International Journal in IT & Engineering (IJITE)
Volume 6 Issue 10, September 2018 ISSN: 2321-1776 Impact Factor: 6.341
Journal Homepage: http://ijmr.net.in, Email: irjmss@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal

MAJOR PROBLEM – SOLVING APPLICATIONS OF DISCRETE MATHEMATICS USING LOGIC OPERATIONS IN COMPUTER SCIENCE: A CRITICAL STUDY

Sidhu[1], Dr. Amardeep Singh[2]

Department of Mathematics

[1,2]Shri Venkateshwara University, Gajraula (Uttar Pradesh)

Abstract

A discrete mathematics structure is included in computer science. It mainly includes the elementary logic and set theory, relations, functions, counting argument, graph theory, circuit graph, probability, and algebra. Discrete mathematics majorly focuses on proofs and problem-solving in computer science. The calculus and algebra are the main utilizations of discrete mathematics. The essential; systems learned in discrete math can be applied to a wide range of fields. Discrete mathematics is the section of science focused on the study of discrete objects. In this article, we critically analyze "problem-solving applications of discrete math with the help of logical operation in a computer". Algorithms are the rules by which a computer operates. These rules is created through the laws of discrete mathematics. A computer programmer uses discrete math to design efficient algorithms. This design includes applying discrete math to determine the number of steps an algorithm needs to complete, which implies the speed of the algorithm. Because of discrete mathematical applications in algorithms, today's computers run faster than ever before.

1. OVERVIEW

The set theory and logic are significant for defining the problem. In the formal analysis, it is a fact that state and determine the set of correct logical structures. The set and functions describe the structure of the problem domain and are used to document requirements and specifications. The set and functions define the entity types of distinct data types, classes of data structures, and database. Albeit, the maps, and set theory define the structure shared by the solution and the problem environment.

Discrete mathematics for programming designing", it ponders that discrete science in positive action of programming building. It presents inadequate solicitations, settled sets and complete selection. The standard of acknowledgment applies to sets with a mentioning continuously complex that the mentioning on the trademark numbers [1].

2. THE CIRCUIT DESIGN

The techniques of discrete mathematics are used throughout computer science. We have seen many small examples of the application of mathematics to computing and we have also used programming to help with mathematics. This section looks at the application of formal reasoning and a mathematical approach to computer science: the use of discrete mathematics to help with the process of designing digital circuits. In addition to applying discrete mathematics, it includes several useful tasks: precise specification of circuits, simulation, correctness proofs, and circuit derivations. Digital circuit design is a vast subject area, and there is not space here to cover all of it. Therefore it is considered to the class of digital circuits (combinational circuits, which don not contain flip flops). However, that restriction is made only to keep the chapter short; discrete mathematics is used heavily throughout the entire subjects of digital circuit design and computer architecture. [2]

3. LOGICAL OPERATIONS AND BOOLEAN FUNCTIONS

Logic operations also known as Boolean functions, part of Boolean algebra, are widely used in computer science, engineering and mathematics. There are different words and expressions used for them, like logic gates or bitwise operations, but the main principle is the same: performing logical operations on bits (0 and 1 values). Electronics is now part of nearly every engineering domain; therefore, engineers must have a minimum understanding of logic, bitwise operations. Most of the physics calculations are performed with decimal numbers. Because, we use decimal numbers for all the physical values (e.g., 10 A, 250 Nm, 120 km).
Computers use binary numbers to perform calculations. To recall how to convert a decimal number in a binary number, we follow the Decimal to Binary Conversion table.
In parallel with arithmetic operations (addition, subtraction, multiplication, division), there are also logic operations. These are used to evaluate if a logical expression is either true or false. In our examples, we are going to use two symbols *A* and *B*, called inputs. Each one of them can have either a true value (1) or a false value (0). After a logic operation is going to be performed on the inputs, we are going to obtain a result, with the symbol *Q*, called output. Similar to the inputs, the output *Q* can only have a true (1) or a false (0) value.
The logical state the value true, also called HIGH, is equivalent to the binary value 1. The logical value false, also called LOW, is equivalent to the binary value 0.

true	HIGH	1
false	LOW	0

The most common logic operations also called gates or operators) are:

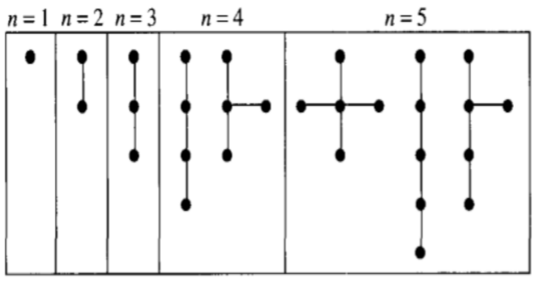
- NOT
- AND
- OR
- NAND
- NOR
- XOR
- XNOR

Each operation has a symbol assigned or a block diagram and a truth table. The symbol are used for designing graphical diagrams of logical operations. There are different standards for the symbols, the most common being ANSI (American National Standards Institute) and IEC (International Electro technical Commission). [3]

4. DEFINITION: TREES

A tree is a connected acyclic graph. A graph in which each connected component is a tree is called a forest. A vertex of degree one in a tree is called a leaf, or a terminal vertex. A vertex of degree higher than one in a tree is called an interior vertex.

The standard definition of trees, the vertex set, may not be empty. Later, a particular application will be facilitated by relaxing this restriction on the definition of a tree. In Figure, we see all the trees with fewer than six vertices. It shows a relationship between the number of vertices and the number of edges in a tree from this example.



A tree is an undirected graph G that satisfies any of the following preferable conditions:

- G is connected and contains no cycles.
- G is acyclic, and a simple cycle is formed if any edge is added to G.
- G is connected, but would become disconnected if any single edge is removed from G.
- G is connected, and the 3-vertex complete graph K_3 is not a minor of G.
- Any two vertices in G can be connected by a unique simple path.

If G has finitely many vertices, say n of them, then the above statements are also equivalent to any of the following conditions:

- G is connected and has n − 1 edges.
- G is connected, and every sub graph of G includes at least one vertex with zero or one incident edges. (That is, G is connected and 1-degenerate.)
- G has no simple cycles and has n − 1 edge.

An internal vertex (or inner vertex or branch vertex) is a vertex of degree at least 2. Similarly, an external vertex (or outer vertex, terminal vertex or leaf) is a vertex of degree 1.

An irreducible tree is a tree in which there is no vertex of degree 2. [4]

5. PROPOSITIONAL FORMULAS

For formulas the order of precedence from high to low is as follows: ¬ ∧,↑ ∨,↓ → ↔, Operators are assumed to associate to the right, that is, a∨b∨c means (a∨ (b∨c)).Parentheses are used only if needed to indicate an order different from that imposed by the rules for precedence and associativity, as in arithmetic where a(b+c) needs parentheses to denote that the addition is done before the multiplication. With minimal use of parentheses, the formulas above can be written: p→q↔¬q→¬p, p→(q↔¬(p→¬q)).

Extra parentheses may always be used to clarify a formula: (p∨q)∧(q∨r). The Boolean operators∧,∨,↔,⊕ are associative so we will often omit parentheses in formulas that have repeated occurrences of these operators: p∨q∨r∨s. Note →,↓,↑ are not associative, so parentheses must be used to avoid confusion. Although the implication operator is assumed to be right-associative, so that p→q →r unambiguously means p→(q →r), we will write the formula with parentheses to avoid confusion with (p→q)→r.[5]

6. PROPOSITIONAL LOGIC: FORMULAS, MODELS, TABLEAUX

Example: The strings associated with the two formulas are: ↔→pq→¬p¬q, →p↔q¬→p¬q and there is no longer any ambiguity.

The formulas are said to be in Polish notation, named after a group of Polish logicians. We see the fix notation more confortable to read because it is familiar from arithmetic, so polish notation is typically used only in the internal representation of arithmetic and logical expressions in a computer.

The advantage of Polish notation is that the expression can be evaluated in the linear order that the symbols appear using a stack. If we rewrite the first formula backward (reverse Polish notation): q¬p¬→qp→↔ , it can be directly compiled to the following sequence of instructions of an assembly language: Push q Negate Push p Negate Imply Push q Push p Imply Equiv The operators are applied to the top operands on the stack which are then popped and the result pushed.

7. OPERATOR ALTERNATES JAVA LANGUAGE

¬~ ! ∧ &&, && ∨ |, ‖ →⊃ ,⇒ ↔≡ ,⇔ ⊕ ≡ ^ ↑|

A Formal Grammar for Formulas

This subsection assumes familiarity with formal grammars. Instead of defining formulas as trees, they can be defined as strings generated by a context-free formal grammar. Definition: Formula in propositional logic is derived from the context-free grammar whose terminals are: • an unbounded set of symbols P called atomic propositions.

Definition: ∨|∧| → | ↔ | ⊕ |↑|↓ A formula is a word that can be derived from the non terminal. The set of all formulas that can be derived from the grammar is denoted F.

Derivations of strings (words) in a formal grammar can be represented as trees.

The word generated by a derivation can be read off the leaves from left to right. Example Here is a derivation of the formula p→q↔¬p→¬q in propositional logic; the tree representing its derivation [6]

8. CONCLUSION

The presentation of propositional logic was carried out in a manner that we will use for all systems of logic. First, the syntax of formulas is given. The formulas are defined as trees, which avoid ambiguity and imply the description of structural induction. The second step is to define the semantics of formulas. An interpretation is a mapping of atomic propositions to the values {T, F}. An interpretation is used to give a truth value to any formula by induction on the structure of the formula, starting from atoms and proceeding to more complex formulas using the definition of the Boolean operators.

The Boolean algebra: Equational reasoning

We now look at the form of equation reasoning. There are two fundamental ideas:

- The two values are the same by buildings up chains of equalities
- Second, we can substitute equals for equals in order to add a new link to the chain.

Chain equality relies on the fact that a = b and b= c, then we can deduce them formally that a= c. Example – the following chain of equations allow us to conclude that a=e:a=b =c=d=e .By substituting equals for equals means that we know x=y and we have an expression that contains x, then we can replace x by y without changing the value of the high expression.

Algorithms are the rules by which a computer operates. These rules are created through the laws of discrete mathematics. A computer programmer uses discrete math to design efficient algorithms. This design includes applying discrete math to determine the number of steps an algorithm needs to complete, which implies the speed of the algorithm. Because of discrete mathematical applications in algorithms, today’s computers run faster than ever before. In this, we have discussed the benefits of including discrete mathematics and have focused on two well-known topics like number theory. The properties of the set of integers are the subject of Number Theory.

The integers are the familiar mathematical structure that has lots of interesting to prove properties. It makes Number Theory an excellent place to start serious practice with the methods of proof. Moreover, Number Theory has turned out to have multiple applications in computer science.

In understanding discrete mathematics, we should comprehend the results presented here, concluded that the DM material should be understood with examples and applications from computer science, because the applications would enhance the understanding of discrete mathematics. The principal aim of logic in computer science is to develop languages to model the situations we encounter as computer science professionals, in such a condition that we can reason about them formally. Reasoning about situations means constructing arguments about them; we want to do this formally, so that the arguments are valid and can be defended rigorously, or executed on a machine.

In understanding mathematics we must comprehend, what makes a correct mathematical argument, a proof is a succession of steps which prompts to the desired conclusion Proofs are used to verify that computer programs produce the correct result, to establish the security of a system and to create artificial intelligence. Logic is interested in true or false statements and how the truth or falsehood of a statement can be determined from other statements

REFERENCES

[1]. Jean Galler (2008) Discrete Mathematics For Computer Science, Research Gate Publication

[2]. Tosuni (2015), "Graph Theory in Computer Science - An Overview", International Journal of Academic Research and Reflection, Vol. 3, No. 4.

[3]. Maurer, Stephen B, & Ralston, Anthony (2004), "Discrete Algorithmic Mathematics", Third Edition (Wellesley, MA: A.K. Peters).

[4]. John O'Donnell, Cordelia Hall and Rex Page (2006) Discrete Mathematics Using A Computer, Springer Publication

[5]. Rosen K. H, (2012), "Discrete Mathematics and Its Applications", 7th Edition available at: https://www.academia.edu/28429136/Rosen_Discrete_Mathematics_and_Its_Applications_7th_Edition, (visited on : september 09, 2019)

[6]. J. M. Wing (2006) Computational Thinking, Communications Of The ACM

[7]. S. Epp. Susanna (2004) Discrete Mathematics with Applications. Thomson-Brooks/ Cole

[8]. Biswa Purna Chandra (2005) Discrete Mathematics And Graph Theory, Prentice Hall Of Private Limited

[9]. Ari Ben Mordechai, "Mathematical logic for computer science, Springer