MATHEMATICAL REASONING IN COMPUTER SCIENCE

Diana GAPONCIC, Dominic FLOCEA

Technical University of Moldova

***Abstract**: Mankind reached all-time highs in what concerns science. This exponential development is especially thanks to mathematics and its contribution to what we know today as Computer Science and programming. Unfortunately, not everyone understands why mathematics are so important and why it's such an asked subject at schools and universities and why it's a must learn subject. Mathematics are practically everywhere now with the development of computer science and programming and we are very fortunate to study this interesting complex subject and contribute to its development as future software engineers.*

***Keywords**: mathematics, computer, science, engineering, programming.*

Introduction

Mathematics can have rather different relationship with people. Some love it, some hate it. But something rather sure is that every person in their life met at some point this "stranger", maybe it had a good impression on them maybe not such a pleasant one, it might be that with time they became friends or enemies. A very relevant question that circles around this "stranger" often between the youth is "Do I need to study this", "Will it help me in life?" or "When would I need to use this in the real life".

Actually, in most cases, people don't see the use of mathematics in their daily routine. That's why it is hard to explain why a person, has to struggle and study integrals, derivatives and other scary sounding for a lot of people words. In fact, the applicability of math in real life is unimaginably huge and if someone thinks of a career related to engineering, architecture, economics etc. he should consider brushing up his math background. "Applied mathematics" is the branch that deals with the application of mathematical methods in different fields like science, engineering, business, industry and only due to precise calculation and rigorously described processes we can trust bridges, breaking systems, we can have electricity, cars.

Let's talk about the historical background of mathematics. It is as old as mankind itself. It all started with counting, studying shapes, observing processes, but gradually, using logic and abstraction, mathematics has become the complex science we have today that had definitely helped us to reach the point we are at this very moment. With the time it became more sophisticated, it developed in many different branches of what we know today – Geometry, Algebra, functions, analytics and many more. It directly influenced the development of science and technological progress of mankind.

With the time a new branch of science was born as a tool for counting. It developed and found its applicability in the most unexpected areas. People used it as the foundation for their daring and courageous ideas making the development of what we know today as "Computer Science" possible. The development of this branch directly boosted, sky rocketed, threw straight into space and made possible complex computations which were too consuming for human beings. As a result, many enthusiastic people came with the idea to use magic black-boxes in order to prove some famous conjectures.

Some of them are:

- The four-color theorem
- the Kepler Conjecture
- Kepler conjecture, the problem of optimal sphere packing in a box
- 17-point case of the Happy Ending problem, 2006
- NP-hardness of minimum-weight triangulation, 2008
- Optimal solutions for Rubik's Cube can be obtained in at most 20 face moves, 2010
- Minimum number of clues for a solvable Sudoku puzzle is 17, 2012

Mathematicians struggled with proving them for years until some program prove them correct (or wrong). You can see how powerful is Computer Science for other areas and especially for mathematics.

Citing Mr O'Regan in book "A Brief History of Computing" [1]: "The increase in productivity due to the more advanced computerized technologies has allowed humans, at least in theory, the freedom to engage in more creative and rewarding tasks" and he is right, machine computation saves a lot of time, time that can be used on other tasks. And there it goes - programming - one of the most powerful tools in the history of mankind, if not the most. It is used everywhere, from gaming consoles to serious military defensive and

unfortunately offensive devices. How it developed in such a short time fascinates and tells a lot about the power we have as humans when we use our brain and work together.

1. Evolution of Programming

Living in this decade means that you are being surrounded by someone else's written code. It is in your microwave, TV, phone and even in the air in form of waves from our routers which again work based on someone else's written code. As software engineers we also have to write code and it is no easy task to master it. The reason being - programming and computer science changes fast and it's constantly evolving, from its very beginning. One of the founders of what we know today as programming is none other than Alan Turing who was the one that understood that programs were simply data and managed to set the foundation of modern computer programming. By 1945, he was writing programs that used loops, branches, stacks. Not even 10 years have passed and coding took big turns in its development. One of very important role in evolution of programming had Fortran (Formula Translation), a language that was developed in 1953 to perform high level mathematical, scientific computations. It became the foundation for many other languages and even now it is still used in automotive industries, in research institutes. In a few decades, programming developed a lot creating more and more languages to work with: Cobol, Basic, C, Pascal, Perl, Python, Ruby, PHP, Java, JavaScript, Ruby on Rails. All this programs have one thing in common, they are based purely on mathematical logic (and this hasn't changed for years), without which they couldn't even exist. Programmers strive to make them as simple and understandable as possible to use by writing languages like the ones previously mentioned.

This brings us to another significant name - Edsger Wybe Dijkstra. In 1960s, while working as a programmer, he realized that programming was getting too hard as the use of mathematics became more and more complex. Well truth to be told, then even the simplest programs, had too many details. Edsger Dijkstra came with the idea to use the the construction of the Euclidean hierarchy of postulates, theorems, corollaries and lemmas the way mathematicians do. That means, that he wanted programmers to use proven structures, to tie them together with their code and then prove the result itself to be true. The challenge was to find an algorithm of proving a program as a truthful one. He used induction and enumeration to prove the concept of iteration and sequential statements. His approaches were based on simple mathematical proofs.

Interesting thing to know about programming is that really it hasn't changed much in the last 50 years or so at its core. The main idea is that our tools with which we write the code became much better with the time. It became easier to write code, but the basic building blocks of a computer program remained the same on the machine level.

Robert C. Martin was correct when he affirmed the following: "(Programs) It's made of the same stuff. It's made of if statements, assignment statements, and while loops." in his book Clean Architecture [3].

In short, mathematics started its development as a counting tool, when Computer Science was something fantastic to people. As time passed, the concept of programming appeared, but the process of code writing was too difficult for a human, and that was the point when people used math in order to make the programs more accessible for people to write. At that moment, programming and Computer Science started to be used in almost everything, in building models, computing complex formulas, in statistics, physics, military etc. The huge development in technology lead to the birth of new concepts and methods in mathematics, that were later on used in Computer Science. So, the help was always mutual, and there is no technology and improvement without math and vice versa.

2. Maths in Development of Computer Science

"Computer science investigates general categories such as determinism, nondeterminism, randomness, information, truth, untruth, complexity, language, proof, knowledge, communication, approximation, algorithm, simulation, etc."

Theoretical Computer Science, Jurak Hromkovic **[**2]

Computer Science cannot be analyzed as a subcategory of a specific science, as it combines aspects from mathematics, engineering, natural science and many more, due to this, it should be considered and studied as an independent entity with a huge applicability. Even so, the importance of math behind most applications cannot be enough emphasized, some of them will be further discussed.

Statistics is about gathering data and trying to predict future outcomes. Mathematical methods that are used for this aim are multiple kind of regressions and fit lines. As the simplest example can serve Least-Square Regression, whose method is based on minimizing the sum of the squares of all point-line deviations. It is tightly bound with probability, which is the science that deals with the concept of likelihood of an event

to happen. Especially, an important application is related to the Bayes' formula, the conditional probability. How are these topics related to computer science? Simple, the answer is Artificial Intelligence and Machine Learning. Even if for non-tech people those computational sciences seem to be Black Boxes that do magic stuff, in fact, the magic lies in math itself. AI is about Matrix Algebra, that helps in converting data into something that an engineer can handle, probability, statistics and of course a lot of calculus, as this science is about training, about "learning" how close a value is to the real one.

Are there any doubts about the importance of mathematics in Computer Science remained? Then let's move on.

Number Theory is a branch of mathematics that deals with the study of integers numbers. Seems simple, doesn't it? Well, Not really. As today is the time when everybody keeps all their information online, the importance of cryptography is as huge as it has never been. Its applications include block chain, data-storing, digital signatures, https and many more. This hard-implementing and vital for online world science is just math, and the study of it begins with multiplications, divisibility, factorizations, primes which later on grow in more complicated things like rings and fields, cryptosystems like RSA, cryptanalysis and factorization.

Another important category of math is linear algebra. Its importance lies in the applicability in big data (which is huge in computer science), along with branches like basic probability, statics and others. Today, the development of computer science is directly related to the ability of processing and analyzing millions of data points, that's why a lot of research is done in the terms of optimization, numerical analysis and approximation algorithms. It is important to reach a point where mathematical models for big data are well defined using discrete math and analysis.

Keeping in mind things above, Graph Theory cannot be overlook, we couldn't fit all of the theory related to this, to put it short: "Graph theoretical ideas are highly utilized by computer science applications. Especially in research areas of computer science such data mining, image segmentation, clustering, image capturing, networking etc." from the book Applications of graph theory in computer science an overview [4].

So, is Computer Graphics and Visualization your kink? Be ready to learn a lot of trigonometry. Interested in Algorithms Analysis? Learn combinatorics. Thinking about a career related to data comprehension? Probability and Statistics are your points of interest in near future.

3. Personal Examples

Studying as a software engineer we stumbled across a lot of mathematical and programming challenges. As majority of software engineers do, we sometimes use programming to solve mathematical problems and at the same time we use mathematics to write code.

During school years we got used to the fact that people approach mathematics and computer science independently. Even if everybody knew that they are related, only a few would have been able to explain how. When coming to university, we started learning mathematics for computer science, and that was when all the pieces came together and formed an overview of the real importance and relation between those two.

We started with functions and sets which in a few lectures advanced in the study of mathematical logic used in computer science. At that moment we figured out how the ambiguities inherent in natural language can become a real problem and why we should define every detail in our reasoning. We started proving theorems, argument our hypotheses using famous methods and principles, we started being aware of the relations between everyday things, and look at everything from a mathematical point of view. We talked about number theory and modular arithmetic making first baby steps in understanding basic concepts of cryptography. We learned about graphs, about possible ways of representing data, about algorithms used unaware on a daily basis by every existing person. And then, when we felt ready enough, we dove into probability, we counted things, simulated experiments and even learned about data compression. We felt excited when we started applying derivative and integrals in our computer science lessons, that was, when it hit us, everything is related to mat

A good example in this context is the lesson when we learned about Method of Least Squares, an approximation method which helps with the prediction of the future data. This approach needs a decent amount of calculations, and while doing them, in order to prove our reasoning, we used induction, matrices and derivatives. On the other hand, after proving our assumptions, with the help of a code we got the results instantly and were able to change the initial data for other results and computations. This saved a lot of time.

That was it, a method with real application and huge role for all companies, that was done, using math and computer science.

It is common among people to think about math and programming as separated things, but they helped each other to develop and the huge success of science, robotics, IT now is a result of math contribution. A real engineer is a person that is able to detect problems, build mathematical models and use code for implementing ideas and understanding the concepts of all kinds of simulations and mutual applications of science.

References

1. Gerard O'Regan, *A Brief History of Computing*(2008)
2. Juraj Hromkovic, *Theoretical Computer Science*(2004)
3. Robert C. Martin, *Clean Architecture*(2017)
4. S.G.Shirinivas, S.Vetrivel, Dr. N.M.Elango, Applications of graph theory in computer science an overview(2010)