A Novel Approach to Classify Nondeterministic Finite Automata Based on More than Two Loops and its Position

Ezhilarasu P[1], Prakash J[2], Krishnaraj N[3], Satheesh Kumar D[4], Sudhakar K[5], Parthasarathy C[6]

Associate Professor, Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, Coimbatore - 641032, India[1.]

Assistant Professor, Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, Coimbatore - 641032, India [2, 4, 5.]

Assistant Professor, Department of Aeronautical Engineering, Hindusthan College of Engineering and Technology, Coimbatore - 641032, India [6.]

Head of the Department, Department of Information Technology, Sree Sastha Institute of Engineering and Technology, Chennai, India[3.]

ABSTRACT: *Generally Finite Automata has 3 positions, particularly beginning state, ending state and intermediate state (except beginning and ending state). Usually we have a tendency to square measure having only one beginning state and one acceptive state. In some cases we have a tendency to square measure having over one acceptive state. By neglecting those cases, we have a tendency to square measure having completely 2 state,s considering beginning and ending state. If we have a tendency to place a loop at those 2 states, the remaining loop is to be placed at intermediate state. Therefore we've got minimum one loop at intermediate state. By inserting one loop, two loops and over 2 loops at intermediate state we have a tendency to get 3 forms of NFA. 1.NFA that contains 2 or more Substring 2. NFA that starts with a substring and contains 2 or more Substring 3. NFA that ends with a substring and contains 2 or more Substring.*

Keywords: *NFA, DFA, Regular Expression, Containing, Starting, Ending.*

I. INTRODUCTION

A. Finite Automaton

A Finite Automaton (FA) is used as a recognizer for accepting or rejecting the input string supported its construction.

This FA might be a mixture of five elements (tuples), those are

• Collection of states.
• Collection of input characters.
• Collection of transition function that comes with a state with an input character as an input and a state as an output.
• a begin state.
• a set of accepting or final states.

B. Categories

Types of FA

1. Nondeterministic Finite Automata (NFA)

2. Deterministic Finite Automata (DFA)

Converting DFA into NFA and the other way around each is feasible.

Ex. Construction of NFA to just accept a string that ends with ‘z’ over alphabet that is shown in Fig. 1.

This downside has following five tuples

- a finite set of states (q_0,q_1)
- a finite set of input symbols referred to as the alphabet (e,z)
- a transition function (q_0,e-> q_0, q_0,z-> q_0, q_1)
- a begin state (q_0).
- a set of accepting states (q_1).

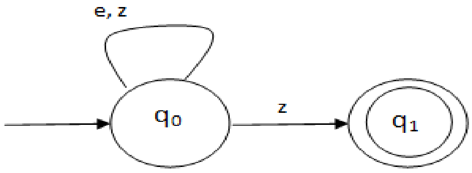

Fig. 1. NFA that accepts any string that ends with z over {e,z}

C. Problem

Regular expression is employed to construct a FA. FA can even be reborn back to Regular expression.

Ex. Regular expression for Fig. 1 is

$(e+z)^*.z$

Based on this regular expression it is attainable to construct decreased DFA, that involves construction of NFA from regular expression using Thompson algorithm, NFA to DFA and DFA to decreased DFA through minimization algorithm. Therefore for construction of DFA from Regular expression desires construction of NFA.

D.Comparison between NFA and DFA.

NFA and DFA may be connected using 2 factors. 1. Transition 2. Construction. In terms of transition NFA has a lot of transition for a given regular expression as compared to DFA. for instance if we have a tendency to take into account A as no input character and R as no of states then most no of transition just in case of DFA could be A * R, for NFA it's $A*R*R(A*R^2)$. Just in case of construction DFA construction is complicated as compared to NFA.

II.RELATED WORK

Ezhilarasu et. al. [2014] classified NFA supported single loop and its position into 3 varieties. Those square measure 1. NFA that accepts the string that starts with explicit Substring. 2. NFA that accepts the string that ends with explicit Substring. 3. NFA that accepts the string that starts with the particular Substring and ends with the particular Substring, as in [1].

III.ABOUT NFA CLASSIFICATION

Based on, as in [2], [3], [4], [5], [6] the non deterministic finite automata can be broadly classified based on more than two loops and its position into three categories, as shown in Fig. 2.

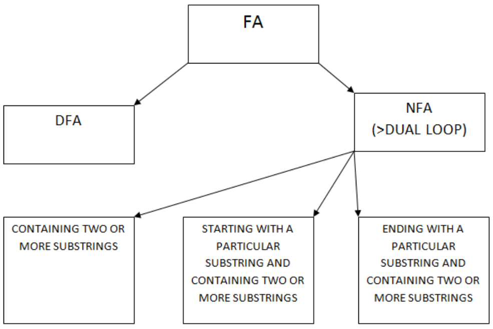
FA
DFA
NFA
(>DUAL LOOP)
CONTAINING TWO OR MORE SUBSTRINGS
STARTING WITH A PARTICULAR SUBSTRING AND CONTAINING TWO OR MORE SUBSTRINGS
ENDING WITH A PARTICULAR SUBSTRING AND CONTAINING TWO OR MORE SUBSTRINGS

Fig. 2. Types of NFA

IV.NFA TYPES

The NFA that contains more than two loops are broadly classified as 1) Containing two or more substrings 2) Starting with a particular substring and containing two or more substrings 3) Ending with a particular substring and containing two or more substrings

A. LOOP AT THE STARTING, INTERMEDIATE AND ENDING STATE

In this type of NFA the loop is present at the starting, intermediate and ending state. That means it can have finite amount of input characters (Substring) between starting and intermediate state, then finite amount of input characters (Substring) between various intermediate states and finite amount of input characters (Substring) between various intermediate and final state. Once it reaches the accepting state it will remain in the same state and it will process the remaining inputs in the accepting state itself, as shown in Fig. 3.

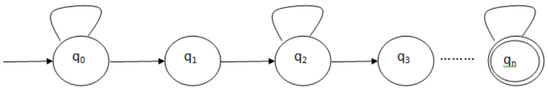

Fig. 3. General form NFA, having loops at starting, intermediate and ending state

GENERAL FORMAT: Self Loop at starting state + Substring + Self Loop at intermediate states + Substring + Self Loop at final state.

Ex. A NFA that accepts a string that contains a first Substring “ez” then Substring “ze” over {e,z}, as shown in Fig. 4.

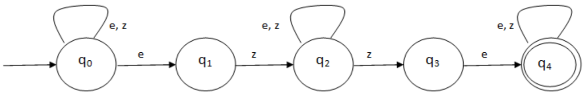

Fig. 4. Containing two or more substrings

B. *LOOP AT THE INTERMEDIATE AND ENDING STATE*

In this type of NFA the loop is present at intermediate and ending state. It can have finite amount of input characters (Substring) between intermediate states and finite amount of input characters (Substring) between the intermediate state and ending states. Once it reaches the accepting state it will remain in the same state, as shown in Fig. 5.

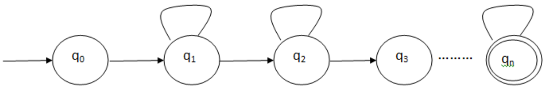

Fig. 5. General form NFA, having loops at intermediate states and final state

.

GENERAL FORMAT: Substring + Self Loop at intermediate states with Substrings between those state + Substring + Self Loop at accepting state.

Ex. A NFA that accepts a string that starts with e and contains a Substring “z” and “ze” over {e,z}, as shown in Fig. 6.

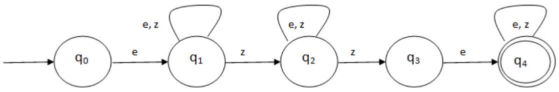

Fig. 6. Starting with a Substring and containing two or more substrings

C. *LOOP AT THE STARTING AND INTERMEDIATE STATE*

In this type of NFA the loop is present at two positions namely 1.Starting state 2. Intermediate states. Initially Substring between starting state and first intermediate state is processed then Substrings between intermediate states are processed, finally Substring between final Intermediate state and accepting state is processed to reach the final state and will remain in the same accepting state, as shown in Fig.7.

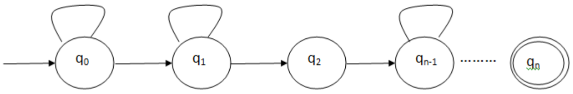

Fig. 7. General form NFA, having loops at Starting and intermediate states

GENERAL FORMAT: Self Loop at starting state + Substring+ Self Loop at Intermediate state with substring between Intermediate states + Selfloop at final intermediate state + Substring .

Ex. A NFA that accepts a string that ends with a substring “ze” and contains a substring “e” “z” over {e,z}, as shown in Fig. 8.

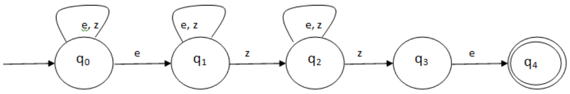

Fig. 8. Ending with a Substring and containing two or more substrings.

V.CONCLUSION

Based on more than two loops and three positions (start, end, intermediate) NFA can be classified into three categories as Two or more containing condition, Two or more containing condition with starting condition and Two or more containing condition with ending condition.

VI.FUTURE WORK

This classification is based on no of loops. This concept may also be implemented using no of substrings.

REFERENCES

[1] Ezhilarasu P, Prakash J, Krishnaraj N, Satheesh Kumar D, Sudhakar K and Dhiyanesh B, “A Novel Approach to Classify Nondeterministic Finite Automata Based on Single Loop and its Position” International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), Volume 1, Issue 4, 2014.

[2] Theory of Computation subject handled by Prof.Dr.P.Ezhilarasu for CSE students Batch 2007-2011, Hindusthan College of Engineering and Technology, Coimbatore.

[3] Formal Languages and Automata Theory subject handled by Prof.Dr.P.Ezhilarasu for CSE students Batch 2008-2012, Hindusthan College of Engineering and Technology, Coimbatore.

[4] Formal Languages and Automata Theory subject handled by Prof.Dr.P.Ezhilarasu for CSE students Batch 2010-2014, Hindusthan College of Engineering and Technology, Coimbatore

[5] Theory of Computation subject handled by Prof.J.Prakash for CSE students Batch 2011-2015, Hindusthan College of Engineering and Technology, Coimbatore.

[6] Theory of Computation subject handled by Prof.J.Prakash for CSE students Batch 2012-2016, Hindusthan College of Engineering and Technology, Coimbatore.