

Construction of a Basic Calculator through the Turing Machine – A Review

Ezhilarasu P

Associate Professor

Department of Computer Science and Engineering
Hindusthan College of Engineering and Technology, Coimbatore
Tamil Nadu – India

ABSTRACT

This paper focuses on the Turing Machine usage as a basic calculator. The basic calculator here refers the four primary arithmetic operations. Those are namely 1.Addition, 2.Subtraction, 3.Multiplication, and 4. Division. The Turing Machine has the favorable circumstances as compared with the other abstract machines the Finite Automata and the Pushdown Automata. The Turing Machine has the vital feature called bi-directional movement. This aspect makes the Turing Machine to support the construction of the basic calculator. The transition diagram and transition table also derived for each calculation. The processing of input for each operation also given.

Keywords:- Turing Machine, Addition, Subtraction, Multiplication, Division

I. INTRODUCTION

A. Turing Machine

The Turing Machine is septuplets. The Turing machine mathematically models a mechanically operating machine on a tape [1]. It has the tape for taking input from the cell. The tape by default has the blank symbol (#) in all the cells. The head for moving in the right or left direction to perform read/write operation. A state register that has the initial value, as a start state for storing the Turing Machine's state. Then a finite table of instruction.

B. Basic Calculator

The basic calculation frequently refers the basic arithmetic operations. Those are

1. Addition (represented by the operator '+')
2. Subtraction (represented by the operator '-')
3. Multiplication (represented by the operator \times or \cdot or $*$)
4. Division (represented by the operator \div or $/$)

II. RELATED WORKS

Turing A M [1936] invented the automatic machine that processes input symbols based on the rules [2]. Many scientists invented the similar computational model [3], [4], [5], [6], and [7]. Post E L [1936] invented the similar machine model [8]. Lin S & Rado T [1965] discussed the Turing Machine problems [9]. Boolos G S & Jeffrey R C [1974] and Davis M [1982]

written a book on the topic the computability [10], and [11]. Hopcraft J E et. al [2001] explained about the subroutine usage in the Turing Machine [12]. This subroutine technique was used to construct the Turing Machine for multiplication and division. Martin J C [2010] explained the Turing Machine to support the implementation of palindrome and the reversing the string [13]. Ezhilarasu P & Sowmya R [2015] constructed the Turing Machine to check the equivalence of the input string for the three input characters [14].

III. ARITHMETIC OPERATIONS

The Turing Machine used to perform the arithmetic operations namely addition, subtraction, multiplication, and division explained using the transition diagram, transition table, and the example input.

A.ADDITION

The Turing Machine that performs the addition operation as shown in figure 1, and table 1.

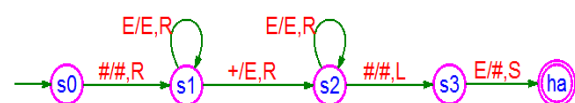


Fig.1. Transition Diagram of the Turing Machine to perform the addition operation

Table. I. Transition Table of the Turing Machine that perform the addition operation

States	Input		
	E	+	#
$\rightarrow s_0$	-	-	#,R, s ₁
s ₁	E,R, s ₁	E,R, s ₁	-
s ₂	E,R, s ₂	-	#,L, s ₃
s ₃	#,S, h _a	-	-

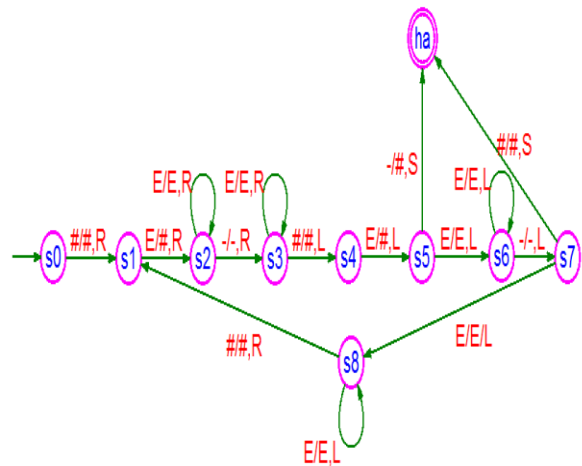


Fig.2. Transition Diagram of the Turing Machine to perform the subtraction operation

INPUT PROCESSING FOR THE ADDITION

The input given here is unary 'E'. Ex. The number 5 represented as EEEEE.

Input : #EEEEEE+EE#

s₀, #EEEEEE+EE#

Step 1: s₁, #EEEEEE+EE#

Step 2: s₁, #EEEEEE+EE#

Step 3: s₁, #EEEEEE+EE#

Step 4: s₁, #EEEEEE+EE#

Step 5: s₁, #EEEEEE+EE#

Step 6: s₁, #EEEEEE+EE#

Step 7: s₂, #EEEEEE#

Step 8: s₂, #EEEEEE#

Step 9: s₂, #EEEEEE#

Step 10: s₃, #EEEEEE#

h_a, #EEEEEE##

The Turing Machine reached the halting state for acceptance. The final input string is #EEEEEE##. i.e seven E's. 5+2=7. Hence the construction of the Turing Machine is correct for addition operation.

B.SUBTRACTION

The Turing Machine that performs the subtraction operation as shown in figure 2, and table 2.

Table. II. Transition Table of the Turing Machine that perform the subtraction operation

States	Input		
	E	-	#
$\rightarrow s_0$	-	-	#,R, s ₁
s ₁	#,R, s ₂	-	-
s ₂	E,R, s ₂	-,R, s ₃	-
s ₃	E,R, s ₃	-	#,L, s ₄
s ₄	#,L, s ₅	-	-
s ₅	E,L, s ₆	#,S,h _a	-
s ₆	E,L, s ₆	-,L, s ₇	-
s ₇	E,L, s ₈	-	#,S,h _a
s ₈	E,L, s ₈	-	#,R, s ₁

INPUT PROCESSING FOR THE SUBTRACTION

In subtraction, we have three types of conditions. Those are 1. The two numbers are same, 2. The first operand(number) greater than the second operand(number), and 3. The second number higher than the first number. Hence, we need to check the three types of input. The input given here is unary 'E'. Ex. The number 3 represented as EEE.

Input Type 1: #EE-EE#

s₀, #EE-EE#

- Step 1: $s_1, \# \underline{E} E \#$
- Step 2: $s_2, \# \# \underline{E} E \#$
- Step 3: $s_2, \# \# E_2 \underline{E} E \#$
- Step 4: $s_3, \# \# E \underline{E} E \#$
- Step 5: $s_3, \# \# E \underline{E} E \#$
- Step 6: $s_3, \# \# E \underline{E} E \#$
- Step 7: $s_4, \# \# E \underline{E} E \#$
- Step 8: $s_5, \# \# E \underline{E} E \#$
- Step 9: $s_6, \# \# E_2 E \# \#$
- Step 10: $s_7, \# \# E \underline{E} E \# \#$
- Step 11: $s_8, \# \# E \underline{E} E \# \#$
- Step 12: $s_1, \# \# E \underline{E} E \# \#$
- Step 13: $s_2, \# \# \# \underline{E} E \# \#$
- Step 14: $s_3, \# \# \# \underline{E} E \# \#$
- Step 15: $s_3, \# \# \# \underline{E} E \# \#$
- Step 16: $s_4, \# \# \# \underline{E} E \# \#$
- Step 17: $s_5, \# \# \# \underline{\#} \# \# \#$
- Step 18: $h_a, \# \# \# \# \# \# \# \#$

The Turing Machine reached the halting state for acceptance. The final input string is $\# \# \# \# \# \# \#$. i.e., zero E's. $2-2=0$. Hence, the construction of the Turing Machine is correct for the subtraction operation of input type 1.

Input Type 2: $\# E E \underline{E} \#$

$S_0, \# E E \underline{E} \#$

- Step 1: $s_1, \# \underline{E} E \underline{E} \#$
- Step 2: $s_2, \# \# \underline{E} \underline{E} \#$
- Step 3: $s_2, \# \# E_2 \underline{E} \#$
- Step 4: $s_3, \# \# E \underline{E} \#$
- Step 5: $s_3, \# \# E \underline{E} \#$
- Step 6: $s_4, \# \# E \underline{E} \#$
- Step 7: $s_5, \# \# E_2 \# \#$

Step 8: $h_a, \# \# E \# \# \#$

The Turing Machine reached the halting state for acceptance. The final input string is $\# \# E \# \# \#$. i.e., one E's. $2-1=1$. Hence, the construction of the Turing Machine is correct for the subtraction operation of input type 2.

Input Type 3: $\# E \underline{E} E \#$

$S_0, \# E \underline{E} E \#$

- Step 1: $s_1, \# \underline{E} \underline{E} E \#$
- Step 2: $s_2, \# \# \underline{E} E \#$
- Step 3: $s_3, \# \# \underline{E} E \#$
- Step 4: $s_3, \# \# \underline{E} E \#$
- Step 5: $s_3, \# \# \underline{E} E \#$
- Step 6: $s_4, \# \# \underline{E} E \#$
- Step 7: $s_5, \# \# \underline{E} E \#$
- Step 8: $s_6, \# \# \underline{E} E \#$
- Step 9: $s_7, \# \# \underline{E} E \#$
- $h_a, \# \# \underline{E} E \#$

The Turing Machine reached the halting state for acceptance. The final input string is $\# \# \underline{E} E \#$. i.e minus followed by one E's. $1-2=-1$. Hence, the construction of the Turing Machine is correct for the subtraction operation of input type 3.

C.MULTIPLICATION

The Turing Machine that performs the multiplication operation as shown in figure 3.

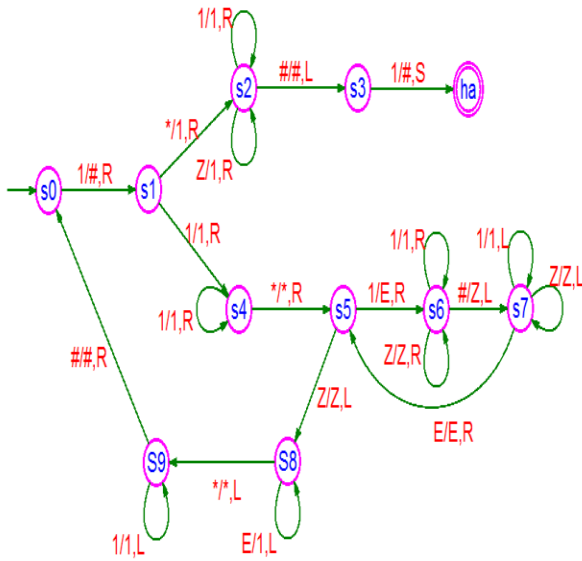


Fig.3. Transition Diagram of the Turing Machine to perform the multiplication operation

Table. III . Transition Table of the Turing Machine that perform the multiplication operation

States	Input				
	1	*	E	Z	#
→s ₀	#,R,s ₁	-	-	-	-
s ₁	1,R,s ₄	1,R,s ₂	-	-	-
s ₂	1,R,s ₂	-	-	1,R,s ₂	#,L,s ₃
s ₃	#,S, h _a	-	-	-	-
s ₄	1,R,s ₄	*,R,s ₅	-	-	-
s ₅	E,R,s ₆	-	-	Z,L,s ₈	-
s ₆	1,R,s ₆	-	-	Z,R,s ₆	Z,L,s ₇
s ₇	1,L,s ₇	-	E,R,s ₅	Z,L,s ₇	-
s ₈	1,L,s ₈	*,L,s ₉	-	-	-
s ₉	1,L,s ₉	-	-	-	#,R,s ₀

INPUT PROCESSING FOR THE MULTIPLICATION

The subroutine concept used to implement the multiplication operation [12]. The input given here is unary '1'. Ex. The number 7 represented as 1111111.

Input : #11*111####

s₀, #11*111####

Step 1: s₁, #1*111#### Step 2: s₄, #1*111####

Step 3: s₅, #1*111#### Step 4: s₆, #1*E11####

Step 5: s₆, #1*E11#### Step 6: s₆, #1*E11####

Step 7: s₇, #1*E11Z### Step 8: s₇, #1*E11Z###

Step 9: s₇, #1*E11Z### Step 10: s₇, #1*E11Z###

Step 11: s₅, #1*E11Z###, Step 12: s₆, #1*EE1Z###

Step 13: s₆, #1*EE1Z###, Step 14: s₆, #1*EE1Z###

Step 15: s₇, #1*EE1ZZ##, Step 16: s₇, #1*EE1ZZ##

Step 17: s₇, #1*EE1ZZ##, Step 18: s₅, #1*EE1ZZ##

Step 19: s₆, #1*EEEZZ##, Step 20: s₆, #1*EEEZZ##

Step 21: s₆, #1*EEEZZ##, Step 22: s₇, #1*EEEZZ##

Step 23: s₇, #1*EEEZZ##, Step 24: s₇, #1*EEEZZ##

Step 25: s₅, #1*EEEZZ##, Step 26: s₈, #1*EEEZZ##

Step 27: s₈, #1*EE1ZZZ#, Step 28: s₈, #1*E11ZZZ#

Step 29: s₈, #1*111ZZZ#, Step 30: s₉, #1*111ZZZ#

Step 31: s₉, #1*111ZZZ#, Step 32: s₀, #1*111ZZZ#

Step 33: s₁, ###*111ZZZ#, Step 34: s₂, ###111ZZZ#

Step 35: s₂, ###111ZZZ#, Step 36: s₂, ###111ZZZ#

Step 37: s₂, ###111ZZZ#, Step 38: s₂, ###111ZZZ#

Step 39: s₂, ###11111Z# Step 40: s₂, ###111111#

Step 41: s₃, ###111111# Step 42: h_a, ###111111#

The Turing Machine reached the halting state for acceptance. The final input string is ###111111#. i.e six 1's. 2*3=6. Hence the construction of the Turing Machine is correct for the multiplication operation.

D.DIVISION

The Turing Machine that performs the division operation as shown in figure 4.

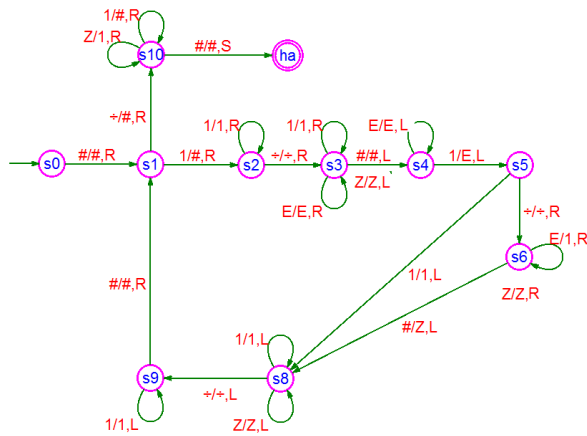


Fig.4. Transition Diagram of the Turing Machine to perform the division operation

Table. IV. Transition Table of the Turing Machine that perform the multiplication operation

States	Input				
	1	÷	E	Z	#
→s ₀	#,R,s ₁	-	-	-	-
s ₁	#,R,s ₂	#,R,s ₁₀	-	-	-
s ₂	1,R,s ₂	÷,R,s ₃	-	-	-
s ₃	1,R,s ₃	-	E,R,s ₃	Z,L,s ₄	#,L,s ₄
s ₄	E,L,s ₅	-	E,L,s ₄	-	-
s ₅	1,L,s ₈	÷,R,s ₆	-	-	-
s ₆	-	-	1,R,s ₆	-	Z,L,s ₈
s ₈	1,L,s ₈	÷,L,s ₉	-	Z,L,s ₈	-
s ₉	1,L,s ₉	-	-	-	#,R,s ₁
s ₁₀	#,R,s ₁₀	-	-	1,R,s ₁₀	#,S, h _a

INPUT PROCESSING FOR THE DIVISION

The subroutine concept used to implement the division operation [12]. The input given here is unary '1'. Ex. The number 4 represented as 1111.

Input : #1111÷11####

s₀#1111÷11####

Step 1: s₁,#1111÷11####, Step 2: s₂,##111÷11####

Step 3: s₂,##111÷11#### Step 4: s₂,##111÷11####

Step 5: s₂,##111÷11####, Step 6: s₃,##111÷11####

Step 7: s₃,##111÷11####, Step 8: s₃,##111÷11####

Step 9: s₄,##111÷11####, Step 10: s₅,##111÷1E####

Step 11: s₈,##111÷1E####, Step 12: s₉,##111÷1E####

Step 13: s₉,##111÷1E####, Step 14: s₉,##111÷1E####

Step 15: s₉,##111÷1E####, Step 16: s₁,##111÷1E####,

Step 17: s₂,###11÷1E####, Step 18: s₂,###11÷1E####

Step 19: s₂,###11÷1E####, Step 20: s₃,###11÷1E####

Step 21: s₃,###11÷1E####, Step 22: s₃,###11÷1E####

Step 23: s₄,###11÷1E####, Step 24: s₄,###11÷1E####

Step 25: s₅,###11÷1E####, Step 26: s₆,###11÷1E####

Step 27: s₆,###11÷1E####, Step 28: s₆,###11÷1E####

Step 29: s₈,###11÷1Z####, Step 30: s₈,###11÷1Z####

Step 31: s₈,###11÷1Z####, Step 32: s₉,###11÷1Z####

Step 33: s₉,###11÷1Z####, Step 34: s₉,###11÷1Z####

Step 35: s₁,###11÷1Z####, Step 36: s₂,###11÷1Z####

Step 37: s₂,###11÷1Z####, Step 38: s₃,###11÷1Z####

Step 39: s₃,###11÷1Z####, Step 40: s₃,###11÷1Z####

Step 41: s₄,###11÷1Z####, Step 42: s₅,###11÷1Z####

Step 43: s₈,###11÷1EZ####, Step 44: s₉,###11÷1EZ####

Step 45: s₉,###11÷1EZ####, Step 46: s₁,###11÷1EZ####

Step 47: s₂,###11÷1EZ####, Step 48: s₃,###11÷1EZ####,

Step 49: s₃,###11÷1EZ####, Step 50: s₃,###11÷1EZ####,

Step 51: s₄,###11÷1EZ####, Step 52: s₄,###11÷1EZ####,

Step 53: s₅,###11÷1EEZ####, Step 54: s₆,###11÷1EEZ####,

Step 55: s₆,###11÷1EZ####, Step 56: s₆,###11÷1EZ####,

Step 57: s₆,###11÷1EZ####, Step 58: s₈,###11÷1ZZ####,

Step 59: s₈,###11÷1ZZ####, Step 60: s₈,###11÷1ZZ####,

Step 61: s₈,###11÷1ZZ####, Step 62: s₉,###11÷1ZZ####,

Step 63: s₁,###11÷1ZZ####, Step 64: s₁₀,###11÷1ZZ####,

Step 65: s₁₀,###11÷1ZZ####, Step 66: s₁₀,###11÷1ZZ####,

Step 67: $s_{10}, \text{#####}1\underline{z}\#\#, \text{Step 68: } s_{10}, \text{#####}11\#\#,$
 $h_a, \text{#####}11\#\#,$

The Turing Machine reached the halting state for acceptance. The final input string is #####11##. i.e two 1's. $4 \div 2 = 2$. Hence the construction of the Turing Machine is correct for the division operation.

IV.CONCLUSION

The Turing Machine bi-directional movement both in forward and backward direction supports the implementation of the basic calculator. This capability can be used to build the Turing Machine for finding modulus value for the given input.

REFERENCE

- [1] Available online at https://en.wikipedia.org/wiki/Turing_machine
- [2] A.M.Turing, "On Computable Numbers, with an Application to the Entscheidungs problem". Proceedings of the London Mathematical Society. Vol.2, No.42, pp. 230–265, 1937.
- [3] N. Chomsky, "Three models for the description of language", IRE Transactions on Information Theory, Vol.2, No.3, pp. 113-124, 1956.
- [4] A. Church, "The Calculi of Lambda-Conversion," Annals of Mathematics Studies, Vol 6, Princeton University Press, Princeton, New Jersey.1941.
- [5] S. C. Kleene, "General recursive functions of natural numbers," Mathematische Annalen 112:5 (1936) 727-742.
- [6] A. A. Markov, "Theory of Algorithms," Trudy Matematicheskogo Instituta imeni V. A. Steklova 42 (1954).
- [7] E. L. Post, "Formal reductions of the general combinatorial decision problem," American Journal of Mathematics 65 (1943) 197-215.
- [8] E. L. Post, "Finite combinatory processes. Formulation I," Journal of Symbolic Logic 1 (1936), 103-105.
- [9] Lin, S. And Radó, T. (1965). Computer Studies of Turing Machine Problems, Journal of ACM, Vol. 12, pp. 196-212.
- [10] Boolos, G. S. And Jeffrey, R. C. (1974). Computability and Logic, Cambridge: Cambridge University Press.
- [11] Davis, M. (1982). Computability and Unsolvability, New York: Mcgraw-Hill.
- [12] John E.Hopcraft, Rajeev Motwani Jeffery D Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education, Second Edition, 2001.
- [13] John C Martin,"Introduction to languages and the theory of computation" Third Edition, Tata Mcgraw Hill, 2010.
- [14] Ezhilarasu P, Sowmya R, "Construction of a Turing Machine to Check the Equivalence of an Input String for the Three Input Characters – A Review", International Journal on Applications in Electrical and Electronics Engineering, Vol. 1, No. 8, pp. 11-13, 2015.