Preface

In computer science, as in other sciences, mathematics plays an important role. In fact, computer science has more in common with mathematics than with the traditional sciences. Consider physics, for example. Clearly, mathematics is right at the core of the subject, used to describe models which explain and predict the physical world. Without mathematics, most of modern physics would have been impossible to develop, since a purely qualitative analysis does not allow one to reason and deduce what logically follows from an observation or a hypothesised model. No manner of qualitative reasoning would have enabled Newton to formulate his law of gravity based on Kepler's claims about planetary motion. Computer science also exploits mathematical reasoning in a similar manner. Given a computer system, one can apply mathematical reasoning to prove that it will always calculate your paycheck correctly.

However, the link is even tighter than this. Computer science is a direct descendant of mathematics, and computers are nothing but the physical embodiment of mathematical systems developed long before the first modern computer was even conceived, let alone built.

In this book, we will be exploring the foundational mathematics which is necessary for the understanding of more advanced courses in computer science. Whether you are designing a digital circuit, a computer program or a new programming language, mathematics is needed to reason about the design—its correctness, robustness and dependability.

There are two distinct approaches used to present mathematical concepts and operators. In the first approach concepts and operators are defined in terms of properties which they satisfy. Based on these definitions, ways of computing the result of applying these operators are then developed and proved to be correct. In contrast, in computer science frequently one takes the opposite approach—one starts by defining ways of calculating the result of an operator, and then proves that the operator one knows how to compute satisfies a number of properties. The two approaches are, in the end, equivalent. However, given that this book is aimed at computer science students, the latter approach is usually adopted.

Finally, most sections are accompanied by exercises which are usually necessary as part of the learning process. Learning mathematics is surprisingly like learning to ride a bicycle. You must try to write formal expressions and proofs before you really understand the concepts you read about. Some of the concepts covered in this book may initially appear to be difficult. However, as you progress through the book and familiarise yourself with the concepts, you will hopefully start requiring less effort to follow the definitions and proofs, and begin to enjoy the beauty of how various concepts in mathematics can be built and reasoned about, based on just a small number of basic concepts.

Buenos Aires, Argentina Gordon J. Pace

http://www.springer.com/978-3-642-29839-4

Mathematics of Discrete Structures for Computer Science

Pace, G.J.

2012, XVI, 296 p., Hardcover
ISBN: 978-3-642-29839-4