Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20240302003
Volume 3, Issue2 (May-August 2024), PP: 38-41.
www.indjcst.com
ISSN No: 2583-5300

Truth Table Generator

Prajval Balte[1], Aryan Inge[2], Sahil.Munot[3], Manasi Dhawale[4]
[1,2,3,4] UG Student, Department of Artificial Intelligence and Data Science, PVG's College of Engineering Technology and GKPIM, Pune, Maharashtra, India.

To Cite this Article: *Prajval Balte [1], Aryan Inge [2], Sahil.Munot[3], Manasi Dhawale [4], "Truth Table Generator", Indian Journal of Computer Science and Technology, Volume 03, Issue 02 (May-August 2024), PP: 38-41.*

Abstract: *Tables of truth are tools used in evaluating logical expressions and proving arguments. The name comes from the factual nature of the mathematical table in which all possible outcomes are represented. The Tractus Logico-Philosophicus, written by Ludwig Wittgenstein in 1918 and published in 1921, is largely credited for creating and popularizing the truth table. Truth Table Generator is a program which provides a truth table for the propositional logic expression entered by the user and also confirms whether the given expression is a tautology or not. Mathematics and science that rely on Boolean logic also use truth tables to show the truth or falsity of expressions or operations. This paper contains the proposed idea for evaluating and generating truth tables for propositions using python.*

Keywords: *Truth Table, Logical Expression, Ludwig Wittgenstein, Tractus, Philosophicus, Tautology, Boolean logic.*

I.INTRODUCTION

By listing all the possible values that the function may achieve, the truth table gives the breakdown of the logical function. It is a form of chart used to determine the validity of arguments and the true values of propositions. As an example, a very basic truth table would simply show the truth value of a proposition p and its negation, or opposite (denoted by the symbol ~ or ¬). There are several rows and columns in such a table, with the top row representing logical variables and combinations, with the number of rows and columns increasing as complexity increases.

To provide this truth table we use a Truth Table Generator. In this paper we study about the Truth Table Generator and how it works using the principles of a Truth Table.

II.LITERATURE REVIEW

Spreadsheet Generation of a Truth Table by John D. Sullivan [1], a fundamental technique for solving and analyzing a propositional logical expression or designing a truth table to determine the expression is discussed. The paper detected that the truth table technique of proving the logical statement will soon be unused because more automation mechanisms are accessible to use. Generating a truth table of expression through spreadsheets on a computer would be a revived method because the accuracy and speed of computation overcomes the tedious task. A spreadsheet can easily create truth tables by repeatedly building each table from the earlier table. Evaluating an expression in truth tables contains a pattern, which helps develop complex tables from an existing table. Evaluation of difficult expressions in complex large truth tables, saving and enforcing them in another large expression can be done effectively with computation methods instead of manually.

A paper by Josje Lodder, Bastiaan Heeren and Johan Jeuring, A Domain Reasoned for Propositional Logic is presented [2] on the topic of rephrasing the propositional formulas by using standard equivalences. The paper's main focus is to assess the feedback and changes on the proposition and review the feedback of other learning techniques provided for reconstructing propositional logic formulas. It guides the propositional expression task knowledge in a learning environment. This paper focuses on the propositional logic system LE

Proof Complexity in classical propositional logic by alasdair urquhart[3] did a notepaper survey about the complexity and difficulty of proofs in propositional logic. In the opinion of this writer, one of the most important properties of proof systems which we consider to be the defining condition is the feasibility of methods for determining if an asserted proof is in fact a proof, and if it is, then of what kind. A common framework for logical proofs of tautologies.

Truth set method and propositional logic by Yongjian Zhan [4] has come up with a different way of comparing the truth table method with the method of processing data they addressed that although the truth set method is mechanical, it is more effective than the truth table method and it is simple to execute for computing because of numerical calculations. According to them we can apply the same data sets and secure the matching outcome as the truth table method. A difference between it and the truth table is that it provides a dissimilar technique of processing data. Proofs of logical equivalence have the ability to authenticate by utilizing the truth set method. The paper has also approached a view that states that a truth set theory is a series of assignment indexes where the theories are true. Here, the main tool utilized to estimate arguments is the truth table technique and the formal proof technique.

Yongjian Zhan [5] has proved in his paper Truth set procedure for solving SAT problems that the design of a truth table is an inappropriate method to check the saturation of a propositional expression. The author proposed an advanced procedure:

"Procedure for setting the truth". It is an advanced truth table method for evaluating complex SAT formulae, which is a first depth search algorithm. Variable truth values are displayed on the left, and formula truth values appear on the right. The truth set method, although simpler, is very close to the truth table method. A truth set is the representation of a statement in the truth set method.

Propositional Logic of Context Saga BuvaE & Ian A. Mason [6] has examined contexts' basic logical characteristics. The paper provides a Hilbert-style proving system for a traditional context-based propositional logic language and discusses its syntax and semantics. a group of in that context specified or relevant propositional logical building blocks. The paper's primary objective is to develop the approach into a complete quantification logic. The advantage of the quantification approach is that it enables us to show relationships among contexts, operations on contexts, and state cancellation rules that explain how a single example from one context may be applied in another.

Propositional logic syntax and semantics by Mahesh Viswanathan [7] has come up with symbolic arrangements that attempt to express the propositions of error free reasoning and truth. They mentioned that to narrate any official language in a specific way there are three bits of knowledge like the symbols used to design sentences in that particular language according to the alphabet described and also the syntax narrates the regulation for grammatically correct examples of a sentence in that language. Lastly, the semantics shows the meaning of phrases in official language.

III.METHODOLOGY

3.1 Proposed System

Truth table generator based on the proposed system is to provide a convenient, user-friendly and computational method to generate truth tables of the following propositional, logical expressions. The system helps the beginner, also researchers and practitioners to study and design logic and proofs in the computer science field. The graphical representation and transition function table will help to provide a detailed and instinctive way. The generator can also select which symbols to use for the two truth values and the connectives. Supports all basic logic operators: negation, and, or, implication, converse of implication, no implication, converse no implication, xnor, tautology, and contradiction. This web application will be a powerful and versatile tool for working with propositional and logical expressions.

3.2 Flow Chart

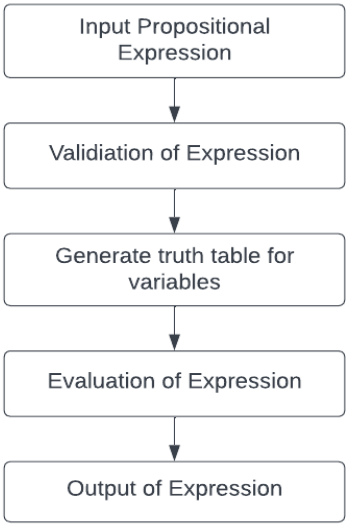

Fig.1 Flow chart of the project

3.3 Algorithm

step 1: Input a propositional logical expression.
step 2: Validate the expression.
Step 3: Generate truth values for the number of variables in a given expression.
step 4: Display the evaluated result with truth values of variables.
step 5: Evaluate the expression based on brackets from left to right.

IV.RESULTS AND DISCUSSIONS

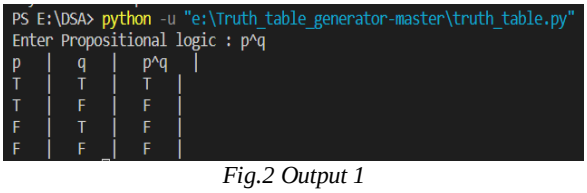

Fig.2 Output 1

In fig 2: The user enters propositional logic of conjunction (p^q), then the truth table for expression will be generated of all four cases.

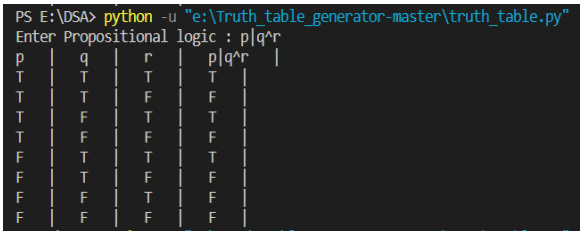

Fig.3 Output 2

In fig 3: Here the user enters propositional logic by using three operands and does conjunction and disjunction then the truth table is shown by machine.

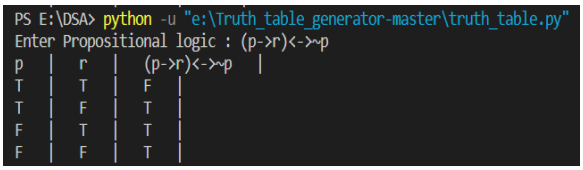

Fig.4 Output 3

In fig 4: Here the user has entered negation, implies an equivalence expression by using two operands.

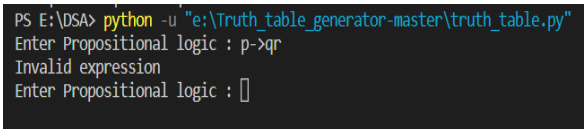

Fig.5 Output 4

In fig 5: If the user enters an invalid expression to generate the truth table, then it'll show the message as invalid expression and we can again enter another valid expression.

V.LIMITATIONS

- In this project, users can only add propositions in logical format.
- There is no suggestion system for auto-completion of propositions.
- Users cannot visualize the results of propositions.

VI.CONCLUSION

From this project we have concluded the Truth table generation of various simple as well as complex propositional statements. The project first validates the propositional expression and then evaluates it according to the precedence using a python model. The user can evaluate the propositional logic by using two or more operands and can implement conjunction, disjunction, negation, implies and conditional on a given expression.

VII.FUTURE SCOPE

As this project gives output after compiling the whole result. In the future, we can calculate the output of propositions generated by our project in real time. Also we can add a "Suggestion" feature for auto completion of logical propositions.

REFERENCE

1. *John D. Sullivan. (2000, September). Spreadsheet Generation of a Truth Table. ARMY RESEARCH LABORATORY.*
2. *Josje Lodder, Bastiaan Heeren and Johan Jeuring. (2016). A Domain Reasoner for Propositional Logic. Journal of Universal Computer Science, vol. 22, no. 8 (2016), 1097-1122*
3. *Alasdair Urquhart. Complexity of Proofs in Classical Propositional Logic. Logic from Computer Science. Proceedings of a Workshop (Nov., 13–17, 1989)*
4. *Zhan, Yongjian, Truth Set Method and Propositional Logic (February 7, 2022). Available at SSRN: https://ssrn.com/abstract=4028578 or http://dx.doi.org/10.2139/ssrn.4028578*
5. *Zhan, Yongjian, Truth Set Procedure for Solving SAT Problems (March 15, 2022). Available at SSRN: https://ssrn.com/abstract=4058278 or http://dx.doi.org/10.2139/ssrn.4058278*
6. *Propositional Logic of Context Saga BuvaE & Ian A. Mason, ga BuvaE & Ian A. Mason Computer Science Department Stanford University Stanford. AAAI-93 Proceedings*
7. *Mahesh Viswanathan, Propositional logic syntax and semantics (2018)- courses.engr.illinois.edu https://courses.engr.illinois.edu/cs498mv/fa2018/PropositionalLogic.pdf*
8. *Liu Yu 'e. (2000.) Discussion on Propositional Logic Incorporating Set Thought into Discrete Mathematics. J. Phys.: Conf. Ser. 1634 012087*

9. *Emil L. Post. Introduction to a General Theory of Elementary Propositions. American Journal of Mathematics Vol. 43, No. 3 (Jul., 1921), pp. 163-185 (23 pages). https://www.jstor.org/stable/2370324*
10. *Byjus, accessed on December 27, 2022. https://byjus.com/question-answer/what-is-truth-table-and-its-significance*
11. *Samruddhi Mumbare, Kunal Shivam, Priyanka Lokhande, Samruddhi Zaware, Varad Deshpande and Kuldeep Vayadande,"Software Controller using Hand Gestures", ITM Web Conf. Volume 50, 2022*
12. *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.*
13. *Vayadande, Kuldeep, et al. "Modulo Calculator Using Tkinter Library." EasyChair Preprint 7578 (2022).*
14. *VAYADANDE, KULDEEP. "Simulating Derivations of Context-Free Grammar." (2022).*
15. *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.*
16. *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).*
17. *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).*
18. *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.*
19. *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.*
20. *Vayadande, Kuldeep, Neha Bhavar, Sayee Chauhan, Sushrut Kulkarni, Abhijit Thorat, and Yash Annapure. 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. *Vayadande, Kuldeep. "Harshwardhan More, Omkar More, Shubham Mulay, Atahrv Pathak, Vishwam Talanikar,"Pac Man: Game Development using PDA and OOP"." International Research Journal of Engineering and Technology (IRJET), e-ISSN (2022): 2395-0056.*
23. *Ingale, Varad, Kuldeep Vayadande, Vivek Verma, Abhishek Yeole, Sahil Zawar, and Zoya Jamadar. "Lexical analyzer using DFA." International Journal of Advance Research, Ideas and Innovations in Technology, www. IJARIIT. com.*
24. *Manjramkar, Devang, Adwait Gharpure, Aayush Gore, Ishan Gujarathi, and Dhananjay Deore. "A Review Paper on Document text search based on nondeterministic automata." (2022).*
25. *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.*
26. *Kuldeep Vayadande, Kirti Agarwal, Aadesh Kabra, Ketan Gangwal and Atharv Kinage," Cryptography using Automata Theory", ITM Web Conf. Volume 50, 2022*