Indian Journal of Science and Technology, *Vol 8(28), DOI: 10.17485/ijst/2015/v8i28/78189, October 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Novel Approach to Design the Finite Automata to Accept the Palindrome with the Three Input Characters

P. Ezhilarasu[1], J. Prakash[1], N. Krishnaraj[2], D. Satheesh Kumar[1], K. Suresh Babu[3] and C. Parthasarathy[4]

[1]Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, Coimbatore - 641032, Tamil Nadu, India; prof.p.ezhilarasu@gmail.com, jeevaprakash86@gmail.com, dsatheeshme@gmail.com
[2]Department of Information Technology, Valliammai Engineering College, Kattankulathur, Chennai - 603203, Tamil Nadu, India; drnkrishnaraj@gmail.com
[3]Department of Electronics and Communication Engineering, Hindusthan Institute of Technology, Coimbatore - 641032, Tamil Nadu, India; ammaskhicet@gmail.com
[4]Department of Mechatronics Engineering, Hindusthan College of Engineering and Technology, Coimbatore – 641032, Tamil Nadu, India; aruncps3000@gmail.com

Abstract

Background/Objectives: In this paper we discuss that, how Finite Automata can accept the palindrome statically. **Methods/Statistical Analysis:** The formula $3^0+3^1+3^2+...+3^n$ used to derive the possible strings. Where 3 represents input character and n represents maximum length of the string. Here the value of n taken as 5. **Findings:** The formula $2*3^1+2*3^2+1*3^3$ used to derive palindrome from the possible strings. **Application/Improvements:** This method shows the extended use of Finite Automata as compared with the Turing Machine.

Keywords: Finite Automata, Input Characters, Palindrome, String Length, Turing Machine

1. Introduction

Theory of computation defined as computation through the abstract machines. The three primary abstract machines used are Finite Automata Pushdown Automata and Turing Machine. The first two abstract machines moves only in unidirectional. i.e reading strings from the left most to the right most. They cannot perform the movement from the right to left. To overcome this problem the Turing machine used. It can perform the operations that are difficult (or) impossible in Finite Automata and Pushdown Automata.

The string called as palindrome if it is same in both forward and backward direction. For checking whether the given input string is the palindrome (or) not we need to check the first leftmost and right most characters. If both are same, then we must check second leftmost and rightmost characters. This step implemented for n/2 times. Where n stands for string length. Ex. For the string "racecar" the string length is seven. 7/2=3.5 then removing the fractional part we will get the value as 3. The first leftmost character r checked with first rightmost character r. Both are same. Then second leftmost character a checked with the first rightmost character r. Both

**Author for correspondence*

are same. Finally, third leftmost character c checked with first rightmost character c. Both are same. The value three reached. The result from step 1 to step 3 is true. Hence, the given string is the palindrome. If any case the result is false then the given string is not a palindrome.

1.1 Finite Automaton

The recognizer that accepts or rejects the input string called as a Finite Automaton (FA). The construction of Finite Automata based on the condition[1–3].

This FA is a quintuple. If FA represented as M then M= (Q, Σ, δ, q0, F).

- Collection of states (Q).
- A finite set of input symbols called the alphabet (Σ).
- A transition function ($\delta : Q \times \Sigma \rightarrow Q$).
- A starting state ($q0 \in Q$).
- A set of accepting state(s) ($F \subseteq Q$).

1.2 Turing Machine

The Turing Machine defined as a hypothetical device used to represent a computing machine[4]. It helps us to know the limitation of mechanical computation. It has seven tuples. Hence called as septuplet. It consists of

- A tape.
- A head.
- A state register.
- A finite table.

The head can move in both directions. This supports, the operations those are not possible in Finite Automata and Pushdown Automata.

Table 1. Replacement logic in the Turing Machine that accepts the palindrome

S. No.	First Character	Replaced Character	Last Character	Replaced Character
1	a	Blank	a	Blank
2	a	Blank	b	No replacement. Reach the halting state for rejection.
3	b	Blank	a	No replacement. Reach the halting state for rejection.
4	b	Blank	b	Blank

John C. Martin[5] constructed a Turing Machine to accept the string that is palindrome[5]. The Turing Machine accepted both the odd length palindrome and even length palindrome. 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 1.

2. Construction of Finite Automata

In this work totally three input characters are used. Those are 'u' 'm' 'a'. The maximum string length restricted to five. The possible combinations of strings given in Table 2.

Table 2. Possible combination of string using three input characters up to string length five

S. No.	String Length	Possible Combination	Example
1	1	3^1=3	"u","m","a"
2	2	3^2=9	"uu", "um", "ua" …
3	3	3^3=27	"uuu", "uum", "uua" …
4	4	3^4=81	"uuuu", "uuum", "uuua" …
5	5	3^5=243	"uuuuu", "uuuum", "uuuua" …

The possible combinations of palindrome given in Table 3.

Table 3. Possible combination of palindrome using three input characters up to string length five

S. No.	String Length	Possible Combination	Example
1	1	3^1=3	"u","m","a"
2	2	3^1=3	"uu", "mm", "aa"
3	3	3^2=9	"uuu", "umu", "uau" …
4	4	3^2=9	"uuuu", "ummu", "uaau" …
5	5	3^3=27	"uuuuu", "uumuu", "uuauu" …

3. Palindrome Derivation

The Finite Automata, which accepts palindrome for three input characters and string length, exactly five is shown

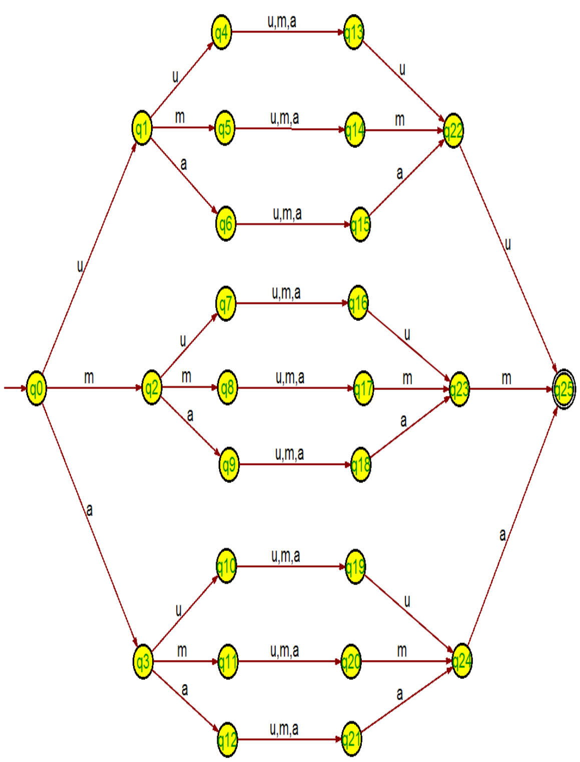

Figure 1. Transition Diagram - Finite Automata for accepting the string of length five exactly which is also palindrome using input characters 'u', 'm', 'a'.

in Figure 1. It accepts 27 possible palindromes out of 243 possible strings. It is possible to construct a Finite Automata that accepts all possible palindrome ranging from string length one to string length five. For that we need totally $3^0+3^1+3^2+3^3+3^4+3^5 = 1+3+9+27+81+243=364$ states as shown in Table 4.

Table 4. Collection of state for getting the string with length one to five

States	Input Characters		
	u	m	a
$\rightarrow q_0$	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_{15}
q_5	q_{16}	q_{17}	q_{18}
q_6	q_{19}	q_{20}	q_{21}
q_7	q_{22}	q_{23}	q_{24}
$^*q_8$	q_{25}	q_{26}	q_{27}
q_9	q_{28}	q_{29}	q_{30}
q_{10}	q_{31}	q_{32}	q_{33}
q_{11}	q_{34}	q_{35}	q_{36}
$^*q_{12}$	q_{37}	q_{38}	q_{39}
$^*q_{13}$	q_{40}	q_{41}	q_{42}
q_{14}	q_{43}	q_{44}	q_{45}
q_{15}	q_{46}	q_{47}	q_{48}
$^*q_{16}$	q_{49}	q_{50}	q_{51}
q_{17}	q_{52}	q_{53}	q_{54}
q_{18}	q_{55}	q_{56}	q_{57}
$^*q_{19}$	q_{58}	q_{59}	q_{60}
q_{20}	q_{61}	q_{62}	q_{63}
q_{21}	q_{64}	q_{65}	q_{66}
q_{22}	q_{67}	q_{68}	q_{69}
$^*q_{23}$	q_{70}	q_{71}	q_{72}
q_{24}	q_{73}	q_{74}	q_{75}
q_{25}	q_{76}	q_{77}	q_{78}
$^*q_{26}$	q_{79}	q_{80}	q_{81}
q_{27}	q_{82}	q_{83}	q_{84}
q_{28}	q_{85}	q_{86}	q_{87}
$^*q_{29}$	q_{88}	q_{89}	q_{90}
q_{30}	q_{91}	q_{92}	q_{93}
q_{31}	q_{94}	q_{95}	q_{96}
q_{32}	q_{97}	q_{98}	q_{99}
$^*q_{33}$	q_{100}	q_{101}	q_{102}
q_{34}	q_{103}	q_{104}	q_{105}
q_{35}	q_{106}	q_{107}	q_{108}
$^*q_{36}$	q_{109}	q_{110}	q_{111}
q_{37}	q_{112}	q_{113}	q_{114}
q_{38}	q_{115}	q_{116}	q_{117}
$^*q_{39}$	q_{118}	q_{119}	q_{120}
$^*q_{40}$	q_{121}	q_{122}	q_{123}
q_{41}	q_{124}	q_{125}	q_{126}
q_{42}	q_{127}	q_{128}	q_{129}
q_{43}	q_{130}	q_{131}	q_{132}
q_{44}	q_{133}	q_{134}	q_{135}
q_{45}	q_{136}	q_{137}	q_{138}

q_{46}	q_{139}	q_{140}	q_{141}
q_{47}	q_{142}	q_{143}	q_{144}
q_{48}	q_{145}	q_{146}	q_{147}
q_{49}	q_{148}	q_{149}	q_{150}
q_{50}	q_{151}	q_{152}	q_{153}
q_{51}	q_{154}	q_{155}	q_{156}
$^*q_{52}$	q_{157}	q_{158}	q_{159}
q_{53}	q_{160}	q_{161}	q_{162}
q_{54}	q_{163}	q_{164}	q_{165}
q_{55}	q_{166}	q_{167}	q_{168}
q_{56}	q_{169}	q_{170}	q_{171}
q_{57}	q_{172}	q_{173}	q_{174}
q_{58}	q_{175}	q_{176}	q_{177}
q_{59}	q_{178}	q_{179}	q_{180}
q_{60}	q_{181}	q_{182}	q_{183}
q_{61}	q_{184}	q_{185}	q_{186}
q_{62}	q_{187}	q_{188}	q_{189}
q_{63}	q_{190}	q_{191}	q_{192}
$^*q_{64}$	q_{193}	q_{194}	q_{195}
q_{65}	q_{196}	q_{197}	q_{198}
q_{66}	q_{199}	q_{200}	q_{201}
q_{67}	q_{202}	q_{203}	q_{204}
$^*q_{68}$	q_{205}	q_{206}	q_{207}
q_{69}	q_{208}	q_{209}	q_{210}
q_{70}	q_{211}	q_{212}	q_{213}
q_{71}	q_{214}	q_{215}	q_{216}
q_{72}	q_{217}	q_{218}	q_{219}
q_{73}	q_{220}	q_{221}	q_{222}
q_{74}	q_{223}	q_{224}	q_{225}
q_{75}	q_{226}	q_{227}	q_{228}
q_{76}	q_{229}	q_{230}	q_{231}
q_{77}	q_{232}	q_{233}	q_{234}
q_{78}	q_{235}	q_{236}	q_{237}
q_{79}	q_{238}	q_{239}	q_{240}
$^*q_{80}$	q_{241}	q_{242}	q_{243}
q_{81}	q_{244}	q_{245}	q_{246}
q_{82}	q_{247}	q_{248}	q_{249}
q_{83}	q_{250}	q_{251}	q_{252}
q_{84}	q_{253}	q_{254}	q_{255}

q_{85}	q_{256}	q_{257}	q_{258}
q_{86}	q_{259}	q_{260}	q_{261}
q_{87}	q_{262}	q_{263}	q_{264}
q_{88}	q_{265}	q_{266}	q_{267}
q_{89}	q_{268}	q_{269}	q_{270}
q_{90}	q_{271}	q_{272}	q_{273}
q_{91}	q_{274}	q_{275}	q_{276}
$^*q_{92}$	q_{277}	q_{278}	q_{279}
q_{93}	q_{280}	q_{281}	q_{282}
q_{94}	q_{283}	q_{284}	q_{285}
q_{95}	q_{286}	q_{287}	q_{288}
$^*q_{96}$	q_{289}	q_{290}	q_{291}
q_{97}	q_{292}	q_{293}	q_{294}
q_{98}	q_{295}	q_{296}	q_{297}
q_{99}	q_{298}	q_{299}	q_{300}
q_{100}	q_{301}	q_{302}	q_{303}
q_{101}	q_{304}	q_{305}	q_{306}
q_{102}	q_{307}	q_{308}	q_{309}
q_{103}	q_{310}	q_{311}	q_{312}
q_{104}	q_{313}	q_{314}	q_{315}
q_{105}	q_{316}	q_{317}	q_{318}
q_{106}	q_{319}	q_{320}	q_{321}
q_{107}	q_{322}	q_{323}	q_{324}
$^*q_{108}$	q_{325}	q_{326}	q_{327}
q_{109}	q_{328}	q_{329}	q_{330}
q_{110}	q_{331}	q_{332}	q_{333}
q_{111}	q_{334}	q_{335}	q_{336}
q_{112}	q_{337}	q_{338}	q_{339}
q_{113}	q_{340}	q_{341}	q_{342}
q_{114}	q_{343}	q_{344}	q_{345}
q_{115}	q_{346}	q_{347}	q_{348}
q_{116}	q_{349}	q_{350}	q_{351}
q_{117}	q_{352}	q_{353}	q_{354}
q_{118}	q_{355}	q_{356}	q_{357}
q_{119}	q_{358}	q_{359}	q_{360}
$^*q_{120}$	q_{361}	q_{362}	q_{363}
$^*q_{121}$	NIL	NIL	NIL
q_{122}	NIL	NIL	NIL
q_{123}	NIL	NIL	NIL
q_{124}	NIL	NIL	NIL
q_{125}	NIL	NIL	NIL
q_{126}	NIL	NIL	NIL
q_{127}	NIL	NIL	NIL

q_{128}	NIL	NIL	NIL
q_{129}	NIL	NIL	NIL
*q_{130}	NIL	NIL	NIL
q_{131}	NIL	NIL	NIL
q_{132}	NIL	NIL	NIL
q_{133}	NIL	NIL	NIL
q_{134}	NIL	NIL	NIL
q_{135}	NIL	NIL	NIL
q_{136}	NIL	NIL	NIL
q_{137}	NIL	NIL	NIL
q_{138}	NIL	NIL	NIL
*q_{139}	NIL	NIL	NIL
q_{140}	NIL	NIL	NIL
q_{141}	NIL	NIL	NIL
q_{142}	NIL	NIL	NIL
q_{143}	NIL	NIL	NIL
q_{144}	NIL	NIL	NIL
q_{145}	NIL	NIL	NIL
q_{146}	NIL	NIL	NIL
q_{147}	NIL	NIL	NIL
q_{148}	NIL	NIL	NIL
q_{149}	NIL	NIL	NIL
q_{150}	NIL	NIL	NIL
*q_{151}	NIL	NIL	NIL
q_{152}	NIL	NIL	NIL
q_{153}	NIL	NIL	NIL
q_{154}	NIL	NIL	NIL
q_{155}	NIL	NIL	NIL
q_{156}	NIL	NIL	NIL
q_{157}	NIL	NIL	NIL
q_{158}	NIL	NIL	NIL
q_{159}	NIL	NIL	NIL
*q_{160}	NIL	NIL	NIL
q_{161}	NIL	NIL	NIL
q_{162}	NIL	NIL	NIL
q_{163}	NIL	NIL	NIL
q_{164}	NIL	NIL	NIL
q_{165}	NIL	NIL	NIL
q_{166}	NIL	NIL	NIL
q_{167}	NIL	NIL	NIL
q_{168}	NIL	NIL	NIL
*q_{169}	NIL	NIL	NIL
q_{170}	NIL	NIL	NIL

q_{171}	NIL	NIL	NIL
q_{172}	NIL	NIL	NIL
q_{173}	NIL	NIL	NIL
q_{174}	NIL	NIL	NIL
q_{175}	NIL	NIL	NIL
q_{176}	NIL	NIL	NIL
q_{177}	NIL	NIL	NIL
q_{178}	NIL	NIL	NIL
q_{179}	NIL	NIL	NIL
q_{180}	NIL	NIL	NIL
*q_{181}	NIL	NIL	NIL
q_{182}	NIL	NIL	NIL
q_{183}	NIL	NIL	NIL
q_{184}	NIL	NIL	NIL
q_{185}	NIL	NIL	NIL
q_{186}	NIL	NIL	NIL
q_{187}	NIL	NIL	NIL
q_{188}	NIL	NIL	NIL
q_{189}	NIL	NIL	NIL
*q_{190}	NIL	NIL	NIL
q_{191}	NIL	NIL	NIL
q_{192}	NIL	NIL	NIL
q_{193}	NIL	NIL	NIL
q_{194}	NIL	NIL	NIL
q_{195}	NIL	NIL	NIL
q_{196}	NIL	NIL	NIL
q_{197}	NIL	NIL	NIL
q_{198}	NIL	NIL	NIL
*q_{199}	NIL	NIL	NIL
q_{200}	NIL	NIL	NIL
q_{201}	NIL	NIL	NIL
q_{202}	NIL	NIL	NIL
*q_{203}	NIL	NIL	NIL
q_{204}	NIL	NIL	NIL
q_{205}	NIL	NIL	NIL
q_{206}	NIL	NIL	NIL
q_{207}	NIL	NIL	NIL
q_{208}	NIL	NIL	NIL
q_{209}	NIL	NIL	NIL
q_{210}	NIL	NIL	NIL
q_{211}	NIL	NIL	NIL
*q_{212}	NIL	NIL	NIL
q_{213}	NIL	NIL	NIL

q_{214}	NIL	NIL	NIL
q_{215}	NIL	NIL	NIL
q_{216}	NIL	NIL	NIL
q_{217}	NIL	NIL	NIL
q_{218}	NIL	NIL	NIL
q_{219}	NIL	NIL	NIL
q_{220}	NIL	NIL	NIL
$^{*}q_{221}$	NIL	NIL	NIL
q_{222}	NIL	NIL	NIL
q_{223}	NIL	NIL	NIL
q_{224}	NIL	NIL	NIL
q_{225}	NIL	NIL	NIL
q_{226}	NIL	NIL	NIL
q_{227}	NIL	NIL	NIL
q_{228}	NIL	NIL	NIL
q_{229}	NIL	NIL	NIL
q_{230}	NIL	NIL	NIL
q_{231}	NIL	NIL	NIL
q_{232}	NIL	NIL	NIL
$^{*}q_{233}$	NIL	NIL	NIL
q_{234}	NIL	NIL	NIL
q_{235}	NIL	NIL	NIL
q_{236}	NIL	NIL	NIL
q_{237}	NIL	NIL	NIL
q_{238}	NIL	NIL	NIL
q_{239}	NIL	NIL	NIL
q_{240}	NIL	NIL	NIL
q_{241}	NIL	NIL	NIL
$^{*}q_{242}$	NIL	NIL	NIL
q_{243}	NIL	NIL	NIL
q_{244}	NIL	NIL	NIL
q_{245}	NIL	NIL	NIL
q_{246}	NIL	NIL	NIL
q_{247}	NIL	NIL	NIL
q_{248}	NIL	NIL	NIL
q_{249}	NIL	NIL	NIL
q_{250}	NIL	NIL	NIL
$^{*}q_{251}$	NIL	NIL	NIL
q_{252}	NIL	NIL	NIL
q_{253}	NIL	NIL	NIL
q_{254}	NIL	NIL	NIL
q_{255}	NIL	NIL	NIL
q_{256}	NIL	NIL	NIL

q_{257}	NIL	NIL	NIL
q_{258}	NIL	NIL	NIL
q_{259}	NIL	NIL	NIL
q_{260}	NIL	NIL	NIL
q_{261}	NIL	NIL	NIL
q_{262}	NIL	NIL	NIL
$^{*}q_{263}$	NIL	NIL	NIL
q_{264}	NIL	NIL	NIL
q_{265}	NIL	NIL	NIL
q_{266}	NIL	NIL	NIL
q_{267}	NIL	NIL	NIL
q_{268}	NIL	NIL	NIL
q_{269}	NIL	NIL	NIL
q_{270}	NIL	NIL	NIL
q_{271}	NIL	NIL	NIL
$^{*}q_{272}$	NIL	NIL	NIL
q_{273}	NIL	NIL	NIL
q_{274}	NIL	NIL	NIL
q_{275}	NIL	NIL	NIL
q_{276}	NIL	NIL	NIL
q_{277}	NIL	NIL	NIL
q_{278}	NIL	NIL	NIL
q_{279}	NIL	NIL	NIL
q_{280}	NIL	NIL	NIL
$^{*}q_{281}$	NIL	NIL	NIL
q_{282}	NIL	NIL	NIL
q_{283}	NIL	NIL	NIL
q_{284}	NIL	NIL	NIL
$^{*}q_{285}$	NIL	NIL	NIL
q_{286}	NIL	NIL	NIL
q_{287}	NIL	NIL	NIL
q_{288}	NIL	NIL	NIL
q_{289}	NIL	NIL	NIL
q_{290}	NIL	NIL	NIL
q_{291}	NIL	NIL	NIL
q_{292}	NIL	NIL	NIL
q_{293}	NIL	NIL	NIL
$^{*}q_{294}$	NIL	NIL	NIL
q_{295}	NIL	NIL	NIL
q_{296}	NIL	NIL	NIL
q_{297}	NIL	NIL	NIL
q_{298}	NIL	NIL	NIL
q_{299}	NIL	NIL	NIL

q_{300}	NIL	NIL	NIL
q_{301}	NIL	NIL	NIL
q_{302}	NIL	NIL	NIL
$^*q_{303}$	NIL	NIL	NIL
q_{304}	NIL	NIL	NIL
q_{305}	NIL	NIL	NIL
q_{306}	NIL	NIL	NIL
q_{307}	NIL	NIL	NIL
q_{308}	NIL	NIL	NIL
q_{309}	NIL	NIL	NIL
q_{310}	NIL	NIL	NIL
q_{311}	NIL	NIL	NIL
q_{312}	NIL	NIL	NIL
q_{313}	NIL	NIL	NIL
q_{314}	NIL	NIL	NIL
$^*q_{315}$	NIL	NIL	NIL
q_{316}	NIL	NIL	NIL
q_{317}	NIL	NIL	NIL
q_{318}	NIL	NIL	NIL
q_{319}	NIL	NIL	NIL
q_{320}	NIL	NIL	NIL
q_{321}	NIL	NIL	NIL
q_{322}	NIL	NIL	NIL
q_{323}	NIL	NIL	NIL
$^*q_{324}$	NIL	NIL	NIL
q_{325}	NIL	NIL	NIL
q_{326}	NIL	NIL	NIL
q_{327}	NIL	NIL	NIL
q_{328}	NIL	NIL	NIL
q_{329}	NIL	NIL	NIL
q_{330}	NIL	NIL	NIL
q_{331}	NIL	NIL	NIL
q_{332}	NIL	NIL	NIL
$^*q_{333}$	NIL	NIL	NIL
q_{334}	NIL	NIL	NIL
q_{335}	NIL	NIL	NIL
q_{336}	NIL	NIL	NIL
q_{337}	NIL	NIL	NIL
q_{338}	NIL	NIL	NIL
q_{339}	NIL	NIL	NIL
q_{340}	NIL	NIL	NIL
q_{341}	NIL	NIL	NIL
q_{342}	NIL	NIL	NIL
q_{343}	NIL	NIL	NIL
q_{344}	NIL	NIL	NIL
$^*q_{345}$	NIL	NIL	NIL
q_{346}	NIL	NIL	NIL
q_{347}	NIL	NIL	NIL
q_{348}	NIL	NIL	NIL
q_{349}	NIL	NIL	NIL
q_{350}	NIL	NIL	NIL
q_{351}	NIL	NIL	NIL
q_{352}	NIL	NIL	NIL
q_{353}	NIL	NIL	NIL
$^*q_{354}$	NIL	NIL	NIL
q_{355}	NIL	NIL	NIL
q_{356}	NIL	NIL	NIL
q_{357}	NIL	NIL	NIL
q_{358}	NIL	NIL	NIL
q_{359}	NIL	NIL	NIL
q_{360}	NIL	NIL	NIL
q_{361}	NIL	NIL	NIL
q_{362}	NIL	NIL	NIL
$^*q_{363}$	NIL	NIL	NIL

The Finite Automata, which accepts palindrome using three input characters and maximum string length, has 51 final states as shown in Table 5.

Table 5. Final States with accepted string from string length one to five

S. No.	String Lengt h	Final State	Accepted String
1	1	q_1	"u"
2	1	q_2	"m"
3	1	q_3	"a"
4	2	q_4	"uu"
5	2	q_8	"mm"
6	2	q_{12}	"aa"
7	3	q_{13}	"uuu"
8	3	q_{16}	"umu"
9	3	q_{19}	"uau"
10	3	q_{23}	"mum"
11	3	q_{26}	"mmm"
12	3	q_{29}	"mam"
13	3	q_{33}	"aua"

14	3	q_{36}	"ama"
15	3	q_{39}	"aaa"
16	4	q_{40}	"uuuu"
17	4	q_{52}	"ummu"
18	4	q_{64}	"uaau"
19	4	q_{68}	"muum"
20	4	q_{80}	"mmmm"
21	4	q_{92}	"maam"
22	4	q_{96}	"auua"
23	4	q_{108}	"amma"
24	4	q_{120}	"aaaa"
25	5	q_{121}	"uuuuu"
26	5	q_{130}	"uumuu"
27	5	q_{139}	"uuauu"
28	5	q_{151}	"umumu"
29	5	q_{160}	"ummmu"
30	5	q_{169}	"umamu"
31	5	q_{181}	"uauau"
32	5	q_{190}	"uamau"
33	5	q_{199}	"uaaau"
34	5	q_{203}	"muuum"
35	5	q_{212}	"mumum"
36	5	q_{221}	"muaum"
37	5	q_{233}	"mmumm"
38	5	q_{242}	"mmmmm"
39	5	q_{251}	"mmamm"
40	5	q_{263}	"mauam"
41	5	q_{272}	"mamam"
42	5	q_{281}	"maaam"
43	5	q_{285}	"auuua"
44	5	q_{294}	"aumua"
45	5	q_{303}	"auaua"
46	5	q_{315}	"amuma"
47	5	q_{324}	"ammma"
48	5	q_{333}	"amama"
49	5	q_{345}	"aauaa"
50	5	q_{354}	"aamaa"
51	5	q_{363}	"aaaaa"

4. Conclusion

The Turing Machine offers a fast construction for accepting a palindrome. Comparing with Turing Machine the possible combination of the path is equal for the Finite Automata. The string length constraint not considered in the Turing Machine. In case of Finite Automata string length plays a vital role in designing an abstract machine. The designing of Finite Automata for accepting the palindrome is an effective resource management technique. Because without Turing Machine it is possible to construct an abstract machine for accepting the palindrome.

5. Future Work

This work further may be implemented for the available palindrome in the dictionary. The number of the palindrome and its length is already available. So it is possible to construct the Finite Automata for the palindrome.

6. References

1. Ezhilarasu P, Thirunavukkarasu E, Karuppusami G, Krishnaraj N. Single substring based classification for non-deterministic finite automata. International Journal on Applications in Information and Communication Engineering. 2015; 1(10):29–31.
2. 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.
3. Ezhilarasu P, Krishnaraj N. Triple Substring Based Classification for Nondeterministic Finite Automata. IJAER. 2015; 10(59):177–82.
4. Available from: https://en.wikipedia.org/wiki/Turing_machine.
5. Martin JC. Introduction to languages and the theory of computation. 3rd ed. Tata Mcgraw Hill; 2010.