The Importance of Discrete Mathematics in Computer Science

from：

https://www.ukessays.com/essays/computer-science/the-importance-of-discrete-mathematics.php

The Importance of Discrete Mathematics in Computer Science

Published: 3rd October, 2016 **Last Edited:** 7th December, 2016

This essay has been submitted by a student. This is not an example of the work written by our professional essay writers.

BIRKBECK UNIVERSITY
The Importance of Discrete Mathematics in Computer Science
Introduction

Computer science is the study of problems, problem solving and the solutions that come out of the problem solving process, B. Miller and D. Ranum (2013). A computer scientist goal is to develop an algorithm, a step by step list of instructions in solving a problem. Algorithms are finite processes that if followed will solve the problem

Discrete mathematics is concerned with structures which take on a discrete value often infinite in nature. Just as the real-number system plays a crucial role in continuous mathematics, integers are the cornerstone in discrete mathematics. Discrete mathematics provides excellent modelling tools for analysing real-world phenomena that varies in one state or another and is a vital tool used in a wide range of applications, from computers to telephone call routing and from personnel assignments to genetics, E.R. Scheinerman (2000) cited in W. J. Rapaport 2013).

The difference between discrete mathematics and other disciplines is the basic foundation on proof as its modus operandi for determining truth, whereas science for example, relies on carefully analysed experience. According to J. Barwise and J. Etchemendy, (2000), a proof is any reasoned argument accepted as such by other mathematicians.

Discrete mathematics is the background behind many computer operations (A. Purkiss 2014, slide 2) and is therefore essential in computer science. According to the National Council of Teachers of Mathematics (2000), discrete mathematics is an essential part of the educational curriculum (*Principles and Standards for School Mathematics*, p. 31). K. H Rosen (2012) cites several important reasons for studying discrete mathematics including the ability to comprehend mathematical arguments. In addition he argues discrete mathematics is the gateway to advanced courses in mathematical sciences.

This essay will discuss the importance of discrete mathematics in computer science. Furthermore, it will attempt to provide an understanding of important related mathematical concepts and demonstrate with evidence based research why these concepts are essential in computer science. The essay will be divided into sections. Section one will define and discuss the importance of discrete mathematics. The second section will focus on and discuss discrete structures and relationships with objects. The set theory would be used as an example and will give a brief understanding of the concept. The third section will highlight the importance of mathematical reasoning. Finally, the essay will conclude with an overview of why discrete mathematics is essential in computer science.

Discrete Mathematics

According to K. H. Rosen, (2012) discrete mathematics has more than one purpose but more importantly it equips computer science students with logical and mathematical skills. Discrete mathematics is the study of mathematics that underpins computer science, with a focus on discrete structures, for example, graphs, trees and networks, K H Rosen

(2012). It is a contemporary field of mathematics widely used in business and industry. Often referred to as the mathematics of computers, or the mathematics used to optimize finite systems (Core-Plus Mathematics Project 2014). It is an important part of the high school mathematics curriculum. Discreet mathematics is a branch of mathematics dealing with objects that can assume only distinct separated values (mathworld wolfram.com). Discrete mathematics is used in contrast with continuous mathematics, a branch of mathematics dealing with objects that can vary smoothly including calculus (mathworld wolfram.com). Discrete mathematics includes graph theory, theory of computation, congruences and recurrence relations to name but a few of its associated topics (mathworld wolfram.com).

Discrete mathematics deals with discrete objects which are separated from each other. Examples of discrete objects include integers, and rational numbers. A discrete object has known and definable boundaries which allows the beginning and the end to be easily identified. Other examples of discrete objects include buildings, lakes, cars and people. For many objects, their boundaries can be represented and modelled as either continuous or discrete, (Discrete and Continuous Data, 2008). A major reason discrete mathematics is essential for the computer scientist, is, it allows handling of infinity or large quantity and indefiniteness and the results from formal approaches are reusable.

Discrete Structures

To understand discrete mathematics a student must have a firm understanding of how to work with discrete structures. These discrete structures are abstract mathematical structures used to represent discrete objects and relationships between these objects. The discrete objects include sets, relations, permutations and graphs. Many important discrete structures are built using sets which are collections of objects K H Rosen (2012).

Sets

As stated by Cantor (1895: 282) cited in J. L. Bell (1998) a set is a collection of definite, well- differentiated objects. K. H Rosen (2012) states discrete structures are built using sets, which are collections of objects used extensively in counting problems; relations, sets of ordered pairs that represent relationships between objects, graphs, sets of vertices and edges that connect vertices and edges that connect vertices; and finite state machines, used to model computing machines. Sets are used to group objects together and often have similar properties.

For example, all employees working for the same organisation make up a set. Furthermore those employees who work in the accounts department form a set that can be obtained by taking the elements common to the first two collections. A set is an unordered collection of objects, called elements or members of the set. A set is said to contain its elements. To denote that a is an element of the set A, we write a € A. For example the set O of odd positive integers less than 10 can be expressed by O = {1, 3, 5, 7, 9}. Another example is, {x |1 ≤ x ≤ 2 and x is a real number.} represents the set of real numbers between 1 and 2 and {x | x is the square of an integer and x ≤ 100} represents the set {0. 1, 4, 9, 16, 25, 36, 49, 64, 81, 100}, (www.cs.odu.edu).

Mathematical Reasoning

Logic is the science for reasoning, Copi, (1971) and a collection of rules used in carrying out logical reasoning. The foundation for logic was laid down by the British mathematician George Boole. Logic is the basis of all mathematical reasoning and of all automated reasoning. It has practical applications to the design of computing machines, to the specification of systems, to artificial intelligence, to computer programming, to programming languages and to other areas of computer science, K H Rosen, (2012 page 1).

Mathematical logic, starts with developing an abstract model of the process of reasoning in mathematics, D. W. Kucker page 1. Following the development of an abstract model a study of the model to determine some of its properties is necessary. The aim of logic in computer science is to develop languages to model the situations we encounter as computer science professionals, in such a way 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 understand what makes a correct mathematical argument, that is, a proof. As stated by C. Rota (1997) a proof is a sequence of steps which leads 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 (www.cs.odu.edu). Logic is represented by symbols to represent arbitrary statements. For example the following statements are propositions “grass is green” and “2 + 2 = 5”. The first proposition has a truth value of “true” and the second “false”. According to S. Waner and S. R Constenoble (1996) a proposition is any declarative sentence which is either true or false.

Many in the computing community have expressed the view that logic is an essential topic in the field of computer science (e.g., Galton, 1992; Gibbs & Tucker, 1986; Sperschneider & Antoniou, 1991). There has also been concern that the introduction of logic to computer science students has been and is being neglected (e.g., Dijkstra, 1989; Gries, 1990). In their article “A review of several programs for the teaching of logic”, Goldson, Reeves and Bornat (1993) stated: There has been an explosion of interest in the use of logic in computer science in recent years. This is in part due to theoretical developments within academic computer science and in part due to the recent popularity of Formal Methods amongst software engineers. There is now a widespread and growing recognition that formal techniques are central to the subject and that a good grasp of them is essential for a practising computer scientist. (p. 373). In his paper “The central role of mathematical logic in computer science”, Myers (1990) provided an extensive list of topics that demonstrate the importance of logic to many core areas in computer science and despite the fact that many of the topics in Myers list are more advanced than would be covered in a typical undergraduate program, the full list of topics covers much of the breadth and depth of the curriculum guidelines for computer science, Tucker (1990). The model program report (IEEE, 1983) described discrete mathematics as a subject area of mathematics that is crucial to computer science and engineering. The discrete mathematics course was to be a pre or co requisite of all 13 core subject areas except Fundamentals of Computing which had no pre requisites. However in Shaw’s (1985) opinion the IEEE program was strong mathematically but disappointing due to a heavy bias toward hardware and its failure to expose basic connections between hardware and software. In more recent years a task force had been set up to develop computer science curricula with the creation of a document known as the Denning Report, (Denning, 1989). The report became instrumental in developing computer science curriculum. In a discussion of the vital role of mathematics in the computing curriculum, the committee stated, mathematical maturity, as commonly attained through logically rigorous mathematics courses is essential to successful mastery of several fundamental topics in computing, (Tucker, 1990, p.27).

It is generally agreed that students in undergraduate computer science programs should have a strong basis in mathematics and attempts to recommend which mathematics courses should be required, the number of mathematics courses and when the courses should be taken have been the source of much controversy (Berztiss, 1987; Dijkstra, 1989; Gries, 1990; Ralston and Shaw, 1980; Saiedian 1992). A central theme in the controversy within the computer science community has been the course discrete mathematics. In 1989, the Mathematical Association of America published a report about discrete mathematics at the undergraduate level (Ralston, 1989). The report made some recommendations including offering discrete mathematics courses with greater emphasis on problem solving and symbolic reasoning (Ralston, 1989; Myers, 1990).

Conclusion

The paper discussed the importance of discrete mathematics in computer science and its significance as a skill for the aspiring computer scientist. In addition some examples of this were highlighted including its usefulness in modelling tools to analyse real world events. This includes its wide range of applications such as computers, telephones, and other scientific phenomena. The next section looked at discrete structures as a concept of abstract mathematical structures and the development of set theory a sub topic within discrete mathematics. The essay concluded with a literature review of evidence based research in mathematical reasoning where various views and opinions of researchers, academics and other stakeholders were discussed and explored. The review makes clear of the overwhelming significance and evidence stacked in favour for students of computer science courses embarking on discrete mathematics. Overall, it is generally clear that pursuit of a computer science course would most definitely need the associated attributes in logical thinking skills, problem solving skills and a thorough understanding of the concepts. In addition the review included views of an increased interest in the use of logic in computer science in recent years. Furthermore formal techniques have been acknowledged and attributed as central to the subject of discrete mathematics in recent years.

References

1. A. Purkiss 2014, Lecture 1: Course Introduction and Numerical Representation, Birkbeck University.
2. B. Miller and D. Ranum 2013. Problem Solving with Algorithms and Data Structures: accessed on [18.01.15]
3. Berztiss, A. (1987). A mathematically focused curriculum for computer science. Communications of the ACM, 30

(5), 356–365.

4. Copi, I. M. (1979). Symbolic Logic (5th ed.). New York: Macmillan

5. Core-Plus Mathematics Project 2014: Discrete Mathematics available at http://www.wmich.edu/cpmp/parentresource/discrete.html [accessed on 25.01.14]

6. D W Kucker Notes on Mathematical Logic; University of Maryland, College Park. Available at http://www.math.umd.edu/~dkueker/712.pdfAccessed on [24.01.15]

1. Denning, P. J. (chair). (1989). Computing as a discipline. Communications of the ACM, 32 (1), 9–23.

2. Dijkstra, E. W. (1989). On the cruelty of really teaching computing science. Communications of the ACM, 32 (12), 1398–1404.

3. Discrete and Continuous Data, (2008). Environmental Systems Research Institute, Inc. Available at http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=Discrete%20and%20continuous%20data [accessed on 18.01.15].

4. Discrete Structures (2010) available at http://www.cs.odu.edu/~toida/nerzic/content/schedule/schedule.html#day3 [accessed on 25.01.15]

5. Edward R. Scheinerman (2000), *Mathematics, A Discrete Introduction* (Brooks/Cole, Pacific Grove, CA, 2000): xvii–xviii." Cited in W. J. Rapaport (2013). Discrete Structures. What is Discrete Maths? available from http://www.cse.buffalo.edu/~rapaport/191/whatisdiscmath.html-20130629 accessed on [25.01.2015]

6. Galton, A. (1992). Logic as a Formal Method. The Computer Journal 35 (5), 431–440

7. Gibbs, N. E., & Tucker, A. B. (1986). A model curriculum for a liberal arts degree in computer science. Communications of the ACM 29 (3), 202–210

8. Goldson, D., Reeves, S., & Bornat, R. (1993). A review of several programs for the teaching of logic. The Computer Journal, 36 (4), 373–386.

9. Gries, D. (1990). Calculation and discrimination: A more effective curriculum. Communications of the ACM. 34 (3). 44–55.

16. http://www.cs.odu.edu/~toida/nerzic/content/intro2discrete/intro2discrete.html : Introduction to Discrete Structures --- What’s and Whys

1. IEEE Model Program Committee. (1983). The 1983 IEEE Computer Society Model Program in Computer Science and Engineering. IEEE Computer Society. Educational Activities Board

2. J. Barwise and J. Etchemendy, Language, Proof and Logic, Seven Bridges Press, New York, 2000, ISBN 1-889119-08-3.

3. J. L. Bell Oppositions and Paradoxes in Mathematics and Philosophy available at http://publish.uwo.ca/~jbell/Oppositions%20and%20Paradoxes%20in%20Mathematics2.pdfaccessed on [25.01.2015]

20. K. H Rosen 2012 Discrete Mathematics and its Applications, 7edn, Monmouth University.

1. Myers, Jr. J. P. (1990). The Central role of mathematical logic in computer science. SIGCSE Bulletin, 22 (1), 22–26.

2. Ralston, A. (Ed.) (1989). Discrete Mathematics in the First Two Years. MAA Notes No. 15. The Mathematical Association of America.

3. Ralston, A., & Shaw, M. (1980). Curriculum '78 - Is computer science really that unmathematical? Communications of the ACM, 23 (2), 67–70.

4. Rota, G.-C. (1997). The phenomenology of mathematical proof. Syntheses, 111:183-196.

5. S. Waner & S. R. Costenoble (1996) Introduction to Logic.

6. Saiedian, H. (1992). Mathematics of computing. Computer Science Education, 3 (3), 203-221.

7. Shaw, M. (Ed.) (1985). The Carnegie-Mellon Curriculum for Undergraduate Computer Science. New York: Springer-Verlag

8. Sperschneider, V., & Antoniou, G. (1991). Logic: A foundation for computer science International Computer Science Series. Reading, MA: Addison- Wesley

9. The National Council of Teachers of Mathematics (2000). Principles and Standards for School Mathematics.

10. Tucker, A. B. (Ed.) (1990). Computing Curricula 1991: Report of the ACM/IEEE-CS Joint Curriculum Task Force Final Draft, December 17. ACM Order Number 201910. IEEE Computer Society Press Order Number 2220