Macmillan Computer Science Series

Consulting Editor:

Professor F.H. Sumner, University of Manchester

- A. Abdellatif, J. Le Bihan, M. Limame, *Oracle - A User's Guide*
Ian O. Angell, *High-resolution Computer Graphics Using C*
Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using FORTRAN 77*
Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using Pascal*
M. Azmoodeh, *Abstract Data Types and Algorithms, second edition*
C. Bamford and P. Curran, *Data Structures, Files and Databases, second edition*
P. Beynon-Davies, *Information Systems Development, second edition*
G.M. Birtwistle, *Discrete Event Modelling on Simula*
Linda E.M. Brackenbury, *Design of VLSI Systems - A Practical Introduction*
Alan Bradley, *Peripherals for Computer Systems*
G.R. Brookes and A.J. Stewart, *Introduction to occam2 on the Transputer*
P.C. Capon and P.J. Jinks, *Compiler Engineering Using Pascal*
Robert Cole, *Computer Communications, second edition*
Eric Davalo and Patrick Naïm, *Neural Networks*
S.M. Deen, *Principles and Practice of Database Systems*
Joyce Duncan, Lesley Rackley and Alexandria Walker, *SSADM in Practice*
D. England *et al.*, *A Sun User's Guide, second edition*
Jean Ettinger, *Programming in C++*
J.S. Florentin, *Microprogrammed Systems Design*
A.B. Fontaine and F. Barrand, *80286 and 90386 Microprocessors*
Michel Gauthier, *Ada - A Professional Course*
M.G. Hartley, M. Healey and P.G. Depledge, *Mini and Microcomputer Systems*
J.A. Hewitt and R.J. Frank, *Software Engineering in Modula-2 - An Object-oriented Approach*
Patrick Jaulent, *The 68000 - Hardware and Software*
M.J. King and J.P. Pardoe, *Program Design Using JSP - A Practical Introduction, second edition*
Bernard Leguy, *Ada - A Programmer's Introduction*
M. Léonard, *Database Design Theory*
David Lightfoot, *Formal Specification Using Z*
A.M. Lister and R.D. Eager, *Fundamentals of Operating Systems, fifth edition*
Elizabeth Lynch, *Understanding SQL*
Tom Manns and Michael Coleman, *Software Quality Assurance, second edition*
B.A.E. Meekings, T.P. Kudrycki and M.D. Soren, *A book on C, third edition*
R.J. Mitchell, *C++ Object-oriented Programming*
R.J. Mitchell, *Microcomputer Systems Using the STE Bus*
R.J. Mitchell, *Modula-2 Applied*
Y. Nishinuma and R. Espesser, *UNIX - First Contact*
Ian Pratt, *Artificial Intelligence*
Pham Thu Quang and C. Chartier-Kastler, *MERISE in Practice*
A.J. Pilavakis, *UNIX Workshop*
E.J. Redfern, *Introduction to Pascal for Computational Mathematics*
F.D. Rolland, *Programming with VDM*
A.G. Sutcliffe, *Human-Computer Interface Design, second edition*
C.J. Theaker and G.R. Brookes, *Concepts of Operating Systems*

continued overleaf

M. Thorin, *Real-time Transaction Processing*

M.R. Tolhurst *et al.*, *Open Systems Interconnection*

I.R. Wilson and A.M. Addyman, *A Practical Introduction to Pascal, second edition*

Other titles

Ian O. Angell and Dimitrios Tsoubelis, *Advanced Graphics on VGA and XGA Cards Using Borland C++*

N. Frude, *A Guide to SPSS/PC+, second edition*

Peter Grossman, *Discrete Mathematics for Computing*

Percy Mett, *Introduction to Computing*

Tony Royce, *COBOL - An Introduction*

Tony Royce, *Structured COBOL - An Introduction*

Mathematical Foundations for Computing

G.P. McKeown and V.J. Rayward-Smith

*School of Information Systems
University of East Anglia, Norwich*

© G.P. McKeown and V.J. Rayward-Smith, 1995

All rights reserved. No reproduction, copy or transmission of this publication may be made without written permission.

No paragraph of this publication may be reproduced, copied or transmitted save with written permission or in accordance with the provision of the Copyright, Designs and Patents Act 1988, or under the terms of any licence permitting limited copying issued by the Copyright Licensing Agency, 90 Tottenham Court Road, London W1P 9HE.

Any person who does any unauthorised act in relation to this publication may be liable to criminal prosecution and civil claims for damages.

First published 1995 by
MACMILLAN PRESS LTD
Houndmills, Basingstoke, Hampshire RG21 2XS
and London
Companies and representatives
throughout the world

ISBN 978-0-333-48855-3
DOI 10.1007/978-1-349-10719-3

ISBN 978-1-349-10719-3 (eBook)

A catalogue record for this book is available
from the British Library.

To our sons,

Peter, Tom, Sam and Ben
and Mark and William

Contents

Preface	ix
1 Fundamentals	1
1.1 Sets	1
1.2 Numbers	15
1.3 Vectors	33
1.4 Matrices	49
1.5 Algebraic Structures	66
2 Functions and Relations	91
2.1 Introduction to Functions	91
2.2 Sequences and Recursion	112
2.3 Functions Over the Reals	132
2.4 Functional Programming	158
2.5 Relations	182
3 Logic	217
3.1 Propositional Logic	217
3.2 Predicate Calculus: Syntax and Semantics	236
3.3 Applications of Logic in Computer Science	256
3.4 Boolean Algebra	274
4 Algorithm Analysis Techniques	291
4.1 Systems of Linear Equations	291
4.2 Probability	307
4.3 Techniques for Solving Recurrence Relations	343
4.4 Algorithm Analysis	369
Index	395

Preface

Both the nature and the level of the mathematics required by students of computing have been the subject of debate for many years. In the 1970's, it became clear that many of the mathematical needs of computing students were not being met by traditional mathematics courses in universities. This realisation gave rise to the development of new courses in "discrete" mathematics, a term coined to cover a melding of such topics as sets, logic, induction and recursion, recurrence relations, combinatorics and discrete probability, graphs and algebras. Over the years, many excellent textbooks have been published in this area. However, we believe that simply substituting discrete mathematics for more traditional mathematics courses such as calculus and linear algebra also fails to meet the needs of computing students, and a major concern of ours is a trend in some quarters away from traditional mathematics. Discrete mathematics is of fundamental importance to computing but a knowledge of continuous mathematics is still essential. Indeed, Donald Knuth has coined a new term, "concrete mathematics", to describe the blend of continuous and discrete mathematics relevant to computing science. Within our text, therefore, we have a mix of discrete and continuous mathematics.

Given that the mathematical foundations of computing include both continuous and discrete mathematics, we now address the issue of the level to which this mathematics should be taught to computing students. There is no doubt that during the last 10 or 15 years, undergraduate degree programmes in the general area of computing have greatly diversified. One of the main features distinguishing one degree programme from another in this plethora is the level of inherent mathematics. Sadly, a very low mathematical content is often presented as a selling point for a programme. Of course, if the aim of a particular degree programme is to train students in the use of a variety of software packages for solving problems in a given application area, then the mathematics required by such students depends mainly on the domain of application. If the latter requires little mathematics, then so should the course. However, in writing this book, we have in mind those who regard themselves as students of computing. It is not a book for those who wish to avoid mathematics and want to treat computing merely as an applied science and who have no interest in rigour, accuracy or justification. Computer scientists need a quite sophisticated level of mathematics if they are to fully master their

subject. All students of computing need much of the mathematics given in this book. Those students wishing to study the analysis of algorithms to an advanced level need everything in this book.

In writing this book, we have tried to give a clear description of the fundamental mathematical concepts used by computer scientists while, at the same time, emphasising the need for careful justification. We therefore include proofs of all the major results. We do not present algorithms as “recipes” but instead carefully develop them and analyse their performance. We try and discuss the difficult issues, not sweeping them under the carpet.

Computer science has used many mathematical concepts but it has also raised new problems in mathematics and thereby generated whole new avenues of mathematical research. Computer science is not easy; some of the problems it raises are extraordinarily difficult. Our intention is to help train a new generation of computer scientists who appreciate that care and attention to detail is a necessary part of their professionalism. We hope that this new generation will feel confident enough to tackle some of the outstanding problems in computing, most notably in the fundamental problems of programming language semantics, program correctness and algorithm analysis and design.

The hardest problem we have faced is what to leave out. There are so many results in mathematics which have an application in computing. We have decided to concentrate on that core material which should see a student through his or her first degree. We hope this book will be used as a course text in University departments for first/second year undergraduate teaching. It should provide material for about a third of a student-year of study. It could either be presented as two short courses or as one longer course. In either case, there are some more advanced topics which can be viewed as optional; these are indicated with stars in section headings.

We also hope that the book will be kept by computer scientists as a useful reference book. We have thus adopted a style whereby definitions are readily accessible through the index and are generally followed by simple examples.

The first chapter of the book covers very fundamental material relating to data types found in computer science – sets, numbers, vectors and matrices. In the last section of this chapter, we introduce binary operators and discuss some classic concepts in abstract algebra. This gives us an opportunity to emphasise the importance of a formal proof in terms of an axiom scheme. If a computer scientist is to make any headway with the theory of the subject, understanding the meaning of “proof” is essential.

Recent years have seen some significant developments in computing which have repercussions for the mathematics syllabus for undergraduate computer scientists. The development of functional programming has motivated us to commit a large part of chapter two to a formal presentation of partial functions and an introduction to functional programming. Within chapter two, we also cover some real analysis and introduce various techniques used

in algorithm analysis. The last section of the chapter is devoted to relations – an essential subject for those involved in relational data bases or in graph theoretical applications.

The widespread use of formal methods, the increasing use of logic programming, expert systems and the need to prove programs correct, especially in safety critical systems, all underline the need for the computer scientist to have a good understanding of formal logic. We give an introduction to first order logic and its applications in chapter three. Again, we spend some time emphasising the construction of formal proofs.

Chapter four covers a number of application areas and some more advanced topics, the underlying theme being the analysis of algorithms. We use the linear algebra of chapter one to solve systems of linear equations, we use concepts from chapter two to solve recurrence relations and we present some probability theory with applications in computing. Finally, we draw all this material together with a number of case studies in algorithm analysis.

We have very much enjoyed writing this book; it has grown out of an earlier text we wrote, "Mathematics for Computing", published by Macmillans in 1982. Our current book, however, is not simply a rewrite of that earlier text; only about a quarter of the original material is to be found in this book. That is a reflection on the changing nature of computing. We hope the new generation of computer scientists will find this new book a help in their studies. We wish them every success!

The authors wish to express their thanks to Dr. Stephen Rush, whose expertise in the use of $\text{\LaTeX}$ greatly eased the task of typesetting this book.