

G.2 DISCRETE MATHEMATICS

G.2.0 General

GRIES, DAVID; AND SCHNEIDER, FRED B. 9407-0412

A logical approach to discrete math.

Springer-Verlag New York, Inc., New York, NY, 1993, 497 pp., \$44.95, ISBN 0-387-94115-0. [Texts and monographs in computer science.]

Gries and Schneider's textbook for the standard undergraduate course in discrete mathematics is a radical departure from the norm. The title says it all. The authors' approach to discrete math is based on formal logical reasoning methods.

The authors claim that students of discrete mathematics should first be taught the syntax and semantics of formal proofs using an equational style. This educational philosophy was expressed by the first author in a *CACM* paper a few years ago [1]. The content of the text reflects this philosophy. The first third of the book is divided into nine chapters that take the student through propositional and predicate calculus. The remainder of the book consists of 11 chapters—"Predicates and Programming," "A Theory of Sets," "Mathematical Induction," "A Theory of Sequences," "Re-

lations and Functions," "A Theory of Integers," "Combinatorial Analysis," "Recurrence Relations," "Modern Algebra," "A Theory of Graphs," and "Infinite Sets"—that apply formal proof techniques to traditional topics of the discrete math course. This organization is in marked contrast to that in most discrete math texts. The standard practice is to introduce logic quantification as one of many topics but then to neglect it as a tool for exploring the remaining topics.

How well does the book fulfill its stated goal? I have taught sophomore-level discrete mathematics for the past seven years from a variety of textbooks. Lately I had grown weary of the shotgun approach, where the course presented a group of seemingly unrelated discrete topics. Using this book convinced me that its approach is the best way to achieve a unified presentation of the topics.

Be forewarned, however, that if you adopt this book having previously taught discrete mathematics, you will experience a major paradigm shift. It will probably involve a trade-off of breadth for depth. You will have to eliminate some topics to make time for the development of calculational skills with formal methods. The new character of the course resembles that of traditional calculus. In the same way that calculus students progress through skill-based exercises based on single-variable differentiation, applications of the chain rule, integration, and so on in a linear fashion, discrete mathematics students progress through skill-based exercises based on textual substitution, Leibniz's rule, Boolean expressions, propositional calculus, proof techniques, general quantification principles, and predicate calculus. Only then are the more traditional topics of mathematical induction, sets, relations, and functions learned.

It makes a great difference, however, when students have the tools to study these topics with a greater depth of understanding. In most discrete math books, proofs are couched in English. It was always difficult to grade proofs

because I never knew if the student really understood a proof, but was simply having difficulty with the English, or if she or he simply did not understand the proof. Now, proofs are not ambiguous, and I can pinpoint errors in syntax as well as logic. Students have a deeper understanding of the proof process, and we can communicate with more precision. One side effect is that grading proofs is easier now, because I can be more rational in the granting of partial credit. Another benefit is that proofs, which used to be the most unpleasant part of the material, are now the favorite.

The authors have improved on standard formal methods notation by using a vertical bar instead of a colon to separate a quantified variable from the range. The usual notation $(\forall x : P : Q)$ is rendered $(\forall x|P : Q)$. This notation is especially nice because quantification concepts carry over to the specification of sets, which are rendered similarly as $\{x|P : E\}$ or, in the more familiar abbreviated form, $\{x|P\}$. Another benefit is that a quantified expression with a true range $(\forall x| : Q)$ is easier to read than $(\forall x :: Q)$.

The book has proven to be serviceable in class because of the quantity and quality of the exercises, and because of the extensive instructor's manual, with 310 pages of teaching

tips, notes, and complete solutions to all the exercises. The manual is especially helpful because this approach to teaching discrete math is so new. The authors have obviously tested the material in class, and their experience as related in the manual is practical.

The book has a few shortcomings, all of them minor. In terms of content, two early chapters on Hilbert-style proofs and formal logical systems are so specialized that they will rarely be used in the undergraduate discrete math course. I would have preferred material on finite state machines or grammar theory in their place. The treatment of digital circuits does not seem natural because the formal method paradigm is forced onto it, rather like a square peg in a round hole. A more conventional approach would probably serve the student better. In terms of form, the latter part of the book does not have enough figures, and the figures that are present are too small and not detailed enough. This is in contrast to (and, unfortunately, usually accompanies) the excellent typesetting in $\text{T}_\text{E}_\text{X}$. The inevitable typographic errors are few for a book of this size and complexity.

This book is a statement by Gries and Schneider on how discrete math should be taught, especially to computer science majors. Its adoption at our institution will probably cause a ripple effect throughout the curriculum. If teachers can assume the facility with formal methods presented in this book, later courses such as data structures and algorithms, programming languages, software engineering, and AI will benefit.

The book is a solid contender for the discrete math course and deserves serious consideration. Time will tell whether this approach takes hold in the undergraduate computer science curriculum. I hope it does.

— J. Stanley Warford, Malibu, CA

REFERENCES

- [1] GRIES, D. Teaching calculation and discrimination: a more effective curriculum. *Commun. ACM* 34, 3 (March 1991), 45–55. See CR, Rev. 9112-0981.

GENERAL TERMS: ALGORITHMS, THEORY