ISSN:0975-9646

A Novel Approach to Design Optimized Deterministic Finite Automata to Accept the Palindrome using Three Input Characters

Abhishek Bhardwaj[#1], Achint Chaudhary[#1], Shankar Z. Thawkar[*2], Jhalak[*2], Vijay S. Katta[#3]

[1-3]*Department of Computer Science & Information Technology,*

Hindustan College of Science & Technology, Mathura-281122, Uttar Pradesh, India

***Abstract*— In this paper we will construct optimized deterministic finite automata to accept palindrome string from the string length one to five. First we will construct a Regular Expression for palindrome strings till maximum length five. The possible strings will be $3^0+3^1+3^2+\ldots\ldots+3^n$. Here 3 represents input character and n denotes maximum length of the string. We consider the value of 'n' as 5. The possible strings of palindrome for regular expression can be drawn by the formula $2*3^1+2*3^2+1*3^3$. Regular expression can be converted into equivalent non deterministic finite automata. By subset construction we can construct the DFA and the DFA can be optimized by tabulation method. In optimized DFA we found only 45 total numbers of states to accept any string of palindrome till maximum length five. This method shows the extended power of Deterministic Finite Automata as compared with the Turing Machine.**

***Keywords*— Regular Expression, Finite Automata, Deterministic Finite Automata, String of input characters, Optimized DFA by tabulation, Palindrome, String Length, Turing Machine**

I. INTRODUCTION

We can define computation in Theory of Automata through different machines. The most popular machines used are Finite Automata. Finite Automata can be deterministic or non deterministic. Pushdown Automata are more powerful machines as compare to Finite Automata and most powerful is Turing Machine. In first two machines the head on tape can move only in one direction i.e. left to right while in Turing Machine the Head can move in both the directions i.e. it can read the characters either from left to right or right to left. Turing Machine can perform the operations which might not be performed in Finite Automata and Pushdown Automata.

The string of characters is called a palindrome if a string 'w' with a reverse string 'w^R ' is taken together as ww^R or wCw^R . Where 'C' is a middle character for odd palindrome. So, to check the given string whether it is a palindrome or not we check the first character from left side and right most character of reverse string. If both are same, then we have to check second left most character of original string with second rightmost character of reverse string. This step is repeated n/2 times. Where n is the length of string. e.g. for the string "nitin" the length of the string is 5. So, 5/2=2.5 then we can leave the fractional part and consider the value as 2.

The left most characters 'n' & 'i' are reversed after 't' to make the palindrome string. So, first leftmost character n will be matched with first rightmost character of reversed string i.e. also n. Then second leftmost character i is matched with the second rightmost character 'i'. Both are same. The middle character t makes the string as odd palindrome as the result of step 1 to step 2 is true. Hence we can find the given string as palindrome. If in any case we do not find the desired result then the given string is not a palindrome. [1]

II. FOUNDATION OF OUR METHOD

A. Regular Expression (R. E.)

Just as finite automata are used to recognize patterns of strings, regular expressions are used to generate patterns of strings. A regular expression is an algebraic formula whose value is a pattern consisting of a set of strings, called the language of the expression. [6]

Operands in a regular expression can be:

- Characters from the alphabet over which the regular expression is defined.
- Variables whose values are any pattern defined by a regular expression.
- Epsilon which denotes the empty string containing no characters.
- Null which denotes the empty set of strings.

B. Finite Automata (FA)

A machine which can accept the input string if it starts from initial state and reaches to some final state or reject the input string if it starts from initial state and stops at some non-final state.

A FA has five tuples. FA Machine M can be described as M= (Q, Σ, q0, F, δ)

- (Q) Denotes finite non empty set of states.
- (Σ) is called a finite non empty set of input alphabets.
- ($q_0 \in Q$) is the initial state.
- ($F \subseteq Q$) is the set of final state(s).
- (δ) is called the next state function.

$Q \times \Sigma \rightarrow Q$ in case of Deterministic Finite Automata

$Q \times \Sigma \rightarrow 2^Q$ in case of Non-deterministic Finite Automata

C. Deterministic Finite Automata (DFA)

A DFA is deterministic machine which have unique transitions on each state of the machine. It is called a deterministic acceptor as it can accept the input string starting from initial state and going through unique transitions to some accepting state. Unique transition means a DFA can have one input one transition from each state.

The transition function of Finite Automata can be given by: $Q \times \Sigma \rightarrow Q$

D. Turing Machine

A Turing machine, denoted TM, is a collection of six things:

1. An alphabet Σ of input letters.
2. A TAPE divided into a sequence of numbered cells each containing one character or a blank.
3. A TAPE HEAD that can in one step read the contents of a cell on the TAPE, replace it with some other character, and reposition itself to the next cell to the right or to the left of the one it has just read.
4. An alphabet, τ, of characters that can be printed on the TAPE by the TAPE HEAD.
5. A finite set of states.
6. A program, which is a set of rules that tell us, on the basis of the letter the TAPE HEAD has just read, how to change states, what to print and where to move the TAPE HEAD.

(letter, letter, direction)

In TM the head can move in both directions which is not possible in Finite Automata.

TABLE I
Logic in TM to Accept the Strings of Palindrome

S. No.	First Character	Replaced Character	Last Character	Replaced Character
1	0	Δ (Blank)	0	Δ (Blank)
2	0	Δ (Blank)	1	Reject
3	1	Δ (Blank)	0	Reject
4	1	Δ (Blank)	1	Δ (Blank)

Danial I. A. Chohen [5] constructed a Turing Machine to accept the string that is palindrome [5]. The TM accept both the odd length palindrome and even length palindromes. He used the two characters for forming the string. In two input characters, we need to check n*n combinations. Where n represents a total number of input characters. Therefore he used 2*2 = 4 combinations. Those are as shown in TABLE I.

III. PROPOSED METHOD

A. Regular Expression for Palindrome string (Maximum Length 5)

0|1|2|00|11|22|000|010|020|101|111|121|202|212|222|0000|0110|0220|1001|1111|1221|2002|2112|2222|00000|00100|00200|01010|01110|01210|02020|02120|02220|10001|10101|10201|11011|11111|11211|12021|12121|12221|20002|20102|20202|21012|21112|21212|22022|22122|22222

The R. E. consist total $2*3^1+2*3^2+1*3^3=51$ combinations of palindrome strings for 3 input characters 0, 1, 2. '|' denotes 'or' operation. The combinations of palindrome are shown in TABLE II.

TABLE II
Possible Combinations of Palindrome with 3 Input Characters up to String Length Five

String Length	Possible Combinations	Example
1	$3^1=3$	0, 1, 2
2	$3^1=3$	00, 11, 22
3	$3^2=9$	000, 010, 020, ……
4	$3^2=9$	0000, 0110, 0220, ….
5	$3^3=27$	00000, 00100, 00200, ……

B. Construction of NFA with empty (λ) moves by Regular Expression of Palindrome

A nondeterministic finite automaton with ε-moves (NFA-ε) or e-transition (also known as NFA-λ) is an extension of nondeterministic finite automaton (NFA), which allows a transformation to a new state without consuming any input symbols. The transitions without consuming an input symbol are called ε-transitions or λ-transitions. In the state diagrams, they are usually labeled with the Greek letter ε or λ. [1], [7]

C. Converting NFA of Palindrome with empty (λ) moves into NFA without empty (λ) moves

For an NFA with ε-moves (also called an ε-NFA), the construction must be modified to deal with these by computing the ε-closure of states: the set of all states reachable from some given state using only ε-moves. Van Noord recognizes three possible ways of incorporating this closure computation in the power set construction. [9]

D. Converting Nondeterministic FA of Palindrome into Deterministic FA

For every NDFA, there exists a DFA which simulates the behavior of NDFA. Alternatively, if L is the set accepted by NDFA, then there exists a DFA which also accepts L. [8]

E. Optimization of Deterministic Finite Automata of Palindrome

A DFA can be optimized by tabulation method. The optimized DFA have minimum number of states in DFA machine. [8]

F. Palindrome Transition Table (Optimized DFA)

The DFA which accepts palindrome for three input characters and string length, one to exactly five is shown in transition TABLE III.

TABLE III
Transition Table for Minimized DFA for Palindrome Strings

DFA's State	Input Characters		
	0	1	2
→[q_0]	q_1	q_2	q_3
[q_1]*	q_4	q_5	q_6
[q_2]*	q_7	q_8	q_9
[q_3]*	q_{10}	q_{11}	q_{12}
[q_4]*	q_{13}	q_{14}	q_{14}
[q_5]	q_{15}	q_{16}	q_{17}
[q_6]	q_{18}	q_{19}	q_{20}
[q_7]	q_{21}	q_{22}	q_{23}
[q_8]*	q_{24}	q_{25}	q_{24}
[q_9]	q_{26}	q_{27}	q_{28}
[q_{10}]	q_{29}	q_{30}	q_{31}
[q_{11}]	q_{32}	q_{33}	q_{34}
[q_{12}]*	q_{35}	q_{35}	q_{36}
[q_{13}]*	q_{37}	q_{44}	q_{44}
[q_{14}]	q_{38}	q_{44}	q_{44}
[q_{15}]*	q_{44}	q_{38}	q_{44}
[q_{16}]	q_{39}	q_{38}	q_{44}
[q_{17}]	q_{44}	q_{38}	q_{44}
[q_{18}]*	q_{44}	q_{44}	q_{38}
[q_{19}]	q_{44}	q_{44}	q_{38}
[q_{20}]	q_{39}	q_{44}	q_{38}
[q_{21}]	q_{40}	q_{39}	q_{44}
[q_{22}]*	q_{40}	q_{44}	q_{44}
[q_{23}]	q_{40}	q_{44}	q_{44}
[q_{24}]	q_{44}	q_{40}	q_{44}
[q_{25}]*	q_{44}	q_{41}	q_{44}
[q_{26}]	q_{44}	q_{44}	q_{40}
[q_{27}]*	q_{44}	q_{44}	q_{40}
[q_{28}]	q_{44}	q_{39}	q_{40}
[q_{29}]	q_{42}	q_{44}	q_{39}
[q_{30}]	q_{42}	q_{44}	q_{44}
[q_{31}]*	q_{42}	q_{44}	q_{44}
[q_{32}]	q_{44}	q_{42}	q_{44}
[q_{33}]	q_{44}	q_{42}	q_{39}
[q_{34}]*	q_{44}	q_{42}	q_{44}
[q_{35}]	q_{44}	q_{44}	q_{42}
[q_{36}]*	q_{44}	q_{44}	q_{43}
[q_{37}]*	q_{39}	q_{44}	q_{44}
[q_{38}]	q_{39}	q_{44}	q_{44}
[q_{39}]*	q_{44}	q_{44}	q_{44}
[q_{40}]	q_{44}	q_{39}	q_{44}
[q_{41}]*	q_{44}	q_{39}	q_{44}
[q_{42}]	q_{44}	q_{44}	q_{39}
[q_{43}]*	q_{44}	q_{44}	q_{39}
[q_{44}]	q_{44}	q_{44}	q_{44}

The DFA, which accepts palindrome using three input characters and maximum string length, has 45 states and 19 final states shown in TABLE III. [q_0] is initial state of the palindrome machine while * marked states are all final states.

TABLE IV
Final States of DFA with Accepted Strings from Length One to Five

S. No.	String Length	Final State	Accepted Strings
1	1	[q_1]*	0
2	1	[q_2]*	1
3	1	[q_3]*	2
4	2	[q_4]*	00
5	2	[q_8]*	11
6	2	[q_{12}]*	22
7	3	[q_{13}]*	000
8	3	[q_{15}]*	010
9	3	[q_{18}]*	020
10	3	[q_{22}]*	101
11	3	[q_{25}]*	111
12	3	[q_{27}]*	121
13	3	[q_{31}]*	202
14	3	[q_{34}]*	212
15	3	[q_{36}]*	222
16	4	[q_{37}]*	0000
17	4, 5	[q_{39}]*	0110, 0220, 1001, 1221, 2002,2112, 00000, 00100, 00200, 01010, 01110, 01210, 02020, 02120, 02220, 10001, 10101, 10201, 11011, 11111, 11211, 12021, 12121, 12221, 20002, 20102, 20202, 21012, 21112, 21212, 22022, 22122, 22222
18	4	[q_{41}]*	1111
19	4	[q_{43}]*	2222

From the above TABLE IV we find total 19 final states. State number q_{39} accept length 4 as well as length 5 palindrome strings.

IV. CONCLUSION

The Turing machine accept palindrome strings with- out any restriction on string length as well as the result of accepting the string is fast as compare to DFA constructed. Constructed DFA will be faster than NFA constructed in the previous work [1] and the number of states in DFA is optimized i.e. only 45 total states and 19 final states accepting palindrome strings length from one to five.

REFERENCES

[1] P. Ezhilarasu, J. Prakash, N. Krishnaraj, D. Satheesh Kumar, K. Suresh Babu and C. ParthasarathyA Novel Approach to Design the Finite Automata to Accept the Palindrome with the Three Input Characters. Indian Journal of Science and Technology, Vol 8(28), DOI 10.17485/ijst/2015/v8i28/78189, October 2015.

[2] Ezhilarasu P, Thirunavukkarasu E, Karuppusami G, Krishnaraj N. Single substring based classification for nondeterministic finite automata. International Journal on Applications in Information and Communication Engineering. 2015; 1(10)29–31.

[3] Ezhilarasu P, Krishnaraj N. Double Substring Based Classification for Nondeterministic Finite Automata. International Conference on Recent Advances in Engineering, Science and Technology – ICON'15. Noorul Islam University, 2015. p. 104–7.

[4] Available from https//en.wikipedia.org/wiki/ Turing_machine.

[5] DANIAL I. A. COHEN Introduction to Computer Theory. 2nd ed. Willey; 2009.

[6] https//www.cs.rochester.edu/~nelson/courses/csc_173/fa/re.html

[7] https//en.wikipedia.org/wiki/Nondeterministic_finite_automaton_with_%CE%B5-moves

[8] K. L. P. Mishra THEORY OF COMPUTER SCIENCE Automata, Languages and Computation. 3rd Edition Prentice'Hall of India; 2008.

[9] Van Noord, Gertjan (2000). "Treatment of epsilon moves in subset construction". Computational Linguistics 26 (1) 61–76. doi10.1162/089120100561638