The Book Review Column[1]
by William Gasarch
Department of Computer Science
University of Maryland at College Park
College Park, MD, 20742
email: gasarch@cs.umd.edu

In this column we review the following books.

1. **Cryptography in C and C++** by Michael Welschenbach. Review by S Terai. This is a book on cryptography with far more emphasis on practical issues then most books on crypto (at least more than those usually reviewed in this column).

2. **A Beginner's Guide to Discrete Mathematics** by W.D. Wallis. Review by James V. Rauff. This is a no-frills discrete math. To quote the review: "The material is presented without sidebars, historical notes, photographs, or other distractions common in lengthier and more expensive discrete mathematics texts."

3. **Handbook of Elliptic and Hyperelliptic Curve Cryptography** Edited by H. Cohen and G. Frey. Review by Larry Washington. Is this Everything you wanted to know about Elliptic and Hyperelliptic Curve Crypto. Or is it more than you want to know? Read the review to find out!

4. **The Game's Afoot: Game Theory in Myth and Paradox** by Alexander Mehlmann. Review by Danny Krizanc, 2008. This is an introduction to Game Theory using examples from literature such as Sherlock Homes.

5. A joint review of

   (a) **Introducing Game Theory and Its Applications** by Eliott Mendelson, and

   (b) **Game Theory and Strategy** by Philip D. Straffin.

   The joint review is by William M. Springer II. These are both elementary books on Game Theory.

6. **Semantic Integration of Heterogeneous Software Specifications** by Martin Grobe-Rhode. Review by Maulik A Dave. What do you do if different parts of your specifications are given in different languages? You would read this book.

Books I want Reviewed

If you want a FREE copy of one of these books in exchange for a review, then email me at gasarchcs.umd.edu

Reviews need to be in LaTeX, LaTeX2e, or Plaintext.

Books on Algorithms and Data Structures

1. *Algorithms on Strings* by Crochemore, Hancart, and Lecroq.
2. *Algorithms for Statistical Signal Processing* by Proakis, Rader, Ling, Nikias, Moonen, Proudler.
3. *Nonlinear Integer Programming* by Li and Sun.
4. *Planning Algorithms* LaValle.
5. *Binary Quadratic Forms: An Algorithmic Approach* by Buchmann and Vollmer.
6. *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis* by Dey

Books on Cryptography, Coding Theory, and Security

1. **Cryptanalytic Attacks on RSA** by Song Yan.
2. *Privacy on the Line: The Politics of Wiretapping and Encryption* by Diffie and Landau (updated and expanded edition).
3. *Concurrent Zero-Knowledge* by Alon Rosen.
4. *Cryptography and Computational Number Theory* edited by Lam, Shparlinski, Wang, Xing.
5. *Coding, Cryptography, and Combinatorics* edited by Feng, Niederreiter, Xing.
6. *Formal Correctness of Security Protocols* by Bella
7. *Coding for Data and Computer Communications* by David Salomon.
8. *Block Error-Correcting Codes: A Computational Primer* by Xambo-Descamps.

Combinatorics Books

1. *Graph Theory* by Bondy and Murty.
2. *A Course in Enumeration* by Aigner.
3. *Enumerative Combinatorics* by Charalambides.
4. *Combinatorics of Permutations* by Bona
5. *Graphs and Discovery: A DIMACS Workshop* Edited by Fajilowicz, Fowler, Hansen, Janowitz, Roberts. (About using software to find conjectures in graph theory.)
6. *Combinatorial Designs: Constructions and Analysis* by Stinson.

7. *Computationally Oriented Matroids* by Bokowski

Auction Theory and Game Theory

1. *Putting Auction Theory to Work* by Paul Milgrom.

2. *Game Theory: Decisions, Interactions, and Evolution* by James Webb.

3. *Superior Beings: If they exist, how would we know?* by Brams. (This is a Game Theory book about Games where God may be one of the players.)

Logic and Verification Books

1. *Elements of Finite Model Theory* by Libkin.

2. *Software Abstractions: Logic, Language, and Analysis* by Jackson.

3. *Formal Models of Communicating Systems: Languages, Automata, and Monadic Second-Order Logic* by Benedikt Bollig.

Misc

1. *How to Prove it: A structured approach* by Velleman.

2. *Automata Theory with Modern Applications* by James Anderson.

3. *Difference Equations: From Rabbits to Chaos* by Cull, Flahive, and Robson.

4. *Chases and Escapes* by Nahin. (This is the math behind pursuits- if a ship is moving and another ship wants to intercept it ....)

5. *Geometric Algebra for Computer Scientists: An Object Oriented Approach to Geometry* by Dorst, Fontijne, and Mann.

Review[2] of Cryptography in C and C++
Author of Book: Michael Welschenbach
Publisher: Apress, 2005. $59.99, Hard Cover, 471 Pages

Reviewer: S Terai, School of Advanced Technology, Algonquin College, Ottawa, Canada
sterai@hotmail.com

1 Introduction

Cryptography is a science of transforming data into an unreadable form. The transformation is done with a key, only an indented recipient(s) who posses the key is able to bring the data back to its readable form. Cryptography, an art for more than two thousand years is now, with a strong foundation in mathematics, a science.It has been a pleasure reviewing the unique book. Addressing mathematical concepts, algorithms and C/C++ code is not a simple task; the author handles them well throughout the book. This book is translated from German by David Kramer. Although well translated there are a few sentences that could be revised, the meaning becomes apparent on a second read. Certain colloquial terms and phrases could have been avoided, considering the academic audience of the book. Comprising of eighteen chapters divided into two sections and an additional section devoted to appendices, chapters are short to facilitate the division of the material.

2 Coverage

2.1 Chapter 1: Introduction

With an introduction on set theory the relevance of number theory in cryptography is highlighted. The author makes references to other chapters in the book that discuss topics in detail. By providing a cursory discussion on algorithms such as the Euclidean Algorithm, Sieve of Eratosthenes, rules, principles, laws and axioms the reader is made aware of the mathematics required in cryptography.Five tables in the chapter list the functions available in the software that is downloadable from the publisher s website www.apress.com. This section also provides some information on compiling the code on certain platforms; the code can be run on at least ten different platforms. The software is titled FLINT/C, an acronym for Functions for Large Integers in Number Theory and Cryptography. I have tried compiling the code on Linux using gcc using the makefile and running the rsademo program, with two compiler warnings, it works.

2.2 Chapter 2: Number Formats

The Representation of Large Numbers in C. A four page chapter justifies the use of static data structures to represent natural numbers several hundred digits long. Dynamic memory allocation is not preferred, because of the overhead. Ordering of the digits is discussed. If the address is constant changes in the number can be made by simply allocating additional digits, the digits are represented in increasing order with increasing memory addresses.

[2] Michael Welschenbach©2008

2.3 Chapter 3: Interface Semantics

The treatment of leading zero s overflow and underflow is discussed. The error codes generated by arithmetic operations are listed. This is a short chapter of two and a half pages.

2.4 Chapter 4. Fundamental Operations

An interesting treatment of arithmetic operations on large integers. Techniques for multiplying large numbers such as Schonhage and Strassen s procedure versus Karatsuba s method are compared; the author chooses to use the Karatsuba s method. A progression from a simple method of multiplication to more optimized techniques is mentioned, focusing on reducing the number of multiplications and consequently CPU time. Generous references are made to Donald E Knuth s Art of Computer Programming Vol 2, Seminumerical Algorithms. Squaring two integers though much simpler than multiplying two integers needs to deal with a carry, code segments are well explained which makes up for minimal comments in the code.

2.5 Chapter 5: Modular Arithmetic- Calculating with Residue Classes

An entire chapter is devoted to this topic considering its extensive use in cryptography.

2.6 Chapter 6: Where All Roads Meet: Modular Exponentiation

Algorithms for $a^e \pmod m$ are illustrated, from elementary binary algorithms, M-ary exponentiations, Montgomery reduction and exponentiation. This serves as a foundation for understanding the computation for RSA and Diffie Hellman s (DH) protocol for key exchange. A section on Cryptographic Application of Exponentiation is included with brief explanation of man-in-the-middle attack. References are made to Stallings Cryptography and Network Security, 2nd edition, considering the book was published in 2005, an updated reference to a new edition, third or forth, would be preferable. ISAKMP/Oakley are mentioned as protocols overcoming DH weaknesses, considering Oakley is an IPSec protocol a cursory treatment is understandable in a book dedicated to Cryptography. Finally Taher El-Gamal protocol and key generation is explained.

2.7 Chapter 7, 8, 9

These chapters deals with bitwise and logical Functions, input, output, assignment, and conversion. These chapters explain the building of a toolbox of arithmetic functions. The term dynamic registers is used to name a structure for data management using dynamic memory allocation.

2.8 Chapter 10: Basic Number-Theoretic Functions

This chapter deals with implementation of algorithms; calculating gcd using extended Euclid s algorithm, lcd, multiplicative properties of residue class rings, identification of quadratic residues, square roots, Chinese remainder theorem and identification of prime numbers is discussed among other topics. Two important theorems in Number theory, Fermat s Little Theorem and Euler s theorem are discussed. As footnote it is mentioned about the non dependence on P and NP completeness. A section is devoted to primality testing a table lists the ten largest known primes. The sieve of Eratosthenes and probabilistic primality using Solvay-Strassen and Miller Rabin methods are illustrated. The code segment printed on page 226 had some typographical errors. The website on Stallings book provides interesting links on prime numbers and forecasts the time frame on finding a prime number consisting of a billion digits.

2.9 Chapter 11: Rijndael: A Successor to the Data Encryption Standard

These chapter Starts with an overview of the selection criteria for AES; the basic for deciding on AES as a standard, security speed, memory requirements and implementation in hardware are discussed. A short comparison with DES and Feistel model is mentioned. I would have liked to see an entry on Feistel model in the index. The use of S-Box and the layering of the algorithm are discussed. Decryption in Rijndael is not the same as encryption, since it does not use the Feistel Model, section 11.9 is devoted to decryption. Finally a short section on modes, with a short explanation on three modes; Counter, CCM and RMAC. ECB, CBC, CFB OFB are mentioned but not explained.

2.10 Chapter 12: Large Random Numbers

This is a well written chapter; it starts with an introduction and mentions the difficulty of generating true random numbers on computers. After a short section on Simple Random Number Generation a large section is devoted to Cryptographic Random Number Generations including using AES. Another section deals with Quality Testing.

2.11 Chapter 13: Strategies for Testing LINT

A large number of test routines are provided which are also listed in the book.

2.12 Chapter 14, 15: Let C++ Simplify Your Life- a tutorial to C++

There are sections on constructors and overloaded operators. These concepts are illustrated with LINT code.

2.13 Chapter 16: Error Handling

Throughout the code error handling is carried out with the use of exceptions. Again, the author tries to explain C++ concepts using examples from LINT.

2.14 Chapter 17: An Application Example– The RSA Cryptosystem

A chapter to celebrate the book. It includes a section on Digital Signatures and weaknesses that a user should be aware of in implementing the algorithm.

2.15 Chapter 18: Do it yourself LINT

Validation of the C++ class LINT and tests for C functions are provided.

2.16 Chapter 19: Approaches for Further Extensions

Suggestions to modifying the code to utilize a 64 bit word length are suggested.Appendices include Directory of C/C++ functions in LINT, Macros, Computation time taken by several C functions, Mathematical notations used in the book. An appendix on packages available on number theory is useful. Intended Readership: The chapters do not have any exercises, review questions or problems. It is suitable as an accompaniment text for a cryptographic course or project work. It does have an extensive list of references

3 Summary

A well written book that addresses the intended purpose. With a good suite of test routines it does serve as an industrial strength API, however some known weakness in algorithms would benefit the reader. A beginner or intermediate level C/C++ programmer can follow the text. Concepts in C++ such as assertions are explained which could serve as a review for a beginner programmer. Understandably the book does not discuss ECC, an industrial strength Crypto API with ECC is a non trivial task. The book focuses on asymmetrical algorithms with one chapter devoted to symmetrical algorithm - Rijndael. Certain concepts such as modes are not explained in this text, which makes it necessary for the reader to refer to other books on Cryptography. The author has devoted a chapter in discussing C++, instead I would have preferred a good treatment of symmetrical encryption.

Review[3] of
A Beginner's Guide to Discrete Mathematics
Author of book: W.D. Wallis
Published in 2003 by Birkhuser, 367 pages

Reviewer: James V. Rauff
Department of Applied Mathematics and Computer Science
Millikin University

Overview

This book introduces the basic topics of discrete mathematics to students of mathematics and computer science. The text is written at a low level of mathematical sophistication. It is appropriate for first-year students in mathematics and computer science. Sample problems and solutions are presented throughout the text. After each sample problem is a similar practice exercise. The solutions to all the practice exercises are collected in the back of the book. In addition, the book provides many exercises for each section of material. Most of these are easy and the answers to the odd-numbered exercises are collected in the back of the book. Wallis' exposition is direct and uncluttered. He says what needs to be said and no more. The material is presented without sidebars, historical notes, photographs, or other distractions common in lengthier and more expensive discrete mathematics texts. Theory is kept at a minimum in favor of applications and technique.

[3] James V. Rauff ©2008

1 Summary of Contents

The text includes most of the topics one would expect of a discrete mathematics text, but some things are unexpectedly absent. Here is a chapter-by-chapter summary.

1. Chapter 1. Properties of Numbers (30 pages). This chapter includes summation notation, number bases and scientific notation. Also included is an introduction to how numbers are stored in computers. The IEEE754 standard is a notable entry in this chapter.

2. Chapter 2. Sets and Data Structures (34 pages) Here we find propositional logic, set theory (including Venn diagrams) and mathematical induction. Wallis discusses alternative proof techniques for set theory.

3. Chapter 3. Boolean algebra & Circuits. (26 pages) The main points of emphasis in this chapter are disjunctive normal form, Karnaugh maps, and combinatorial circuits. Included is a very brief discussion of the half-adder and full-adder.

4. Chapter 4. Relations and Functions. (20 pages) This chapter includes equivalence relations, partial and total orders, functions and inverses of functions. The approach is set-theoretic.

5. Chapter 5. The Theory of Counting. (44 pages) Elementary counting techniques are introduced in this chapter. Included are the multiplication principle, tree diagrams, using Venn diagrams to count the number of elements in sets, one-to-one correspondences, countably infinite sets, permutations, combinations, the binomial theorem and the principle of inclusion and exclusion.

6. Chapter 6. Probability. (50 pages) This chapter is a brief introduction to discrete probability theory. It uses the techniques from the previous chapter in calculating the familiar probabilities associated with cards, coins, and jars of marbles. Also included are Bernoulli trials and Bayes' formula.

7. Chapter 7. Graph Theory. (46 pages) Wallis' chapter on graphs and trees is shorter than corresponding chapters and sections in most other discrete mathematics texts. Included are the topics of connected graphs, the complement of a graph, Eulerian and Hamiltonian circuits, the traveling salesperson problem, spanning trees, Prim's algorithm, Djikstra's algorithm, and the nearest neighbor algorithm. Noticeably absent is any discussion of planar graphs. Also omitted are tree traversals, binary trees and Huffman codes.

8. Chapter 8. Matrices. (34 pages) Here we find vectors, matrix arithmetic, systems of linear equations, matrix inverses, and adjacency matrices of graphs.

9. Chapter 9. Number Theory and Cryptography (42 pages) This final chapter includes a short survey of encryption techniques and the bits of number theory needed to understand these techniques. Topics include prime numbers, the Euclidean algorithm, modular arithmetic (including zero divisors and inverses), the Chinese Remainder Theorem, the Caesar and Vignre ciphers, substitution ciphers and the RSA system.

2 Opinion

I used A Beginner's Guide to Discrete Mathematics as the text for my discrete mathematics course during the Spring 2006 semester. The students in the class were mostly first-year students who listed computer science, mathematics, or mathematics education as their intended majors. The book is very user-friendly. My students found that they could actually read and understand the exposition. Wallis' practical approach to the topics appealed to my students. I liked the number and variety of exercises as well as the practice problems that were scattered throughout the text. Nevertheless, this text does have some weaknesses. The greatest weakness is the overabundance of typographical and printing errors. For example, the figures accompanying the Karnaugh map discussion (pp.78-84) contain stray rectangular printing marks that seem to be some sort of printer's glitch associated with the apostrophe. These marks are as obvious as they are meaningless and caused some of my students distress as they attempted to decipher their meaning. Other notable errors include an incorrect summation formula stated as a theorem (p.12) and a lengthy sample problem using Dijkstra's minimal path algorithm (pp.228-230) that refers to a figure in which the edge weights are omitted. Wallis has put together a nice text for beginning students, but his editors have failed him. The other major problem that I have with A Beginner's Guide to Discrete Mathematics is the absence of a mathematical discussion of algorithms and computational complexity. Although Wallis makes some remarks about computer searches for Hamiltonian cycles, he doesn't consistently examine running times or space requirements for major algorithms. Big-Oh notation is an important aspect of discrete mathematics that could have been easily included in the text.

A Beginner's Guide to Discrete Mathematics is a well-written (although poorly edited) text that is accessible to first-year students. The text includes much of the standard topics in discrete mathematics, but many instructors will feel the need to supplement it. If you teach discrete mathematics at the beginning level to students with minimal mathematical preparation, I recommend that you take a look at this text.

Review[4]
Handbook of Elliptic and Hyperelliptic Curve Cryptography
Editors: H. Cohen and G. Frey
Publisher: Chapman & Hall/CRC, 2006, Hardcover
ISBN 1-58488-518-1
Price: $99.95 (US)

Reviewed by: Lawrence C. Washington
University of Maryland, College Park, MD.

1 Overview

Elliptic curve cryptography was introduced in the mid 1980s and is now finding applicability in many public key situations. In particular, it provides a level of security comparable to that of classical methods while employing smaller numbers, which results in significant advantages in systems with limited computing resources such as smart cards.

[4] ©2008, Larry Washington

Most cryptographic applications depend on the difficulty of solving the discrete logarithm problem. In the classical situations, there is a large prime p and numbers a and b are given such that $a^x \equiv b \pmod{p}$ for some x. The problem is to find x. The elliptic curve analogue starts with an elliptic curve E defined over a finite field and points A and B on E. There is a law of addition for points on E, so that if x is an integer (assume it is positive, for simplicity), then we can repeatedly add A to itself ($A + A + \cdots + A$ with x summands) to obtain a point denoted xA. Suppose we know that $xA = B$ for some x. The elliptic curve discrete logarithm problem is to find x. In most situations, this problem is more difficult to solve than the classical discrete logarithm problem.

Classical cryptographic applications of the discrete logarithm problem, for example Diffie-Hellman key exchange and ElGamal digital signatures, carry over to the case of elliptic curves. However, the existence of pairings on elliptic curves allows additional applications such as tripartite key exchange and identity-based public key systems.

Why elliptic curves? First, we need a finite set having a law of composition in order to state the discrete logarithm problem. Second, this law needs to be computable by some efficient method. Generally, this means that there should be algebraic formulas describing the composition of two elements. The most significant examples satisfying these two requirements are the multiplicative group of a finite field (yielding, for example, the classical discrete logarithm problem) and elliptic curves. A more complicated example is the Jacobian of an algebraic curve (usually assumed to be hyperelliptic to make computations feasible). There are of course other examples given by various finite groups, but these generally have not proved as useful in cryptography.

2 Summary

The present book gives a very thorough and up to date treatment of many aspects of elliptic curves and Jacobians of hyperelliptic curves in cryptography. Despite the fact that the various chapters were written by different authors, the exposition is fairly uniform. This does not mean that the level of the mathematics is uniform. There is a very broad range of topics. For example, there are chapters on smart cards, including their physical and chemical properties. There are chapters on algorithms for elementary arithmetic operations, both for the integers and for finite fields. On the other hand, there are sections discussing the Tate-Lichtenbaum pairing for Abelian varieties over local fields (yielding a pairing-based attack on the discrete logarithm problem, for example).

The organization of the topics in the book is destined to cause some controversy among its readers. Some will love it, others will hate it. A reader who simply opens the book and tries to read it without first reading the detailed preface is sure to be puzzled by the ordering of the topics. The preface explains that the book is written on three levels. The first is the mathematical background, which gives brief descriptions of groups, rings, fields, p-adic numbers, algebraic curves and their Jacobian's, pairings, Weil descent, and some cohomology. Much of this is intended not for systematic reading but rather for reference purposes when reading later chapters. The second level concerns algorithms and their implementations. The reader who was overwhelmed by the first level will probably feel more comfortable here. There are many explicit algorithms for calculations with integers, finite fields, p-adic numbers, elliptic curves, hyperelliptic curves, pairings, point counting, and discrete logarithms. This level is the heart of the book and comprises slightly more than half the total pages for the book. The algorithms are given explicitly, but the surrounding text also explains the mathematics involved, so most readers will find something of value. It is easy to imagine a mathematically oriented reader paired with a programmer using this section.

The algorithms are explicit enough to implement, but mathematical guidance and motivation are useful. The third level concerns cryptographic applications. It starts with issues such as choosing appropriate curves and covers various cryptographic protocols plus applications to factoring and primality testing. It then treats concrete topics involving smart cards, side-channel attacks, and random number generation.

The main purpose of a handbook is for someone to find out the basic idea of a topic or to find out details about various parts of an overall scheme. Suppose the reader needs information about an algorithm such as the elliptic curve version of the ElGamal signature scheme. At the basic level, the ElGamal signature scheme is given for an arbitrary finite group in Section 1.6. A quick review of groups is given in Section 2.1. Of course, a reader of the present book probably wants to use the group of points on an elliptic curve. In Chapter 4, a lot of algebraic geometry is presented, in particular Abelian varieties. However, a better strategy is to go directly to Chapter 13, where the group law on an elliptic curve and many possible implementations are discussed thoroughly. Chapters 19 to 22 could be consulted to learn about various attacks on the elliptic curve discrete logarithm problem. The reader might want to look at Chapter 23 in order to see how to choose a good elliptic curve to avoid these attacks. To set up the ElGamal algorithm, the order of the group of points needs to be calculated. This can be done by one of the point counting techniques described in Chapter 17. Also, the ElGamal algorithm needs a random number. Chapter 30 discusses how to generate them. An ambitious reader might want to use ElGamal signatures with hyperelliptic curves. The necessary techniques, similar to those needed for elliptic curves, are also treated in the book. In summary, it's all here, if the reader has the persistence to find it.

The book is not meant to be a systematic textbook. The typical reader may have learned some of the basics of elliptic curves and groups from other sources. In this case, the various sections will work well to explain the ingredients, as needed.

The index is good – an important feature for a handbook. The bibliography is extensive, but cannot and does not include all papers on the subject. However, it does have the very useful feature that each bibliographic entry includes a listing of the pages in the handbook where it is referenced.

3 Opinions and Comparisons

There are two other recent handbooks on elliptic curve cryptography. The book [1] has a large overlap with the present book. Moreover, it also has individual chapters written by experts. Since it is much smaller, it does not contain as much detail as the book under review. However, some people will prefer the more traditional systematic organization of topics in [1]. The book [2] gives many practical details on implementing elliptic curve cryptography, but does not contain the advanced topics of the present book, except in notes at the end of each chapter. For example, it does not discuss pairings, a topic covered in great detail in [1] and in the book under review. But someone who simply wants to make an elliptic cryptosystem without worrying about the underlying theory might find [2] more congenial than the present book.

In summary, the book contains a wealth of material. Every time I opened it, I found something interesting. Generally, the book is aimed at people with a good mathematical background, or people who have such a person nearby. This is partly due to the nature of the subject. Anyone with a basic knowledge of elliptic curves and cryptology will find a lot of valuable insights in the book. On the other hand, someone who thinks that most cryptology books are too mathematical should avoid it. But anyone seriously interested in working with elliptic or hyperelliptic curve cryptography should have this book on the shelf.

References

[1] I. Blake, G. Seroussi, N. Smart, “Advances in Elliptic Curve Cryptography,” Cambridge Univ. Press, New York, 2005.

[2] D. Hankerson, A. Menezes, and S. Vanstone, “Elliptic Curve Cryptography,” Springer-Verlag, New York, 2004.

Review[5] of
The Game’s Afoot: Game Theory in Myth and Paradox
Author of Book: Alexander Mehlmann
Publisher: American Mathematical Society, 2000
ISBN 0-8218-2121-0, Paperback, Pages: 159, $27.00 (US)

Reviewed by: Danny Krizanc, Wesleyan University, Middletown, CT 06459

4 Overview

Type “Game Theory” and “Computer Science” into Google as search terms and you get approximately 759,000 hits. Ten years ago the subject “Computational/Algorithmic Game Theory” barely existed if at all. Now a search of the web yields dozens of courses on this topic. Game theory is quickly becoming an important tool in many sub-disciplines of computer science, from artificial intelligence to networking. At the same time, techniques from algorithmic analysis and computational complexity are having an impact on traditional game theory which until recently did not consider the complexity issues raised by many of the strategies it studied.

As a result of this convergence of interests, many computer scientists are motivated to learn something about traditional game theory. This book might be the place for them to start. The author’s intent is for the book to be an accessible introduction to the field for the intelligent layperson willing to think logically and follow elementary arguments that use nothing beyond high school algebra. At the same time, the book is meant to be entertaining especially to those who enjoy literary allusions ranging from Homer (not Simpson!) to Blackadder.

5 Summary of Contents

The book is divided into two parts consisting of four chapters each. The first part is an nontraditional introduction to the basic concepts of game theory. The first chapter uses a pair of simple games (one of them being “paper, rocks, scissors”) to introduce the normal form and game tree representations of two person games as well as some of the standard terminology associated with them, e.g., pure and mixed strategies, zero-sum and equilibrium. The second chapter introduces the concept of Nash equilibrium and these are computed for some simple games. The third chapter contains a “bestiary” of example games all with different “animals” associated with them. The subject of chapter four is differential games, or games where the element of time has been added.

[5] ©Danny Krizanc, 2008

All of the examples introduced are analyzed using a mostly semi-formal approach with a minimum of algebra. The setup for each example is generally quite entertaining. For example, the concept of a Truel - a generalization of the duel - is introduced using the characteristics of John Wayne, Gary Cooper and Clint Eastwood (though, personally, I think the author missed a great opportunity to reference the final scene from *The Good, the Bad and the Ugly*).

The second part emphasizes the "myth and paradox" mentioned in the subtitle. Of special interest to the author seems to be examples where analysis and practice don't quite match up. Chapter five examines variations on perhaps one of the most studied games in psychology, the Prisoner's Dilemma. Embedded in this, is the idea of how learning can effect the outcome in repeated games. Chapter six concerns "backward induction" and how it can lead to what appears to be the wrong conclusion when taken too far. An example of how a chain store (think MacDonald's) should rationally respond to a competitor (think Burger King) entering into competition in multiple markets, one after the other, is analyzed. If the number of markets is large and the cost of preventing competition is not excessive then killing the competition early seems a good strategy but backwards induction leads to the opposite conclusion. In chapter seven, the idea of rationality is hinted at, with an example of a game an owner plays against his dog and loses, the conclusion being the dog is clearly not rational! The concept of common knowledge and its effect on games is also discussed. Finally, chapter eight consists of an interesting analysis of a story taken from the myth of Odysseus.

Anecdotes and asides from the history and development of game theory appear throughout the book. An appendix containing a list of classic texts as well as web-pointers is added at the end. The list of references can be seen as a reasonably comprehensive bibliography of the most important papers in traditional game theory.

6 Opinion

Readers of SIGACT News who might be interested in buying this book probably fall into one of two categories. The first of these are teachers of courses on computational game theory who are looking for entertaining examples and historical asides to spice up their lectures or problem sets. Here I think they will find a wealth of new material, especially, those with the right literary background to get some of the more obscure references. The second category would be novices to the field who are interested in learning some of the basic notions and terminology but would like to get this information in an entertaining fashion, i.e., those looking for an "easy read." Personally, I think this second group might feel a bit short-changed by the lack of rigor and depth in the presentation but for those with just the right sense of humor, this might be the perfect book. For those looking for a more traditional introduction to game theory, the book by Osborne and Rubinstein [1] appears regularly on recommended reading lists.

References

[1] M. J. Osborne and A. Rubinstein, "A Course in Game Theory," MIT Press, Cambridge, 1994.

Joint Review[6]
Introducing Game Theory and Its Applications
Author of Book: Elliott Mendelson
CRC press, 259 pages
and
Game Theory and Strategy
Author of Book: Philip D. Straffin
MAA press

Reviewer: William M. Springer II, wmspringer@gmail.com

1 Introduction

The prisoner sat quietly in his cell. Soon the jailer would return, and he must decide: should he confess, or not? Certainly his partner would remain silent, and soon they would both be set free . . . or would he crack? What to do, what to do . . ..

The prisoner's dilemma, partially given above, is a classic problem in game theory. Game theory is applicable when a situation arises where two or more parties find themselves in a situation where the outcome for each of them depends on what actions both of them take. The actors could be individuals, companies, nations one could even be chance or a force of nature. Game theory uses mathematical methods to find (or prove the existence or nonexistence of) optimal solutions for games, which can range to actual simple games to complex situations in business, economics, politics, and other areas.

Games are broken down into two main types. The first type is zero-sum games, in which a win for one player is offset by a corresponding loss for the other player. In this case, the player's interests are strictly opposed; a win for one player results in a loss of equal magnitude for the other. A chess game, for example, is zero-sum; either the first player wins and the second loses, which we can represent as (1,-1), the second wins and the first loses (-1,1) or the game ends in a draw (0,0). Note that in each case, the sum of the payoffs is zero thus, a zero-sum game.

In the second type, non-zero-sum games, the outcomes are not all diametrically opposed; win-win and lose-lose situations may be possible. The prisoner's dilemma is the archetypal non-zero-sum game. Two prisoners are interrogated separately; no communication between them is possible. If both refuse to confess to the crime, the police will only be able to convict them of a minor charge, and both will receive a 6-month prison sentence. If one confesses while the other remains silent, the prisoner who refused to talk receives a 10-year prison term, while the confessor goes free. If both confess, they each receive a 2-year prison term. The dilemma is that while the optimal solution for the two prisoners as a group is to both remain silent (thus receiving a total sentence of only a year in prison), the best move for either prisoner, seeking only to minimize his own sentence, is to betray the other.

2 Review of Introducing Game Theory

Introducing Game Theory is divided into four chapters: combinatorial games, two-person zero-sum

[6] ©2008 William M. Springer II

games, the simplex method and fundamental theorem of duality, and non-zero-sum and k-person games. The introduction gives some basic rules for game theory: every game must have two or more players, have rules which specify how the game is to start, and have one or more terminal positions which end the game with a specified payoff to each player. Games can be broken down into deterministic games, which have no random moves (e.g., chess) and nondeterministic games, which do (such as flipping a coin). Additionally, games may or may not have perfect information, meaning that the outcome of every possible move is known to all players. Game theory may be used to determine a strategy, which is a specification telling the player what to do in any situation that might arrive during the game; sample strategies may include winning, non-losing, and win-or-draw strategies.

Chapter one defines a combinatorial game as a deterministic, finite, zero-sum game with perfect information and exactly two players; an example of a combinatorial game would be tic-tac-toe. A proof [Zermelo12] is given that in all combinatorial games, either one player has a winning strategy (and can always ensure a win) or both players have a non-losing strategy, so that if both play perfectly the game will always end in a draw. (If no draw is possible, then one player has a winning strategy) Thus, the author classifies all combinatorial games as either unfair or uninteresting; however, in many cases, we do not know which games are which! A number of examples are given, including various forms of Nim and Hex. One of the strengths of the book, in fact, is the large number of sample problems given every few pages; selected answers are at the back of the book. This chapter contains very little mathematical terminology and can be easily followed by someone with no more background than high school math; the remaining chapters, however, are much more complex.

In chapter two, we are still considering two-person zero-sum games, but players may no longer have perfect information and games may be nondeterministic. Only games with a finite number of strategies are considered; for games where a move may have an infinite number of outcomes, readers are referred to [Burger63][Dresher81]. While a wide variety of games may be considered, the details of the games are unimportant; the only information required is a list of strategies available to the players and the payoffs corresponding to each choice of strategies. Thus, given player A s strategies A1,. . .,Am and player B's strategies B1,. . .,Bn, the game may be described with an $m \times n$ matrix of real numbers in which the entry in the ith row and jth column, P(Ai, Bj), describes the payoff for A given the selected strategies. (As we are considering zero-sum games, B's payoff, of course, will be - P(Ai, Bj)). Given this payoff matrix, we can search for optimal strategies for each player. If the matrix contains at least one saddle point (an entry that is the minimum in its row and the maximum in its column) then that point is an equilibrium pair for the game; if one player chooses his equilibrium strategy, the other player cannot do any better (and may well do worse) by not choosing his strategy in the equilibrium pair. However, if a game matrix has no saddle points, then if either player consistently chooses one of his strategies, the other player can take advantage of that information for maximum return. Thus, players need to randomize their choices; each choice should be picked with a specific probability, determined by the payoffs, to make the strategy unexploitable. (A strategy that includes several different pure strategies is called a mixed strategy). Although not every matrix has a saddle point, every game matrix does have at least one equilibrium pair of mixed strategies [Von Nuemann28][Ville38][Loomis46]. Chapter three covers linear programming, including the simplex method and the fundamental theorem of duality. This chapter discusses programming methods for finding solutions for two-player zero-sum games. Chapter four is a continuation of chapter two in which games may be non-zero-sum and/or have more than two players. This chapter discusses Nash equilibria and Pareto-optimal strategies. Finally, the book includes three short appendixes covering finite probability theory, utility theory, and Nash's theorem.

3 Review of Game Theory and Strategy

Game Theory and Strategy contains much the same information as Introducing Game Theory, arranged similarly. It is divided into three chapters: two-person zero-sum games, two-person non-zero-sum games, and n-person games. Each chapter is divided into a number of short sections, ideal for covering in a short study session. While the previous book focused largely on theoretical and recreational games, Game Theory and Strategy is largely based around actual applications of game theory, including anthropology, business, economics, and war.

Chapter one defines a game and covers saddle points, mixed strategies, game trees, and utility theory. Chapter two covers Nash equilibria and non-cooperative solutions, the prisoner's dilemma, and strategic moves (in which there is communication between players); the remainder of the chapter is applications. Chapter three includes n-person games, strategic voting, and bargaining sets. The book concludes with a section on the value of game theory, and answers to all the exercises.

4 Comparisons

I enjoyed reading both books, although I preferred the early sections of Introducing Game Theory. However, Game Theory and Strategy has the advantage of being understandable throughout with very little mathematical maturity and, as previously mentioned, is fully broken down into easily digestible chunks. Speaking as someone with almost no prior experience with game theory, I found these books to be very comprehensible. I would happily recommend either book for either a class or independent study.

5 References

1. Burger, E., Introduction to the Theory of Games, Prentice-Hall, 1963.

2. Dresher, M., The Mathematics of Games of Strategy: Theory and Applications, Dover, 1981

3. Loomis, L.H., On a theorem of von Neumann, Proceedings of the National Academy of Sciences, USA, Vol. 32, 1946, pp. 213-215

4. Villie, J., Sur la thorie gnrale des jeux o intervient l'habilit des joueurs, in Applications des Jeux de Hasard, Vol. 4, E. Borel et al. (1925-1939), 1938, Fasc. 2, pp. 105-113

5. Von Neumann, J., Zur Theorie der Gesellschaftsspiele, Math AnnalenVol. 100, 1928, pp. 295-300.

6. Zermelo, E., Ueber eine Anweding der Mengenlehre auf die Theorie des Schachspiels, Proceedings of the Fifth International Congress of Mathematicians, Vol. 2, Cambridge, 1912, pp. 501-504

Review[7] of
Semantic Integration of Heterogeneous Software Specifications
Author: Martin Grobe-Rhode
Publisher: Springer-Verlag, 2004

Reviewer: Maulik A. Dave

1 Overview

Software specifications and modeling are extensively used in software engineering. The variations in different specification methods are found not only at language level, but also at theoretical level. The book offers to solve the problem of integrating specifications developed with different methods.

2 Summary of Contents

After literature survey, the first chapter introduces concepts such as viewpoint model of software modeling, and reference models. It stresses on language and method independent integration. The transformation systems reference model is introduced informally, and briefly. The second chapter is on data spaces. Transformations on data spaces are formalized. They are explained by taking linked list as an example. Data spaces from other specification frameworks such as graph grammars, petri nets, states as algebras, and abstract state machines are discussed. Control flow specifications are discussed in third chapters. Rewriting of algebras with the application of transformation rules, is explained.

The models also get developed using other abstract models. The operations required to do this, are described in forth chapter. The chapter begins by describing operations such as restriction, view, exclusion, hiding, extension, and reduction on transformation systems. The category formed from these operations, and refinement are also described. In the end, institution of transformation systems is presented in formal manner, and discussed. The component based development of models is dealt in fifth chapter. Composition of transformation systems is formalized. The connection relations, required for compositions, are defined. A connection consists of identification, and synchronization. Composition of more than two transformation systems requires limits to be placed. These limits are described in details.

One of the most practical languages used for software modeling, is Unified modeling language (UML). The theories developed, are applied to UML software specifications in sixth chapter. The discussion focuses on class diagrams, state machines, and sequence diagrams. Semantics of these, are discussed in details. The concluding chapter presents the summary of the work, future directions, and an overview of other approaches for integration.

3 Style

The book is highly theoretical in nature. Mathematical notations are extensively used. In particular, notations of tuples, sets, and functions are used throughout. Set theory, algebraic theory, category theory, lambda calculus, and theory of institutions are the major theories used. The required fundamentals of partial algebra are supplied in appendix. The formalism uses definitions, propositions, theorems, and lemmas as usual. Examples, and sections on discussions are included for informal explanation.

4 Opinion

The book establishes mathematical foundation for integration of software specifications written in different frameworks. The stress is on semantics. Transformation theory is presented as a solution. For practitioners, there is a chapter on UML. The chapter can serve as an excellent guide for the application of the theories developed in the book. To understand the book, a general idea of software specifications, and an elementary knowledge of mathematical areas such as semantics, algebras, set theory, and category theory are sufficient. Future directions, and a list of references can be used for further research to build upon the foundation in the book.