A Novel Approach to Classify Nondeterministic Finite Automata Based on Dual Loop 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 is classified as Nondeterministic Finite Automata and Deterministic Finite Automata. In this paper we classify Nondeterministic Finite Automata based on two factors. 1. Dual loop 2. Position, namely starting state, ending state and intermediate state (except starting and ending state). Using the aforementioned factors, loops are placed at any two positions out of available three positions. As a result we can get three NFA types, namely 1. Containing condition NFA 2. Containing and starting condition NFA 3. Containing and ending condition NFA.**

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

I. INTRODUCTION

A. Finite Automaton

A Finite Automaton (FA) also known as recognizer does any one of two operations, namely accepting or rejecting the given string based on its construction.

This FA is a combination of five tuples (Q, Σ, δ, q0, F)

- a finite set of states (Q)
- a finite set of input characters called the alphabet (Σ)
- a transition function (δ : Q × Σ → Q)
- a start state (q0 ∈ Q)
- a set of accepting states (F ⊆ Q)

B. Categories

Types of FA

1. Nondeterministic Finite Automata (NFA)

2. Deterministic Finite Automata (DFA)

Converting NFA into DFA and vice versa both is possible.

Ex. Construction of NFA to accept a string that ends with 1 over alphabet {0 and 1} which is shown in Fig. 1.

This problem has following five tuples

- a finite set of states (p,q)
- a finite set of input symbols called the alphabet (0,1)
- a transition function (p,0->p, p,1->p,q)
- a start state (p ∈ Q)
- a set of accept states (q ⊆ Q)

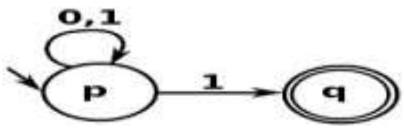

Fig. 1. NFA that accepts any string that ends with 1 over {0,1}

C. Problem

In general the NFA problem involves designing of recognizer for the given string. Strings are generally represented in terms of regular expression.

Ex. Regular expression for Fig. 1 is

$(0+1)^*.1$

Based on this regular expression it is possible to construct minimized DFA, which involves construction of NFA from regular expression using Thompson algorithm, NFA to DFA and DFA to minimized DFA through minimization algorithm. So for construction of DFA from Regular expression needs construction of NFA.

D.Comparison between NFA and DFA.

NFA and DFA can be related using two factors. 1. Transition 2. Construction. In terms of transition NFA is having more no of transition for a given regular expression as compared to DFA. For example if we consider M as no input character and N as no of states then maximum no of transition in case of DFA is M * N, for NFA it is $M*N*N(M*N^2)$. In case of construction DFA construction is complex as compared to NFA.

II. RELATED WORK

Ezhilarasu et. al. [2014] classified NFA based on single loop and its position into three types. Those are 1. NFA that accepts the string which starts with particular Substring 2. NFA that accepts the string that ends with particular string 3.NFA that accepts the string which starts with the particular string and ends with the particular string, 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 dual loop and its position into three categories, as shown in Fig. 2.

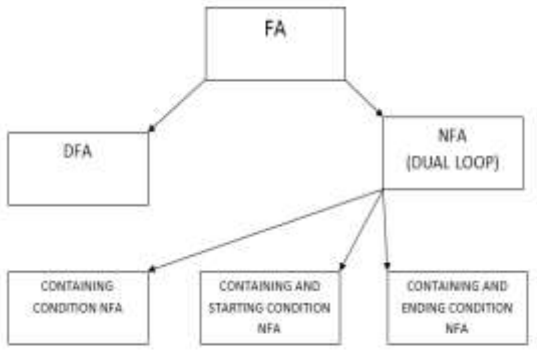

Fig. 2. Types of NFA

IV.NFA TYPES

The NFA that contains dual loops are broadly classified as 1) loop at the starting and ending state, 2) loop at starting and intermediate state 3) loop at ending and intermediate state

A. LOOP AT THE STARTING AND ENDING STATE

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

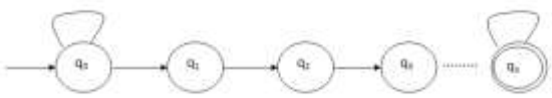

Fig. 3. General form NFA, having loops at starting and ending state

GENERAL FORMAT: Self Loop at starting state + Substring + Self Loop at ending state

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

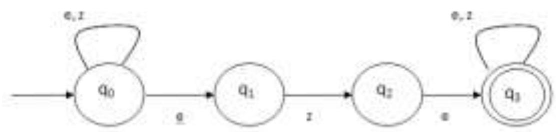

Fig. 4. Containing condition NFA.

B.LOOP AT STARTING AND INTERMEDIATE STATE

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

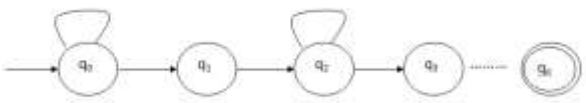

Fig. 5. General form NFA, having loops at starting and intermediate state

.

GENERAL FORMAT: Self Loop at starting state + Substring + Self Loop at intermediate state + Sub String.

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

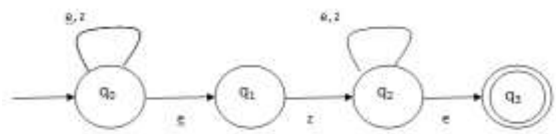

Fig. 6. Containing and Ending condition NFA.

C. LOOP AT ENDING AND INTERMEDIATE STATE

In this type of NFA the loop is present at two positions namely 1. Intermediate state 2. Ending State. Initially Substring with starting condition is processed then string that contains a substring is processed to reach the accepting state. In accepting state it will process any number of remaining input character and will remain in the same accepting state, as shown in Fig. 7.

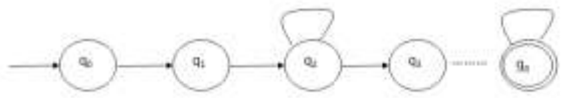

Fig. 7. General form NFA, having loops at Ending and intermediate state

GENERAL FORMAT: Substring + Self Loop at intermediate state + Substring+ Self Loop at Ending state.

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

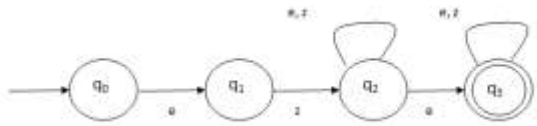

Fig. 8. Starting and Containing condition NFA.

V.CONCLUSION

Based on Double loop and three positions (start, end, between start and end) NFA can be classified into three categories as Containing condition, Containing with starting condition and Containing with ending condition.

VI.FUTURE WORK

This classification further can be extended by using triple or more number of loops at various positions.

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.