Simulation of Turing Machine

Ajinkya Ghusarkar[1], Girish Nikam[2], Atharva Thokal[3], Sourabh Shintre[4]

[1,2,3,4] *Student, Dept. of Information Technology ,Vishwakarma Institute of Technology ,Maharashtra, India*

Abstract - *A Turing machine is a mathematical model of computation that can be used to simulate the logic of any algorithm that can be expressed as a set of rules. The simulation of a Turing machine allows for the study of algorithms and their computational complexity, as wellas the development and analysis of various machine learning techniques. In this paper, we present asimulation of a Turing machine using a software program that allows for the creation, manipulation, and execution of Turing machine programs. We discuss the implementation of the simulation and demonstrate its capabilities through a series of examples. Finally, we discuss the potential applications and future directions for this simulation, including its use in education, research, and industry.*

Key Words: ***Turing machine, Automata theory, Python.***

1. INTRODUCTION

A Turing machine is a theoretical device that is used to model the behavior of a computer algorithm. It is named after Alan Turing, the mathematician who first described the concept in his 1936 paper, "On Computable Numbers,with an Application to the Entscheidungsproblem."

Simulating a Turing machine involves creating a virtual version of the machine that can be run on a computer. This allows researchers to study the behavior of the machine and how it processes input. It can also be used to test different algorithms and compare their efficiency andperformance.

In this research paper, we will explore the use of simulation to study the behavior of a Turing machine. Wewill describe the basic principles of a Turing machine and its components, including the tape, head, and state transition function. We will also discuss the various techniques and algorithms used to simulate a Turing machine and the benefits and limitations of this approach.Finally, we will present some case studies demonstratingthe usefulness of simulation in understanding the behavior of a Turing machine and its applications in computer science.

2. LITERATURE REVIEW

B. A. Trakhtenbrot in 1963, [1] has published a research paper to study the behavior of Turing machines under different conditions and to evaluate their computational power. In the following years, many researchers have used simulation to study the complexity and efficiency of Turing machines, as well as their ability to solve various computational problems.

J. N. Tsitsiklis and M. Athans [2] used simulation to study the behavior of Turing machines in at distributed setting and found that their performance was significantly affected by the communication and synchronization among the different machines.

T. H. Cormen et al. [3] used simulation to study the behavior of Turing machines in the presence of noise anderrors, and found that their performance was significantlyaffected by the probability and magnitude of the errors.

K. S. Trivedi et al, [4] used simulation to study thebehavior of Turing machines with continuous inputs and outputs, and found that their performance was significantly affected by the resolution and accuracy of the inputs and outputs.

The paper by researchers from Imperial College London [5] found some computational and thermodynamic constraints which would be faced if we are building a Universal Turing Machine. A model was proposed by them which implements multiple distinct computations inparallel but this makes the model complicated.

3. METHODOLOGY

For this model, we researched various existing papers and understood the functioning of the Turing machineand found no clear simulation of a Turing machine ina programming language. Considering this we came forward with a model designed using python which simulates the Turing machine and performs simple binary operations by following all the rules of the Turing machine. The model proposed in the paper uses the features of object oriented programming like inheritance, using all this a complete Turing machine is simulated.

3.1 Proposed System

To get the 1's compliment of a binary number, follow these steps:

1. Write down the binary number you want to find the1's compliment of.

2. Determine the length of the binary number. For example, if the binary number is 1010, the length is 4.

3. Create a new string of zeros with the same length asthe binary number. For example, if the binary number is 1010, create a new string of 0000.

4. For each bit in the binary number, write down its opposite value in the corresponding position in the new string of zeros. If the bit is a 1, write down a 0 in the corresponding position. If the bit is a 0, write down a 1 in the corresponding position.

5. Add 1 to the new string of zeros. To do this, start atthe rightmost (least significant) bit I and add 1 to it. Ifthe result is greater than or equal to 2, carry the 1 overto the next most significant bit and add it to that bit. Continue this process until you reach the most significant bit.

3.2 Flowchart

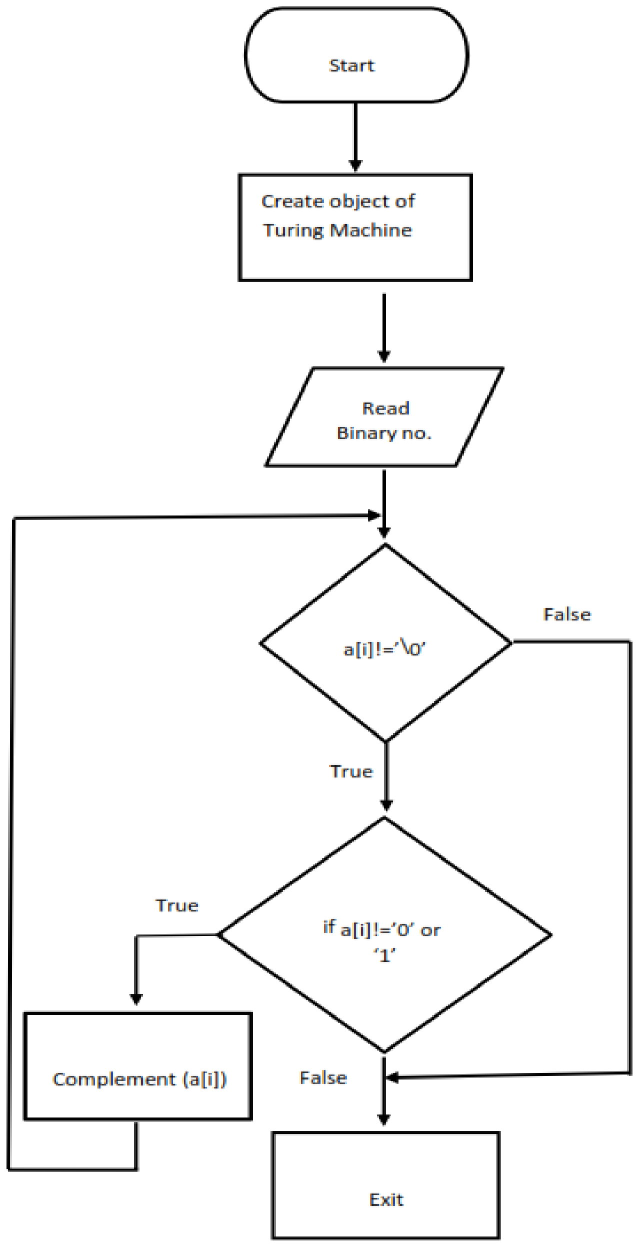
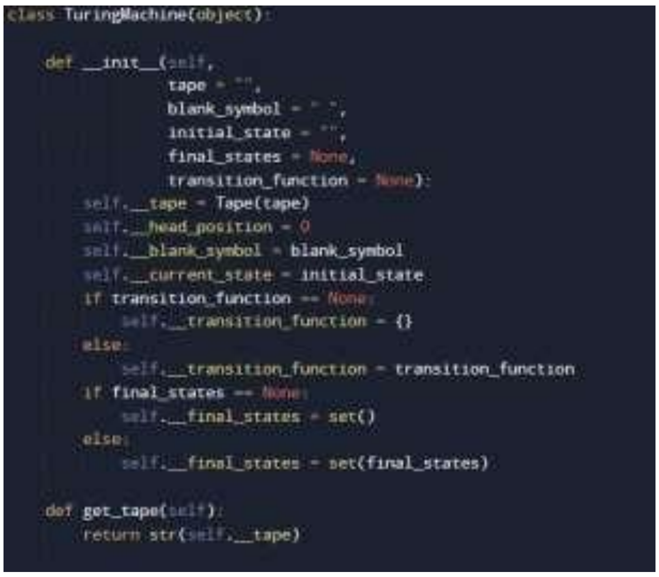


Fig-1: Flowchart to find 1's complement using Turing Machine

4. RESULTS AND DISCUSSIONS

In fig 2: the class Turing Machine performs the operation of creating a tape, a head to move on tape and the states of the Turing Machine.



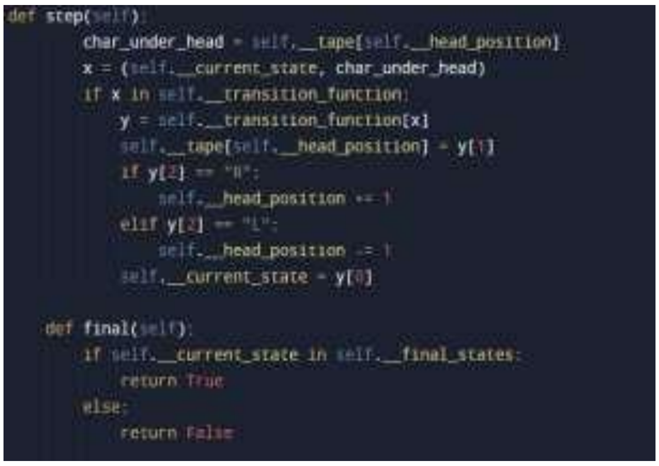
```
class TuringMachine(object):

    def __init__(self,
                 tape = "",
                 blank_symbol = " ",
                 initial_state = "",
                 final_states = None,
                 transition_function = None):
        self.__tape = Tape(tape)
        self.__head_position = 0
        self.__blank_symbol = blank_symbol
        self.__current_state = initial_state
        if transition_function == None:
            self.__transition_function = {}
        else:
            self.__transition_function = transition_function
        if final_states == None:
            self.__final_states = set()
        else:
            self.__final_states = set(final_states)

    def get_tape(self):
        return str(self.__tape)
```

Fig-2: Class Turing Machine

Fig 3: shows the conversion of the input string to its 1's complement by reading the string from the tape and converting 1's to 0's and 0's to 1's.

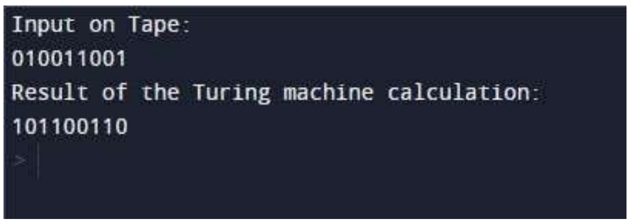


```
def step(self):
        char_under_head = self.__tape[self.__head_position]
        x = (self.__current_state, char_under_head)
        if x in self.__transition_function:
            y = self.__transition_function[x]
            self.__tape[self.__head_position] = y[1]
            if y[2] == "R":
                self.__head_position += 1
            elif y[2] == "L":
                self.__head_position -= 1
            self.__current_state = y[0]

    def final(self):
        if self.__current_state in self.__final_states:
            return True
        else:
            return False
```

Fig-3: Actual Conversion Logic

Fig 4: shows the output which is generated by the model. Here a simple binary operation of finding 1's complement of the number is done. The input binary string is converted to its 1's complement using the mechanism of the Turing Machine.



```
Input on Tape:
010011001
Result of the Turing machine calculation:
101100110
>
```

Fig-4: Output

5. LIMITATIONS

[1] The model proposed in the paper takes only a binary input to demonstrate the implementation of the Turing machine.

[2] This model performs only a single operation onthe given input and returns one's complement as anoutput.

6. CONCLUSION

We investigate existing models on the Turing machine and implemented a model using python that successfully simulates the Turing machine. Came concludes that a mechanism of the Turing machine can be implemented in a programming language.

7. FUTURE SCOPE

[1] Implementing more computation operations into the model.

[2] Adding a graphical user interface would make it more user-friendly.

8. REFERENCES

[1] Mohamed Hamada, "Turing Machine and Automata Simulators", Procedia Computer Science 18(2), May 2013, Doi: 10.1016/j.procs.2013.05.314

[2] Max Datchet, "Simulation of Turing machines by a regular rewrite rule", Theoretical Computer Science 103(1992) 409-420 Elsevier.

[3] F. C. Hennie and R. E. Stearns, "Two-Tapesimulation of multi tape Turing machines", Journal of the Association for Computing Machinery, Vol. 13, No. 4(October 1966), pp. 533-546

[4] B. Jack Copeland, Richard Sylvan, "Beyond the Universal Turing Machine", Australasian Journal of Philosophy, 1999.

[5] Rory A. Brittain, Nick S. Jones, Thomas E. Auldridge, "What would it take to build a thermodynamically reversible Universal Turing Machine? Computational and thermodynamic constraints in a molecular design", arXiv: 2102. 03388, 2021.

[6] Kuldeep Vayadande, Aditya Bodhankar, Ajinkya Mahajan, Diksha Prasad, Shivani Mahajan, Aishwarya Pujari and Riya Dhakalkar, "Classification of Depression on social media usingDistant Supervision", ITM Web Conf. Volume 50, 2022.

[7] Kuldeep Vayadande, Rahebar Shaikh, Suraj Rothe, Sangam Patil, Tanuj Baware and Sameer Naik," Blockchain-Based Land Record System", ITM WebConf. Volume 50, 2022.

[8] Kuldeep Vayadande, Kirti Agarwal, Aadesh Kabra, Ketan Gangwal and Atharv Kinage," Cryptographyusing Automata Theory", ITM Web Conf. Volume 50, 2022

[9] Samruddhi Mumbare, Kunal Shivam, Priyanka Lokhande, Samruddhi Zaware, Varad Deshpande and Kuldeep Vayadande," Software Controller usingHand Gestures", ITM Web Conf. Volume 50, 2022

[10]Preetham, H. D., and Kuldeep Baban Vayadande. "Online Crime Reporting System Using Python Django."

[11]Vayadande, Kuldeep B., et al. "Simulation and Testing of Deterministic Finite Automata Machine." International Journal of Computer Sciences and Engineering 10.1 (2022): 13-17.

[12]Vayadande Kuldeep, et al. "Modulo Calculator Using Tkinter Library." EasyChair Preprint 7578(2022).

[13]Vayadande, kuldeep. "Simulating Derivations of Context-Free Grammar." (2022).

[14]Vayadande, Kuldeep, Ram Mandhana, Kaustubh Paralkar, Dhananjay Pawal, Siddhant Deshpande,and Vishal Sonkusale. "Pattern Matching in File System." International Journal of Computer Applications 975: 8887.

[15]Vayadande, Kuldeep, Ritesh Pokarne, Mahalakshmi Phaldesai, Tanushri Bhuruk, Tanmay Patil, and Prachi Kumar. "Simulation Of Conway's Game Of Life Using Cellular Automata."SIMULATION 9,no.01 (2022).

[16]Gurav, Rohit, Sakshi Suryawanshi, Parth Narkhede, Sankalp Patil, Sejal Hukare, and Kuldeep Vayadande. "Universal Turing machine simulator." International Journal of Advance Research, Ideas, and Innovations in Technology, ISSN (2022).

[17]Vayadande, Kuldeep B., Parth Sheth, Arvind Shelke, Vaishnavi Patil, Srushti Shevate, and Chinmayee Sawakare. "Simulation and Testing of Deterministic Finite Automata Machine." International Journal of Computer Sciences and Engineering 10, no. 1 (2022):13-17.

[18]Vayadande, Kuldeep, Ram Mandhana, Kaustubh Paralkar, Dhananjay Pawal, Siddhant Deshpande,and Vishal Sonkusale. "Pattern Matching in File System." International Journal of Computer Applications 975: 8887.

[19]Vayadande, Kuldeep B., and Surendra Yadav. "A Review paper on Detection of Moving Object in Dynamic Background." International Journal of Computer Sciences and Engineering 6, no. 9 (2018): 877-880.

[20]Vayadande, Kuldeep, Neha Bhavar, Sayee Chauhan, Sushrut Kulkarni, Abhijit Thorat, and YashAnnapure. Spell Checker Model for String Comparison in Automata. No. 7375. EasyChair, 2022.

[21]Vayadande, Kuldeep, Harshwardhan More, Omkar More, Shubham Mulay, Atharva Pathak, and Vishwam Talnikar. "Pac Man: Game Development using PDA and OOP." (2022).

[22]Preetham, H. D., and Kuldeep Baban Vayadande. "Online Crime Reporting System Using Python Django."

[23]Vayadande, Kuldeep. "Harshwardhan More, Omkar More, Shubham Mulay, Atahrv Pathak, Vishwam Talanikar,"Pac Man: Game Development usingPDA and OOP"." International Research Journal of Engineering and Technology (IRJET), e-ISSN (2022): 2395- 0056.

[24]Ingale, Varad, Kuldeep Vayadande, Vivek Verma, Abhishek Yeole, Sahil Zawar, and Zoya Jamadar. "Lexical analyzer using DFA." International Journalof Advance Research, Ideas, and Innovations in Technology, www. IJARIIT. com.

[25]Manjramkar, Devang, Adwait Gharpure, Aayush Gore, Ishan Gujarathi, and Dhananjay Deore. "A Review Paper on Document text search based on nondeterministic automata." (2022).

[26]Chandra, Arunav, Aashay Bongulwar, Aayush Jadhav, Rishikesh Ahire, Amogh Dumbre, Sumaan Ali, Anveshika Kamble, Rohit Arole, Bijin Jiby, and Sukhpreet Bhatti. Survey on Randomly Generating English Sentences. No. 7655. EasyChair, 2022.