A Simplified Solution To Design DFA That Accept Strings Over input symbol { a, b} Having At Least x Number Of A And Exactly x Number Of B.

Danish Ather[#1] Ashendra Kumar Saxena[#2] Ambuj Kumar Agarwal[#3]
Department Of Computer Applications, Teerthanker Mahaveer Univerisity, Moradabad, India.
[1]danishather@gmail.com, [2]ashendrasaxena@gmail.com, [3]ambuj4u@gmail.com

Abstract- *Today, it is very difficult to understand the designing concepts of deterministic machine. One question is arise? How it is possible to understand the concepts of deterministic machine in a very easy manner. In this paper, we have design a DFA and develop an algorithm with suitable graphs and examples that how DFA machine works in a simply manner. For it, we consider that a DFA machine takes the input string {a, b} having atleast x number of a and y number of b. The objective of this paper to understand the concepts of deterministic machine in easy manner. We have design a finite automata machine and developed an algorithm to accepts strings of input {a, b} having atleast x number of a and y number of b.*

Keyword - DFA, Transition Table, Transition Graph(TG), Input Symbol

I. INTRODUCTION

A deterministic Finite automaton (also known as deterministic finite state machines) is the system to accomplish many tasks in Computer Science. To increase the computational power of existing computers, it is based not only to increase the frequency of CPU but also we use other modern technologies. The finite automata implementations are used to consider these types of technologies. For example, multiple CPU core is one of the latest technologies which is used now. We can represent DFA by digraphs which is also called state transition diagram. In this digraph the vertices are denoted by single circles of a transition diagram which represent the states of the DFA and the arcs are labeled with an input symbol correspond to the transitions. We represent accepting states by double circles.
We had simulated the results and verified it using JFLAP an open source tool[14] .

II. MYTHOLOGY

Automata theory is a branch of theoretical computer science, DFA is known as Deterministic Finite Automata. A finite state machine accepts or rejects finite strings of symbols and gives a unique computation for each input string[15]. McCulloch and Pitts were among the first researchers to introduce a concept similar to finite automaton in 1943.[5]

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[15]. For example, a DFA can model software that decides whether or not online user-input such as email addresses are valid.

Finite Automata (M) is defined as a set of five tuples (Q, Σ, δ, q_0, F)
Where

Q= a finite, non-empty set of states

Σ= a finite, non-empty set of inputs

δ is the state-transition function:

$\delta: Q X \Sigma \rightarrow Q$

q_0 is the initial state

F is the set of final states, a 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 Graph as the approach to represent δ. A transition diagram is a special kind of directed labeled graph: the vertices are labeled by the states Q; there is an arrow labeled from the vertex labeled q ε Q to the vertex labeled q ε Q on some Σ .The input Σ causes the

ISSN: 2348 – 3326

acceptor change from state q ε Q to q ε Q. The initial state and final states are distinguished as we mark the initial state by an inward-pointing arrow, the final states by double circles.[4]

III. ALGORITHM TO DRAW TG

- Number of a in input string = x
- Number of b in input string = y

FINITE AUTOMATA M=(Q, ∑, δ, q_0, F)

Where

- Q= {q_{11}, q_{12}, q_{13}…. q_{21},…… q_{ij}} i=x+1 and j=y+1
- Input Symbol ∑={a,b}
- δ:QX∑→Q{Represented by Transition Graph }
- q_0 = q_{ij} when i=j and i=j=1. i.e. q_{11}
- F=q_{ij} when i=x+1 and j=y+1.
- Let Q be the set of states in finite automata such that Q = {q_{11}, q_{12}, q_{13}…. q_{21},…… q_{ij}} where i = 1 to x+1 and j= 1 to y+1
- Design a directed transition graph having (x+1)*(y+1) states.
- Label each node as q_{11}, q_{12}, q_{13}…. q_{21},…… q_{ij}
  Where i = 1 to x+1; j= 1 to y+1; x= n_a & y=n_b
- FOR i= 1 to x
- do
- FOR j= 1 to y
- do
- if i=j=1 then q_{ij} ε q_0 (Initial State)
- else there exist a edge E such that δ(q_{ij},a)→$q_{i(j+1)}$
- done inner loop
- done outer loop
- FOR i= 1 to x
- do
- FOR j= 1 to y
- do
- if i=j=1 then q_{ji} ε q_0 (Initial State)
- else there exist a edge E such that δ(q_{ji},b)→$q_{(j+1)i}$
- if i=x and j=y then $q_{(i+1)(j+1)}$ is the final state.
- done inner loop
- done outer loop
- FOR i= 1 to y+1
- do
- there exist a edge E such that δ($q_{i(x+1)}$,a)→$q_{i(x+1)}$
- done
- DFA "M" will accept all string with atleast x no of "a" and exactly y number of "b" if all the inputs are consumed and halting state is the final state.

IV. IMPLEMENTATION

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

Let the resultant DFA is M= (Q, ∑, δ, q_0 , F)

Q = {q_{11}, q_{12}, q_{13}, q_{14}, q_{21}, q_{22}, q_{23}, q_{24}, q_{31}, q_{32}, q_{33}, q_{34}, q_{41}, q_{42}, q_{43}, q_{44}}

∑ = {a,b}

δ is given by transaction graph in Figure 1

q_0 = { q_{11}}

F = {Q_{44}}

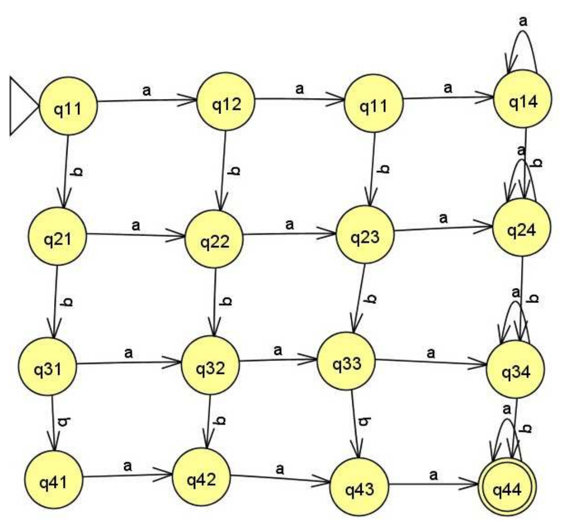

Figure 1 DFA that accepts a string having atleast 3 a's and exactly 3 b's

Design a DFA that accept Strings Over Input Symbol a, b having atleast two a's & exactly one b.

Let the resultant DFA is M= (Q, ∑, δ, q_0 , F)

Q = {q_{11}, q_{12}, q_{13},q_{21}, q_{22}, q_{23}}

∑ = {a, b}

δ is given by transaction graph in Figure 2

q_0 = { q_{11}}

F = {q_{23}}

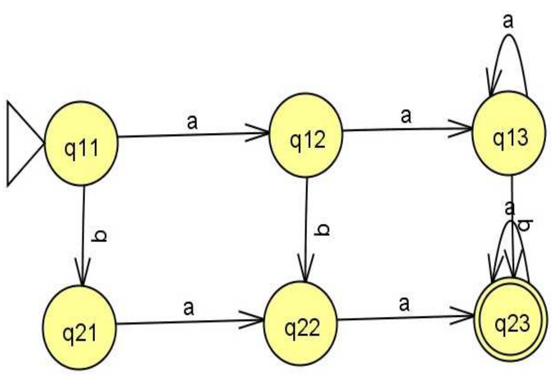

Figure 2 DFA that accepts a string having atleast 2 a's and exactly 1 b's

Design a DFA that accept Strings over Input Symbol a, b having atleast one a &exactly two b.

Let the resultant DFA is M= (Q, ∑, δ, q_0 , F)

Q = {q_{11}, q_{12}, q_{21}, q_{22}, q_{31}, q_{32}}

∑ = {a,b}

δ is given by transaction graph in Figure 3

q_0 = { q_{11}}

F = {q_{32}}

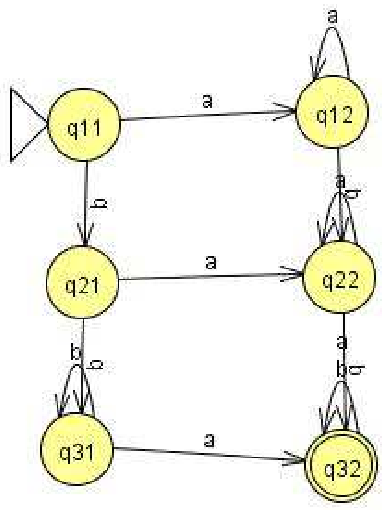

Figure 3 DFA that accepts a string having atleast 1 a's and 2 b's.

V. RESULT

The simulation results presented in this section were run with Java Formal Languages and Automata (jFLAP)[22]. Various DFA design were implemented under section IV based on the purposed algorithm mentioned under section III. These implementations were simulated and the results are verified using jFLAP open source tool whose snapshot are mentioned below under Figure No. 4,5,6.

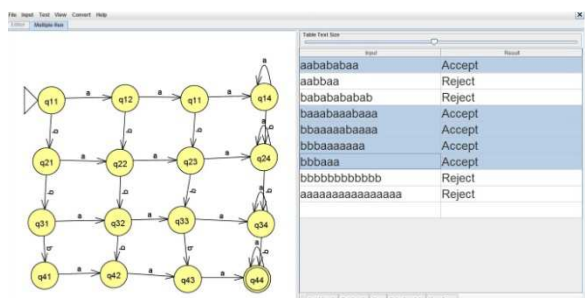

Figure 4 Multiple string test result for atleast 3a's and exactly 3 b's.

ISSN: 2348 – 3326

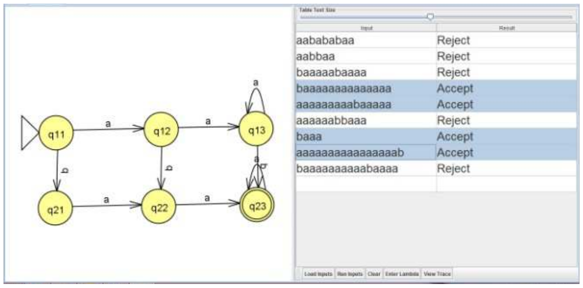

Figure 5 Multiple string test result for atleast 2a's and exactly 1 b's.

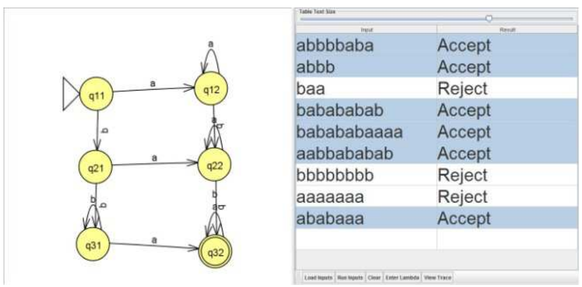

Figure 6 Multiple string test result for atleast 1a's and exactly 2 b's.

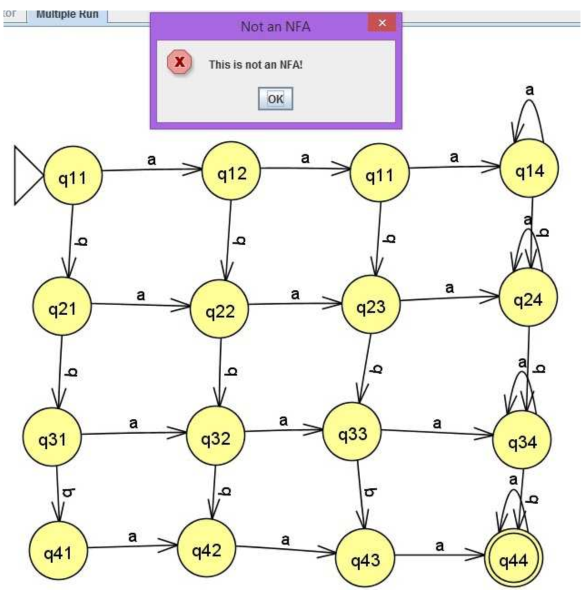

Figure 7 Test for existence of NFA in DFA that accept Strings over Input Symbol a, b having atleast 3 a & exactly 3 b.

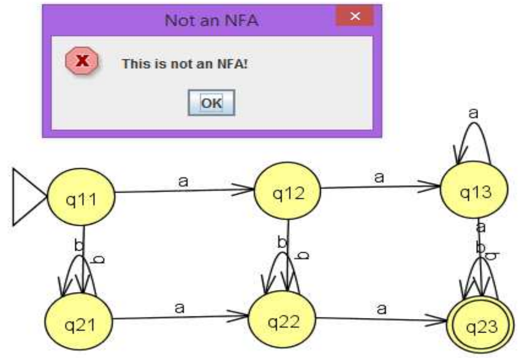

Figure 8 Test for existence of NFA in DFA that accept Strings over Input Symbol a, b having atleast 2 a & exactly 1 b.

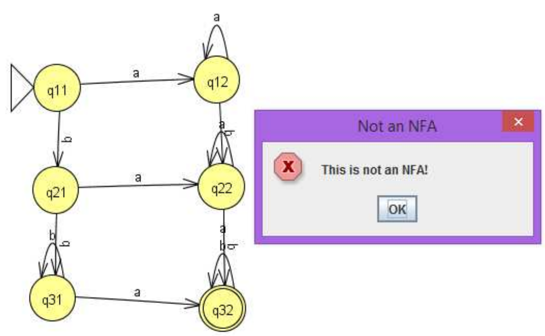

Figure 9 Test for existence of NFA in DFA that accept Strings over Input Symbol a, b having atleast 1 a & exactly 2 b.

VI. CONCLUSION

Examples based on the purposed algorithm is being verified using JFLAP in figure no 1,2,3 and test of acceptability/rejection of the string is done which is shown in figure no 4,5,6. Moreover test for existence of NFA is done which is shown in figure 7,8,9. The purposed method will help in easy and accurate design of a transition graph of finite automata that accept strings over input symbol a, b having atleast x number of a & y number of b.

ACKNOWLEDGMENT

This research is a part of my research entitled “Optimized simplified machine design model of computational research”. 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 Ather Educational Scoiety.

REFERENCES

[1.] Danish Ather, Raghuraj Singh, 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. IJCA- Foundation of Computer Science U.S.A Volume 91 - Number 7 April 2014.

[2.] 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- Foundation of Computer Science U.S.A Volume 60 - Number 10 December 2012.

[3.] Danish Ather, Raghuraj Singh, Vinodani Katiyar. An Efficient Algorithm to Design DFA That Accept Strings Over Input Symbol a, b Having exactly x Number of a & y Number of b. in IEEE Catalogue Number: CFP1362U-CDR, March,2013

[4.] Danish Ather, Raghuraj Singh, Vinodani Katiyar. An Efficient Algorithm to Design DFA That Accept Strings Over Input Symbol a, b Having Atmost x Num-ber of a & y Number of b. JNIS World Science Publishers, U.S.A. Feb 2013

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

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

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

[8.] J. E. Hopcroft and J. D. Ullman, Introduction to Automata The-ory, Languages,and Computation, Addison-Wesley, Reading, MA, 1979.

[9.] J. E. Hopcroft, R. Motwani, and J. D. Ullman, Introduction to Automata Theory,Languages, and Computation, 2nd Edition, Addison-Wesley, Reading, MA, 2001.

[10.] D. Jurafsky and J. H. Martin, Speech and Language Processing, Prentice-Hall,Englewood Cli®, NJ, 2000.

[11.] B. Khoussainov and A. Nerode, Automatic presentations of structures, LectureNotes in Computer Science 960 (1995), 367-392.

[12.] A. Kornai (editor), Extended Finite State Models of Language, Cambridge Uni-ersity Press, London, 1999.

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

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

[15.] www.jflap.org

[16.] en.wikipedia.org/wiki/Deterministic_finite_automaton