ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 1, Issue 4, December 2014

A Novel Approach to Classify Nondeterministic Finite Automata Based on Single Loop and its Position

Ezhilarasu P[1], Prakash J[2], Krishnaraj N[3], Satheesh Kumar D[4], Sudhakar K[5], Dhiyanesh B[6]
Associate Professor, Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, Coimbatore - 641032, India[1.] prof.p.ezhilarasu@gmail.com
Assistant Professor, Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, Coimbatore - 641032, India[2, 4, 5, 6.] jeevaprakash86@gmail.com
Head of the Department, Department of Information Technology, Sree Sastha Institute of Engineering and Technology, Chennai, India[3.] drnkrishnaraj@gmail.com

Abstract: The Finite Automata consists of two major categories, which are NFA and DFA. In this paper we propose a novel idea to classify NFA based on single loop into two major categories 1) Unlooped 2) Looped (Single Loop). The looped NFA is further classified as 1) Loop at the starting position 2) Loop at the ending position 3) Loop between starting and ending position. The Loop at the starting position NFA is termed as NFA that ends with sub string. The Loop at the ending position NFA is termed as NFA that starts with sub string. The Loop between starting and ending position NFA is termed as NFA that starts with sub string and ends with sub string.

Keywords: NFA, DFA, REGULAR EXPRESSION, UNLOOPED, LOOPED.

I. INTRODUCTION

A. Finite Automaton

A Finite Automaton (FA) is a simple idealized machine used to recognize patterns within input taken from some character set (or alphabet) C as shown in Fig. 1. The job of an FA is to accept or reject an input depending on whether the pattern defined by the FA occurs in the input, as in [1].

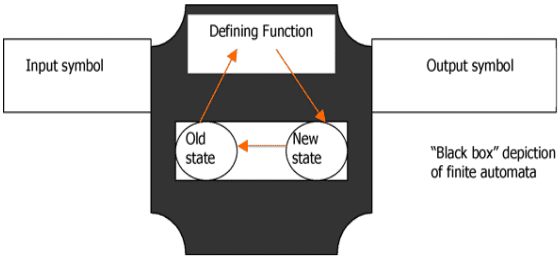

Fig. 1 Finite State Machine

FA is a combination of five tuples, those are

- Set of states (finite).
- Collection of input symbols.
- Transition function between various states through input symbols.
- A start state.
- Final or accepting states.

B. Categories

FA primarily categorized as, as in [2]

1. Nondeterministic Finite Automata (NFA)
2. Deterministic Finite Automata (DFA)

C. Comparison

In case of transition function generally NFA is having more transition function than DFA. Because NFA at a time may make more than one transition function in its

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 1, Issue 4, December 2014

state. But DFA can make maximum one transition function in its state.

D. Problem

The general problem in FA is construction of minimized DFA from the given regular expression, as in [3], [4], [5] and [6].

Construction of NFA is less complex as compared to construction of DFA, as in [7] and [8].

For Example.
Regex: (a)*|(a)*
Regex length: 9
Regex to NFA loops: 4
Number of NFA nodes: 8
NFA to DFA loops: 12
Number of unminimised DFA nodes: 2
Minimize DFA loop: 1
Number of minimized DFA nodes: 2
DFA to regex loops: 5

E. Steps

Steps for converting Regular Expression into minimized DFA, as in [9].

1. Convert Regular expression into epsilon NFA using Thompson algorithm.
2. Convert Epsilon NFA into NFA
3. Convert NFA into DFA.
4. Minimize the obtained DFA using minimization algorithm.

II. Related Works

Juraj Hromkovic et. al. [1997] described a language L(n) by a regular expression of size O(n) such that every NFA accepting L(n) is of size (n log n), as in [10].

Yongzhi Cao and Yoshinori Ezawa [2012] generalized nondeterministic finite automata into fuzzy automata to handle fuzzy uncertainty in system modeling. They categorized nondeterministic fuzzy automata into (deterministic) fuzzy automata, nondeterministic fuzzy automata, and nondeterministic fuzzy automata with ϵ-moves by providing a mathematical representation of nondeterministic dynamic fuzzy systems, as in [11].

Jan Holub and borivoj Melichar [1999] categorized NFA into two types. The first one is the transformation to the equivalent deterministic finite automaton and the second one is the simulation of the run of *NFA*, as in [11].

Henning Bordihn et. al. [2006] defined Extended finite automata as finite state automata equipped with the additional ability to apply an operation on the currently remaining input word, depending on the current state. Hybrid extended finite automata can choose from a finite set of such operations, as in [11].

Hing Leung [2006] defined a structurally unambiguous finite automaton (SUFA) to be a nondeterministic finite automaton (NFA) with one starting state q_0 such that for all input strings w and for any state q, there is at most one path from q_0 to q that consumes w. The definition of SUFA differs from the usual definition of an unambiguous finite automaton (UFA) in that the new definition is defined in terms of the transition logic of the finite automaton, and is independent of the choice of final states, as in [14].

III. About NFA Classification

The non deterministic finite automata can be broadly classified into looped and unlooped NFA as shown in Fig. 2.

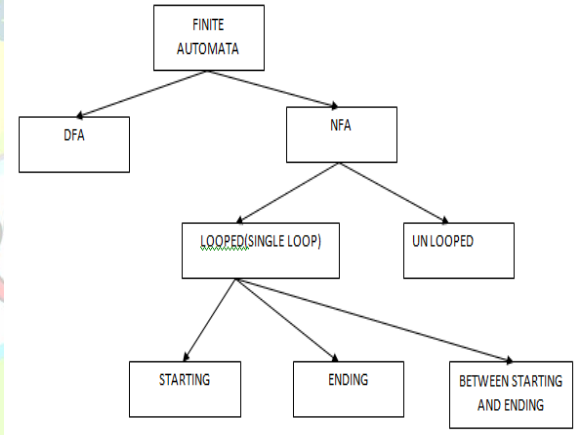

Fig. 2 Types of NFA

IV. Unlooped

In this type, NFA doesn't contain any loop. The input string will end with accepting state without any loop.

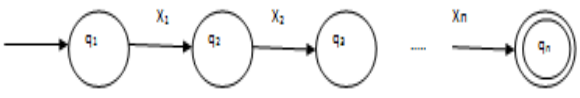

Fig. 3 General form of unlooped NFA

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 1, Issue 4, December 2014

GENERAL FORMAT: Sub string with No loop as shown in Fig. 3.
Ex. A NFA that accepts a string aba over {a,b}.

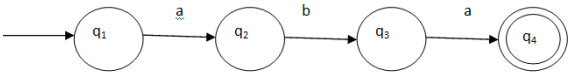

Fig. 4 Example for General form of unlooped NFA

V. LOOPED

The NFA that contains loops are broadly classified as 1) Loop at the end, 2) Loop between start and end, 3) Loop at the start.

A. Loop at the End

In this type of NFA the loop is present at the ending state. That means it can have any number of input character before the accepting state. Once it reached the accepting state it will remain in same state and it will process the remaining inputs in the accepting state itself.

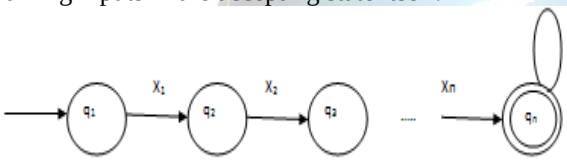

Fig. 5 General form of looped NFA, loop at the end

GENERAL FORMAT: Sub string + Self loop at the end position as shown in Fig. 5.
Ex. A NFA that accepts a string that starts with a sub string aba over {a,b}.

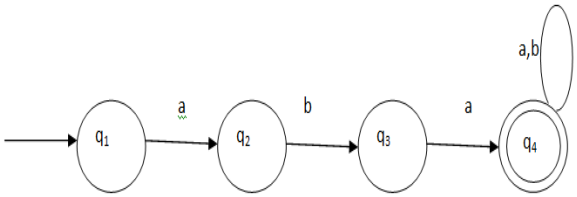

Fig. 6. Example for General form of looped NFA, loop at the end.

B.Loop Between Start And End

In this type of NFA the loop is present in between the starting state and ending state of the string. That means it can have any number of input character before the loop get started and it will process any number of input character within same state and finally it will process some finite number of input string to reach accepting state. Once it reached the accepting state it will remain in same state.

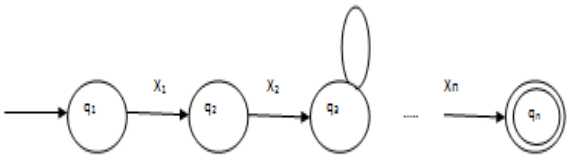

Fig. 7 General form of looped NFA, loop between start and end of NFA

GENERAL FORMAT: Sub string + Self loop + Sub string as shown in Fig. 7.
Ex. A NFA that accepts a string that starts with a sub string ab and ends with a sub string a over {a,b}.

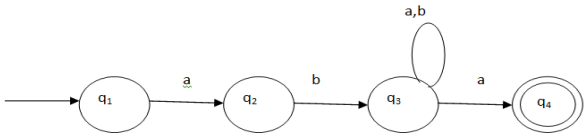

Fig. 8 Example for General form of looped NFA, loop between start and end of NFA

C.Loop at the Start

In this type of NFA the loop is present at the starting state. That means it can have any number of input character at starting state. After start state it will process some finite number of input character then it will reach accepting state.

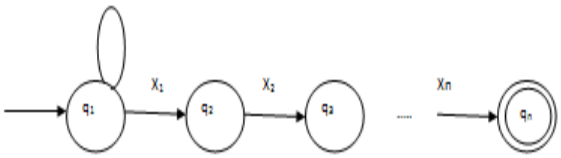

Fig. 9 General form of looped NFA, loop at the start

GENERAL FORMAT: Self loop at the starting position + Sub string as shown in Fig. 9.
Ex. A NFA that accepts a string that ends with a sub string aba over {a,b}.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 1, Issue 4, December 2014

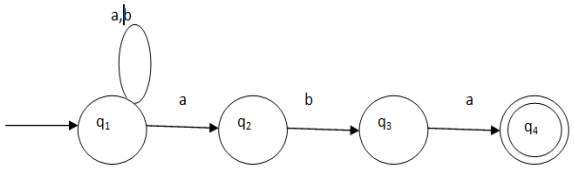

Fig. 10 Example for General form of looped NFA, loop at the start

VI. CONCLUSION

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

VII. FUTURE WORK

This classification further can be extended by using double loop at various positions.

REFERENCE

[1]. http://www.cs.rochester.edu/~nelson/courses/csc_173/fa/fa.html

[2]. https://www.cs.umd.edu/class/fall2014/cmsc330/lectures/05-automata2.pdf

[3]. Berry, G. and R. Sethi [1986]. "From regular expressions to deterministic automata". Theoretical Computer Science. vol. 48, no. 1, pp. 117-126.

[4]. Bruggemann-Klein A. [1993]. "Regular expressions into finite automata". Theoretical Computer Science. vol. 120, no. 2, pp. 197-213.

[5]. Chang, C. H. and R. Paige [1992]. "From regular expressions to DFAs using compressed NFAs". In Proceedings of the 3rd Annual Symposium on Combinatorial Pattern Matching. Lecture notes in Computer Science no. 644. Springer-Verlag, London. pp. 90-110.

[6]. Hopcroft, J. E. and J. Ullman [1979]. Introduction to Automata Theory, Languages and Computation. Addison-Wesley Longman Publishing Company, Inc. Boston, MA, USA.

[7]. http://regex-automaton.com/complexity.php

[8]. http://www.cs.uml.edu/~giam/91.304/Lectures/1-2.pdf

[9]. Introduction to Automata Theory, Languages, and Computation John E. Hopcroft , Rajeev Motwani Jeffrey D. Ullman Prentice Hall; 3 edition (29 June 2006).

[10]. Juraj Hromkovic , Sebastian Seibert and Thomas Wilke [1997]. " translating regular expression into small epsilon-free non deterministic Automata" STACS '97 Proceedings of the 14th Annual Symposium on Theoretical Aspects of Computer Science Pages 55-66.

[11]. Yongzhi Cao and Yoshinori Ezawa [2012]. "Nondeterministic fuzzy automata", Information Sciences, Volume 191, 15 May 2012, Pages 86–97.

[12]. Jan Holub and borivoj Melichar [1999]. "Implementation of Nondeterministic Finite Automata for Approximate Pattern Matching", Automata Implementation Lecture Notes in Computer Science Volume 1660, 1999, pp 92-99.

[13]. Henning Bordihn, Markus Holzer and Martin Kutrib [2006]. Hybrid Extended Finite Automata implementation and Application of Automata Lecture Notes in Computer Science Volume 4094, 2006, pp 34-45.

[14]. Hing Leung [2006]. "Structurally Unambiguous Finite Automata", implementation and Application of Automata Lecture Notes in Computer Science Volume 4094, 2006, pp 198-207.