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. **P, NP, and NP-Completeness** by Oded Goldreich. Review by S.V.Nagaraj. This is a simplified and version of the authors text on complexity theory. It is aimed at novices.

2. **Bijective Combinatorics** by Nicholas Loehr. Review by Miklós Bóna. Despite the title the book is really about Enumerative Combinatorics, in which bijection proofs play a significant role.

3. **Surveillance or Security?** By Susan Landau. Review by Tal Moran. Technology has forced all societies to revisit legal questions about surveillance. The usual question is *Where is the balance of power between allowing the government to catch criminals and terrorists, and the citizens need for privacy?* This book challenges that tradeoff. She claims that the downsides of building surveillance into devices is not worth the alleged security benefits.

4. **Spyware and Adware** by John Aycock. Review by Dave Werden Spyware is when someone uses your own computer to spy on you. Adware is when someone uses your own computer so try to sell you something. This book discusses both in terms of how they can be installed and how to defend against them, and other information. The book does not require a technical background.

5. **Burdens of Proof** by Jean-François Blanchette. Review by Harry Lewis. This book is a study of how cryptographic (digital) signatures travelled into the legal world to become a forensic technology, endowed with the power to transform bits into admissible evidence. The book raises issues with respect to the physicality of computing and the use of models in mathematical practice.

6. **Boolean Models and Methods in Mathematics, Computer Science, and Engineering** Edited by Yves Crama and Peter L. Hammer. Review by Marcin Kamiński. A *Boolean function* is arguably one of the most fundamental and also most studied objects in discrete mathematics. Boolean functions appear in many different contexts and have been studied in several fields before a common theory emerged. In combinatorics, a hypergraph, that is, a collection of subsets of a given set, can be encoded as a Boolean function. Many properties of hypergraphs (transversals, stable sets, colorings) translate to properties of corresponding Boolean functions. This book is a collection of papers on the application of Boolean functions to many different fields including integer programming, operations management, decision theory/making, voting theory, game theory, circuit complexity, circuit design, machine learning, social choice theory, neural networks, and more.

7. **Algorithmic Randomness and Complexity** by Downey and Hirschfeldt. Review by Jason Teutsch. This book investigates the connections between randomness and computability. There is already an obvious connection: a string x is random if (informally) there is no Turing machine of length much less than $|x|$ that prints it. This is a fast growing field and this book collects up much of what is known.

8. **Information Retrieval** By Buettcher, Clarke, Cormack. Review by Paul Rubin. This book is a guide to search engine implementation. This is a vast topic involving both human-computer interaction and algorithms.

9. **Models of Conflict and Cooperation** by Rick Gillman and David Housman. Review by Mark C. Wilson. This book covers Game theory in detail for a non-math audience.

10. **Deterministic Extraction from weak random sources by Ariel Gabizon**. Review by Marius Zimand. This book, based on the authors PhD thesis, is about devices (extractors) that take a weak random source and produce a strong random source. They are useful for derandomization.

11. **Applied Information Security by David Basin, Patrick Schaller, and Michael Schläpfer**. Review by Jonathan Katz. This book promises a hand-on approach to real security. Does it deliver? Read the review to find out!

12. **Introduction to Bio-Ontologies** by Peter N. Robinson and Sebastian Bauer. Review by Mohsen Mahmoudi Aznaveh. How do we handle the vast amount of data in the field of bioinformatics? One way is to make ontologies. This book discusses how to do that and the math involved in doing it.

13. **The Dots and Boxes Game: Sophisticated Child's Play** by Elwyn Berlekamp. Review by Omar Shehab. Do you want to be an expert at the so-called child's game of Dots and Boxes? If so then read this book! The game is more sophisticated than you think.

BOOKS I NEED REVIEWED FOR SIGACT NEWS COLUMN
Logic and Computability

1. *Computability and Complexity Theory (2nd Edition)* by Homer and Selman.
2. *Proof Analysis: A Contribution to Hilbert's Last Problem* by Negri and Von Plato.
3. *Programming with Higher Order Logic* by Miller and Nadathur.
4. *Graph Structure and Monadic Second-Order Logic: A language -theoretic approach* by Courcelle and Engelfriet.
5. *Software Abstractions: Logic, Language, and Analysis* by Daniel Jackson.

Algorithms, Cryptography, and Coding

1. *Integrated Methods for Optimization (second edition)* by John Hooke
2. *Algebraic Shift Register Sequences* by Goresky and Klapper.
3. *Information Theory and Best Practices in the IT industry* by Sanjay Mohapatra.

Misc

1. *A Wealth of Numbers: An anthology of 500 years of popular math writings* Edited by Benjamin Wardhaugh.
2. *The logician and the engineer: How George Boole and Claude Shannon created the information age* by Paul Nahin.
3. *History of Mathematics: Highways and Byways* by Amy Dahan-Dalmedico and Jeanne Peiffer.

Review of[2]
P, NP, and NP-Completeness
by Oded Goldreich
Cambridge University Press, 2010
214 pages, Hardcover

Review by
S.V.Nagaraj svnagaraj@acm.org
RMK Engg. College, India

1 Introduction

Complexity theory is a branch of theoretical computer science which in turn is a sub-discipline of computer science. Complexity theory is concerned with the study of computational resources required for solving problems of an algorithmic nature. This book is an introduction to complexity theory. It studies one of the fundamental questions of theoretical computer science: the P versus NP question. It also focuses on the theory of NP-completeness. A million dollar prize is associated with the famous P versus NP question question. P is defined as the class of all problems for which a solution is easy to find (polynomial time in complexity theory parlance). NP is the class of problems for which a proposed solution can be verified in polynomial time. The P versus NP question asks whether or not the classes P and NP are indeed identical. There are many textbooks that deal with complexity theory, however, this book offers a good introduction to complexity theory in general and the P versus NP question in particular. This book is an enhanced version of a chapter in another advanced book on complexity theory by the same author: Computational Complexity - A Conceptual Perspective, Cambridge University Press, 2008. This simplified book is aimed at novices as well as undergraduate students. The book is available both in hardback and paperback editions. The respective ISBNs are 9780521192484 and 9780521122542. The prices are 60 British pounds and 25 British pounds respectively.

2 Summary

The book is made up of five chapters. The first chapter deals with computational tasks and models. The author introduces the concepts of search, decision and promise problems. For solving problems one must assume a particular model of computation. The author introduces uniform and non-uniform models of computation on the basis of which problems may be solved.

The second chapter is concerned with the P versus NP question. As explained earlier, this is a very fundamental question whose resolution will have many important implications in complexity theory and computer science. The author explains the classes P and NP in terms of search problems and decision problems. There is also a discussion of the equivalence of the definitions in terms of nondeterminism and verification.

The third chapter studies polynomial-time reductions. The general notion of reduction is first introduced. The notions of polynomial time reductions due to Cook, Karp and Levin are then

illustrated. The author looks at ways of reducing optimization problems to search problems. He then introduces the notion of self-reducibility of search problems.

The fourth chapter is concerned with the theory of NP-completeness. The amazing existence of NP-complete problems is shown. Several natural NP-complete problems are then discussed. The author also discusses the positive and negative applications of NP-completeness. He demonstrates that there are NP problems that are not NP-complete if P and NP happen to be distinct.

The fifth chapter is titled "three relatively advanced topics". The topics are promise problems, an optimal search algorithm for NP, and the class co-NP.

3 Opinion

The book offers an interesting introduction to complexity theory with focus on the P versus NP question and the theory of NP-completeness. The author has in fact improvised a chapter in one of his more advanced books on complexity theory in order to produce this book as mentioned before. The author has attempted to simplify the presentation so that beginners to complexity theory will find it more appealing. The numerous exercises at the end of chapters will definitely help the learners as well as the teachers. This book will be very suitable for an introductory course on complexity theory. The author is a well-known expert in the field of complexity theory and so is well-qualified to bring out this book which will serve as a very good introductory textbook. The focus on search problems and promise problems in this book is to be appreciated since many books neglect these topics. Complexity theory by itself can be a very good course for beginners who are often exposed to courses on computability alone. The author has done a good job by focusing on the P versus NP question. The resolution of this question will have several implications on problem solving. The introduction to the theory of NP-completeness has also to be valued. Given the size of the book it is no surprise that the author has not focused on approximation algorithms and meta-heuristics for solving purportedly hard problems. I feel that the chapter on NP-completeness could have been a bit longer with more examples of NP-complete problems. Nonetheless, I feel this is a very good introductory book on complexity theory, the P versus NP question and the theory of NP-completeness. I have one complaint though. The fonts used in the book are barely readable.

Review of[3]

Bijective Combinatorics
by Nicholas Loehr
CRC Press - Chapman Hall
2010, 612 pages, 85 dollars

Review by Miklós Bóna

When I first heard that there is a book entitled "Bijective Combinatorics", I was surprised. As all combinatorial enumerators, I love bijective proofs. But an entire book on them, with 570 pages? What kind of book would that be? A collection of the nicest known bijective proofs could not be quite that long.

The short answer is that the title is somewhat misleading. This is an informative textbook on Enumerative Combinatorics, in which bijective proofs certainly get their fair share.

Another central question that a reviewer must answer about the book he reviews is "just what kind of book is it", in the sense of level, targeted audience, and style of presentation. That question is particularly difficult to answer for this book, since it can be viewed as several books in one.

In the first four chapters, you find all the topics that you would find in other introductory textbooks on combinatorial enumeration, such as counting permutations, words, subsets, with or without repetitions, graphical enumeration, the formula of inclusion-exclusion, and the Möbius inversion formula. The discussion is somewhat deeper than what one finds in undergraduate textbooks, though probably not quite as advanced as that of the classic graduate books. There are plenty of exercises.

Chapter 5, Ranking and Unranking, is more interesting. This reviewer has not seen this topic in any other combinatorics textbooks, and suspects that theoretical computer scientists will be more familiar with the subject than most combinatorialists. If certain combinatorial objects, such as permutations of length 25, are listed in some order, and we are asked where in that lists a given permutation p is located, then we are in fact asked what the *rank* of that permutation is in the ranking specified by this listing. It goes without saying that we would like to find a method that involves as few steps as possible, so going through the whole list is not a desirable option. This family of problems is called *ranking*. Not surprisingly, *unranking* is the procedure of computing the ith element of a list from the rules that were used to create that list.

After defining these intriguing goals, the chapter continues with a rather long list of introductory results, then finally reaches some interesting instances of the problem, such as ranking in the set of n^{n-2} trees on n labeled vertices, and finding the next element in an ordered set of Dyck paths of a given length.

Chapter 6, Counting weighted structures, is in some sense a chapter that prepares the reader for the upcoming chapters on formal power series. Instead of simply permutations, or lattice paths, of a given length, we can count them according to some statistics, such as the number of inversions, or the area below the lattice path, or between the lattice path and the main diagonal. If these statistics are defined in an appropriate way, then they behave like "weights", that is, if we unite two small structures, then their weights get added or multiplied together. The author explains q-binomial(or Gaussian) coefficients, which are the most frequently used weighted version of binomial coefficients, and the far less known generalization into *Quantum multinomial coefficients* as well.

Chapters 7 and 8 are devoted to generating functions, the former to the concept of Formal

Power Series, and the latter to their combinatorial applications. This, of course, belongs to every serious textbook on enumerative combinatorics, and there are not too many decisions an author has to make as far as topical coverage is concerned. The discussion is rigorous, certainly more appropriate for graduates than undergraduates. There is not a lot of bijective combinatorics here, except a section in Chapter 8 about bijections on integer partitions.

Chapter 9 is a fairly typical chapter on Enumeration under group action. It goes far deeper into the topic than most undergraduate textbooks, while it does not seem to be too challenging for a good graduate student.

Chapter 10 and 11 are the most advanced parts of the book. Their subject is symmetric and antisymmetric functions and tableaux. With 120 large pages, they could provide enough material for a graduate course on their own. A symmetric polynomial is a polynomial in several variables that is invariant to permutations of its variables, such as $xyz + x^2 + y^2 + z^2$. Symmetric polynomials have a number of standard bases, such as the elementary symmetric polynomials, the complete symmetric functions, and the power sum symmetric functions. Algebraic combinatorialists are interested in these functions because of their applicability to the combinatorics of *tableaux*.

A tableau is a Ferrers shape filled out with positive integers. The most studied kind of tableau is the *Standard Young Tableau* (SYT) that is, a tableau whose boxes are bijectively filled with the integers $1, 2, \cdots, n$ so that the numbers increase from left to right in each row and top to bottom in each column.

The most spectacular result on SYT, the famous Robinson-Schensted correspondence, is a celebrated bijection that maps each permutation p of length n into a *pair* $(P(p), Q(p))$ of SYT on n boxes so that $P(p)$ and $Q(p)$ have the same shape. This bijection has many very useful features. For instance, the image of p^{-1} is the pair $(Q(p), P(p))$. In particular, $P(p) = Q(p)$ if and only if $p = p^{-1}$, that is, when p is an *involution*. As a consequence, the number of involutions on n elements is equal to the number of SYT on n boxes.

We are shown these classic results and many of their variations. Then the author establishes the connection between tableaux and symmetric polynomials by presenting theorems that show that certain numbers counting tableaux can also be obtained as coefficients of certain symmetric functions in the appropriate basis.

The chapter is impressive because it contains a lot of information, but it manages to do so without being intimidating– a feat that many books on this subject could not accomplish. Chapter 11 is a natural counterpart to Chapter 10, covering antisymmetric functions instead of symmetric ones. An antisymmetric polynomial is a polynomial in several variables with the property that applying a permutation p to the variables has the effect of multiplying the value of the polynomial by the sign of p. So if f is an antisymmetric polynomial in three variables, then $f(y, z, x) = f(x, y, z)$, while $f(x, y, z) = -f(y, x, z)$.

Chapter 12 is a collection of unrelated sections of interesting enumeration problems that did not fit into previous chapters, such as parking functions, alternating permutations, and cyclic shifts of lattice paths.

There is a very extensive list of exercises throughout the book, averaging well over 100 per chapter. None come with full solutions, but about half of them have short answers at the end of the book.

As we mentioned in the introduction, the book is difficult to categorize. It is certainly a book on Enumerative Combinatorics. Each chapter has a long introduction, often taking up more than half of the chapter, that is accessible for undergraduates. This is true for the book as whole as

well– the first four chapters are appropriate for undergraduates. The rest of the chapters, and the rest of the book, is at graduate level. As we have recently said, Chapters 10 and 11 could be used for a special topics graduate course on symmetric and antisymmetric functions.

For all these reasons, and for the size of the book, (590 big pages), this reviewer thinks that there will be more people who use the book as reference material than people who teach a class from it. The book can certainly be used as a textbook, but the instructor will have to make many decisions as to what to cover.

Review of[4]
Surveillance or Security?
by Susan Landau
The MIT Press, 2010
383 pages, Hardcover

Review by
Tal Moran[5] (talm@seas.harvard.edu)

1 Introduction

Long-distance communication was subject to interception and surveillance even when it consisted mainly of parchment and ink, hand-carried to its destination. Today, as in previous eras, surveillance is used by governments to help ensure the safety and security of their citizens, but also to suppress free speech and civil liberties.

As communication technology evolved, so did the techniques for secretly monitoring it. In the past few decades, however, there has been a qualitative change in the capabilities of the watchers: at least in theory, it is now technically possible to monitor and record most communication of a large fraction of the population simultaneously. Did you just send an instant message to a friend? Even if you're using encryption software, your ISP probably has a log of who you're talking to (or at least where they're located) and how much you had to say. Did you use a cellphone? Your wireless carrier also recorded exactly where you were when you said it.

In the United States (and elsewhere), law-enforcement agencies have been pushing for even more information collection and easier access to this information, arguing that wiretapping Internet communication is essential in fighting terrorism and organized crime (not to mention its usefulness in more "mundane" criminal cases).

The question most often posed about government surveillance is where to find the balance between its positive effects — increased security — and citizens' right to privacy. This book does not answer that question. In fact, the main premise of the book is that its implied assumption is a false dichotomy: making eavesdropping and wiretapping easier for law-enforcement can actually harm our overall security.

In this book, Susan Landau does not argue against (legal) surveillance in and of itself. Rather, her point is that purposefully introducing wiretapping "backdoors" into our communications infrastructure has significant costs (in addition to the obvious loss of privacy) and, at least in the case of the Internet, these outweigh the benefits.

2 Summary

The book's first seven chapters provide the technical, legal and policy background required to understand the problems that wiretapping was meant to solve and the new problems it creates. Chapter 1 contains a general introduction. Chapters 2 and 3 give technical details about the telephone

[5]In the interests of full disclosure, I am currently a fellow at the Center for Research on Computation and Society at Harvard, with which Susan Landau is also affiliated.

network, the development of the Internet and the difficulties in ensuring its security. In Chapter 4, Landau discusses the history and current state of wiretapping laws (and their application) in the United States, while in Chapter 5 she compares the effectiveness of wiretapping in different contexts (e.g., national-security cases, organized crime or apprehending escaped prisoners). Chapter 6 describes the evolution of communications technologies, up to the current state-of-the-art and predictions for the near future. In Chapter 7, Landau talks about the threats facing the Internet: from individual hackers, malware producers and "botnet" operators to attacks against "critical infrastructure", industry and military targets carried out by nation-states.

Chapters 8 and 9 are, in my opinion, the most interesting in the book. Building on the background developed in the previous chapters, these chapters contain the main arguments against mandating wiretapping capabilities as part of the communication infrastructure. In Chapter 8, Landau surveys the "security risks" (as opposed to "policy risks") associated with widespread surveillance. Some examples:

- Built-in wiretap capability is, by definition, an architected security breach. If there is a bug in its implementation, it opens new avenues for attack that did not previously exist. Even if the backdoor itself is implemented perfectly (which software rarely is), it increases the overall system complexity, making the system as a whole much more likely to be vulnerable.

- The governments of other countries (including U.S. allies) make use of their foreign intelligence operations for the purpose of economic espionage. Additional vulnerabilities in the communication infrastructure would make it easier to attack U.S. companies and decrease their global competitiveness.

- The added requirements for building new systems would have the effect of stifling innovation: small companies would be hard-pressed to ensure that their products comply, while large companies may prefer to minimize changes to existing products in order to avoid the costs of recertification.

- Technical standards for our communication equipment today will have effects many years down the road (e.g., the current telephone system still supports phones built in the 1950s). If we embed vulnerabilities into our infrastructure, they will remain even if policy (and law) have changed in the meantime.

Chapter 9 contains a discussion of the "policy risks" that arise from wiretapping. This chapter deals with issues such as wiretapping the press and the importance of the "rule of law" (in the context of the illegal wiretapping that occurred in the U.S.) to long-term U.S. interests. I found very interesting the contention that cooperation with local communities can be more effective than wiretapping in discovering and foiling nascent terrorists; excessive wiretapping, especially if it is perceived to be targeted at minorities, can backfire by harming relations between law-enforcement and these communities. Another interesting argument is the value of civil-sector communication security to intelligence collecting: NGOs and human-rights organizations operating in repressive regimes critically rely on the security of their communications. These groups also provide significant (sometimes the only) intelligence about the situation "on the street". If surveillance capabilities were embedded in commercial communication equipment, NGOs would become more vulnerable to wiretapping by hostile governments, reducing their effectiveness.

In Chapter 10, Landau approaches the problem from a slightly different angle, and talks about communication systems for first-responders in crisis situations. Explaining that a critical missing ingredient in first-responder communication systems is interoperability (e.g., police and firefighters after the 9/11 attacks on the Twin Towers were not able to communicate effectively, possibly contributing to the death toll when the north tower collapsed). Landau makes the case that solving the interoperability problem on a nationwide scale would require off-the-shelf systems, available to everyone — and since these must provide secure (encrypted) communications to first-responders, their wide availability would make this type of untappable communication open to anyone.

Finally, Chapter 11 contains technical and policy recommendations for the future. Some of these are "standard" best-practices agreed on by most security professionals (e.g., the sensitivity of data on a networked device should be coupled to the broadness of who may access that device). Others are more thought-provoking: for example, requiring that software used for wiretapping be made available for public auditing.

3 Opinion

Speaking as a security expert but a policy layman, this is a book I wish every policy-maker involved in security policy would read (and I believe anyone interested in security policy would enjoy). The case this book makes against widespread surveillance seems counter-intuitive, but Susan Landau gives convincing arguments (backed by detailed research — the book contains over 80 pages of notes and citations) and a clear explanation of the required technical background for the non-technical reader.

The book is focused mainly on the United States, especially when discussing the legal and economic framework for wiretapping (and some of the legal/moral arguments against it). However, I think it would be interesting even for readers in other countries: both because the technical problems are the same regardless of where you are located, and because U.S. policy has effects far beyond its borders.

Personally, I found this book fascinating for its wealth of anecdotes; there's a big difference between understanding, theoretically, that embedding eavesdropping capability into software could introduce security weaknesses, and learning that senior members of the Greek government were wiretapped for ten months by "parties unknown" using exactly such a vulnerability.

Review of[6] of
Spyware and Adware
by John Aycock
Springer, 2010
145 pages, Hardcover

Review by
Dave Werden dwerden@acm.org
Augusta, GA

1 Introduction

Spyware and Adware are two of the most common threats that internet users have to deal with. Anyone who has dealt with the surprisingly malicious, but generally real-looking, "Your computer is infected" popup has experienced this first hand. These types of programs can execute tasks ranging from the monitoring of online browsing habits to the malicious harvesting and transmitting of a user's personal data. Additionally, even if these programs aren't profiling users or stealing information, they can be equally malicious by using up a large portion of a system's resources, rendering that system almost completely inoperable.

2 Summary

Spyware and Adware, written by John Aycock, is a book published by Springer and is part of an 85-book series "Advances in Information Security." This particular title in the series is a reference to the definition, characteristics, and actions specific to today's most common spyware and adware. Aycock uses this small sized book to present a large quantity of data on this topic. He divides this information between nine chapters that are organized in a logical manner. This organization begins with an introduction and discussion on how these programs can get installed and how they behave, while moving through topics such as "phoning home" and "tracking users." Chapters 2 through 8 present the bulk of the material.

"Getting There" is the title of chapter 2 and this is an informative discussion on how spyware and adware can be installed and executed. The installation methodologies discussed include those of voluntary installation, drive-by downloads, and installation through other malware. The second focus of this chapter is that of the execution of these programs, which can be as covertly started during kernel startup, or started by a well-intentioned mouse-click in an infected GUI screen. It is worth noting that while there are some complex topics in this chapter, Aycock uses a style of writing and illustrations that allow for an ease of understanding. Chapter 3 discussed the logical follow-on to the installation and execution chapter, which is that of avoiding detection or, "Staying There." Aycock discusses some of the methodologies used to avoid detection by the user as well as some of those methods used in avoiding detection of anti-virus/anti-spyware programs.

Chapters 4 through 8 discuss the actual primary activities of spyware and adware. Chapter 4 presents informative information on key logger, software (and sometimes hardware) that captures

and records the order and value of each keyboard key pressed by the user. These applications have been effective over the years with providing a means to garner usernames and passwords, in addition to possible financial, corporate, or private information. Key loggers have been a commonly malicious tools used to gather information to send back to someone with generally criminal intents. Chapter 5 describes the way in which this information can typically leave the users home or work computer: "Phoning Home." Phoning home is the term used to describe the activity of adware and/or spyware communicating back to a location controlled by someone other than the user. As described by Aycock, there are four general aspects to phoning home: Push/Pull, steganography, finding "home," and defense of information leakage. Pushing is a method in which spyware transmits user information back to, who Aycock names as, the "adversary." Pulling is the opposite action in which the adversary will establish a connection with the spyware on the user's machine and subsequently attempt to exfiltrate date back to the adversaries location. Aycock tackles steganography, the art of hiding information in graphic, with relative ease. Steganography can be a rather complex topic but Aycock uses simple to understand illustrations and figures that really compliment his discussion on the topic. After discussing steganography, the discussion in chapter 5 moves to "Finding Home." There are multiple methods that spyware can use to find its "Home." Aycock presents a quick discussion on some of the more common tactics such as changing/overwriting the hosts file and "fast-flux" DNS lookups. Fast-flux is a term used to describe the logic used in the spyware to quickly cycle through a large(r) number of IP address to communicate with. Chapter 5 concludes with a discussion of protection against information leakage, which is the over-arching term describing the unauthorized movement of user information to the adversary.

Chapters 6 and 7 address adware specifically. Chapter 6 deals with the specific types of advertisements, such as annoying pop-ups, Tear-backs, and banners. Adware is not generally illegal, but, as Aycock points out: "it enjoys enough of a poor reputation that mainstream advertisers may always avoid its use." Aycock spends a larger amount of his efforts in this chapter and the next, which is probably due to a belief that two parties are involved in advertisement: the advertiser and the user. It is in the beginning chapter 6 that Aycock discusses advertisements from the user's perspectives. The differences in implementations are actually discussed in Chapter 7.

In Chapter 7 the four primary implementation locations of advertisements is discussed: On the User Machine, In the Network, Near the User Machine, and On the Server. With these four locations, it is of some importance to note that Aycock does provide some differentiation between the "annoying" popups and the use of popups by ISPs to transmit emergency messages. It is here that some sample JavaScript is presented explaining a basic pop-up, with methods such as focus() and blur() explained in layman's terms. The four different locations addressed in this chapter are discussions on where the advertisement software is executed from. Sprinkled in these sections are real-world examples of adware such as LinkMaker and Typhoid.

Chapter 8 is the final discussion in Aycock's book before his summation in Chapter 9. It is here in Chapter 8 that he focuses on methodologies for tracking users. In this discussion Aycock presents one of the better explanations on cookies, how they are used, and the privacy implications. Following the detailed explanation on cookies are some smaller discussions of other methods to track users such as Cascading Style Sheets (CSS's) and third-party tracking sites. What readers may find of interest here are the short discussions on Social Networks (Facebook, MySpace, etc) and Physical Location. In the case of physical location, Aycock makes the reader aware reverse geolocation.

3 Opinion

Aycock begins Chapter 1 with: "Most of us don't parade around nude". Although he is using an analogy here to open the discussion of privacy, his apparently relaxed approach makes this book open to a larger audience base than the "usual" technology books in print today. Point in fact, Aycock's style throughout this work his would allow for an audience ranging from a high-school aged computer novice to an experienced computer science professor. In summary, Spyware and Adware is a well-written book that stays on focus and presents the reader with easy-to-understand explanations of each of the sub-topics. As someone who has worked network security for almost 15 years, I still found this book to be educational and accurate and I would recommend it to anyone who has even a slight interest in the topics of adware and spyware.

Review of[7] of

Burdens of Proof
By Jean-François Blanchette
MIT Press, 2012
264 pages, Hardcover, $30.00

Review by
Harry Lewis lewis@harvard.edu
Harvard University

Some cryptographers will not enjoy this analysis of what has gone wrong with public-key cryptography, but it's a refreshing splash of cold water for romantic idealists about the use of mathematical abstractions in social processes. It's also a useful reminder that when there is money to be made by turning a theorem into a product, those who stand to profit have an incentive to fudge the boundary conditions in which the invention has to function.

The author, Jean-François Blanchette, is a professor in the Department of Information Studies at UCLA. He approaches his subject not as a computer scientist but as an analyst of the practice of science in society. Blanchette makes the case that cryptographers became too enamored with the mathematical profundities stemming from Diffie and Hellman's stunning *New Directions in Cryptography* paper of 1976. Especially as protocols became more complex, the lovely abstraction of Bob and Alice grew too distant from flesh and blood, the bits of their electronic keys too disembodied from atoms of real computing and communication devices. Blanchette all but says that it served the cryptographic community right when Paul Kocher startled everyone in 1999 by sniffing the electromagnetic radiation leaking from a processor chip carrying out a cryptographic computation. These systems function in the real world, which cares not for the mathematicians' simplifying assumptions.

It is not just the material world that Blanchette argues got lost in the shuffle of conferences and research publications about mathematical cryptography, but more importantly the social world. The book opens with a compelling example of how far society yet is, 36 years after *New Directions*, from believing what we are told is important, authentic truth: Barack Obama's birth certificate. The adaptation of cryptography to digital signatures was supposed to resolve all doubt about such things, wasn't it? Yet here we have as much mistrust as ever about the authenticity of signed documents, and instead of Bob and Alice behaving as the math papers say they should, we have (p. 140) the absurd image of two French notaries signing a "digital document" not via Diffie-Hellman key exchange but with video cameras capturing images of each manually signing an electronic tablet using a capacitive stylus.

France, Blanchette good-naturedly notes, is an ideal place to look at case studies of digital documents, since it is the *country that gave bureaucracy its name, where citizens must carry at all times their* papiers d'identité, *where administrative procedures are synonymous with intricate theatrical performances of forms, stamps, signatures and countless* pièces justificatives (p. 93). It is, moreover, a top-down society. What the government says, goes—even if it is that the *Minitel* is the future of communication technology. Blanchette's account reminds us how different friendly democracies can be. Imagine any elected leader in the U.S. proudly calling for *a political vision of*

the Information Society, as Prime Minister Lionel Jospin did in 1997. Surely, France should be the perfect place to lead the transition to a paperless society with fully verifiable documents.

Not quite. The paper system is too deeply entrenched. It is the source of pride—and revenue—for entire professions. The mathematical purity of the new cryptography—the whole numbers being the work of God, after all, whoever might have devised the rest of mathematics—was of little interest to the politicians and the bureaucrats. Quoting another metaphor for excessive abstraction from reality, Blanchette summarizes his thesis as follows: *I offer an empirical counterpoint to this fetishism of the "geometry of the line" by following the deployment of electronic signatures within the very professions entrusted with the production and management of documentary evidence* (p. 7).

The market, Blanchette argues, has offered its own judgment on the products of mathematical cryptography. *Digicash, a celebrated start-up that sought to commercialize anonymous electronic cash eventually filed for bankruptcy* (p. 60). *Throughout the 1990s, the business opportunities entailed by such a deployment [of a public key infrastructure consisting of certificates, centralized public file authorities, and revocation mechanisms] would be repeatedly predicted explosive growth, in spite of lethargic market uptake* (p. 77).

And then there is the awkward fact that the entire enterprise stands on an unproven number-theoretic foundation—one of the missing pieces referred to in the title of the book. Blanchette promises to *tread very lightly* in his account of the epistemological controversies over the Random Oracle Model, but can't help quoting one cryptographer's self-serving, *realpolitik* rationale for accepting it as *the only way we know of to justify the security of ordinary RSA signatures* (p. 167).

Blanchette's assault—scholarly, historically founded, dryly Gallic, and meticulously documented, but an assault nonetheless—reminds me of an earlier paper, *Social Processes and Proofs of Theorems and Programs* by Richard A. De Millo, Richard J. Lipton, and Alan J. Perlis (Communications of the ACM, May, 1979). That paper generated a firestorm of controversy, the heat of the fire enhanced by the distinction of the computer scientists who wrote it. In a word, it argued that proofs of programs would, in practice, never be like proofs of mathematical theorems. Thirty-three years later, watching a trading program place $440 million in spurious trades on the day it went live perhaps vindicates their position—though one could also argue that the triumphs of circuit verification, for which Ed Clarke and his colleagues won the Turing Award, should be success enough. Either way, it's a reminder that computer science needs its skeptics, and we all benefit when someone steps outside the cocoon and asks about the big picture.

The book is an easy read—there are virtually no equations. Almost all of the first half covers material that will be familiar to anyone who has taken a course in cryptography, though it is presented in a nice historical narrative that is picked up again through out the rest of the book. The focus then narrows to digital signatures, ignoring the ubiquitous use of SSL almost entirely, and failing to align the thesis of market failure with EMC's 2006 acquisition of RSA Data Security for $2.1 billion. (SSL is mentioned only in an endnote, which quotes Bruce Schneier's characterization of its security as *a complete sham* since in practice, nobody verifies certificates—credit card transactions are secure, he says, only because the credit card companies let us repudiate transactions.) The case studies laying out the heart of Blanchette's evidence occupy some 35 pages. The writing is clear, with mercifully few slips, though there are a few: *if one inversed the procedure* (p. 66); and most dramatically, *as the fumes of the dot-com crash cleared, much of cryptography's project laid* [sic] *in the rubles* (p. 5)—probably a spell-correction artifact, not an observation about the Russian economy!

Review of[8]

Boolean Models and Methods in Mathematics, Computer Science, and Engineering
by Yves Crama and Peter L. Hammer, Eds.
Cambridge University Press, 2010
759 pages, HARDCOVER

Review by
Marcin Kamiński
Department of Mathematics,
Computer Science and Mechanics
University of Warsaw
mjk@mimuw.edu.pl

1 Introduction

A *Boolean function*, arguably one of the most fundamental and also most studied objects in discrete mathematics, is a $\{0,1\}$-valued function of a finite number of $\{0,1\}$-valued variables. The interest in Boolean functions initially came from propositional logic. If we interpret the variables as basic propositions, then the value of the function corresponds to the logical value of the sentence. The theory of Boolean functions takes a different perspective. While in logic symbols "0" and "1" are interpreted as logical values, in the theory of Boolean functions they are numerical values.

Boolean functions appear in many different contexts and have been studied in several fields before a common theory emerged. In combinatorics, a hypergraph, that is, a collection of subsets of a given set, can be encoded as a Boolean function. Many properties of hypergraphs (transversals, stable sets, colorings) translate to properties of corresponding Boolean functions.

It is difficult to underestimate the importance of Boolean functions in computer science, with binary computations and digital circuits. They also appear naturally in a number of different fields, like integer programming, operations management, decision theory/making, voting theory, game theory, circuit complexity, circuit design, machine learning, social choice theory, or neural networks.

The book explores applications of Boolean models in all these fields and more. The volume is a collection of chapters written by recognized experts in their fields. As such, it is a companion volume to the monograph edited by Yves Crama and Peter Hammer (who are also the editors of the volume under review), *Boolean Functions: Theory, Algorithms, and Applications*. Cambridge University Press, Cambridge, UK 2010.

2 Summary

The book is divided into five parts. The first part explores algebraic approaches to Boolean functions and the second one looks at Boolean functions from the logical perspective. The third part studies applications in learning theory and cryptography and the fourth investigates graph representations and efficient models of computation. The last fifth part is devoted to applications of Boolean functions in engineering.

Part I. Algebra.

The book begins with a chapter on a classic topic of complete functions. A set of Boolean functions F is *complete* if every Boolean function is a composition of functions from F. It is a *clone* if it is closed under composition and contains all projections. Chapter 1, by Reinhard Pöschel and Ivo Rosenberg, is an introduction to the theory of complete functions and clones. It presents the classic results of Post in this field (with proofs).

Chapter 2, by Jan C. Bioch, deals with decompositions of Boolean functions, a topic whose beginnings go back to the switching theory developed in 1950s and 1960s. The chapter presents the theory of modular sets of general Boolean functions and a refinement of this theory to the class of positive Boolean functions. The chapter also discusses applications and briefly treats computational aspects of the topic.

Part II. Logic.

Proof Theory is the topic of Chapter 3 contributed by Alasdair Urquhart. The chapter discusses equational calculus, traditional axiomatic proof systems in the style of Frege and Hilbert. It concludes with an analysis of complexity of proofs in resolution systems and a number of exercises for the reader.

The probabilistic and average-cases analysis may give insight into what algorithms for testing satisfiability might be effective and why. John Franco talks about "Probabilistic Analysis of Satisfiability Algorithms" in Chapter 4 and provides a comprehensive picture of tools available in this area: from basic methods to the most advanced ones.

John Hooker, in Chapter 5, describes applications of optimization methods in logic. The departure point it interpreting 0 and 1 as numerical values rather than "false" and 'true" and interpreting $\wedge$ as $\cdot$ (multiplication) and $\vee$ as $+$ (addition). The chapter discusses integer programming formulations, their linear programming relaxations, cutting planes, Benders decomposition, and Lagrangian relaxations and many others. It includes a few exercises.

Part III. Learning Theory and Cryptography.

In Chapter 6, Martin Anthony explores learnability of Boolean functions in the probabilistic context. Given values of a Boolean function f on some specific points together with the information that f belongs to a particular family of functions, we want to infer information about f. Probabilistic means that the samples of data are drawn randomly and learning is successful in the probabilistic sense.

Chapter 7, by Robert H. Sloan, Balázs Szörényi, and György Turán, is devoted to the topic of "Learning Boolean Functions with Queries". In this model of machine learning, it is assumed that the function belongs to a given class of functions and the goal is to identify the function exactly by asking questions.

Boolean functions appear naturally in coding theory and cryptography. Claude Carlet discusses applications in these two fields in Chapter 8. The idea is to identify properties that must be satisfied by error-correcting codes and cryptographic functions and see what these properties mean in terms of Boolean functions. In Chapter 9, the same author extends this theory to vectorial (that is, multi-output) Boolean functions and while most chapters in this book can be read separately, Chapter 8 is a prerequisite for Chapter 9.

Part IV. Graph Representation and Efficient Computation Models.

A decision diagram of a function f is a directed acyclic graph that provides a concise representation for f. Binary Decision Diagrams are studied by Beate Bolling, Martin Sauerhoff, Detlef Sieling, and Ingo Wegener in Chapter 10. Structural restrictions placed on Binary Decision Diagrams define Boolean functions with certain properties and this is one of the topics studied in this chapter.

Chapter 11 by Matthias Krause and Ingo Wegener is a short but comprehensive introduction to circuit complexity, a topic that has obvious connections to Boolean functions. The chapter covers such topics as efficient circuits for arithmetic functions, lower bounds on circuit size and depth, and voting polynomials. It finishes with a list of open problems.

Chapter 12 by Jehoshua Bruck explores the spectral representation of Boolean functions and describes techniques of Fourier analysis and its applications to threshold circuit complexity. The chapter also contains a short but useful bibliographic guide.

In Chapter 13, Martin Anthony investigates artificial neural networks and the relationship between different types of artificial neural networks and classes of Boolean functions a given type of network can compute. In Chapter 14, the same author introduces decision lists which is another way of representing Boolean functions. Natural restrictions on the type of decision lists lead to interesting classes of Boolean functions that are also described in this chapter.

Part V. Applications in Engineering.

Chapter 15, by J.-H. Roland Jiang and Titziano Villa, offers an introduction to the area of formal verification and studies the problem of "Hardware Equivalence and Property Verification" with a focus on Boolean functions. It covers a wide range of topics in this very important practical area.

In Chapter 16, Titziano Villa, Robert K. Brayton, and Alberto L. Sangiovanni-Vincentelli develop the theory of multilevel Boolean networks, a field of theoretical and practical importance. The main motivation for the study is to minimize the area occupied by the logic gates, delay, and power consumed, and also to improve testability.

In Chapter 17, Charles J. Colbourn explores "Boolean Aspects of Network Reliability". He is mostly interested in computing or bounding the reliability based on the combinatorial structure of operating or failed states.

3 Opinion

As a graduate student I took a course on Boolean and pseudo-Boolean functions taught by Peter Hammer (one of the co-editors of the book) and was fascinated by the multitude of neat and elegant results in the field. Ever since, even though I have no active research interest in this discipline, I enjoy reading and learning about the topic.

"Boolean Models and Methods in Mathematics, Computer Science, and Engineering" is likely to become the reference book for applications of Boolean functions. A common effort of twenty two authors and two editors, leading experts in their fields, the monograph, over 700 pages long, covers all important topics and presents state-of-the-art of applications of Boolean functions. This well-written volume can be useful to graduate students, researchers in discrete mathematics, computer science, or other fields, as well as practitioners of operations research and combinatorial optimiza-

tion. Also readers like me, who want to learn more about Boolean functions, will find this volume very informative.

The book is self-contained, but as mentioned before, is designed as a companion and complement to another volume that focuses on the theory of Boolean functions. The focus of this book is on applications and it will probably to the benefit of the reader to look at both volumes.

While writing styles of the authors naturally differ, much work has been put by the editors into making this volume consistent. Each chapter can be read separately but sometimes definitions should be looked up in previous chapters. This would be much easier if the book came with an index. Another improvement to consider in future editions, is to have a list of exercises for each chapter (now only few have it).

Review[9] of
Algorithmic Randomness and Complexity
Authors of Book: Downey and Hirschfeldt
List Price $99.00, Price on Amazon: $33.31
Springer, Hardcover, 883 pages[10]
Year: 2010
Author of Review: Jason Teutsch teutsch@cse.psu.edu

1 Introduction

The study of algorithmic randomness begins with an innocuous question, namely what is a random sequence? To illustrate this problem, the authors of *Algorithmic Randomness and Complexity* present the reader with the following two infinite sequences, one of which was generated via coin flips:

$$10110101110101011110000101010001011 1\ldots$$
$$10101010101010101010101010101010 10\ldots$$

Intuitively it is obvious that that only the first of these two sequences could possibly be considered "random," yet probability theory does nothing to confirm this feeling: any series of coin flips generated by fair coins is equally likely to occur. Most likely, the reader arrived at an opinion based on one or more of the following heuristics:

A random sequence is unpredictable. A gambler cannot become wealthy by gambling on the bits of a random sequence.

A random sequence is incompressible. A sequence with compressible prefixes has some recognizable pattern, and patterns are not random.

A random sequence is typical. Random have no distinguishable properties except those which also belong to most other sequences.

All three of these notions turn out to characterize the same classes of sequences, and the first half of *Algorithmic Randomness and Complexity* explores this phenomenon in detail. One can rigorously formalize the three randomness paradigms in terms of algorithms, whence the field "algorithmic randomness" derives its name.

Over the past decade, algorithmic randomness has become a remarkably productive research area, both in terms of popularity and number of results. The field not only tries to reconcile various formalizations of randomness but also tries to understand the interaction between randomness content and computational power. In terms of Kolmogorov complexity and computability theory, one might say that highly random sequences contain a lot of information but not much useful information. On the other hand, this perspective oversimplifies the situation. For example, the halting probability, a real number equal to the probability that a randomly selected (prefix-free) computer program halts, admits a computable approximation from below and is Martin-Löf random yet contains enough information to decide whether or not an arbitrary computer program halts.

[9]
[10] 883 Pages, $33.31 - this is no a typo

2 Summary

Downey and Hirschfeldt's book provides a comprehensive and detailed overview of the field of algorithmic randomness. The book collates the majority of results available at the time of publication, standardizes their notation, fills in literature gaps with folklore results and previously unpublished theorems, adds new proofs and results, and includes valuable historical perspective. This book serves as a go-to reference for the field, providing clear statements for theorems previously only accessible via ambiguous, missing, or contentious citations.

Although Downey and Hirschfeldt's book covers advanced topics, it is completely self-contained. The book starts out with a clear yet fast-paced 100-page introduction to computability theory, along the lines of Soare [5] and Odifreddi [3, 4], followed by a streamlined discussion on complexity of finite strings, including plain Kolmogorov complexity, prefix-free complexity, halting probabilities, and other complexity measures for finite strings (see Li and Vitányi's book [1] for an introduction). After the first four chapters on these topics, Downey and Hirschfeldt move on to areas not covered in other books, with the exception of Nies's book [2] which admits substantial overlap on some topics. Chapter 5 gives a gentle introduction to left-r.e. reals, or reals which have computable approximations from below, with some emphasis on presentations.

Chapters 6 and 7 introduce the central randomness notions of Martin-Löf randomness, computable randomness, Schnorr randomness, and Kurtz randomness, each of which are characterized in terms of the three paradigms mentioned earlier. Some of these characterizations are more appealing than others, yet their formalizations are uniform: each characterization consists of definitions involving martingales, initial segment complexity, or effective sets of measure zero (or one). Martin-Löf randomness, being the most-studied randomness notion, receives a chapter all to itself and has earned its popularity due to various nice properties, including Miller and Yu's Ample Excess Lemma and van Lambalgen's Theorem. Martin-Löf randomness, however, lacks a nice characterization in terms of the unpredictability paradigm, a shortcoming which leads to a central open problem of this field: is Martin-Löf randomness the same as Kolmogorov-Loveland randomness?

Chapter 8, the longest non-introductory chapter of this book, contains core results relating randomness to information content via Turing degrees. Results here include:

- the Kučera-Gács Theorem, which says that any set is computable relative to some Martin-Löf random,
- a previously unpublished result from Kautz's thesis which says there is a Martin-Löf random such that every noncomputable set which is computable from it is Turing equivalent to a Martin-Löf random (this also follows from Demuth's Theorem),
- a theorem of Nies, Stephan, and Terwijn which says that the Martin-Löf random reals, computable random reals, and Schnorr random reals can be separated within the high Turing degrees but coincide outside of them, and finally
- a result of de Leeuw, Moore, Shannon, and Shapiro that if the Lebesgue measure of sequences X such that A is c.e. in X is nonzero, then A is itself c.e.

Chapters 9 and 10 investigate other measures of relative randomness and includes the elegant Kučera-Slaman Theorem which says that a real is left-c.e. and Martin-Löf random iff it is the weight of some universal prefix-free machine's domain.

I was especially pleased to find a clear exposition on the deep theorem of André Nies which states that the following statements are equivalent:

- X is K-trivial (the n^{th} prefix of X contains no more information than the number n, which is as little as possible)
- X is low for K (X does not make it easier to compress strings), and
- X is low for random (X does not help with derandomization).

Chapter 11 is dedicated to discussion of this important result and K-trivials in general. For example, every K-trivial set is superlow, and therefore any noncomputable c.e. K-trivial set has incomplete Turing degree. There exists such a set with a single line construction which provides not only an injury-free but priority-free solution to Post's Problem. Randomness frequently simplifies ideas from recursion theory; what once required an involved construction to build a weak truth-table complete but non-truth-table complete set can now be done instead by choosing any left-c.e. Martin-Löf random real.

Chapter 13 discusses algorithmic dimension, which deals with sequences which are only "partially" random, as measured in various ways. Effective Hausdorff dimension, unlike its classical analogue in analysis, has multiple characterizations in terms of randomness which indicate that the concept is intuitive and robust. These characterizations include a definition in terms of open covers (like the classical definition), a definition in terms of martingales, and a definition in terms of Kolmogorov complexity. Prior publications asserted that these notions were equivalent but a cohesive explanation seemed missing. I appreciate Downey and Hirschfeldt's effort to piece these results together as well as their simple example showing that a sequence with full dimension need not be Martin-Löf random.

Downey and Hirschfeldt examine other topics as well, such as Kummer's Gap Theorem on the Kolmogorov complexity of c.e sets, and his theorem that the set of Kolmogorov random strings are truth-table complete.

3 Opinion

Anyone with some knowledge of computability theory and/or Kolmogorov complexity will enjoy browsing through this book, and anyone doing research in these areas will find this reference essential. The book is both well-written and well-organized, and having the results from this book as a systematic treatment brings one's attention to some facts which one might otherwise overlook or even fail to find in the literature. The authors meticulously attribute mathematical ideas with citations, and the index for the book is outstanding.

The authors of this book frequently argue proofs from primitive notions, as opposed to creating long chains of proofs which refer recursively back to previous results. They often improve on the notation of previous papers and sometimes simplify arguments found elsewhere in the literature. The result is a highly readable book.

While I am thrilled with both the writing and the topic of this book, the book itself is physically unwieldy at just under 900 pages. The authors completed *Algorithmic Randomness and Complexity* in the wake of Nies's groundbreaking result on K-trivials, and consequently many people chose to study lowness properties at that time. In keeping up with the field, Downey and Hirschfeldt wrote

more than three chapters on this topic (Chapters 11, 12, 14, and some of 15) and I find the book slightly overweighted in this area. I would have preferred to see a bit more on relationship between randomness and differentiability, randomness and ergodic theory, randomness and numberings, sets of minimal indices, and integer-valued martingales, however some of these connections and topics were not yet known at the time of publication.

I give this books two thumbs up for making a large amount of fascinating new material accessible to the mathematics and computer science communities.

References

[1] Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications*. Texts in Computer Science. Springer, New York, third edition, 2008.

[2] André Nies. *Computability and Randomness*. Oxford University Press, Inc., New York, NY, USA, 2009.

[3] P. G. Odifreddi. *Classical recursion theory. Vol. II*. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Co., Amsterdam, 1999.

[4] Piergiorgio Odifreddi. *Classical recursion theory*, volume 125 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1989. The theory of functions and sets of natural numbers, With a foreword by G. E. Sacks.

[5] Robert I. Soare. *Recursively enumerable sets and degrees*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987. A study of computable functions and computably generated sets.

Review of[11] of
Information Retrieval
By Buettcher, Clarke, Cormack
MIT Press, 2010
632 pages, Hardcover, $40.00

Review by
Paul Rubin `http://paulrubin.com`

1 Introduction

This is generally a good book. It might have saved me quite a lot of pain if I'd read it several years ago, always a good reason to appreciate a book. It does leave some unfulfilled desires, which I'll get to further down.

The book is a comprehensive guide to search engine implementation, covering the different stages from input tokenization through index construction and optimization, query handling, a lot of material about relevance ranking, and a little info on the care and feeding of large distributed search systems indexed using MapReduce. There are also a few chapters about non-search topics that still use related methods, such as text classification for language categorization or spam filtering. While it presents some sophisticated ideas, the book's mathematical level is not too demanding. It assumes a working knowledge of basic probability theory (the reader should understand concepts like probability distributions and Bayes' theorem) but (except in a few places) not much else. Although the technical mathematical background is not terribly high, the reader should have some experience in following mathematical arguments. Since a lot of the book is about low-level algorithms, some programming experience and basic understanding of algorithm complexity should also be considered prerequisite.

Several types of readers may be interested in the book: 1) CS or LIS (Library and Information Science) students examining search engines as part of a broader course of study; 2) developers of professional, high-performance search systems (perhaps employed, like the book's authors, at places like Google); and 3) application programmers wanting to incorporate search functionality into some surrounding application, typically based on a specific document collection, but who need more detailed understanding of search engine internals than they'd get from just running a canned search package and reading its manuals. For disclosure, the reviewer is in this last category, which will color this review.

2 Summary of Contents

There are 5 sections: 1) "Foundations" (3 chapters, about 100 pages); 2) Indexing (4 chapters, 150 pages); 3) Retrieval and Ranking (4 chapters, 150 pages); 4) Evaluation (2 chapters, 80 pages); and Applications and Extensions (3 chapters, 100 pages); plus a brief appendix about computer performance.

Each chapter of the book has a summary section that has references to further sources that look fine to me (I'm not familiar enough with the literature to evaluate the thoroughness of the

references), along with a bibliography. There is also a set of exercises with each chapter, some of which check the reader's understanding and others of which suggest programming projects. Most of the exercises looked pretty straightforward to me, though I only attempted a few of them. The "project" exercises might be more time-consuming than they look, since they call for processing Wikipedia dumps, which are relatively large (IIRC the article-only dump is around 50GB uncompressed XML) so they could be slow to process, especially in interpreted programming languages. It might be helpful if there was some indication of this type of issue along with the exercises.

Section 1 starts with a good introductory chapter, that describes at a high level what search and IR systems do and the problems that implementers face. For example, desktop search systems face higher update load and need to know more about file formats than web search systems, to support operations like searching mail folders for messages ordered by date. These aren't earth-shattering observations but the reader benefits from having the authors' hard-won experiences distilled into useful and potentially mistake-preventing advice. There is also a brief (1 page) list of non-search applications of IR, such as text classification (e.g. separating news stories as "politics", "business", "lifestyle") or multimedia IR (extending IR methods to images and video). Text classification is treated at more length later in the book, but most of the other applications don't show up again in any significant way. The rest of the section introduces IR concepts in somewhat more detail, describing IR terminology and the basic issues of document types, relevance ranking, and performance evaluation. Tokenization, stemming, stopwords, etc. are treated in detail sure to be useful to implementers.

Section 2 discusses index construction and retrieval in detail, giving lots of implementation and encoding tricks for both static and dynamic indexes. This is standard stuff of building inverted indices by sorting and compressing postings lists and so forth. A lot of compression methods are discussed, including some intricate ones that don't seem practical on current CPU's but might be suitable for FPGA implementation, perhaps on future reconfigurable processors that include FPGA fabric.

One issue with this section is that its techniques evolved over decades based on economic tradeoffs between RAM and disk space, described on p. 104. Those in turn reflect the characteristics of rotating disk drives, which these days for many applications are being displaced by solid state (SSD) flash drives with 100x less seek latency that drastically improve search responsiveness [Eskildsen 2009]. I only saw flash memory mentioned (briefly) in one place in the whole book, on p. 493. I think usage of flash memory is likely to affect indexing strategy, especially for applications like faceted search that (depending on implementation approach) may need much more seeking than normal lookups. As another matter, in commodity data center servers these days, the RAM capacity of a machine is limited by the number of RAM sockets in the motherboard as well as the cost. Only unusual motherboards can affordably take more than 32GB of ram, but adding much more capacity of disk or flash is routine. Beyond flash, Ousterhout et al (and others) have proposed multi-TB persistent data stores directly in the RAM of big server clusters ("RAMclouds", [Ousterhout2009]).

Faceted search is itself also not discussed, though it has become an important and expected feature in many search systems. Generally, though, this was a good section, and the parts about proximity ranking might yield some helpful improvements to software I've worked with.

Most of the benchmarks for this and other sections are based on the TREC (Text REtrieval Conference, http://trec.nist.gov) conferences and data collections. These now go up to about 1TB in size (p. 25).

Section 3 is primarily about ranked retrieval, and is the most important part of the book. As

a programmer who has tried in the past to grapple with the ranking problem without the benefit of much knowledge, I found the section both informative and frustrating. A quick summary of the book's approach to ranking is that each document is treated as a bag of opaque tokens. Scoring the results for ranked presentation is then treated as a problem in applied probability, based on (e.g.) frequencies of terms in particular documents as compared with the whole collection. The basic idea is to tune a scoring function to give scores matching human-supplied relevance judgments for some sample queries against the document corpus. For example, a set of such judgments is included in the TREC data. Different methods of extracting info from this sort of analysis (e.g. maximum-likelihood estimation of scoring parameters) is presented at length and in detail.

The treatment in this section strikes a good balance between solid grounding its ideas in probability basics, and stating some more advanced results without getting bogged down in detailed proofs. It gives concrete presentations of topics like relevance feedback (ch. 8.6). And yet it seems very traditional, trying to infer all the ranking info from the document content itself. To be sure, the book mentions use of the document structure (e.g. boost words found in the title field) for ranking (p. 54), and discusses PageRank (Google's use of the link graph between web pages for ranking web search) and so forth. But that seems (in the book) to be the exception rather than the rule.

In practice I think search implementers concerned in any serious way with ranking also have to be document curators, identifying semantic features of the documents that indicate relevance in the specific application, and figuring out how to extract those features and use them to best advantage, rather than going by statistics alone. PageRank is an example of this and is probably the biggest success story in search ranking. As a lower-impact example, for bibliographic ranking, I and co-workers had to come up with a number of ad-hoc tricks like "boost the rank of books by authors who have published multiple titles". I would have liked a much longer discussion of techniques like this rather than just a few mentions. Sections 12.5 (nontraditional effectiveness measures, 12 pages), 15.4 (p. 535 ff), and a few other places in the book have some more discussion along these lines, but I'd have leapt into 100's of pages on this type of thing with glee.

As an aside, the material about human assessment of search relevance (p. 73), suggesting that an assessor should be able to make a judgment in 10 seconds or less, might be reasonable for some collections but is unrealistic for others. Searching a large library catalog for a term like "calculus" retrieves hundreds of indistinguishable hits for different books with that title. The only way to usefully rank them is to actually relate the books to the real world, by finding published reviews, sales or circulation figures, journal or web page citations, etc. The implementation I worked on used a number of scoring factors weighted by manual adjustment, and while changing these weights led to different search ranks, it was quite hard to tell whether a given change made the results better or worse. One item on my agenda was to try regression-fitting the weights against Wikipedia page-view data for the 15,000 or so books in the catalog that were the subjects of Wikipedia articles, but I didn't manage to implement that while I was involved with the project. I would like to have seen advice about how to do things like that.

Chapter 10 of this section is about categorization (text classification, such as for language identification) and filtering rather than IR per se, again using a probabilistic approach. The chapter is worthwhile and informative. I liked that it discusses spam filtering, and the later web search chapter touches on this too.

Chapter 11 discusses fusion and metalearning, treating some machine learning techniques. I still haven't read this chapter carefully but as with the rest, it looks solidly informative without

being too theory-heavy.

Section 5 discusses parallel information retrieval, some further approaches to relevance ranking, web crawling, and other such topics. The discussion of how distributed search systems can be set up is informative. It discusses the basics of parallel indexing (including MapReduce) and two orthogonal ways of organizing distributed indexes for retrieval (partition by document or by term). It mentions (p. 503) that the literature on parallel IR is quite sparse compared with other topics in the book. Lin and Dyer [Lin2011] have a promising-looking new book on MapReduce that was presumably not available when the book under review went to press, that might help with this shortage of info. There is also a substantial ecology of code and documentation around the Apache Hadoop implementation (hadoop.apache.org) by now.

3 Opinion

Conclusion: I think this book is good as a class text to present an overview of search engines, or as an implementation guide for general purpose search systems using standard methods. Its mathematical level is about right for these purposes, but probably a bit too low for readers trying to develop and evaluate ranking strategies of their own (the situation I was in, though I still learned a lot from it). Developers of "industrial strength" search software will have to study further sources (and existing software), but the book is a good way to get started. People just looking for canned software to index moderate sized collections (up to 100GB, say) should probably just download Apache Solr. Solr's online docs are now better than they used to be, but as of a couple years ago Smiley and Pugh's book about Solr [Smiley2009] was very helpful.

The main areas I felt could use improvement were:

1. The computer technology coverage is a somewhat out of date, as mentioned above. Basically no mention of SSD's–all timings are with hard discs; and the parallel IR section is good but should be expanded. The section on free/open-source software on p.27 is also a bit weak, e.g. it doesn't mention Hadoop (distributed indexing) or Mahout (machine learning), both Apache projects, or Xapian, a probabilistic IR system that has been around since the 1980's.

2. Some discussion of "cloud" deployment could also be helpful: this again is a new technological development, making it possible to rent large numbers of remote servers by the minute rather than by the month, making distributed indexing available to "the masses". Being able to spin up 100 servers for a few hours at a time is very helpful to someone who must occasionally index a few TB of documents and can't afford a dedicated cluster for this purpose, and doesn't want to devote weeks to each index task on a handful of scrounged servers.

3. The book was apparently not written all at once since there is occasional apparent chronological skew between sections, but this is minor.

4. I wish there was more emphasis about how to gather and use semantic info from the documents for relevance ranking, that go beyond the bag of words. That is what I hoped for the most. I eagerly reached chapter 9, about language modeling, which I hoped would be about extracting document semantics from grammar or morphology. But while it was useful and interesting (it nicely explains the Kullback-Leibler divergence for comparing two probability distributions), it was just about more statistical processing of terms in the documents and queries.

5. (This is minor) The book would be more self-contained if it introduced probability concepts as it went along. However, it's all easy and standard material that instructors can present from other sources if students aren't familiar with it.

6. I also would have liked if user privacy was discussed more, though as a technical book, it's understandable that not much space is spent on this type of issue. There -are- a few mentions of the subject (e.g. pages 535 and 541 mention gathering relevance data from toolbars (with user permission) but it's not discussed at any length that I noticed, and "privacy" is not in the index.

Some roughly similar books I've seen include [Witten1999] (an older book, I'm familiar only with the earlier edition from the 1980's) and [Manning2009], which has less deep treatment of probabilistic ranking and somewhat different emphasis in other areas, but has the advantage of being available online in pdf form at no charge.

There is a body of literature that I'm not familiar with, about semantic analysis of text, that I think has to make its way into future open retrieval systems (it's surely already in closed ones). Statistical machine translation might also have some relevant techniques to apply to IR.

Note: This review was written in June 2011.

References

[Eskildsen2009] SSD Lightning talk at Code4Lib, 2009,
`http://code4lib.org/files/Flash-lightning-talk.pdf`.

[GM1982] Shafi Goldwasser and Silvio Micali, 1982, quoted by Oded Goldreich, "Foundations of Cryptography" draft, p. 365.
`http://www.wisdom.weizmann.ac.il/~oded/PSBookFrag/enc.ps`

[Lin2011] Jimmy Lin and Chris Dyer, Data-Intensive Text Processing with MapReduce, ISBN 1608453421,
`http://www.umiacs.umd.edu/~jimmylin/book.html`.

[Manning2009] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze, Introduction to Information Retrieval, Cambridge University Press. 2008;ISBN: 0521865719;
`http://www.informationretrieval.org`

[Ousterhout2009], The Case for RAMClouds: Scalable High-Performance Storage Entirely in DRAM. Appears in SIGOPS Operating Systems Review, Vol. 43, No. 4, December 2009, pp. 92-105.
`http://www.stanford.edu/~ouster/cgi-bin/papers/ramcloud.pdf`.

[Slashdot2010] How Google Trends & News Pollute the Web,
`http://tech.slashdot.org/story/10/07/28/0034232/`.

[Smiley2009] David Smiley and Eric Pugh, Solr 1.4 Enterprise Search Server, Packet Publications, ISBN 1847195881.
`http://www.packtpub.com/solr-1-4-enterprise-search-server/book`.

[Witten1999] Ian H. Witten, Alistair Moffat, and Timothy C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images (second edition), Morgan Kaufmann Publishing, ISBN 1558605703.

Review of[12]

Models of Conflict and Cooperation
by Rick Gillman and David Housman
American Mathematical Society
417 + xi pages, hardcover

Review by
Mark C. Wilson, mcw@cs.auckland.ac.nz
University of Auckland

1 Introduction

Game theory, which deals with modeling (optimal) decisions made by sets of interacting agents, has proved useful in many fields, in particular economics, political science, and biology. Although its predictive value is in many cases much less than one would hope for, it remains an essential tool for analyzing strategic interaction. In the last decade or so applications in computer science, under the name of Algorithmic Game Theory, have been prominent, and this has led to a renewed interest in computational questions in economics.

The standard textbooks on game theory are almost all written by economists with little computational perspective. More recently the short introduction by Shoham and Leyton-Brown has appeared, and it should prove useful for researchers from graduate student onward. Despite the appearance of more popular accounts of game theory and game theorists, a book that thoroughly covers the basics of the subject in detail for a general non-mathematical audience did not exist, to my knowledge, until the book under review.

2 Summary

The book is organized into 8 chapter of which the last 3 are more difficult than the others. Chapter 1 discusses "deterministic games" (sequential games of perfect and complete information, where win, loss or draw is the only outcome for a player). The key example is the game of Nim which is completely solved after a thorough discussion of heuristics and strategies. Backward induction and Zermelo's theorem are presented in detail. Chapter 2 gives a detailed discussion of player preferences, giving 5 types including ordinal and cardinal utilities. Chapter 3 deals with strategic (simultaneous or sequential) games, introducing such basic concepts as dominated strategies, maximin solution, Nash equilibrium, game trees, equivalence between game tree and payoff matrix representations. Chapter 4 introduces "probabilistic" (mixed) strategies" and shows how to find Nash equilibria in two-player games, when one of the players has at most 2 pure strategies. Chapter 5 moves on to discuss cooperation, the Prisoners' Dilemma and its repeated version. Chapter 6 examines how binding agreements can lead to improved payoffs for the players, giving detailed discussion of bargaining solutions such as those named after Raiffa and Nash. Chapter 7 deals with "coalition" (cooperative) games and solution concepts such as the Shapley allocation method

and the nucleolus. Chapter 8 concludes the book with a treatment of the problem of fair division, dealing with 5 "fairness" axioms including efficiency and envy-freeness, and four solution methods.

The authors have taught a course for general undergraduates based on the material in this book for over ten years. A notable feature is that several chapters open with an extensive Dialogue between characters (presumably meant to be undergraduate students) which previews and motivates the material in the chapter by means of a concrete problem. Emphasis throughout is on the process of mathematical modeling, followed by investigation of basic properties of the model, and then (in the later chapters) axiomatic characterizations and theorems.

3 Opinion

The book succeeds admirably in presenting material to its intended audience, which is, roughly speaking, North American undergraduates in the final two years of a degree who have a general interest in the topic. Explanations are careful and detailed, and the exercises contain a huge number of interesting applications. The material is developed logically, in a leisurely conversational style and with regard to what the reader can absorb. Each section comes with an explicitly stated learning goal. The mathematical requirements are minimal and much of the book would be appropriate for interested high school students. Students with very limited mathematical background would presumably develop confidence in elementary mathematics as a byproduct of studying the book, and this is apparently an aim of the book. It would also be very appropriate for self-study. The text is very clear and free from errors. The table of contents, index and typesetting are all good. More mathematically-oriented readers impatient with the slow pace would still learn much, as the key results are clearly signposted.

Apart from the deep thought which has clearly gone into the presentation, some of the content is unusual for a book at this level. In particular, the chapter on preferences covers very nicely a topic that is skipped over in most texts. I recommend this book very highly.

For computer science students and (potential) researchers interested in proceeding further, clearly considerable extra reading will be required. One possible minor difficulty would be that the book under review occasionally uses idiosyncratic (or at least nonstandard) terminology.

List of recommended further reading after finishing the book under review:

- T. Roughgarden, *Algorithmic Game Theory*, Comm ACM, July 2010;
- T. Roughgarden, *Computing Equilibria: A Computational Complexity Perspective*, invited survey for Economic Theory, 2010;
- K. Leyton-Brown, Kevi and Y. Shoham, *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*, Morgan & Claypool Publishers 2008 (also available free online);
- Y. Shoham, *Computer Science and Game Theory*, Comm. ACM, August 2008;
- *Algorithmic Game Theory*, edited by Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay Vazirani, Cambridge University Press 2007 (also available free online).

Review [13] of
Deterministic Extraction from weak random sources
by Ariel Gabizon
Springer Verlag, 2011
148 pages, HARDCOVER

Review by
Marius Zimand (mzimandtowson.edu)
Towson University, Department of Computer and Information Sciences

1 Introduction

Virtually all sources of randomness (such as the decay time of radioactive particles, atmospheric noise, the last digit of stock indices, etc.) produce sequences that do contain some randomness, but, alas, of rather poor quality as some of their bits are biased and correlated. More formally, such sources can be viewed as distributions that, while having positive entropy, are far from being perfectly random. For this reason, these sources are called *weak random sources*. A *randomness extractor* is a procedure that takes as input a weak random source, with the promise that it belongs to a certain class $\mathcal{C}$ of distributions (guaranteeing some properties and parameters of the input), and outputs a string that is almost perfectly random. In addition to the primary utilization described above, extractors have found many applications in derandomization, cryptography, data structures, Ramsey theory, and other fields. Naturally, one would like to have a polynomial-time extractor for a large and general class of sources. Specifically, one would like to have an extractor for the class $\mathcal{C}_k$ consisting of all distributions over the set of n-bit strings (or, in some situations, over the set of n-sized vector with elements from a field $\mathbb{F}$) having min-entropy at least k, where $k < n$ is a parameter (we use min-entropy because it is more useful in applications than Shannon entropy). Unfortunately, there exists no algorithmical procedure (not even an inefficient one) that can extract from all distributions in $\mathcal{C}_k$. Therefore one has to consider either (a) the case when the extractor in addition to a distribution in $\mathcal{C}_k$ also has, as a second input (called the seed), a few perfectly random and independent bits (such extractors are called seeded extractors), or (b) the case when the input sources besides being in $\mathcal{C}_k$ are guaranteed to have certain structural properties. Extractors for case (b) are generically called *deterministic extractors*. Ariel Gabizon's book presents the construction of several deterministic extractors.

2 Summary

The book is based on the author's doctoral thesis at Weizmann Institute. Each of the Chapters 2,3,4, and 5, represents an article by Gabizon and coauthors. An Introduction (Chapter 1) and an Appendix with two sections complete the book.

Chapter 1 introduces the concept of randomness extractor and presents one of the main technique, the recycling paradigm, that will be a recurring theme in the following chapters. The recycling paradigm states that, in certain circumstances, bits extracted from a source behave in

the same way as random bits that are independent of the source and thus can be used as the seed in a seeded extractor. The main idea is illustrated in this chapter through a simple and short (less than one page) example.

Chapter 2 shows the construction of a polynomial-time deterministic extractor for the class of *bit-fixing sources*. An (n, k) bit fixing source is a distribution over the set of n-bit strings, in which $n-k$ bits are constant and the remaining k bits are random and independent. Compared to previous constructions in the literature, the extractor presented here improves two essential parameters: (a) it is able to extract even if the min-entropy of the source is very low (more precisely, for $k = \text{polylog}(n)$), and (b) the number of extracted bits is asymptotically equal to the min-entropy of the source.

Chapter 3 considers the case of deterministic extractors for *affine sources*. In this setting, the input distributions range over $\mathbb{F}^n$ and an (n, k) affine source is a distribution that is uniformly distributed over a k-dimensional affine space of $\mathbb{F}^n$. The chapter shows the construction of a polynomial-time deterministic extractor for the case when the field $\mathbb{F}$ is large, having size at least n^c for some constant $c > 0$, and for arbitrary low k. Previous extractors for affine sources were for the more basic case when the field is GF[2], but, on the other hand, were either extracting just one bit, or worked only for min-entropy k equal to a constant fraction of n.

Chapter 4 constructs polynomial-time deterministic extractors for *polynomial sources*. A polynomial source is a distribution over $\mathbb{F}^n$ sampled by n low degree polynomials with k variables. Such sources generalize affine sources and randomness extraction for them has not been studied before. The construction uses deep tools from algebraic geometry, which are very interesting because they are not common in constructive combinatorics.

Chapter 5 focuses on *zero-error dispersers*. A zero-error disperser is a procedure that takes as input a distribution from a class of weak sources and outputs a distribution that assigns non-zero probability mass to each element in its domain (but not necessarily in a balanced way). The chapter presents a general method of increasing the output length of zero-error dispersers. For some type of sources (including bit-fixing sources and affine sources over large fields), the output length is a constant fraction of the source min-entropy.

Appendix A presents some technical details regarding the sampling method used in one of the constructions in Chapter 2. Appendix B reviews concepts and techniques from Algebraic Geometry. This provides the technical background underlying the constructions in Chapter 4.

3 Opinion

The book is focused on a narrow topic that is at the forefront of current research, which is typical for books based on a good doctoral dissertation. Consequently, its primary audience consists of researchers in computational complexity and combinatorics, which will enjoy the coherent presentation of the strong results obtained by Ariel Gabizon (and his coauthors) in a difficult field. The good quality of the writing, the friendly Introduction, and the Appendix on Algebraic Geometry (an area unfamiliar to most SIGACT members and which has a clear potential for interesting applications) are some of the factors that may attract a larger category of readers.

Review [14]

Applied Information Security
by David Basin, Patrick Schaller, and Michael Schläpfer
Springer, 2011, 216 pages, $36.50
Reviewed by Jonathan Katz, Dept. of CS , U. of MD

Teaching computer security can be a challenge. While there are several reasons for this — the topic could be an essay in itself — one reason is the difficulty of developing projects/labs for the students to work on. Here I am not only referring to the step of *imagining* a project that will be simultaneously interesting, instructive, and doable in a reasonable amount of time (though this can be tough, especially once you get beyond the buffer-overflow and web-scripting attacks that everyone seems to use); I mean also the time-consuming job of actually *implementing* the lab and getting it ready for students to use. To some extent, this is a challenge shared by any system class, though my impression is that for courses in networking, operating systems, and the like the standard textbooks come with a set of lab projects for instructors to use as part of their course. Computer security, unfortunately, seems to suffer from a lack of good textbooks and project materials in the first place.[15]

It was therefore with great anticipation that I picked up this book; in fact, I was hoping to use it as part of a *Cybersecurity Lab* class I am teaching this semester. The book promises a "hands-on approach," and contains links to virtual machines (VMs) that can be freely downloaded and used to run the labs outlined each chapter. Finally (or so I thought), teaching computer security would be easier.

1 Summary of the Book

The book begins with a discussion of "security principles" in Chapter 1. There is nothing new here — the authors mainly cover the principles outlined by Saltzer and Schroeder over 30 years ago — but the treatment is adequate and the examples illustrating the various principles are up-to-date and stimulating. Chapter 2 gives instructions on setting up the Linux VMs to be used in the remainder of the book.

In Chapter 3, the authors provide step-by-step instructions for using Nmap (a port-scanning tool) and other programs to scan a remote system and learn information about it. Wireshark (a network-level analyzer) is used to observe the network packets sent and received during the course of these scans. The vulnerability scanner OpenVAS is used to identify potential attack vectors, and Metasploit is introduced as a means of exploiting these vulnerabilities. Commendably, the authors also cover the defensive side of things, focusing on a network administrator's view during the course of the above attacks, and describing actions that can be taken to minimize risk. I have a feeling students would enjoy working through the examples covered in this chapter.

Chapter 4 turns to the problems of access control and authentication. The differences between telnet (which sends passwords in the clear) and ssh (which does not) are explained, and students are given the opportunity to see this in action using the provided VMs. This is followed by a discussion of basic permissions in Linux (for which a special-purpose VM is not really needed), and coverage of setuid/setgid, shell-script security, and chroot, among other topics.

[14]
[15] An exception are the labs provided by the SEED Project; see `http://www.cis.syr.edu/~wedu/seed/` .

In Chapter 5 the authors discuss mechanisms for logging system-level information and performing basic analysis of such logs. This leads to a brief digression about intrusion detection, and techniques adversaries might use to hide files.

Chapter 6 deals with web-application security. Covered here are SQL-injection attacks, basic authentication techniques, cookie management, and cross-site scripting (XSS) attacks. The chapter concludes with a brief discussion of SSL. Public-key cryptography and digital certificates are covered in Chapter 7, and the lab components here deals with generating keys and certificates. The final chapter of the book treats the topic of risk management; this part does not have an associated lab.

The book ends with several appendices that provide guidelines for using the text in a course, along with a review of basic Linux tools. Answers to all the exercises are also included.

I was planning to try out several of the lab exercises as part of this review. Disappointingly, I was unable to do so. Three VMs need to be downloaded in order to run some of the labs; each of these takes a significant amount of time to download and set up. I did manage to download one of the VMs, but I found it unacceptably slow. I don't (entirely) blame the authors for this — I'm sure, with more time, I could have gotten it to work properly — but it does indicate some of the difficulties that would be involved with using this book in a course.

2 Recommendation

As mentioned earlier, I was hoping to use this book (or the lab exercises) as part of a course I am teaching this semester; I also teach computer security regularly and would be thrilled to have a lab-based textbook to use there. In the end, I decided against using the book this semester, though I could imagine using one or two of the labs in the future. I do not think I could use this as the primary textbook in a computer security class.

One of the problems is that the book is lacking any sort of foundational material. The book talks about the differences between telnet, where the password is sent in the clear, and ssh, where it is not, but if you want to find out anything about what the ssh protocol *is* you will have to look elsewhere. Changing access permissions in Linux is described, but no information about access control in general, or different access-control models, is given. That is a general feature of the book: it provides just enough information to run the commands needed to execute the lab, but not much more beyond that.

In fact, I found the book on the whole rather thin. Chapter 3, which covers a range of topics, is only 18 pages long! (Entire books have been written about Nmap and Metasploit alone.) Chapters 3–7, which constitute the core "technical" chapters I could cover in class, are only 85 pages in total (including Exercises). There is simply not enough "meat" here for an entire semester.

On the positive side, the book and associated labs could be useful for a student learning computer security on their own. The student would have to supplement the material here with material from other sources, but the labs would provide a solid way of testing what the student has learned.

Rebuttal from the Author David Basin:

As explained in the book's preface: "Our goal in writing this book is to provide a hands-on experimental counterpart to the more theoretically-oriented textbooks available. We approach Information Security from the perspective of a laboratory where students carry out experiments, much like they do in other courses such as Physics or Chemistry. [...] Just like with other lab courses, this book is not intended to be a replacement for a theory course and associated textbooks; it is complementary and has the aim of building on and extending the students' knowledge."

Given this, the reviewer's critique of the book as missing foundations is off target. Students/Readers are expected to have already learned the necessary foundations, or to be able to find the required information on their own.

Concerning download and setup, it is true that the virtual machines are large. We decided against distributing them with the book as this would increase the book's price. If you wish to use our virtual machines at your university, we suggest downloading them once and mirroring them on your own infrastructure or providing them on a memory stick or any other medium during the first lecture. We have received positive feedback from instructors and students at numerous universities, all of whom have downloaded and used the virtual machines without problems.

Review of[16] of
Introduction to Bio-Ontologies
by Peter N. Robinson and Sebastian Bauer
CRC Press, Taylor & Francis Group, 2011
488 pages, HARDCOVER

Review by
Mohsen Mahmoudi Aznaveh (mahmoudi.mohsen@gmail.com)

1 Introduction

The amount of data in the field of bioinformatics is increasing daily. This data needs to be processed and refined so that useful information can be obtained. The need to manage and understanding this data in biology leads researchers to Ontologies. Ontologies help scientists to automate the processing of information quickly and (hopefully) free of error. Several different types of ontologies have been studied. Gene Ontology was one of the most successful integrations. Ever since other type of ontologies have been looked at.

2 Part I

Chapter 1 gives a rough view of ontology including applications o bioinformatics. Chapter 2 gives the reader the requisite mathematical knowledge needed for this book. This includes propositional logic, first order logic, description logic, and Set Theory. Chapter 3 is on basic statics. It includes Bayes theorem and some of its application in the field were explained, Graph theory, and Bayesian network. Chapter 4 offers an overview of two important ontology languages, OBO and OWL. A brief description of XML in the first chapter leads us to better understand these languages stanzas. They then compare the two languages.

3 Part II

In this part some important ontologies and their different use are shown. The most used ontologies discussed in the chapter and at the end is an interesting survey.

Gene Ontology (GO) is the most famous and successful ontology in the field. Even so, a huge amount of effort is needed to improve it. GO lets researchers classify genes in terms of their functional properties. GO has three subontologies: Molecular function, biological function and biological process. Different types of relation in GO are also explained in this chapter, for example **is_a**,**instance_of** and **part_of** are different types of relation and their exact definitions.

Chapter 6 provides the reader with upper-level bio-ontologies which deals with interaction of relation of different ontologies. In this chapter two important upper-level ontologies: (1) Basic Formal Ontology and (2) Relation Ontology, are discussed. Chapter 7 illustrates some current bio-ontologies and explains the way new ontologies are born. It also discusses what makes good ontologies and what is the best way to make an ontology.

4 Part III

In this part graph algorithms used in bio-ontologies are introduced. Chapter 8 starts with overrepresentation analysis, which is the application of bio-ontologies tor molecular biology. This chapter shows some improvement and also explains that there is not one "right" answer in overrepresentation analysis; the answer depends on the researcher's view. Hence results and/or their interpretations may differ.

Chapter 9 explain model-based approaches to GO analysis. This problem is a kind of optimization problem. This method uses Bayesian network model to find the optimum answer. The algorithm mostly discussed in the chapter is MGSA which can address the problem effectively.

Chapter 10 is about semantic similarity. The amount of information in specific ontology relates to Shannon theory. Entropy of two different ontologies can make a good view of information in them. Chapter 19 discusses algorithms by the authors of the book that are for searching results in a database.

5 Part IV

This part discusses computational reasoning and logic. Inference in GO was discussed in the chapter 12. Bonding between different ontology and knowledge gleaning was also discussed in the same chapter. Chapter 13 introduce formal semantic concepts and entailment rules in RDF and RDFS. Derivation of these rules are also explained in this chapter. Chapter 14 is about semantics in properties, Classes and schema vocabulary. A simple example is used throughout the chapter to demonstrate inference rules for OWL ontologies. Chapter 15 introduces algorithmic foundation of interference. It starts with tableau algorithm and introduces implementation aspects of the algorithm. In the last chapter of the book SPARQL is introduced. SPARQL is a simple query language for RDF graphs. Querying exact data is important in bio-informatic and SPARQL is designed to address this problem.

6 Opinion

The applications of ontology to bio-informatic is a fast growing field of research. Hence the lack of a good resource book is felt. This book is one of the first source books in the field; it is well written and coherent. Its introduction gives the reader a good taste of what comes next and it also contains good exercises. In my opinion the only drawback of the book is lack of consistency between mathematical and biological terms, which make it sometimes hard to follow.

Review of[17] of
The Dots and Boxes Game: Sophisticated Child's Play
by Elwyn Berlekamp
A K Peters, 2006
132 pages, Softcover

Review by
Omar Shehab (shehab1@umbc.edu)
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County

1 Introduction

It is very difficult to find someone who has never played the Dots and Boxes game in any part of her life. It may also be very difficult to find someone who has worked out all the mathematics behind this game just for the sake of playing it. Elwyn Berlekamp, the author of this book is one such person. If you are now wondering whether you would have beaten your classmates more often if you had known the math, this book is for you. This will set you up with fun and confidence if you are planning to arrange an elementary school reunion anytime soon. This book tells how a carelessly chosen move, made may be decades ago on the back of your notebook, had crushed your opponent at school. It tells how some of the mistakes you made then could have been avoided if you knew the amazing theory of the ever changing game board. This book is about the secret role of mathematics played during some of the fun you had long ago.

2 Summary

The book is divided in twelve chapters. Half of them are dedicated for discussing various ideas while the rest are just worked out problems based on previous chapters. The audience will rediscover one of their favorite childhood games throughout the book. The main approach is to characterize the Dots and Boxes game from a mathematical point of view. Based on the characterization a number of theorems are proven and several corollaries are drawn. There are dedicated chapters for discussing opening, middle and end games. So, chess inclined readers may find a familiar experience. The author did a nice job by splitting the discussion and following worked out problems in separate chapters. To some readers, this might be the best part of the reading. At the end of the book, a bunch of unsolved problems are presented to challenge the reader with the experience she just had. Here is a summary of all the chapters.

The book is divided in twelve chapters.

2.1 Chapter 1

The Dots and Boxes game is introduced in the first chapter. The author then introduces several tricks and openings which increases the chance of wining the board. It starts with 'double crossing',

a strategy to trap the opponent over a short term gain and take over the control. The immediate next thing to do is to keep the control as much as you can. The chapter claims that keeping long chains open allows a player to put the opponent into pressure in the long run. The formulae, based on which the strategies are developed, are fairly simple and intuitive. Later several options for opening and variants of the game are also discussed.

2.2 Chapter 2

This chapter is very short which means literally two pages. So, any one can finish it in jiffy and carry on. It explains how the Strings and Coins game is equivalent to the Dots and Boxes game. But with strings, coins and scissors it is more real life. Coins are glued to the string so that any string is attached to two coins on both ends or to the ground at best on one end. A player gets the coin when her cut makes it fall on the ground. Winner is who earns the most. Theoretically both games are equivalent but people differ occasionally on how much they like one over another.

2.3 Chapter 3

The third chapter is no talk and only work. There are sixteen problems to make it sure that you got the idea of keeping long chains open right. The meat of the discussion is already mentioned in the first chapter. So, it is good to have a sand box to get some hands on before getting into deeper theories. Virtually, it is the exercise on the topics covered so far.

2.4 Chapter 4

This chapter justifies and elaborates the claim made in the first chapter that long open chains are preferable. The idea of 'longness' is expanded for loops too. The constraints for long loop and chain are formalized in this chapter. Getting the opponent to create a long loop or long chain is highly desirable. Those blunders are also classified in this chapter in three 'loony' moves. In terms of chess, we can say the middle games are elaborated in this chapter. In a middle game, the number of long chains are not always obvious from visual inspection. Mathematical formulae are provided to determine the number. This information is very important to the player who has the next move. Several scenarios illustrate the discussion in depth.

2.5 Chapter 5

This chapter is another no talk and only work one. Seventeen worked out problems elaborate the ideas mentioned in the previous chapter in different scenarios. Working out those problems by the reader herself is strongly recommended before she proceeds to the following chapters.

2.6 Chapter 6

This chapter introduces the game of Nimstring. It starts with proving that Nimstring is a special scenario of the Strings and Coin game. The strategies for winning an end game are derived from the strategies of the Strings and Coins game. An interesting claim made at this point that the winning strategy of a Strings and Coins game can be reduced to that of a Nimstring. Based on this point, the book keeps on making the reader a master of the later game. The start and end game strategies are well elaborated. For most of the strategies, equivalent strategies for Strings and

Coins game are also inferred. To formalize the strategies, ideas like Nimber value and Nimstring graph are introduced. A recursive 'Mex' function, i. e. **M**inimal **ex**cludant, is used to get the Nimber value. Finally the Nimber value is used to assess the situation while playing an end game. It was said at the starting that to ace the Strings and Coins game one has to master the Nimstring game. So, the chapter aptly ends with establishing the connection between the Nimstring game and counting the number of long chains in Strings and Coins.

2.7 Chapter 7

This chapter makes it sure that the reader gets the theories of the Nimstring game right. So, again, it is a no talk and only work chapter with eight problems elaborating the concept of Nimber value.

2.8 Chapter 8

This chapter presents the advanced ideas of a Nimstring game. It was suggested that when the length of a chain is at least three, the Nimstring value can be considered constant. This implies the concepts of 'mutation', 'chopping' and 'changing'. A theorem and several corollaries are drawn for mutation in different scenarios. The sub concept of 'vine' is also introduced. Vine is a sub part of the Nimstring graphs. Scenarios in a Nimstring game can be commonly interpreted as different classes of vines. For example, there are Dawson's-vines and Kayles-vines. Finally, the chapter introduces specialized formulae to calculate the Mex function when the class of a vine, i.e. the sub graph of Nimstring, is known.

2.9 Chapter 9

This chapter carries the exercise version legacy of the previous chapter. Four advanced Nimstring problems are ready to entertain the reader with the assurance of understanding the following materials better.

2.10 Chapter 10

This is virtually the last chapter of the book. As expected it is dedicate to the end games. A game gets more fascinating when it is almost over and the scores are very close. The chapter addresses this scenario in particular. The strategies are developed around the so called 'canonical move'. It is defined as the move which lowers the number of the Nimber value of the associated region. It is followed by three strategies about how to use the canonical move to win a game. The mathematical proofs of the claims made in the strategies are also presented with due importance. The special case of nodes with fewer connections are also addressed in the perspective of canonical moves. Finally an instance of a large board game is considered to illustrate the ideas.

2.11 Chapter 11

This is one of the largest chapters consisting twenty four worked out end game problems. The reader should be thrilled by looking at some of them.

2.12 Chapter 12

The last chapter is the one the reader has been waiting for from the very beginning. Yes, I am talking about the unsolved problems. This will challenge the reader with both imagination and the knowledge she has gathered so far. Nineteen problems are presented covering the whole material discussed throughout the book. The reader will certainly go back to previous chapters again and again while working them out.

3 Opinion

According to a quick web search, this is probably the only book dedicated to the Dots and Boxes game. Although some of the materials were not very intuitive to the reviewer, it was over all fun! It surely will intrigue the adults who still carries the love for the game. An advanced young student will find it as an amazingly different window to look at her presumably known world. Organization of the book is very assuring to the not-so-math-savvy audience as any chapter ends within a few pages. It will be an exciting journey for anyone who wants to rediscover their once favorite childhood pass time. So, get, solve and love it.

The P versus NP problem is a major unsolved problem in computer science. It asks whether every problem whose solution can be quickly verified can also be solved quickly. It is one of the seven Millennium Prize Problems selected by the Clay Mathematics Institute, each of which carries a US$1,000,000 prize for the first correct solution. The informal term quickly, used above, means the existence of an algorithm solving the task that runs in polynomial time, such that the time to complete the task varies NP-complete is a special category of NP problems that have time complexities greater than polynomial time, are verifiable in polynomial time, and belong to a set of problems known as NP-hard. NP-hard problems are essentially those that are at least as hard as the hardest NP problem, but donâ€™t need to be verifiable in polynomial time.Â This is why P = NP matters. We canâ€™t know for certain, but there is every reason to believe that the answer to that question runs right through NP-complete. First, any algorithm that returns a solution to an NP-complete problem in polynomial time can be modified to solve every single NP-complete problem in polynomial time, since they are all the same problem at their core.