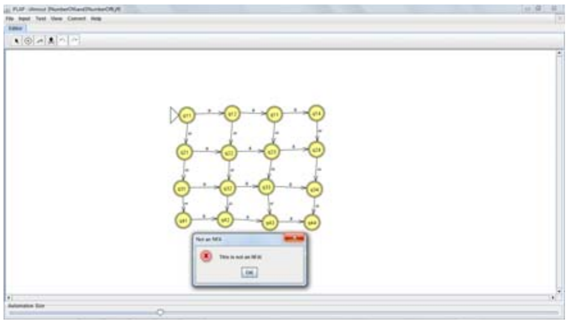

Figure 7 Test for existence of NFA in DFA based on 5.4.1

4.5 Test for existence of NFA in DFA that accept Strings Over Input Symbol a, b having atmost two a's & only one b.

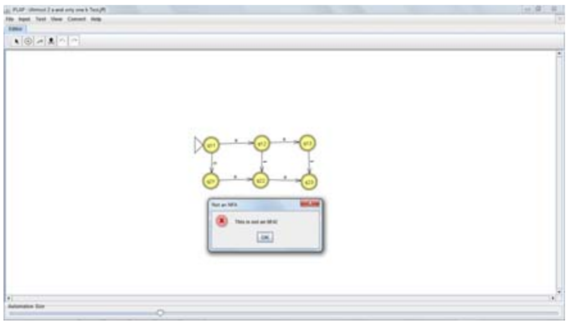

Figure 8 Test for existence of NFA in DFA based on 5.4.2

4.6 Test for existence of NFA in DFA that accept Strings over Input Symbol a, b having atmost only one a & two b.

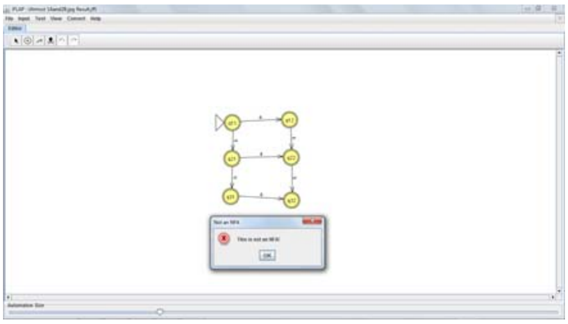

Figure 9 Test for existence of NFA in DFA based on 5.4.3

It has been observed that except the case of unary base number in algorithm 4.1 this computation model works appropriately and producing adequate results

5 CONCLUSION

This research will definitely enhance the teaching learning environment of theory of computation and helps engineering students to design DFA. Current model work with a limitation of null string and the algorithm works well with two input variable only.

ACKNOWLEDGMENT

This research is a part of my research entitled “Optimized simplified algorithm of computational theory”. We also thank R. K. Dwivedi for his help and for the discussions on some of the topics in this paper. The paper is partially supported by Teerthnaker Mahaveer University.

REFERENCES

[1.] Hopcroft, John E.; Motwani, Rajeev; Ullman, Jeffrey D. (2001). Introduction to Automata Theory, Languages, and Computation (2 ed.). Addison Wesley. ISBN 0-201-44124-1. Retrieved 19 November 2012.

[2.] Lawson, Mark V. (2004). Finite automata. Chapman and Hall/CRC. ISBN 1-58488-255-7. Zbl 1086.68074.

[3.] McCulloch, W. S.; Pitts, E. (1943). "A logical calculus of the ideas imminent in nervous activity". Bulletin of Mathematical Biophysics: 541–544.

[4.] Rabin, M. O.; Scott, D. (1959). "Finite automata and their decision problems.". IBM J. Res. Develop.: 114–125.

[5.] Sakarovitch, Jacques (2009). Elements of automata theory. Translated from the French by Reuben Thomas. Cambridge: Cambridge University Press. ISBN 978-0-521-84425-3. Zbl 1188.68177.

[6.] Sipser, Michael (1997). Introduction to the Theory of Computation. Boston: PWS. ISBN 0-534-94728-X.. Section 1.1: Finite Automata, pp. 31–47. Subsection "Decidable Problems Concerning Regular Languages" of section 4.1: Decidable Languages, pp. 152–155.4.4 DFA can accept only regular language

[7.] C. Allauzen and M. Mohri, Finitely subsequential transducers, International J. Foundations Comp. Sci. 14 (2003), 983-994

[8.] M.-P. Béal and O. Carton, Determinization of transducers over finite and infinite words, Theoret. Comput. Sci. 289 (2002), 225-251

[9.] Bruggemann-Klein, Regular expressions into finite automata, Lecture Notes in Computer Science 583 (1992), 87-98

[10.] J. Carroll and D. Long, Theory of Finite Automata, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[11.] M. V. Lawson, Finite Automata, CRC Press, Boca Raton, FL, 2003.

[12.] D. Perrin, Finite automata, in Handbook of Theoretical Computer Science, Volume B (editor J. Van Leeuwen), Elsevier, Amsterdam, 1990, 1-57.

[13.] D. Perrin and J. E. Pin, Infinite Words, Elsevier, Amsterdam, 2004.

[14.] Ch. Reutenauer, Subsequential functions: characterizations, minimization, examples, Lecture Notes in Computer Science 464 (1990), 62-79.

[15.] J. Sakarovitch, Kleene’s theorem revisited, Lecture Notes in Computer Science 281(1987), 39-50.

[16.] E. Roche and Y. Schabes (editors), Finite-State Language Processing, The MIT Press, 1997.

[17.] B. W. Watson, Implementing and using finite automata toolkits, in Extended Finite State Models of Language (editor A. Kornai), Cambridge University Press, London, 1999, 19-36.

[18.] Danish Ather, Raghuraj Singh, Vinodani Katiyar. To Develop an Efficient Algorithm That Generalize the Method of Design of Finite Automata That Accept “N” Base Number Such That When “N” Is Divided By “M” Leaves Reminder “X”. IJCA ISBN December 2012

[19.] Tarun Kumar Sharma, Millie Pant, V.P. Singh, Improved Local Search in Artificial Bee Colony using Golden Section Search, Journal of Engineering, 1:1(2012) 14-19.

[20.] Tarun Kumar Sharma, Millie Pant, V.P. Singh, Adaptive Bee Colony in an Artificial Bee Colony for Solving Engineering Design Problems, Advances in Computational Mathematics and its Applications, 1:4(2012) 213-221.

[21.] Avanaksh Singh Sambyal and Prikhshayat Singh Routing Misbehavior in MANets and How it Impact QoS!, Advances in Computer Science and its Applications,1:1(2012) 84-88.

[22.] S.Balaji, Lakshmi.A, V.Revanth, M.Saragini and V.Venkateswara Reddy, Authentication techniques for engendering Session passwords with colors and text, Advances in Information Technology and Management, 1:2(2012) 71-78.

[23.] C.Narasimha and B.Jalaja Kumari, Secured Multicasting over MANET’s through EGMP, Advances in Information Technology and Management, 1:2(2012) 90-96.

[24.] M. Sreedevi,.C.Narasimha and R.Seshadri, Efficient Data Delivery Over MANET’s through Secured EGMP, Advances in Asian Social Science, 2:3(2012) 512-516.