*International Journal on Applications in Electrical and Electronics Engineering*
*Volume 1: Issue 8: August 2015, pp 11-13. www.aetsjournal.com* ISSN (Online) : 2395-3527

# Construction of a Turing Machine to Check the Equivalence of an Input String for the Three Input Characters – A Review

Ezhilarasu P, Sowmya R

***Abstract***— This paper explores on equality checking of an input string through the Turing Machine. The input string has three unique input characters. The string categorized as 1. Ordered sequence, and 2. Mixed sequence. The transition diagram and the transition table derived for the two categories. The accept and reject condition also checked using the sample input strings. The three input characters can create $3^3$= 27 unique input string. The mixed sequence needs only 3 * 2 = 6 unique input string.

***Keywords***— Turing Machine, equivalence, the input string, ordered sequence, mixed sequence.

## I. Introduction

### *A. Turing Machine*

A Turing Machine (TM) is an enhanced abstract machine. The finite control of a Turing Machine can move both in forward and reverse direction. This advantage is not available in the Finite State Machine (FSM) and the Pushdown Automata(PDA). This feature supports the Turing Machine to execute several operations as compared to FSM and PDA.

The Turing Machine (M) has septuplets [1]. M = <Q, Γ, b, Σ, δ, $q_0$, F> where:

- Q is a collection of states
- Γ is a collection of tape alphabet/symbols
- # ∈ Γ is the blank symbol
- Σ ⊆ Γ - {b} is the set of input characters/symbols
- $q_0$ ∈ Q is the starting state
- F ⊆ Q is the collection of final (accepting) states
- δ: Q - F × Γ → Q - F × Γ × {L, R, S} is the transition function, where L is the left shift, R is the right shift, and S is Stop.

### *B. Categories of Input String*

The checking of equivalence for three input characters falls under two categories. Those are 1. ordered sequence, and 2. Mixed sequence. The input that follows ordered sequence has ‘n' number of first input characters followed by ‘n' number of second input characters. Then finally ‘n' number of thrid input characters. Ex. uuuuummmmmaaaaa. The input that follows mixed sequence has an equal number of input characters, but not in proper order. Ex. umaamumauuumaam.

## II. Related Work

Turing A M [1937] introduced an abstract machine called the Turing Machine [2]. Post E L [1936] discovered machine model similar to the Turing Machine [3]. The similar computational model invented by many scientist [4],[5],[6],[7], and [8]. Lamport L [1977] explained about the multiprocess program and proved its correctness [9]. Lin S & Rado T [1965] explained the Turing Machine problems [10]. Boolos G S & Jeffrey R C [1974] and Davis M [1982] written a book on computability [11], and [12]. Copeland B J & Proudfoot D [1999] explored the Turing’s forgotten ideas [13].

Ezhilarasu et. al. [2014] categorized the Nondeterministic Finite Automata using a number of the loop [14],[15] and [16]. Those are Nondeterministic Finite Automata with 1. Single loop 2. Dual loop 3. More than two loops. In this three positions are used for placing the loop. Those are 1.Starting State 2. Final State 3. Intermediate State. Ezhilarasu and Krishnaraj [2015] categorized the Nondeterministic Finite Automata through a number of the substring in the given string [17], [18], and [19]. Those are 1. Single substring 2.Double Substring 3. More than double substring.

## III. Methodology

Based on, as in [14], [15], [16], [17], [18], and [19] Ezhilarasu et. al. [2015] implemented the construction of Minimized Deterministic Finite Automata for implementing the equality of input character for the given string [20]. The counter concept used to implement the construction of minimized Deterministic Finite Automata. The similar concept used to implement the Turing Machine for checking the equality. The each loop counts the three unique input characters at a time. If processing of input string over and at each loop three unique conversions occurred; then the given input string has the equal amount of unique input characters.

## IV. Construction Of The Turing Machine

### *A. Ordered Sequence*

The transition diagram(TD) and the transition table(TT) of the Turing Machine that accepts ordered sequence input string with equal amount of the input characters, as shown in Figure 1, and Table 1.

Ezhilarasu P, Associate Professor, Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, Coimbatore - 641032, India. ( Email: prof.p.ezhilarasu@gmail.com)
Sowmya R , PG Scholar, Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, Coimbatore - 641032, India.

International Journal on Applications in Electrical and Electronics Engineering
Volume 1: Issue 8: August 2015, pp 11-13. www.aetsjournal.com ISSN (Online) : 2395-3527

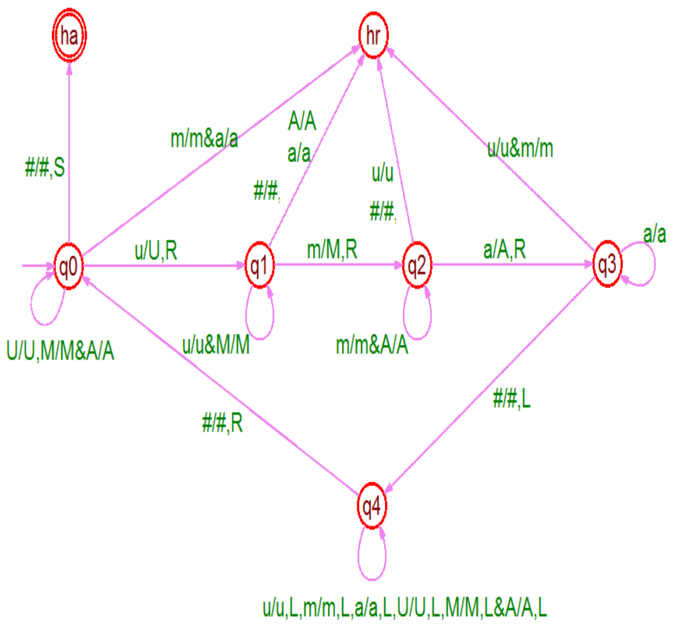


Fig.1.Transition Diagram of the Turing Machine that accepts ordered sequence input string with equal amount of the input characters

Table. I . Transition Table of the Turing Machine that accepts ordered sequence input string with equal amount of the input characters

| States | Input | | | | | | |
|---|---|---|---|---|---|---|---|
| | u | m | a | U | M | A | # |
| → $q_0$ | U,R, $q_1$ | $h_r$ | $h_r$ | U,R, $q_0$ | M,R, $q_0$ | A,R, $q_0$ | #,S,$h_a$ |
| $q_1$ | u,R, $q_1$ | M,R, $q_2$ | $h_r$ | - | M,R, $q_1$ | $h_r$ | $h_r$ |
| $q_2$ | $h_r$ | m,R,$q_2$ | A,R, $q_3$ | - | - | A,R, $q_2$ | $h_r$ |
| $q_3$ | $h_r$ | $h_r$ | a,R,$q_3$ | - | - | - | #,L,$q_4$ |
| $q_4$ | u,L,$q_4$ | m,L,$q_4$ | a,L,$q_4$ | U,L,$q_4$ | M,L,$q_4$ | A,L,$q_4$ | #,R,$q_0$ |

PROCESSING AN INPUT STRING

Accepting Condition: Input string : uuummmaaa

$q_0$, #uuummmaaa#

□ $q_1$, #Uuummmaaa#

□ $q_1$, #Uuummmaaa#

□$q_1$, #Uuummmaaa#

□$q_2$, #UuuMmmaaa#

□$q_2$, #UuuMmmaaa#

□$q_2$, #UuuMmmaaa#

□$q_3$, #UuuMmmAaa#

□$q_3$, #UuuMmmAaa#

□$q_3$, #UuuMmmAaa#

□$q_4$, #UuuMmmAaa#

(Step 11-20) □$q_0$, #UuuMmmAaa#

(Step 21-30) □$q_4$, #UUuMMmAAa#

(Step 31-40) □$q_0$, #UUuMMmAAa#

(Step 41-50) □$q_4$, #UUUMMMAAA#

(Step 51-60) □$q_0$, #UUUMMMAAA#

(Step 61-69) □$q_0$, #UUUMMMAAA#

Step 70 $h_a$ (String accepted)

Rejecting Condition

Input string : uumaaa

$q_0$, #uumaaa#

(Step 1-7) □$q_4$, #UuMAaa#

(Step 8-14) □$q_0$, #UuMAaa#

Step 15-17□$q_1$, #UuMAaa#

Step 18□$h_r$(String Rejected)

*B. Mixed Sequence*

The transition diagram and the transition table of the Turing Machine that accepts the mixed sequence input string with the equal amount of the input characters, as shown in Figure 2, and Table 2.

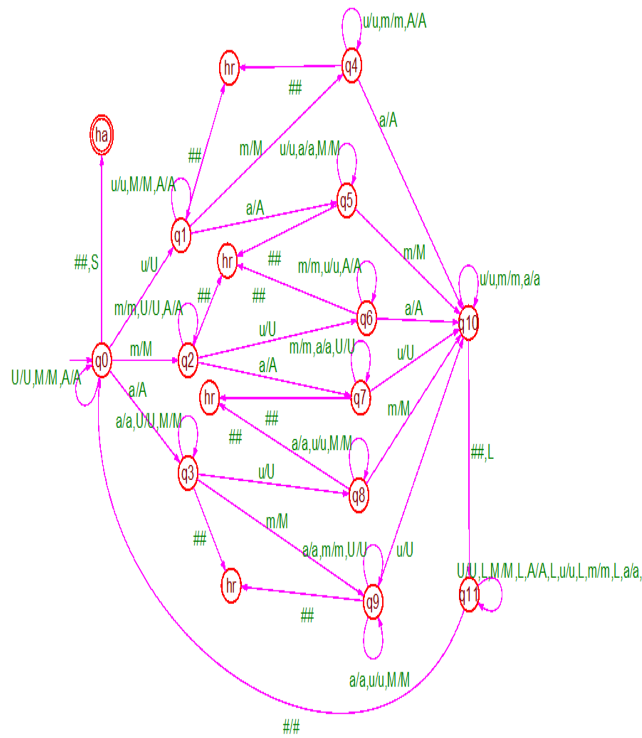


Fig.2.Transition Diagram of the Turing Machine that accepts mixed sequence input string with equal amount of the input characters

Table. II . Transition Table of the Turing Machine that accepts mixed sequence input string with equal amount of the input characters

| States | Input characters | | | | | | |
|---|---|---|---|---|---|---|---|
| | u | m | a | U | M | A | # |
| → $q_0$ | U,R,$q_1$ | M,R, $q_2$ | A,R, $q_3$ | U,R, $q_0$ | M,R, $q_0$ | A,R, $q_0$ | #,S,$h_a$ |
| $q_1$ | u,R,$q_1$ | M,R, $q_4$ | A,R, $q_5$ | - | M,R, $q_1$ | A,R, $q_1$ | #,R,$h_r$ |
| $q_2$ | U,R, $q_6$ | m,R,$q_2$ | A,R, $q_7$ | U,R, $q_2$ | - | A,R, $q_2$ | #,R,$h_r$ |
| $q_3$ | U,R, $q_8$ | M,R, $q_9$ | a,R,$q_3$ | U,R,$q_3$ | M,R,$q_3$ | - | #,R,$h_r$ |
| $q_4$ | u,R,$q_4$ | m,R,$q_4$ | A,R, $q_{10}$ | - | - | A,R,$q_4$ | #,R,$h_r$ |
| $q_5$ | u,R,$q_5$ | M,R, $q_{10}$ | a,R,$q_5$ | - | M,R,$q_5$ | - | #,R,$h_r$ |
| $q_6$ | u,R,$q_6$ | m,R,$q_6$ | A,R, $q_{10}$ | - | - | A,R,$q_6$ | #,R,$h_r$ |
| $q_7$ | U,R, $q_{10}$ | m,R,$q_7$ | a,R,$q_7$ | U,R,$q_7$ | - | - | #,R,$h_r$ |
| $q_8$ | u,R,$q_8$ | M,R, $q_{10}$ | a,R,$q_8$ | - | M,R,$q_8$ | - | #,R,$h_r$ |
| $q_9$ | U,R, $q_{10}$ | m,R,$q_9$ | a,R,$q_9$ | U,R,$q_9$ | - | - | #,R,$h_r$ |
| $q_{10}$ | u,R, $q_{10}$ | m,R, $q_{10}$ | a,R, $q_{10}$ | - | - | - | #,L,$q_{10}$ |
| $q_{11}$ | u,L, $q_{11}$ | m,L, $q_{11}$ | a,L, $q_{11}$ | U,L, $q_{11}$ | M,L, $q_{11}$ | A,L, $q_{11}$ | #,R,$q_0$ |

PROCESSING AN INPUT STRING

Accepting Condition: Input string : uaamum

$q_0$, # uaamum #

□ $q_1$, # Uaamum #

□ $q_5$, # UAamum #

□$q_5$, # UAamum #

□$q_{10}$, # UAaMum #

□ $q_{10}$, # UAaMum #

□ $q_{10}$, # UAaMum #

□$q_{11}$, # UAaMum #

(Step 8 -14) $q_0$, # UAaMum #

(Step 15 -17) $q_3$, # UAAMum #

(Step 18 -19) $q_8$, # UAAMUm #

(Step 20) $q_{10}$, # UAAMUM #

(Step 21) $q_{11}$, # UAAMUM #

(Step 22 -28) $q_0$, # UAAMUM #

(Step 29 -34) $q_0$, # UAAMUM #

Step 35 $h_a$(String accepted)

Rejecting Condition

Input string : uum

$q_0$, #uum#

(Step 1-3) □$q_4$, # UuM #

(Step 4) □$h_r$(String Rejected)

## V. CONCLUSION

The loop makes three unique conversions at a time. The three unique input characters can produce $3^3$ = 27 possible input string with the length three. Based on the unique three conversions we need only n!(n=number of unique input characters) number of input string. So we need only 3! = 3 * 2 * 1 = 6 input string. The input is in lower case. During conversion the input converted into upper case. Therefore the input string remains same after checking the equality.

## VI. FUTURE WORK

This work can be utilized for the construction of the Turing Machine for input string with more than three unique input characters.

## REFERENCES


[1] Available online at : https:// en.wikipedia.org /wiki/ Turing_machine

[2] A. M. TURING, "On computable numbers, with an application to the Entscheidungsproblem," Proceedings, London Mathematical Society 2:42 (1936-1937), 230-265. Errata appear in 2:43 (1937), 544-546.

[3] E. L. POST, "Finite combinatory processes. Formulation I," Journal of Symbolic Logic 1 (1936), 103-105.

[4] N. CHOMSKY, "Three models for the description of language," *IRE Transactions on Information Theory* 2:3 (1956) 113-124.

[5] A. CHURCH, "The Calculi of Lambda-Conversion," *Annals of Mathematics Studies* 6 (1941) Princeton University Press, Princeton, New Jersey.

[6] S. C. KLEENE, "General recursive functions of natural numbers," *Mathematische Annalen* 112:5 (1936) 727-742.

[7] A. A. MARKOV, "Theory of Algorithms," *Trudy Mathematicheskogo Instituta imeni V. A. Steklova* 42 (1954).

[8] E. L. POST, "Formal reductions of the general combinatorial decision problem," *American Journal of Mathematics* 65 (1943) 197-215.

[9] L. Lamport, Proving the correctness of multiprocess programs, IEEE Transactions on Software Engineering SE-3 (2) (1977) 125-143.

[10] Lin, S. and Radó, T. (1965). Computer Studies of Turing Machine Problems, Journal of ACM, Vol. 12, pp. 196-212.

[11] Boolos, G. S. and Jeffrey, R. C. (1974). Computability and Logic, Cambridge: Cambridge University Press.

[12] Davis, M. (1982). Computability and Unsolvability, New York: Mcgraw-Hill.

[13] Copeland, B.J. and Proudfoot, D. (1999), ‘Alan Turing’s Forgotten Ideas in Computer Science’, *Scientific American* 280 (April), pp. 76–81.

[14] 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.

[15] Ezhilarasu P, Prakash J, Krishnaraj N, Satheesh Kumar D, Sudhakar K, Parthasarathy C, "A Novel Approach to Classify Nondeterministic Finite Automata Based on Dual Loop and its Position ", International Journal of Engineering Trends and Technology (IJETT), V18(3),147-150 Dec 2014. ISSN:2231-5381. www.ijettjournal.org. published by seventh sense research group

[16] P. Ezhilarasu, J. Prakash, N. Krishnaraj, D. Satheesh Kumar, K. Sudhakar, and C. Parthasarathy, “A Novel Approach to Classify Nondeterministic Finite Automata Based on More than Two Loops and its Position”, SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE), Volume 1, Issue 10, 2014, pp. 46-49.

[17] P.Ezhilarasu, E.Thirunavukkarasu, G.Karuppusami, N.Krishnaraj, “Single Substring Based Classification for Nondeterministic Finite Automata”, International Journal on Applications in Information and Communication Engineering, Vol 1(10) pp. 29-31. 2015.

[18] P.Ezhilarasu, N.Krishnaraj, “Double Substring Based Classification for Nondeterministic Finite Automata”, International Conference on Recent Advances in Engineering, Science & Technology – ICON’15”, 2015.

[19] P.Ezhilarasu, N.Krishnaraj, “Triple Substring Based Classification for Nondeterministic Finite Automata”, International Journal of Applied Engineering Research (IJAER), Vol.10(59), pp.177-182, 2015.

[20] Ezhilarasu P, Prakash J, Krishnaraj N, Satheesh Kumar D, Sureshbabu V, Parthasarathy C, ”Construction of Minimized Deterministic Finite Automata for Finding the Equal Number of Input Characters”, International Journal on Applications in Information and Communication Engineering, Vol. 1, No 11, pp. 17-20,2015