International Journal of Research in Circuits, Devices and Systems

E-ISSN: 2708-454X
P-ISSN: 2708-4531
IJRCDS 2022; 3(1): 13-16
© 2022 IJRCDS
www.circuitsjournal.com
Received: 24-10-2021
Accepted: 08-12-2021

Siddharth M
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India

Correspondence
Siddharth M
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India

Analysis and research on turing machines and automata simulator

Siddharth M

Abstract

A Turing machine is a model that computes mathematically on machines with dedicated symbols according to a set of rules enlisted in a table. Inspite of being a simple model, a Turing machine can be constructed in such a way that it can decode the algorithm behind it and perform the act of simulation. Its working helps in the learning of computational models. This is further established by showing how it could be used to analyse functions in the case of modern computers. Here the endeavor aims to make the learner easily build, modify a Turing machine and visualize the machine's actions. Finite state automata represent a mathematical model for many devices. Automata has many applications in multiple devices. In software designing, it can be used right from a basic editing of text to a more complex compiler. In computer related gaming, it can be used to frame quizzes, games and many others. In hardware designing, it can be used in modelling, the functioning of various machines like vending, escalators, playing, cooking of cereals etc. Finite state machines are the basic explanation to a many courses. The pushdown Automata which is a finite automata which can use the uppermost part of the stack to finalize the transition it would take is exemplified. Explanation of Pushdown Automata Simulator and Turing machine Simulator facilitating the visual learning through illustration of automata and performance of necessary operations is performed. The combined contribution of this work with the help of modern technology furthers the experimental evaluation and improves learning. A Directed graph is generally used to represent a Turing Machine. It is majorly used to simulate the universal turing machine as some sort of a device that converts variations in physical quantity to a electrical signal of the Turing machine and it consists of many inputs.

Keywords: Turing machine, finite automata, pushdown automata, simulators

1. Introduction

It is Dr. Alan Turing who first described the functions and working of Turing machine in a paper that he wrote during the year 1936. He wanted to explain to the people about the concept of what computability is and how we can compute things. This was the main reason behind why he wrote the paper and the explanation given to that. So the meaning of computational models or computation indicates whether there exists a typical algorithm which can be used to solve a particular problem or not.

He infamously remarked that should an algorithm exist for solving a particular problem then the Turing machine will undoubtedly be able to implement the same. This also helped to identify what an algorithm is. Right from the time of Euclid 2000 years ago nobody has been able to tell what is the proper definition for algorithm. So this idea is formally defined by the Turing Machine [1]. It can clearly tell whatever can be performed by this and whatever can't with the help of its functioning capacities.

1. Components

A Turing machine consists of following components.

1. An infinite tape

This Tape is partitioned into a number of cells with each of them having an input character which is taken by the machine either as an input or as an output. These which is displayed are a total subset of control alphabets [2]. The empty cells in the tape will have whichever characters possessed by it.

2. Control state

This belongs to a finite set of states Q. This has the responsibility to control whatever is being read from the current cell and give a well-defined detailed input regarding it.

3. Head

The Head will point to a cell in the machine. It can perform various operations like reading writing and altering the existing content whatever is carried currently inside the cell.

Any language that has a typical alphabet, grammar and ways to identify is called as a formal language. The list of strings on which some relevant conditions can be enforced to follow to involve the making of a string is called formal language [3]. The automata is the way to analyze a language mathematically. In this automata we take few states and rules to see how the language behaves and then convert it into a model that has the same rules that are being followed by this language. Then we take it by identifying the data with the help of the tape symbol. In case it is approved by the Turing Machine it will be referred to as a recursive finite language [4]. They comprise of a recursive finite enumerated set of languages.

1.1 Universal turing machine

A universal turing machine or UTM as it is commonly abbreviated is a commonplace to provide long-lasting solutions for problems related to computability. It has the details and also will be simulating the computation of the Machine on w. So because this many number of turing machines can be able to work together in a very synchronous manner and that will reduce the space complexity rather than each working individually.

It is defined by M = (Q, Σ, B, q0, F, δ, Ι) Where, Q = collection of internal states, Σ = input alphabets, B = unique symbol called blank, Ι = finite set of the tape alphabets, δ = transition function, F = final state, q0 = initial state. And transition function according to which Turing Machine performs its operation is defined by these 7 and the behaviors are directly dependent on these [5].

1.2 Working principle of the UTM

It's power and capabilities are equivalent to that if a computer. As we referred to earlier this Turing Machine has totally 3 elements namely the input output tape. Reading writing head that is bidirectional. Finite state controls which will point towards it. Here we observe that the Turing machine is also a finite automata that is connected to read and write head which can move either right to left or left to right. It can examine the cell in just one move. The header of the tape will analyze the symbol that has been obtained from the tape and will rewrite the symbol on the same tape. The input string can also be manipulated using these tapes that process such kind of information. When we refer to the tape we can observe that it's bidirectional information from Right R to Left L and vice versa will obtain a input. This input which when fed into the Turing machine will certainly alter the present symbol by following the transition rule which is categorically mentioned in details of the tape completing the procedure as per the preexisting rules. Then it moves on from one state to another checks the rule and ratified them [6]. It proceeds till the last state and performs the same operations aforementioned.

1.3 Simulating the turing machines in the universal turing machine

So in case of the Turing Machines, there is a string that is needed as an input and the altered string of some random Turing Machine is provided. When this activity of whatever is altered performs the same activity as that of the Turing machine it is testified. Turing machine will have its performance and maintain some sort of database and maintain a track record of so many things. It will need to maintain functions and important characteristics w.r.t the Turing Machines like it's the current symbol of input current location and state and all that [7].

Here we have to remember that there are some functions that can be assessed by the Turing Machines thoroughly and completely. There are some functions that cannot be. They are respectively referred to as computable and uncomputable functions.

2. Computable and uncomputable sets and functions

In theory of computation there are problems that the Turing machine by advancement of technology and invention can find algorithms to solve a particular problem. There are some cases where it cannot be done. So we can take the example of certain sets and functions to analyse what do these definitions actually mean.

2.1 Computable

They are those kinds of functions with which theory of computation and its algorithms can be computed or said to have a very unique and well defined elaborative algorithm. It has very clear guidelines on how it has to be computed. It must be coded using the computational model adhering to the rules.

2.2 Uncomputables

There are also functions which cannot be computed using regular rules and guidelines. This is because real numbers are uncountable or cannot be counted. The set of numbers in this case cannot be computed like the problem involving halting.

2.3 Real numbers

A real number is some number that continuously has a value. A real number can be computable if it can be approximated by some function that can be computed in such that given any integer n if it produces a finite integral values [8].

3. Finite automata

In layman terms, a finite state machine is something resembling a machine with a definite quantity of states and a control unit that can alter the present machine state to a different state in response to some provision of input. It is totally independent of the previous actions whose responses totally does not affect the style of its functioning [9].

When we notice the way in which it responds whenever an input is fed in to it we can classify into DFA called deterministic finite automata which has the ability to change itself across two different states in which most cases are either 0 or 1. There is another called NFA or non-deterministic finite automata which can move towards multiple states depending upon the situation of how the input is being fed into it. There are also two other cases called lambda NFA and it's a subset of NFA [10].

In fact NFA has 5 tuples where Q is a finite set of states q0 is a subset of q which also functions as the initial and starting state of the machine and F which is the subset of Q refers to whichever final and accepting states w.r.t the machine. We can observe that a series of input can be converted into the other keeping certain conditions constant. That is DFA can be converted into NDFA. It can also be involved through a common place [11].

3.1 Pushdown automata

A pushdown automaton (PDA) is a finite automata that makes use of an arrangement containing data. Pushdown automata differs from regular finite state machines in enlisted ways [12]:

1. They can use the uppermost part of the stack to take a call on which transition to take.
2. They can alter the stack as part of performance of a transition.

The controlling unit will notice the symbols given will scan provided in the feeder. While it is also maintain the relevance of stack using push and pop operations. It will also make distinctions between the push and pop operations using every single element and also regulate the flow of input [13].

4. Simulators

4.1 Pushdown automata simulator

The Pushdown Automaton Simulators gives opportunity for the learners to design an automaton in virtual mode and then use operations on it. At the time of these operations they can notice any changes that may occur to the automation. Taking a case they can check either whether it is accepting or rejecting the feeded inputs given while noting the respective changes in the automation states and in the content that is found in the stack [14]. They can also barge in and out and do self-outlaying to provide a better view for the automation. These ultimate operations are generally useful when the concerned automation is large [15].

The numbers from 1 to 7 in the pushdown automata are explained as follows.

1. **Number 1:** Denotes the action seeing area. Here in the zone the student can see the PDA state changing in every step during the PDA operating in a given input.
2. **Number 2:** Represents the stack view area. Here in the zone the user can watch the stack changing in every step while the PDA operating in a given input.
3. **Number 3:** Represents the control panel. In the control panel the user will be able to control the PDA animation speed, zooming towards and outwards,.
4. **Number 4:** Represents the input editing or displaying area. In this zone the learner will be able to insert any acceptable string as an input for the PDA. This input string will be analyzed totally by the PDA from left towards the right.
5. **Number 5:** Represents the PDA editing or displaying area. In this zone the user will be able to create or modify the PDA every state and every link, defining the start and final states, connecting the states, moving the states, changing the colour of the state, etc.
6. **Number 6:** Represents the controlling of the edit area. In the area the user will be able to save or load PDAs, choosing action to perform towards the edited PDA, choosing the editing color, etc.
7. **Number 7:** Represents a reserved zone for probable future extensions. After designation of the automaton, we can start simulation with the new input strings. The new input window will be displayed after we push the "start" button. Input strings present in the window and click "OK" button. Then PDA Simulator will be able to determine whether that automation will accept or reject those input strings. Users will be able to see the simulation in a step-by step manner or in TOTO [16].

5. Turing machine simulator

Generally physical explanations for Turing machines might not be sufficient for the students to understand the concepts related to it so we will have to simulate a Turing machine and explain its working and analysis a working virtual model in order to help the students in understanding the process of working of Simulator.

5.1 Using the automata simulator

Once the automation editing process is finished, the researcher can get an idea of algorithms NFA and DFA and more. So using this the researcher can feed in a random code and obtain the output using the given rules using interface animation and sound effects that further enable to the student and make it more appealing for the students to interact with the system. It is done using the automata simulator button which when switched on will simulate the input strings. There are different colors to represent different colors. The remaining chunk on the input in the string will be used to improve the visual clarity of the simulation which conjoins robot based motion

5.2 Using the robot simulator

The Robot Simulator will give the reader an opportunity to give a detailed description of the robot and helps the user to begin automation with the help of four distinct states. This gives it a new dimension and improve the visual clarity. The new versions of these will allow motion in different angles and go beyond the four distinct states and it will also give the new learner a virtual experience and help him to discover a new input [17].

In the last, in [18-25] readers, students, researchers can be found interesting works on emerging topics like Industry 4.0, Society 5.0, etc., and their importance for the smart era.

6. Conclusion

In this research paper we have extensively dealt with research and analysis on Automata Simulator. We thought of introducing the automata simulators those who wanted to know about simulation of automata and theory of computability. This will help the reader to know and analyze the thoughts that he had about automata simulators and working of Turing machines. It was done after taking several references into account and making the fundamentals clear to those who wanted to understand the concept in detail. We have also intended to educate the applications of Turing Machine and how it can be used in real life and understand the working of its components with typical examples. We understand that many activities in our real life including arithmetic calculator is an application of the above stated concept.

In studies pertaining to complexity and algorithmic information theory, testing and analyzing the efficiency of a software that was needed the concepts related to Turing Machines are applied. Though, due to the lack of infinite tape, we can effectively conclude that Turing machine is a conceptual or a hypothesis based device and not realizable totally, we can observe the studying it's functioning will enable us to apply this concept in identical cases.

7. References

1. Head E. ASSIST: A Simple Simulator for State Transitions. Master Thesis. State University of New York at Binghamton. Related website Available, 1998.
2. Java Team, Buena Vista Univ., http://sunsite.utk.edu/winners_circle/education/EDUHM01H/applet.html, 2008. 9 M. Mohri, F. Pereria, and M. Riley, AT&T FSM Library. Software tools, 2003. Available at:
3. Rodger S. Visual and Interactive tools. Website of Automata Theory tools at Duke University, 2006.
4. Transforming undergraduate education in science, mathematics, engineering and technology. In "Committee on Undergraduate Science Education", Center for Science, Mathematics, and Engineering Education. National Research Council ed. Washington, DC: National Academy Press, 1999.
5. Dr. G. Wilson. Ed., Constructivist Learning Environments: Case Studies in Instructional Design. Englewood Cliffs, NJ: Educational Technology, 1998. Wikipedia:
6. Herken R. (ed.) The Universal Turing Machine: A half-Century Survey, New York, Oxford University Press, 1988.
7. Jarvis J, Lucas JM. Understanding the Universal Turing Machine: An Implementation in JFLAp, Journal of ACM. 2008;23(5):180-188.
8. Kleene SC. General Recursive Function of Natural Numbers, Mathematics Annalen. 1936;112:727-742.
9. Kleinberg J. Computability and Complexity. In Computer Science: Reflections fromi the Field, Academics Press, 2004.
10. Lewis HR, Papadimitrinu CH. Elements of the Theory of Computation, Englewood Cliffs, N.S. Prentice-Hall, 1981.
11. Lin S, Radó T. Computer Studies of Turing Machine Problems, Journal of ACM. 1965;12:196-212.
12. Minsky M. Computation: Finite and Infinite Machines, Prentice-Hall, Inc., N.J., 1967.
13. Petrzold G. The Annotated Turing Machine: A Guided Tour through Alan Turing‘s Historic Paper on Computability and Turing Machines, Indianapolis, Indiana, Wiley Publisher, 2008.
14. Radó T. On Non-Computable Numbers, with an Application to the Entscheidungsproblem. In Proceedings of London Mathematical Society, Ser. 2. 1962;42:230-265.
15. Sumitha CH, Geddam KO. Implementation of Recursive Enumerable Languages in Universal Turing Machine. Int‘l journal of Computer Theory and Engineering. 2011;3(1):1793-8201, 153-157.
16. Turing AM. On Computable Numbers with Application to the Entscheidungsproblem. In Proceedings of London Mathematical Society. 1936;42:230-265; correction in 1937;43:544-546; reprinted in (Davis, 1965, 115-154).
17. Turing AM. Computability and λ-Definability. The Journal of Symbolic Logic. 1937;2:153-163.
18. Nair MM, Tyagi AK, Sreenath N. The Future with Industry 4.0 at the Core of Society 5.0: Open Issues, Future Opportunities and Challenges, International Conference on Computer Communication and Informatics (ICCCI), 2021, 1-7. Doi: 10.1109/ICCCI50826.2021.9402498.
19. Tyagi AK, Fernandez TF, Mishra S, Kumari S. Intelligent Automation Systems at the Core of Industry 4.0. In: Abraham A, Piuri V, Gandhi N, Siarry P, Kaklauskas A, Madureira A. (eds.) Intelligent Systems Design and Applications. ISDA 2020. Advances in Intelligent Systems and Computing, Springer, Cham., 2021, 13-51. https://doi.org/10.1007/978-3-030-71187-0_1.
20. Goyal Deepti, Tyagi Amit. A Look at Top 35 Problems in the Computer Science Field for the Next Decade, 2020. 10.1201/9781003052098-40.
21. Rekha G, Malik S, Tyagi AK, Nair MM. Intrusion Detection in Cyber Security: Role of Machine Learning and Data Mining in Cyber Security, Advances in Science, Technology and Engineering Systems Journal. 2020;5(3):72-81.
22. Mishra S, Tyagi AK. Intrusion Detection in Internet of Things (IoTs) Based Applications using Blockchain Technolgy, Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2019, 123-128. Doi: 10.1109/I-SMAC47947.2019.9032557.
23. Tyagi Amit Kumar, Nair Meghna Manoj, Niladhuri Sreenath, Abraham Ajith. Security, Privacy Research issues in Various Computing Platforms: A Survey and the Road Ahead, Journal of Information Assurance & Security. 2020;15(1):1-16.
24. Madhav AVS, Tyagi AK. The World with Future Technologies (Post-COVID-19): Open Issues, Challenges, and the Road Ahead. In: Tyagi AK, Abraham A, Kaklauskas A. (eds.) Intelligent Interactive Multimedia Systems for e-Healthcare Applications. Springer, Singapore, 2022. https://doi.org/10.1007/978-981-16-6542-4_22.
25. Amit Kumar Tyagi, Aghila G. "A Wide Scale Survey on Botnet", International Journal of Computer Applications. 2011;34(9):9-22. (ISSN: 0975-8887).