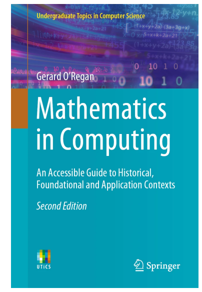


Mathematics in Computing:
An Accessible Guide to Historical, Foundational and Application Contexts, 2nd ed.
by **Gerard O'Regan**

Springer Nature Switzerland, 2020
458 pages, Softcover

Mini-review by [1]
David J. Littleboy (djl@alum.mit.edu)
Technical translator, retired
Tokyo, Japan

1 Overview

I think this book succeeds and would be useful as a text for an introductory and/or overview course for the mathematics used in computer science.

The book consists of 28 (yes, 28!) short chapters, each one of which consists of a short introduction, an exposition of the topic at hand, some review questions, and a summary. Each chapter could be covered in one or two lectures plus a TA-led session, and the professor could pick and choose from the later chapters. The questions are excellent, mostly conceptual, with an occasional calculation problem to keep the students on their toes.

The author deserves praise for his content selection. The first three chapters are necessary background/introductory material, and then individual chapters cover individual topics: algorithms, number theory, algebra, combinatorics, and a lot more. I was particularly pleased that the author covers both number

Mini-review by [1]
S. V. Nagaraj (svnagaraj@acm.org)
Vellore Institute of Technology
Chennai Campus, India

1 Summary

This book, authored by a prolific writer, is about the applications of mathematics in computing and places emphasis on historical contexts where applicable. Aimed at undergraduate students of computer science, the book provides helpful pedagogical features: key topics covered in each chapter are listed at the beginning, followed by an introduction, a summary, questions for review, and references at the end. Its second edition includes 28 chapters and runs to 458 pages, a significant expansion of the first edition, which has 16 chapters and runs to 288 pages. The chapter on foundations of computing introduces binary numbers, early computers such as the analytical engine, and symbolic logic. The chapter on algebra looks very briefly at structures such as monoids, groups, rings, fields and vector spaces. The chapter on coding theory focuses on block codes. The chapter on advanced topics in logic contains a brief description of theory and abstract algebra, not just as incidental material relevant to computer science, but as independent intellectual areas. I also liked the author's example of a proof using strong induction. I've seen too many books that fail to explain where strong induction is needed and used.

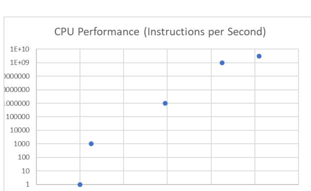
While I did notice some editing issues and typos, I like both the overall design of the book and the content covered. The writing style is on the dense, punchy, and declarative side, but not excessively so. Lastly, the book makes a point of introducing and crediting the figures who made the major contributions to the areas discussed, that is, it keeps its eye on the historical and foundational aspects it aims to.

2 A Historical Aside

While I strongly believe that the brevity of the chapters in this book is a feature, it doesn't allow space for deep historical analysis. In the introductory historical material, the author emphasizes the uninspired performance of early digital computers. How slow were they, actually?

While electro-mechanical computers could perform roughly one operation per second, the early tube computers could perform around 1,000 operations per second. That's three order of magnitude. Not too shabby.

The next gain of three orders of magnitude took another 30 years for mainframes and another 10 years after that for personal machines. And it took another 20 years for the next three orders of magnitude speed improvement (see the plot below); many commentators see those 20 years as a period of unequaled technological progress.



fuzzy logic, temporal logic, intuitionistic logic and undefined values. The chapter on overview of formal methods discusses topics such as the need for formal methods, approaches, the Z specification language, the B method, and model checking. The chapter on automata theory is very brief, running to just ten pages; finite state machines, pushdown automata and Turing machines are considered. The epilog essentially summarizes the concepts discussed in the chapters. The list of figures, glossary, and the index are helpful. The bibliography surprisingly has just one reference by the same author.

2 Opinion

It was not easy to find out what changes were made in the second edition of the book. The companion website gives some information, although inadequate: "This fully updated new edition has been expanded with a more comprehensive treatment of algorithms, logic, automata theory, model checking, software reliability and dependability, algebra, sequences and series, and mathematical induction."

Since 458 pages span the 28 chapters, readers will find that some of the chapters are very abbreviated. However, given the fact that entire books have been devoted to the topics covered in each of the chapters, and that the book is aimed at undergraduate students in computer science, the shortcoming may be excused. The book achieves its objective of providing a flavor of the mathematics used in computing, so the general reader will likely benefit from it as well.