Simplifying Designing Techniques: To Design DFA that Accept Strings over ∑= {a, b} having at least x Number of a and y Number of b

Danish Ather
Teerthanker Mahaveer Univerisity,
Moradabad, India

Raghuraj Singh
Computer Science & Engineering Department, H.B.T.I.,
Kanpur, India

Vinodani Katiyar
SRCET Computer Science & Engineering Department,
Lucknow, India

ABSTRACT
Being a faculty of Theory of computation it had been observed by me that students face difficulty while designing different DFA. It is always being an issue for the students to understand. This paper present an easy way of learning and designing a Deterministic finite automata that accept strings over input symbol {a, b} having atleast x number of a & y number of b. Objective of the research is to make the method of teaching learning easier , simpler and understandable for students . In this paper we develop an algorithm to design finite automata that accepts strings of input {a, b} having atleast x number of a & y number of b .

Keywords
DFA, Transition Table, Transition Graph(TG), Input Symbol

1. INTRODUCTION
Objective of this paper is to give an easy way of learning and designing finite automata that accept strings over input symbol {a,b} having atleast x number of a & y number of b . In the implementation phase several example of different string are discussed under section V Implementation moreover we had simulated the results and verified it using JFLAP an open source tool [6] .

2. 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 [7]. 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[7]. 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, ∑, δ, q_0, F)

Where

Q= a finite, non-empty set of states

∑= a finite, non-empty set of inputs

δ is the state-transition function:

δ:QX∑→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 ∑ .The input ∑ causes the 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][1][2][3]

In transition graph final state is represented by double circle and initial state by a initial arrow where as other states are represented are represented by circle. A edge in transition graph is generally drawn by an arrow from one state to another labeled by a input symbol.

3. 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_{22},……q_{ij}} i=x+1 and j=y+1

∑= {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_{22},……q_{ij}}

Where i = 1 to x+1

j= 1 to y+1

- Input Symbol $\sum$={a,b}
- q_{11} is the initial state.
- Design a directed transition graph having (x+1)*(y+1) states.
- Label each node as $q_{11}, q_{12}, q_{13}, \ldots q_{21}, q_{22}, \ldots\ldots q_{ij}$

  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 $\delta(q_{ij},a) \rightarrow q_{i,j+1}$
- done inner loop
- done outer loop
- FOR i= 1 to x
- do
- FOR j= 1 to y
- do
- if i=j=1 then q_{ij} ε Q_0 (Initial State)
- else there exist a edge E such that $\delta(q_{ji},b) \rightarrow q_{j+1,i}$
- if i=x+1 and j=y+1 then q_{ij} is the final state
- done inner loop
- done outer loop
- FOR j= 1 to y+1
- do
- there exist a edge E such that $\delta(q_{ix+1},a) \rightarrow q_{ix+1}$
- done
- FOR j= 1 to x+1
- do
- there exist a edge E such that $\delta(q_{y+1i},b) \rightarrow q_{y+1i}$
- done
- DFA "M" will accept all string with equal no of "a" & "b" if all the input is consumed and halting state is the final state.

4. IMPLEMENTATION

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

Let the resultant DFA is M= (Q, $\sum$, 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}}

$\sum$ = {a,b}

δ is given by transaction graph in Figure 1

q_0 = { q_{11}}

F = {Q_{44}}

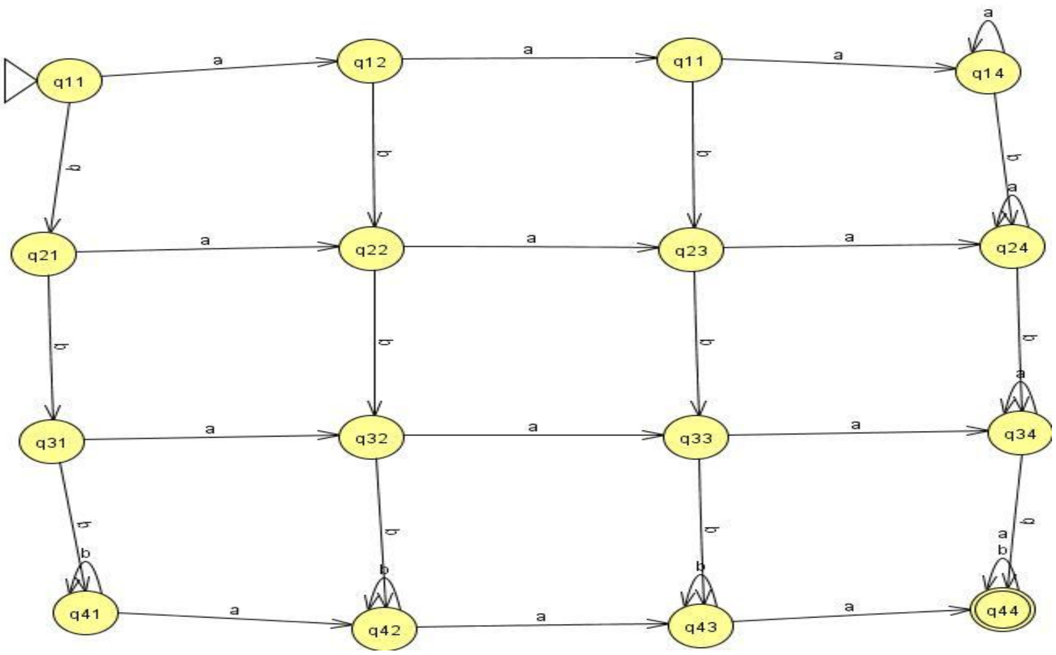


Figure 1 DFA that accepts a string having atleast 3 a's and 3 b's

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

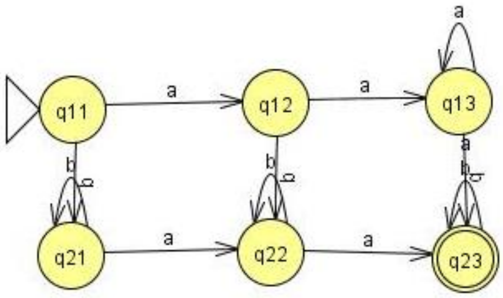


Figure 2 DFA that accepts a string having atleast 2 a's and 1 b's

Design a DFA that accept Strings over Input Symbol a, b having atleast one a & 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}}

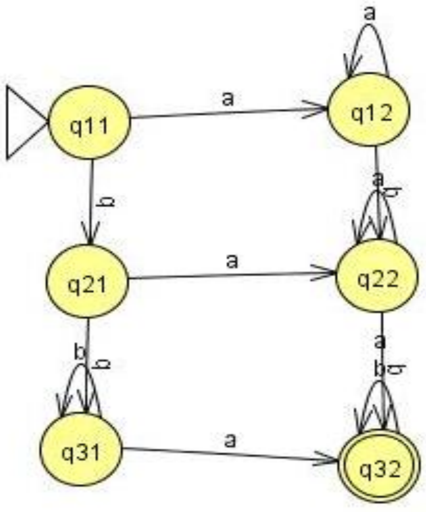
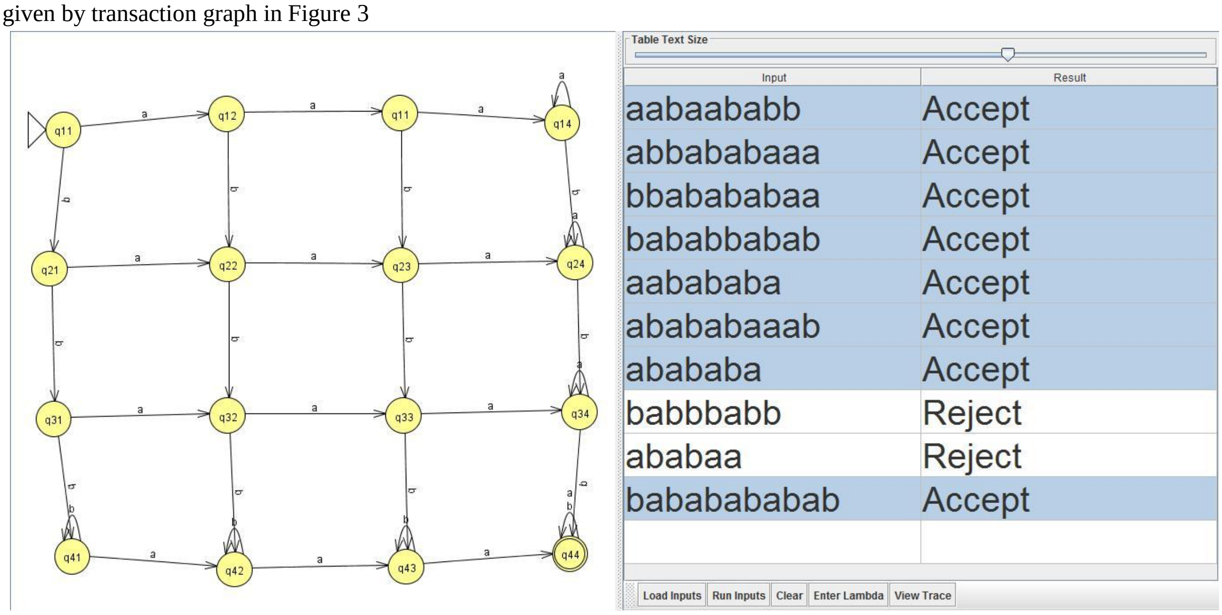


Figure 3 DFA that accepts a string having atleast 1 a's and 2 b's.

5. 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.



Input	Result
aabaababb	Accept
abbababaaa	Accept
bbababaaa	Accept
bababbabab	Accept
aabababa	Accept
abababaaab	Accept
abababa	Accept
babbbabb	Reject
ababaa	Reject
bababababab	Accept

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

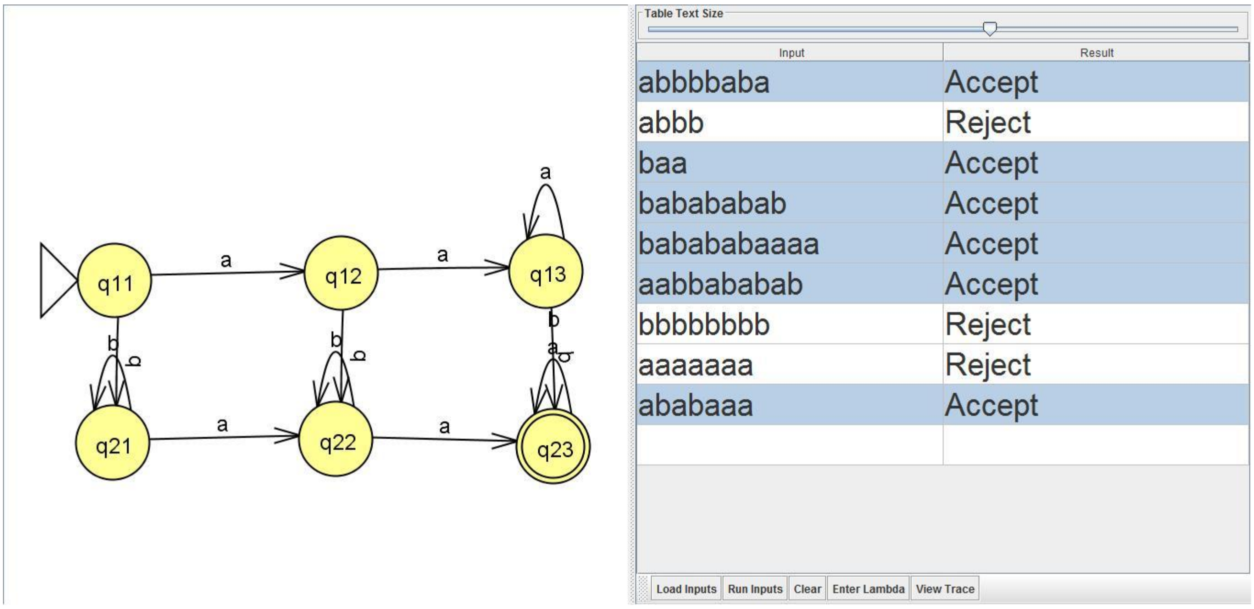


Figure 5 Multiple string test result for atleast 2a's and 1 b's.

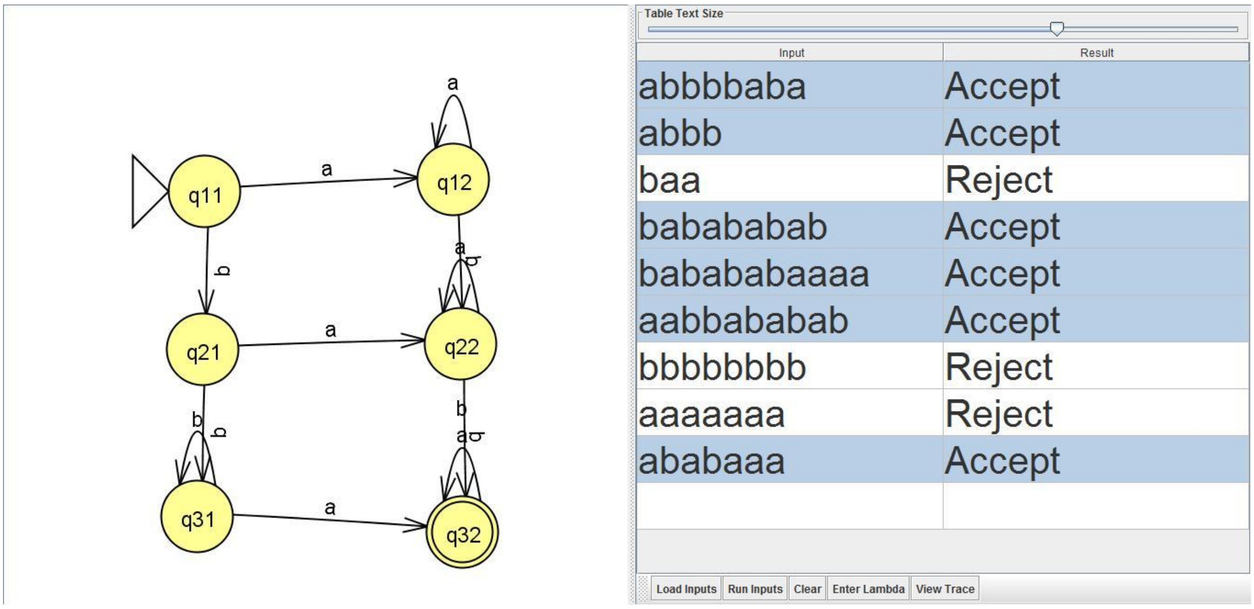


Figure 6 Multiple string test result for atleast 1a's and 2 b's.

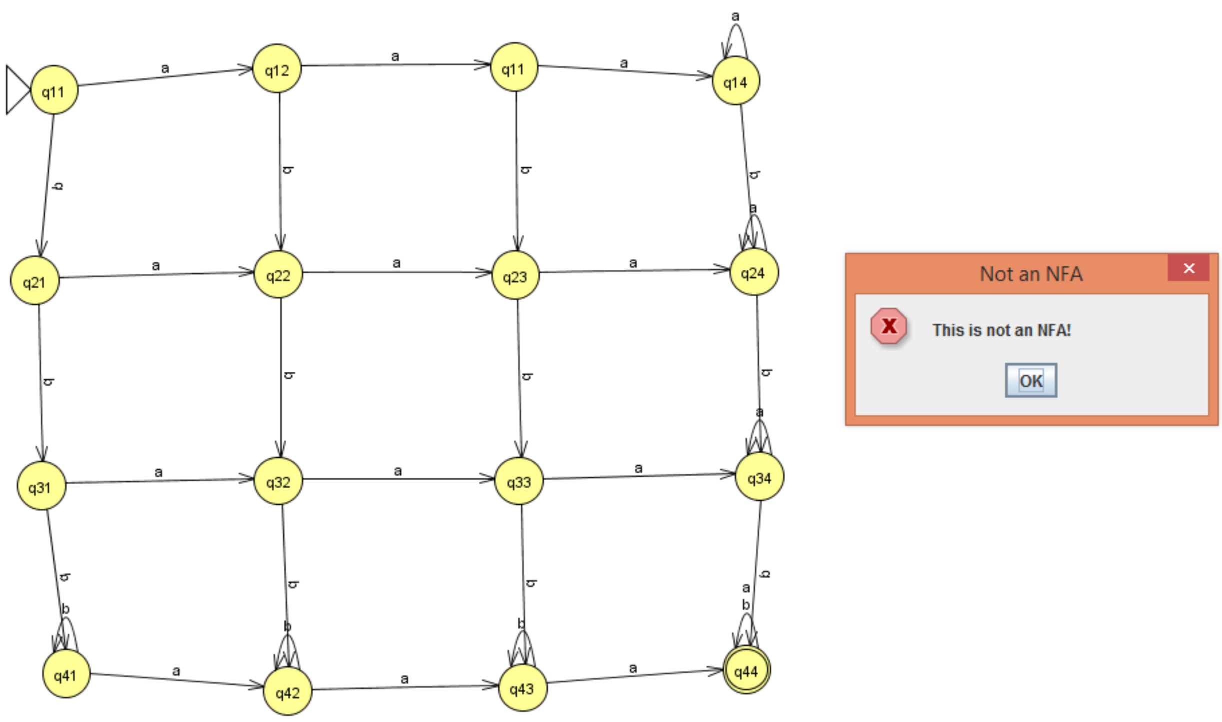


Figure 7 Test for existence of NFA in DFA that accept Strings over Input Symbol a, b having atleast 3 a & 3 b.

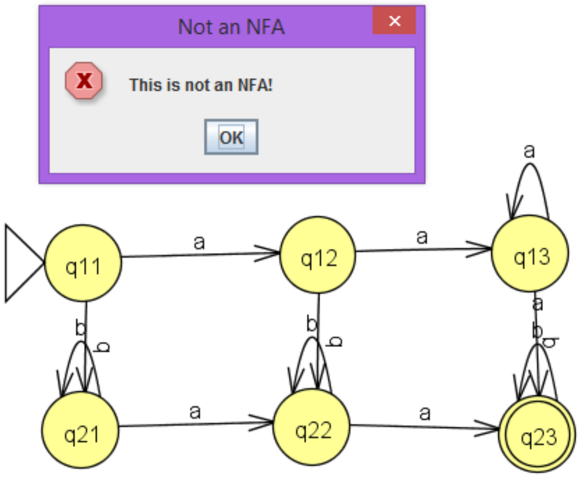


Figure 8 Test for existence of NFA in DFA that accept Strings over Input Symbol a, b having atleast 2 a & 1 b.

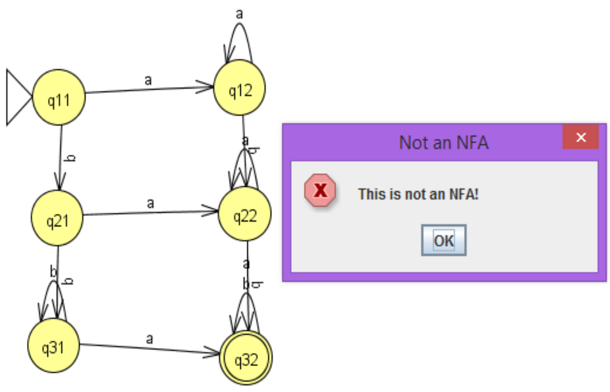


Figure 9 Test for existence of NFA in DFA that accept Strings over Input Symbol a, b having atleast 1 a & 2 b.

6. 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.

7. 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 Teerthnaker Mahaveer University.

8. REFERENCES

[1] 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.

[2] 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

[3] 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

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

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

[6] JFAP tool for simulating results and validation. [Online]. Available: http://www.jflap.org

[7] Deterministic finite automaton [Online] Available at http://www.wikipedia.org/wiki/Deterministic_finite_automaton