The Book Review Column[1]

by William Gasarch
Dept of Computer Science
Univ. of MD at College Park
College Park, MD, 20742
email: `gasarch@cs.umd.edu`

Slight change in information given: in the past I have provided price and ISBN numbers of books. The notion of price is ill-defined: the website www.bestbookbuys.com gives a range of prices from place to purchase books on the web. (amazon.com is usually *not* the cheapest, and its usually not even close.). Providing ISBN numbers is also not needed in this electronic age where such information is available at the touch of a button. If the price is notable– either rather high or rather low, this might be mentioned in the review.

Welcome to the Book Reviews Column. We hope to bring you at least two reviews of books every month. In this column five books are reviewed.

1. **The Codebreakers: The story of secret writing** , by David Kahn reviewed by William Gasarch. This is a rather complete history of crypto up to World War II. The key question for us is, should someone doing research in cryptography read this book?
2. **The Code Book: The Evolution of Secrecy from Mary Queen of Scots to Quantum Cryptography** , by Simon Singh. This book review originally appeared in the Notices of the American Mathematical Society, March 2000, and is reprinted here with permission. Review by Jim Reeds. This is a recent book on the history of crypto that is quite popular. But is it accurate? Read the review to find out.
3. **DNA Based Computers V (Proceedings of a Dimacs Workshop)** edited by Eric Winfree and David K. Gifford, and reviewed by Mitsunori Ogihara. This is a collection of articles on DNA computing that were presented at the fifth DIMACS workshop on this topic.
4. **Basic Proof Theory** by A.S. Troelstra and H. Schwichtenberg, reviewed by Jeremy Avigad. This is a graduate level text in proof theory. It fills a gap since most works in this field are more specialized.
5. **Complexity and Real Computation** by L. Blum, F. Cucker, M. Shub, and S. Smale, reviewed by Tim McNicholl. This book sets forth a complexity theory over the reals analogous to the familiar one over strings.

I am looking for reviewers for the following books

If you want a FREE copy of one of these books in exchange for a review, then email me at gasarchcs.umd.edu If you want more information about any of these books, again, feel free to email me. Reviewing a book is a great way to learn a field. I have personally reviewed books and then went on to use what I learned in my research.

1. *Analysis of Algorithms: An Active Learning Approach* by McConnell.
2. *Combinatorial Optimization: Theory and Algorithms* by Korte and Vygen.

[1]© William Gasarch, 2001.

3. *Computational Commutative Algebra* by Kreuzer and Robbiano.
4. *Petri Net Algebra* by Best, Devillers, and Koutny.
5. *Model Checking* by Grumberg and Peled.
6. *Graph Theory* by W.T. Tutte.
7. *Introduction to Distributed Algorithms* by Gerald Tel.
8. *Computational and Geometric Aspects of Modern Algebra* Edited by Atkinson, Gilbert, Howie, Lipton, and Robertson. (Proceedings of a workshop)
9. *The Clausal Theory of Types* by Wolfram.
10. *Introduction to Process Algebra* by Fokkink.
11. *Learning Automata: Theory and Applications* by Najim and Poznyak.
12. *Algorithmic Geometry* by Boissonnat and Yvinec.
13. *Algorithms Sequential and Parallel: A unified approach* by Miller and Boxer
14. *Parallel Processing and Parallel Algorithms: Theory and Computation* by Roosta.

The following are DIMACS workshop books which are collections of articles on the topic in the title.

1. Constraint Programming and Large Scale Discrete Optimization.
2. Discrete Mathematical Problems with Medical Applications.
3. Discrete Mathematical Chemistry.
4. Randomization Methods in Algorithm Design.
5. Multichannel Optical Networks: Theory and Practice.
6. Networks in Distributed Computing.
7. Advances in Switching Networks.
8. Mobile Networks and Computing.
9. Robust Communication Networks: Interconnection and Survivability.

Review of
The CodeBreakers:
The Story of Secrete Writing
Author: David Kahn
Publisher: Scribner, over 1000 pages!
List Price $65.00, but cheaper on www.bestbookbuys.com
Reviewer: William Gasarch (`gasarch@cs.umd.edu`)

1 Overview and Summary

The Codebreakers chronicles the history of crypto (both making codes and breaking them) from antiquity until the advent of public key cryptography. The first edition was published in 1967– before most of the World War II documents were made public. The second edition was published in 1996 and includes more info on World War II and some on public key. The later treatment is sketchy; hence the book is best viewed as a history up through World War II.

The first chapter discusses crypto around the time of the attack on Pearl Harbor. The chapter is longer then it needs to be but does give an informed and complete picture. As such it is a metaphor for the entire (1301 page) book.

Chapters 2-19 are (roughly) chronological. Chapter 2, titled "The first 3000 years" tells of crypto from 2900 BC until 1600 AD. Of most interest here: The Arab world knew about frequency analysis around 1350 but this knowledge was lost. This chapter also contains a good explanation of frequency analysis. Chapters 3,4,5, and 6 discuss the rise of The West in terms of crypto. Many code systems and the context in which they were used are discussed. These chapters cover the years 1300–1850. Chapters 7,9,10,11,14,15,16, and 17 cover the use of codes in a variety of wars. All of these chapters portray crypto as the key to a battle or even an entire war. The problem of key distribution is seen here in very real terms. Chapters 8 and 12 cover the use of crypto in diplomacy. The ethics of reading someones mail is mentioned; however, most countries had no problem with this. Chapter 13 discusses the use of codes in business. The 1-time pad was first used here (in the early 1920's). Chapter 18 discusses the history of Russian crypto. It is a prelude to chapter 19 which is about the cold war and the founding of the NSA.

Chapters 20-27 are an assortment of topics. These include the following.

1. The type of people are good at this work.
2. Crypto has been used to decode old encoded diplomatic documents. This has been of help to historians.
3. Crypto and linguistics have been used to read old writings in dead languages that were not ciphered (but are still quite difficult) . This has been of help to historians and anthropologists.
4. The inverse problem of crypto— making a message so clear that others (e.g., space aliens) can read it.
5. Public Key crypto.

2 Opinion

The key question for readers of this column is "If I want to work in or teach a course in crypto, then should I read this book?" The answer is an emphatic YES for the following reasons.

1. The need for public key crypto is realized vividly when contrasted with the history of the field. One can say that public key crypto has solved an 3000-year old open problem.

2. Many "unbreakable code" have been broken. This reinforces the need for rigor and at the same time gives us a healthy skepticism of that rigor.

3. There are many stories here to tell a class to liven up a lecture.

4. It is good to have a notion of which systems were actually used. My class was surprised that 1-time pads were ever actually used— they had assumed they were just some weird theory-thing.

In summary, I recommend this book for anyone interested in teaching or doing research in crypto. Too often we forget the origins of the problems we are working on. This book is an excellent solution to that problem in the field of crypto. There is one caveat: the books material on post WW II material is incomplete. For this there are other books (see some of the books referred to in the next review).

The next book review originally appeared in the Notices of the American Mathematical Society, March 2000, and is reprinted here with permission. Copyright 2000 by Jim Reeds.

Review of
The Code Book:
The Evolution of Secrecy from Mary Queen of Scots to Quantum Cryptography
Author: Simon Singh
Publisher: Anchor Books, 410 pages
List Price $14.00, but cheaper on www.bestbookbuys.com
Reviewer: Jim Reeds (`reeds@research.att.com`)

It is hard to write a good book about the history of cryptography. The subject is technical enough to be a turnoff for many readers. The evidence a historian of cryptography works from is often suspect. Because much of the practice and research in the field was carried out in secret, any particular document or interview must be viewed with suspicion: did the author or interviewee know the full truth? Healthy suspicion about the competency of sources is of course appropriate in all branches of historical research, but in the history of cryptography the proportion of misinformed or deceptive sources is probably greater than generally found in the history of science or of technology. The historian's standard technique of precise and thorough citation of documentary evidence is therefore especially important in the history of cryptography. Unfortunately, for popular works this technique can mean death to readability.

In cryptography technical developments often came in reaction to events and activities which were at the time secret, or, conversely, had ceased to be secret. If we do not understand the "who knew what when" details correctly, our reconstructed time tables for technical progress seem to show puzzling fits and starts, apparently unconnected with contemporary events. This makes coherent exposition difficult. Almost every war, however, has notable instances where some cipher message solution foils a plot or wins a battle. Here it is easy to connect the cryptographic or cryptanalytic technicalities with particular historical events, but a book that relies too much on such instances becomes in effect no more than an adventure story anthology.

So it is no surprise that there are few general surveys of the history of cryptography, and fewer good ones. The rule of thumb seems to be one new book every thirty years.

In 1902 and 1906 Alois Meister published his immensely scholarly *Die Anfänge der Modernen Diplomatischen Geheimschrift* and *Die Geheimschrift im Dienste der Päpstlichen Kurie*, reproducing and summarizing texts relevant to cryptography in the late medieval and early modern periods. The readership cannot have been large.

At the opposite extreme of readability was the 1939 *Secret and Urgent: The Story of Codes and Ciphers* by the journalist and naval affairs commentator Fletcher Pratt. The book presented a breezy series of thrilling anecdotal historical episodes involving ciphers and code breaking exploits. Each episode came complete with Sunday supplement-style character sketches and just the right amount of technical information about the cipher or cryptanalysis in question. The technical discussions were not always tightly bound to the factual historical setting: although they were always illustrative of the type of ciphers involved in this or that historical episode, they were not necessarily verbatim transcripts of documents in archives. Pratt thus managed to make the technicalities — always clearly explained — seem important and managed to teach a bit of history in the nonrigorous way a historical movie or novel might teach a bit of history. Like many others, I was inspired by this book when I read it in my early teens. It was only much later that I came to realize that its lack of bibliography and detailed footnotes made it useless as a serious history of cryptography.

In 1967, about thirty years after Pratt's book, a much more serious book appeared, *The Codebreakers*, by David Kahn, also a journalist, but one with a far sterner approach to standards of documentation. Where Pratt had two pages of notes and no literature references, Kahn gave 163 pages. Kahn's method (which he pursued over many years with great energy) seems to have been simply this: to read everything about cryptography in all the world's libraries and archives, to interview all cryptographers, and to write it all down as sensibly, as accurately, and with as much detail as possible; his book has 1,180 pages. This is the book I read when I was in college. By then I had grown up enough to appreciate Kahn's comment in his preface that although his love for cryptography had also been sparked by Pratt's book, he was disappointed in the book. Kahn bemoaned Pratt's "errors and omissions, his false generalizations based on no evidence, and his unfortunate predilection for inventing facts."

Unfortunately Kahn's book was published a short time before the facts about the Polish and British success in breaking the German Enigma cipher of World War II became publicly known and also a short while before the amazing invention of the new number-theoretical "public key cryptography" techniques now pervasive in computers and the Internet. As a result, these interesting and important topics received no treatment.

Now, 30 years after Kahn's book, a new history of cryptography has appeared, again by a journalist: Simon Singh's *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots to Quantum Cryptography*, a best seller in England in its first months of publication. Singh states in his preface that "In writing *The Code Book*, I have had two main objectives. The first is to chart the evolution of codes . . . the book's Second objective is to demonstrate how the subject is more relevant today than ever before."

Singh's first five chapters cover the history of cryptography up through the end of the second World War, summarizing material found in earlier books and journal articles, presented by the episodic snapshot method. His remaining three chapters are based mostly on personal interviews with leading participants. Chapter 6 describes the invention and early development of public key cryptography by W. Diffie, M. Hellman, R. Merkle, R. Rivest, A. Shamir, and L. Adleman in the US, and independently but in secret, by J. Ellis, C. Cocks, and M. Williamson in the UK. Chapter 7 describes the current controversy about the proper role of cryptography in a free society: personal freedom versus the interests of the state, privacy versus wiretapping, key escrow, export of strong cryptography, and so on. The final chapter describes quantum cryptography, the new system of communications made untappable by exploiting the fact that the polarization of a photon is altered when it is measured.

The good news is that Singh's book has all the good qualities of Pratt's. Unfortunately Kahn's criticism of Pratt's book also applies to Singh's book. Almost every page has small errors of fact. In many places it is clear that Singh does not really understand the material he copies from his sources. Many of these errors are of little consequence when taken individually, but their cumulative effect is to destroy a knowledgeable reader's confidence in the author's standards of accuracy.

Here are just a few examples:

- On page 128 Singh describes the wired code wheels of the Enigma cipher machine (the "rotors," which he oddly calls "scramblers"): "The scrambler, a thick rubber disc riddled with wires ..." But the Enigma's rotors were *not* made of rubber but of aluminum, brass, and Bakelite. Singh may have misunderstood a sentence on page 411 of Kahn's book: "The body of a rotor consists of a thick disk of insulating material, such as Bakelite or hard rubber ..." accurately describing the rotors, not of an Enigma machine but of a different cipher machine.

- On page 168 Singh states that A. M. Turing (in his 1937 paper "On computable numbers, with an application to the Entscheidungsproblem") called "this hypothetical device a *universal Turing machine* [Singh's italics]." But of course the terms "Turing machine" and "universal Turing machine" were *not* used by Turing himself; a glance at his paper shows he used "computing machines" and "universal machines".

- On pp. 187–88, Singh states that the British WWII codebreaking organization, the "Government Code and Cypher School," was disbanded after the war, and then replaced by another, the "Government Communications Headquarters," or GCHQ. In fact the change occurred in 1942, and was one in name only.

- On page 191 Singh claims the American breaking of the Japanese "Purple" cipher enabled the naval victory at Midway and the assassination of Admiral Yamamoto. In fact these were due to the breaking of the "JN-25" code. "Purple" was a machine cipher, roughly equivalent to the German Enigma, "JN-25" was a hand system relying on code books and random number tables.

Singh's unfamiliarity with the technical vocabulary used by his sources seems to have led him into a more serious mistake in the first two chapters. To explain this, I must first summarize material in Kahn's chapters 3 to 6.

From before 1400 until about 1750 only one kind of encryption method was widely used (although others were discussed in books). This method used what were called at the time "ciphers" or "keys". A cipher was a collection of arbitrary symbols or numbers that substituted for letters, syllables (or other letter sequences), names, and common words and phrases found the in plain text. By 1700 ciphers with as many as 1,000 or 2,000 substitutions were common, and even larger ones with as many as 10,000 or 50,000 substitutions were in use later on. Although the general trend was towards greater size and complexity, throughout this period ciphers of widely varying size and complexity were used. Modern scholars have used a variety of terms — more or less interchangeably — for this cryptographic genre, including "homophonic cipher", "nomenclator", "code", "code chart", and so on, as the original terms "cipher" and "key" are no longer precise enough to distinguish these methods from more modern ones.

At the same time a theory for another kind of cryptography was being developed, discussed, and elaborated in successive printed cryptography books all through the 1500s and into the 1600s. The set-piece example of this new kind of cryptography, the "Vigenère" cipher, a.k.a. *chiffre indéchiffrable*, was more algebraic in nature, based on Latin squares and what we now know as modular arithmetic. This kind of cryptography was slow to gain acceptance: although available for use in 1575 it was not actually used until the mid 1600s, and then only sparingly. Even at the end of the 1700s Thomas Jefferson's adoption of the Vigenère cipher by the U.S. State Department was an innovation, and when he left office the department reverted to the older nomenclator technology. Only in the nineteenth century did the Vigenère cipher come into common use and serve as a basis for further technical developments.

Singh, however, seeing one author use the term "nomenclator" to describe a cipher in use in 1586, and another author using the term "homophonic cipher" to describe one in use in 1700, supposes the two ciphers to be different kinds of things. And he invents a theory explaining why the latter kind was devised: he says on page 52 that the "homophonic cipher" was invented in Louis XIV's reign to serve as a more practical alternative to the *chiffre indéchiffrable.* But Kahn (whose book appears in Singh's list of references), on page 107, shows an example of a homophonic cipher, labeled as such, from 1401, about three centuries before Singh's invented invention.

A different kind of misunderstanding occurs in the discussion of the attack on the German Enigma machine in the early 1930s. The mathematical basis for the initial Polish success was the well known fact that the cycle type of a permutation is invariant under conjugation: when one writes the permutations τ and $\sigma\tau\sigma^{-1}$ as the products of disjoint cycles, the same lengths appear with the same multiplicities. On pages 148–154 Singh explains very clearly how Marian Rejewski applied this fact to the problem of recovering German Enigma keys. If ever there was a real–world story problem handed to mathematics teachers on a silver platter, this would be it.

The sample permutation Singh uses to illustrate the Enigma application decomposes into cycles of length 3, 9, 7, and 7. (Here of course the permutation is a permutation of the 26 letter alphabet: $3+9+7+7=26$.) But here is the kicker. The permutations τ which actually occur in the Enigma application are of the form $\tau = \alpha\beta$, where α and β are each the products of 13 disjoint 2-cycles. This forces τ to have even cycle length multiplicities, which Singh's example does not have. That is, Singh presents an imitation example, not an example of an actual Enigma τ permutation he has worked out. This is perfectly adequate for illustrating the mathematical fact of the invariance of cycle type under conjugation, but will not do for illustrating the historical facts of Rejewski's solution of the Enigma cipher.

This is as if a historian of trigonometry, describing some early work, wrote: "given a right triangle with sides of lengths 2, 3 and 4, the angle opposite the side of length 2 was found by taking the inverse sine of the ratio of the opposite side to the hypotenuse, in this case $\arcsin(2/4) = 30°$."

The formula is correctly stated and worked out, but applied in an impossible context. Which is the worse fault: Singh not bothering to use an actual historical — or even realistic — example, or not knowing that his example is unrealistic?

Singh does better in the remaining chapters, where the story line and technical explanations derive from interviews. His interviewees' personalities are clearly visible, and the technical explanations are usually comprehensible.[2]

Chapter 6, about the invention of public key cryptography, repeats the stories which have been told in public lectures by Diffie, Hellman, and Shamir about their discovery of the basic ideas of one way functions and public key cryptography, as well as their discovery of the number-theoretic examples based on modular exponentiation and the difficulty of factoring. More interesting is Singh's description of the secret and somewhat earlier independent discovery of these ideas by Ellis, Cocks, and Williamson at the GCHQ, the secret British government cryptography organization. GCHQ has recently "gone public" in this matter, making Cocks a media celebrity by GCHQ standards. (The chronology of this matter is somewhat hard to assess because not all the relevant GCHQ files have been made available. One result, the Diffie–Hellman exponential key exchange, seems *not* to have been first discovered by GCHQ.)

In this chapter Singh spends many pages discussing the matter of priority of scientific discovery, exulting in the recent declassification of the earlier GCHQ work as if an injustice had been righted. This vision of the abstract reward of "credit", based on strict chronological priority distracts Singh from looking at the historically more interesting questions of influence of ideas. These include: how were the initial GCHQ discoveries understood by the discoverers' colleagues at the time, how were these ideas developed, and how were they used? The available evidence is scanty, but it seems likely that they were regarded within GCHQ as impractical curiosities and ignored until the rediscoveries on the outside alerted GCHQ to their importance.

The historiographic issue is neatly illustrated in an example at the end of the chapter, referring back to an episode in Chapter 2, which Singh takes as a parallel foreshadowing. One of the techniques for breaking the Vigenère *chiffre indéchiffrable* was first published in 1863 by F. Kasiski, but apparently some time in the 1850s Charles Babbage had worked out the same method in private. Singh claims (on no evidence whatsoever) that Babbage did not publish his results because of the interests of military secrecy during the Crimean War of 1854. But now Babbage's injustice is also righted: he gets the credit in the end. Regardless of the reasons for Babbage's failure to publish, the following seems clear: Babbage's discovery, since it was unpublished, had no influence on the further development of cryptography. That he made this discovery tells us something about Babbage's mental capabilities; that it was independently rediscovered tells us something (but not much) about its intrinsic level of difficulty. Babbage might have been first but (in this matter) he was historically unimportant. Society uses credit and priority as a reward to encourage the dissemination of new ideas, and it is not at all clear that a researcher who fails to publish a new idea — whether out of diffidence, patriotism, or employment at a secret research laboratory — is done an injustice when not awarded credit. Righting such imagined wrongs is not what history is about.

Chapter 7, based on interviews with the PGP (Pretty Good Privacy) programmer, Phillip R. Zimmermann, concentrates on the currently unsettled matter of the proper role of cryptography in a free society. Zimmermann represents the libertarian side: the people should use — must use, if

[2]Whitfield Diffie, however, has complained in a book review (*Times Higher Education Supplement,* 10 Sept. 1999) that not everything he told Singh was accurately reported.

they don't trust their government — the strongest kind of cryptography they can. Governments, however, remembering the invaluable results of cryptanalysis during the Second World War (and presumably since then) would wish to somehow keep the strongest forms of cryptography out of the hands of potential enemies. As the target of a grand jury investigation, Zimmermann suffered from the American government's embarrassingly inept way of trying to make up its mind on this public policy issue.

The final chapter returns to the purely technological, with a discussion of quantum cryptography. Here again, interviewees (D. Deutsch and C. Bennett) carry the story along. The description of the basics of quantum mechanics is painfully incoherent, that of quantum computing is superficial and vague, but the explanation of how polarized photons can carry untappable information is fairly clear.

In the preface — justifying his rejection of a pedantically more accurate title for his book — Singh states "I have, however, forsaken accuracy for snappiness." With hindsight this is ominous. His carelessness with facts will not harm those readers who pick up the book, skim it, and find the subject not to their taste. Nor will it harm the enthusiasts (like myself), who will seek out other, more reliable books.[3] But most, I suspect, will fall in the middle ground: interested readers who will rely on this book alone for their information about cryptography. This group, which will mine Singh's book for years, if not decades, for term paper and lecture material, and possibly material for other books, will be disserved by the author's lax standards of accuracy.

DNA Based Computers V
Editors: Eric Winfree and David K. Gifford
Publisher: American Mathematics Society
Series: DIMACS Series in Discrete Mathematics and TCS Volume 54
ISBN: 0-8218-2053-2, Hardcover, 249 pages
Review by Mitsunori Ogihara,Dept of Computer Science, Univ. of Rochester

1 Overview

DNA based computation (or molecular computation) is a new field that integrates computer science and molecular biology. The field studies the use of biological molecules (such as DNA and RNA) for conducting massively parallel computation. In a seminal paper published in 1994, Adleman [Adl94] developed a method for solving the Hamilton Path Problem in liquid-phase DNA. Adleman confirmed his idea with a successful biological experiment. Lipton [Lip95] generalized Adleman's idea and developed a DNA-based method for solving 3SAT. These papers raised the hope that molecular computers could solve NP problems much more efficiently than silicon-based computers. Unfortunately, in both the Adleman method and the Lipton method the total amount of DNA required for computation grows exponentially with the input size. Since the Avogadro number (6.02×10^{23}) essentially limits the number of molecules that can be put in a test tube, the

[3]My favorites: instead of Singh's Chapters 1–3, people should read D. Kahn, *The Codebreakers: The Story of Secret Writing* (Macmillan, 1967) and F. Bauer, *Decrypted Secrets : Methods and Maxims of Cryptology* (Springer, 1997). Instead of Singh's Chapter 4, read F. H. Hinsley and A. Stripp, *Codebreakers: The Inside Story of Bletchley Park* (Oxford, 1993), and G. Welchman, *The Hut Six Story* (McGraw-Hill, 1982). Instead of Chapter 7, read W. Diffie and S. Landau, *Privacy on the Line: The Politics of Wiretapping and Encryption* (MIT, 1998). All but one of these books are in Singh's "Further Reading" list, pages 388–393.

size of the largest instance that could be solved using their methods is in the range of 60 to 70. One then wonders whether there is a killer application of molecular computation. The question has been the driving force of the field.

Although there have been a lot of new ideas fused into the field, there is no accessible book that covers the whole spectrum of interest, ranging from molecular biology to computer science. This book does not serve the purpose. Rather, it offers a compendium of articles representing the current status of the field. All the nineteen papers included in the book were presented at the DIMACS Workshop on DNA Based Computers at MIT in June 1999.

2 Contents

The subjects of the articles in the book can be broadly classified into four classes: development of new molecular computation models (six papers), design and analysis of molecular algorithms and methods (ten papers), development of tools for supporting molecular algorithm design (three papers), and biological experiments for testing feasibility of molecular algorithms and methods (six papers). Of course some articles cover multiple subjects. All the papers in the book are of good quality, but there are a few that I think are worth mentioning.

The opener of the book (Faulhammer *et al.*, pages 1–8) shows an excellent proof-of-concept of RNA-based computation, the model proposed by the same set of authors one year earlier [CFLL98]. Ostensibly RNA-based computation is a straightforward analogue of DNA-based computation because all the enzymatic operations used for DNA-based computation are available for RNA as well. The earlier paper observes that strong resistance of RNA strands to cleavage, which results from formation of secondary structure, can be a big obstacle to execution of reliable RNA-based computation. In the paper contained in this volume, the authors report a carefully crafted RNA-based algorithm for solving the Knight problem—the problem of finding all configuration for putting knights on an $n \times n$ Chess board so that no knight is attacking any other knight. In this algorithm each possible solution (viewed an n^2 bit string) is synthesized as an RNA-strand and then all incorrect (i.e., having a knight attacking another) solutions are eliminated via cleavage and length-specific separation. The algorithm was tested on the 3×3 board, where all incorrect solutions were destroyed and all but one of a total of 43 correct solutions were recovered. This is a remarkable achievement not only because the level of accuracy that was achieved is very high but also because the solution space of nine bits is the largest one that has been biologically tested in molecular computation.

A 1995 paper by Reif [Rei95] suggested the use of a compartmentalized architecture for massively parallel molecular computation. Basically, each compartment keeps a solution of molecules without leakage, and, from time to time, new sequences are added to compartments and strands are transferred from one compartment to another. Kurtz *et al.* [KMRS99] suggested realization of such an architecture using membranes. Following up on it, the paper by Bloom and Bancroft (pages 37–48) proposes the use of liposomes for building compartments. A liposome is a phospholipid membrane bilayer (i.e., two thin layers of fat that are placed on top of each other) that is folded into a spherical structure with an aqueous cavity. The use of liposomes has many advantages. They are cheap and easy to produce. They have a long lifetime and are thermostable. Furthermore, there is a set of operations for manipulating them: (i) one can fuse liposomes by controlling temperature, pH, and ion concentration; (ii) certain proteins attached on liposomes can direct fusion; (iii) certain viruses can insert DNA strands into liposomes; and (iv) detergents can solubilize membrane molecules. The paper shows how these operations can be implemented and how the liposome mediated molecular computation can improve efficiency of previously proposed molecular algorithms (such as DNA hairpin [SGK$^+$00], two-dimensional DNA crystals [WLWS98], and DNA-based Boolean logic gates [OR99]).

A big problem to advancement in the field is that large-scale experiments cost a large amount of money and time. In order to study applicability of molecular algorithms it is crucial that their scalability is tested, but it may not be possible to do so due to resource constraints. One naturally thinks of the use of software simulation of large-scale biochemical reactions. The book contains two pieces of work on the topic (Garzon *et al.*, pages 91–100, and Hartemink *et al.*, pages 111–121). These two papers present frameworks for such simulation with a high level of biological details incorporated in the design. If fully developed, their software will become a powerful tool for all molecular computation researchers.

The closing article of the book (Gehani *et al.*, pages 233–249) discusses DNA-based cryptosystems. A few years earlier in this workshop, Boneh *et al.* presented a DNA-based method for attacking DES [BDL96]. The purpose of the present paper is the opposite—to design a secure cryptosystem using DNA. In the codebook-based cryptography a plaintext is transformed to a cipher-text by dividing it into blocks of words and then converting each individual block to a codeword based upon a given codebook. The plaintext is then recovered from the cipher-text by applying the reverse transformation to each block. The paper shows that it is possible to design such a system using a modern technology for extending and cutting thousands of DNA strands that are mobilized on a surface. It is unclear the type of DNA-based cryptography proposed here will be superior to the current state-of-the-art silicon-based cryptography, but the idea is very amusing.

3 Conclusion

Overall this volume offers very enjoyable reading for someone who is interested in this topic and has a certain level of background in biology. As the editors conclude in the foreword, the future of the field looks bright.

References

[Adl94] L. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, 1994.

[BDL96] D. Boneh, C. Dunworth, and R. Lipton. Breaking DES using a molecular computer. In R. Lipton and E. Baum, editors, *DNA Based Computers*, pages 37–65. The AMS DIMACS Series in Discrete Mathematics and TCS Volume **27**, 1996.

[CFLL98] A. Cukras, D. Faulhammer, R. Lipton, and L. Landweber. Chess game: a model for RNA-based computation. In *Preliminary Proceedings of 4th DIMACS Workshop on DNA Based Computers*, pages 27–37, 1998.

[KMRS99] S. Kurtz, S. Mahaney, J. Royer, and J. Simon. Active transport in biological computing. In L. Landweber and E. Baum, editors, *DNA Based Computers II*, pages 171–180. The AMS-DIMACS Series in Discrete Mathematics and TCS Volume **44**, 1999.

[Lip95] R. Lipton. DNA solutions of hard computational problems. *Science*, 268:542–545, 1995.

[OR99] M. Ogihara and A. Ray. Simulating boolean circuits on DNA computers. *Algorithmica*, 25:239–250, 1999.

[Rei95] J. Reif. Parallel molecular computation. In *Proceedings of 7th ACM Symposium on Parallel Algorithms and Architecture*, pages 213–223. ACM Press, 1995.

[SGK$^+$00] J. Sakamoto, H. Gouzu, K. Komiya, D. Kiga, S. Yokoyama, T. Yokomori, and M. Hagiya. Molecular computation by DNA hairpin formation. *Science*, 288:1223–1226, 2000.

[WLWS98] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–544, 1998.

Review of
Complexity and Real Computation[4]
Authors: Blum, Cucker, Shub, and Smale
Published by Springer-Verlag
\$34.95 on amazon.com, but see bestbuys.com
Hardcover, 432 pages
ISBN 0387982817

Reviewed by Timothy H. McNicholl[5]

In [3], Penrose posed the question "Is the Mandelbrot set decidable?" That is, is there an algorithm that given a point as input will determine if that point is in the Mandelbrot set. Part of the problem with this question is figuring out its proper interpretation. The traditional theory of computability deals only with computations over the natural numbers. Computations over the integers and rationals can be modeled as computations over the natural numbers. But since the reals are uncountable, Penrose's question is not amenable to this approach. One could answer the question only for points with rational coordinates. However, this approach does not work for related questions. For example, the curve $x^3 + y^3 = 1$ has no rational points. Yet, the question of its decidability can not be dismissed as meaningless or trivial. A traditional approach would be to use the field of computable real numbers (see, for example, [4]). That is, the set of real numbers whose decimal expansion can be listed by some algorithm. However, there is a severe flaw here due to the fact that there is no algorithm which can determine if two (indices of) computable real numbers are identical. Thus, the unit circle, which certainly ought to be considered computable, turns out to be non-computable.

These considerations lead the authors of this book to formulate their theory of computability over the reals and more generally over any ordered ring. The basic idea is simple. Take flowchart computability (see, for example, section I.5 of [2]) and allow the computation boxes to use polynomials with real coefficients. The decision boxes test conditions of the form '$x \geq 0$'. One can then define a set of reals (or a set of n-tuples of reals) to be *computably enumerable* (*c.e.*)[6]

if it is the halting set for some 'machine' over $\mathbf{R}$. As in the classical theory, it turns out that a set is computable if and only if it is *c.e.* and *co-c.e.*. It is shown that a *c.e.* subset of $\mathbf{R}^n$ is a countable union of *semi-algebraic* sets. These sets (semi-algebraic) are solution sets of finite conjunctions of inequalities. Hence, any *c.e.* subset or of $\mathbf{R}^n$ has countably many connected components. However, the Mandelbrot set is uncountable and totally disconnected and so is not even *c.e.*!

Traditional computability theory developed along two lines. The first might be called the *classical* theory which deals with such topics as Turing reducibility and the lattice of computably enumerable sets under inclusion. What characterizes this branch of the discipline is an almost complete lack of concern for issues dealing with number of steps performed by an algorithm or the amount of memory it uses. The other branch of the theory, *complexity*, deals almost solely with these issues and eschews non-computable sets. As the title suggests, this book pursues mostly the latter direction rather than the former. Many of the results are 'liftings' of results from traditional complexity theory to computability over the reals. For example the class $P_{\mathbf{R}}$ of problems over $\mathbf{R}$ decidable in polynomial time is characterized by fixed point logic and $NP_{\mathbf{R}}$ is characterized using second-order logic (*c.f.* [1]). However, the authors are not merely knocking down straw men.

[4]Timothy H. McNicholl©2001
[5]Dept of Mathematics, University of Dallas, Irving, TX 75062, tmcnicho@acad.udallas.edu
[6]Some readers may still use the old terminology "r.e." for "recursively enumerable".

Since the underlying sets are uncountable, the techniques of traditional complexity theory do not apply. The techniques used are based primarily on analysis and topology, although some model theory is used. In addition, there are some new results. For example, it is shown that the Hilbert Nullstellensatz over any *field* is NP-complete.

The book is well written and uses examples liberally. It is generally speaking, self-contained. In particular, no knowledge of complexity theory or traditional computability theory is necessary. However, knowledge of basic analysis and topology is necessary.

One of the wonderful things about this new field is that it is so new and while much is known, much is not known. The are ample opportunities for new research. The following investigations come to mind.

1. Explore the structure of the Turing degrees over $\mathbf{R}$. Is there a Turing-complete *c.e.* set? Is there a Turing-intermediate *c.e.* set? The usual methods for answering these sort of questions are strongly based on mathematical induction and so do not apply.

2. Continuing with the above line of inquiry, does every 'sufficiently low' Turing degree contain a Julia set?

3. Computable analysis developed in the context of the field of computable real numbers. Perhaps this investigation should be re-opened along the lines of this new work. For example, is there an algorithm for finding the maxima and minima of a continuous computable function over a closed interval?

References

[1] N. Immerman, Descriptive Complexity. Springer-Verlag, 1998.

[2] P.G. Odifreddi, Classical Recursion Theory, Volume 1. Elsevier, 1992.

[3] R. Penrose, The Emperor's New Mind. Penguin, 1991.

[4] H.G. Rice, Recursive real numbers. *Proc. Am. Math. Soc.* 5 (1954) 784-790.

[5] R. Soare, Computably Enumerable Sets and Degrees. Springer-Verlag, 1987.

Review of[7]
Basic Proof Theory (second edition)
Book by
A. S. Troelstra and H. Schwichtenberg
Published by Cambridge University Press
Paperback, 430 pages
Review by Jeremy Avigad (avigad@cmu.edu)

[7]Jeremy Avigad, ©2001

1 Overview

Beweistheorie, or “Proof Theory,” was the phrase that David Hilbert used to describe the program by which he hoped to secure the foundations of mathematics. Set forth in the early 1920’s, his plan was to represent mathematical reasoning by formal deductive systems, and show, using safe, “finitary,” methods, that such reasoning could never lead to contradiction. This particular goal was shown by Gödel to be infeasible. But the more general goal of using formal methods to explore various aspects of mathematical provability, including the relationship between classical and constructive methods in mathematics and the strengths and limitations of various axiomatic frameworks, has proved to be remarkably robust. Today, these goals represent the traditional, *metamathematical* branch of proof theory.

Since Hilbert’s time, the subject has expanded in two important respects. First, it has moved well beyond the study of specifically mathematical reasoning. Proof theorists now consider a wide range of deductive systems, designed to model diverse aspects of logical inference; for example, systems of modal logic can be used to model reasoning about possible states of affairs, knowledge, or time, and linear logic provides a means of reasoning about computational resources.

The second change is that now more attention is paid to specific features of the deductive systems themselves. For the Hilbert school, deductive calculi were used primarily as a means of exploring the deductive consequences of various axiomatic theories; from this point of view, the particular choice of a calculus is largely a matter of convenience. In much of modern proof theory, however, deductive systems have become objects of study in their own right. For example, in the branch of proof theory known as *proof complexity*, one aims to determine the efficiency of various systems with respect to various measures proof length, just as the field of computational complexity explores issues of computational efficiency with respect to various measures of complexity.

In the field of *structural proof theory*, the focus is on the particular syntactic representations, axioms, and rules. A central theme is the exploration of transformations, or “reductions,” that preserve the validity of a proof. One of the subject’s typical goals is to show that proofs in a given calculus can be brought (e.g. via cut-elimination or normalization) into particularly nice canonical forms. There are various reasons for doing so: for example, from a proof in canonical form, it is often possible to extract additional information; and knowing that every proof has a canonical form means that in searching for proofs it is sufficient to search for a canonical representative.

This emphasis on syntax makes structural proof theory a black sheep in the mathematical logic community. Learning to appreciate the beauty of the subject takes some effort, but one is, in the end, rewarded by elegant combinatorial symmetries and theorems that are often striking or unexpected. Moreover, the subject is closely linked to computational applications. The fields of artificial intelligence, hardware and software verification, logic programming, and automated deduction are strongly dependent on logical inference, and any computational task that involves searching for and manipulating proofs in a serious way requires a solid understanding of deductive systems and their properties. There are also substantial interactions with the theory of programming languages. Implicit in the early development of intuitionistic logic, in the hands of Brouwer, Heyting, and Kolmogorov, is the notion that an intuitionistic proof *is* a program, returning a type of evidence that is specified by the proof’s conclusion. A formal analysis of the correspondence between proofs and programs gives rise to constructive type theory, which is of foundational importance to the design of functional programming languages. Given the inevitable tensions between pure and applied branches of any discipline, structural proof theory is a remarkably happy marriage of theory and practice.

Basic Proof Theory is a thorough introduction to structural proof theory. It offers a unified and comprehensive account of the core fundamentals of the subject, and, in doing so, it fills a major expository gap in the literature. The book also presents good deal of additional information in a clear and organized manner, with historical notes and references at the end of each chapter. It serves admirably as a standard reference for the subject, and is likely to maintain that role for a number of years.

2 Contents

The book's eleven chapters can be divided neatly into two parts. The first six chapters provide a comprehensive overview of the subject's essentials. The wealth of information in these chapters may overwhelm the novice, but the exposition is crisp and clear, and those making it through these chapters are thereby licensed to proof-theorize. Topics treated include all of the following (not necessarily in the order indicated):

- the three basic versions of propositional and first-order logic, i.e. classical logic, intuitionistic logic, and minimal logic; various treatments of equality; logic with partial (possibly non-denoting) terms; and the introduction of defined function symbols;
- a thorough treatment of various deductive calculi, including axiomatic calculi, natural deduction, and sequent calculi, with numerous variations, and translations between the various formalisms;
- various double-negation translations, which serve to interpret classical logic in intuitionistic or minimal logic;
- cut-elimination, with variations and extensions, and both upper bounds and lower bounds on the increase in proof length;
- applications of the cut-elimination theorem: Herbrand's theorem, the explicit definability and disjunction properties, interpolation theorems, and conservation results;
- proof terms and the Curry-Howard isomorphism; strong normalization and the Church-Rosser property for the simply typed lambda calculus; weak normalization for full first-order logic; and applications of normalization.

The remaining five chapters make up what I am calling the second part, and survey a number of related topics. In each case, the goal is not to provide a comprehensive introduction, but, rather, to convey the flavor of the subject, and indicate some of the interactions with the core ideas and methods of structural proof theory. Topics treated include the following:

- Resolution: the focus is on linear resolution, which is important to logic programming, but there is also a discussion of the more general form of resolution, and the relationships to other deductive calculi;
- Categorical logic: deduction graphs, the relationship between cartesian-closed categories and the simply typed lambda calculus, and some coherence theorems;

- Modal logic: the focus is on propositional S4, with a cut-elimination theorem, and embeddings of intuitionistic logic;
- Linear logic: cut-elimination, embeddings of classical and intuitionistic logic, and proof nets;
- The ordinal analysis of arithmetic;
- Second-order logic, including the second-order lambda calculus, and Girard's proof of strong normalization.

3 Opinions

The preface indicates that the book is intended as a first step beyond standard introductions to logic, providing adequate preparation for understanding contemporary literature and more advanced monographs on the subject. As such, it will appeal to both computer scientists and mathematical logicians, and graduate students in these disciplines (as well as, perhaps, some very advanced undergraduates). But the text is by no means light reading, and calls for a fair amount of mathematical maturity, although no specific mathematical background is required. The proofs are compact, and the motivation behind some of the definitions and lemmata must often be inferred from their ultimate use. Routine syntactic details are frequently left to the reader.

The treatment of the basics of structural proof theory in the first six chapters is quite detailed. For example, both two-sided and one-sided sequent calculi are considered, using either sets or multisets as sequents, and with many variations of the rules of inference. This encyclopedic treatment is sometimes at odds with the pedagogical goals of an introductory text: someone coming to the subject from scratch may well prefer to avoid this breadth and focus on one or two representative variations. In that respect, the book calls for selective reading. The advantage to having the additional information at hand is that the text can grow with the reader's understanding. Furthermore, it allows the authors to include a number of fundamentally important results that are not found in other expository texts, because they are either too new, or folklore, or buried in the technical literature.

If the first part of the book runs the risk of including too much information, the second part runs the risk of treating its topics too lightly. For example, most of the material in the chapter on categorical logic has been around since the 1970's, and the overall treatment does not do justice to the importance of categorical methods in the study of intuitionistic first- and higher-order logic, or in the study of simple and dependent type theory. Similarly, though linear logic plays an important role in the analysis of concurrency, resource-bounded computation, and games, the book's brief presentation is largely unmotivated, and the applications to the analysis of classical and intuitionistic logic given as examples are, by themselves, not very satisfying. Of course, everyone will have his or her own quibbles with other aspects of the presentation. For example, I would have chosen to present the more general form of resolution first, with linear resolution as a special case; I found the motivational introduction in Section 7.1 a little confusing. Those familiar with linear logic may be put off by the authors' nonstandard choice of symbols. And the book's presentation of the ordinal analysis of arithmetic uses an infinitary extension of natural deduction, which is technically less smooth than presentations using an infinitary sequent calculus (such as the second author's article in the *Handbook of Mathematical Logic*).

Taken in isolation, the text does not provide a comprehensive introduction to any of the subjects covered in the last five chapters. But it is unfair to criticize the book for failing to provide an adequate introduction to seven subjects, when it only aims to provide an introduction to one. The survey offers a good starting point for exploring the additional topics from a proof-theoretic point of view, and, for those who come to this book with prior or independent interest in these topics, it offers a good sense of what proof-theoretic analysis can do.

In the preface, the authors make it clear that the two related branches of proof theory, i.e. proof complexity and metamathematical proof theory, are not the focus of attention. They were wise to limit the scope in this way, and *Basic Proof Theory* nicely complements books devoted to these other subjects. Of course, there is a good deal of overlap between the various branches of proof theory, and the book is attentive to related issues. For example, it provides exponential lower bounds on cut-free propositional provability and hyperexponential lower bounds in the first-order case, which are fundamental results in proof complexity, due to Statman and Orevkov; and it presents Gentzen's results on the ordinal analysis of arithmetic, which is equally fundamental to the metamathematical branch of the subject.

In the second edition, substantial additions and revisions have been made, and a number of errors have been corrected. Monographs of this sort are often obnoxiously overpriced, so the authors and publisher are to be commended for keeping this volume pleasantly affordable ($34.95 in the US). I hope that the shortcomings discussed in this section do not obscure the bottom line: this book is a welcome and important contribution to the expository literature on structural proof theory, and anyone whose professional or intellectual interests involve formal deductive systems in any way will want to keep a copy close at hand.