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. **Symbolic Asymptotics** by John R. Shackell. Review by James C. Beaumont. Given two functions $f(x)$ and $g(x)$, how to you compare their growth as x goes to infinity? This is a deep question that has its roots in analysis but is often applied in analysis of algorithms. This is an advanced, well written book on this question.

2. **Complexity and Cryptography. An Introduction** by John Talbot and Dominic Welsh. Review by Jörg Rothe. Complexity theory deals with functions or sets that are hard to compute. Cryptography is a balancing act- they want functions that are easy for Alice and Bob but hard for Eve. This book brings these two intertwined fields together.

3. **Complexity Theory and Cryptology: An Introduction to Cryptocomplexity** by Jörg Rothe. Review by Piotr Faliszewski. This covers the same ground as the above book. (Note: the author of *Complexity Theory and Cryptography: An Introduction to Cryptocomplexity* is indeed the reviewer of *Complexity and Cryptography: An Introduction*. This is not a typo.)

4. **Three Blogs by Theorists**. Three Blogs by theorists are reviewed. The blogs are by Lance Fortnow, Scott Aaronson, and Luca Trevisan. The review is by Bill Gasarch.

Books I want Reviewed

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

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

Books on Algorithms and Data Structures

1. *Combinatorial Optimization: Packing and Covering* by Cornuejols.

2. *Distributes Systems: An Algorithmic Approach* by Ghosh.

3. *An Introduction to Data Structures and Algorithms* by Storer.

4. *Nonlinear Integer Programming* by Li and Sun.

5. *Biologically Inspired Algorithms for Financial Modelling* Brabazon and O'Neill.

6. *Planning Algorithms* LaValle.

7. *Prime Numbers: A Computational Perspective* by Crandall and Pomerance.

Books on Cryptography

1. *Concurrent Zero-Knowledge* by Alon Rosen.
2. *An Introduction to Cryptography (second edition)* by Richard Mollin.
3. *Algebraic Aspects of the Advanced Encryption Standard* Cid, Murphy, Robshaw.
4. *Cryptographic Applications of Analytic Number Theory: Complexity Lower Bounds and Pseudorandomness* by Shparlinski.
5. *Cryptography and Computational Number Theory* edited by Lam, Shparlinski, Wang, Xing.
6. *Coding, Cryptography, and Combinatorics* edited by Feng, Niederreiter, Xing.

Books on Security

1. *Computer Viruses and Malware* by Aycock.
2. *Network Security Policies and Procedures* by David Frye.
3. *Data Warehousing and Data Mining Techniques for Cyper Security* by Anoop Singhal.
4. *Electronic Postal Systems: Technology, Security, Economics* by Gerrit Bleumer
5. *Cryptographic Algorithms on Reconfigurable Hardware* by Rodriguez-Henriques, Saqib, Perez, Koc.
6. *Preserving Privacy in On-Line Analytical Processing (OLAP)* by Wang, Jajodia, Wijesekera.
7. *Secure Data Management in Decentralized Systems* Edited by Ting Yu and Sushil Jajodia.
8.

Books on Coding Theory

1. *Coding for Data and Computer Communications* by David Salomon.
2. *Block Error-Correcting Codes: A Computational Primer* by Xambo-Descamps.
3. *Algebraic Coding Theory and Information Theory: DIMACS Workshop* Edited by Ashikhmin and Barg.

Combinatorics Books

1. *Graphs and Discovery: A DIMACS Workshop* Edited by Fajilowicz, Fowler, Hansen, Janowitz, Roberts. (About using software to find conjectures in graph theory.)
2. *Combinatorial Designs: Constructions and Analysis* by Stinson.
3. *Computationally Oriented Matroids* by Bokowski

Logic and Verification Books

1. *Elements of Finite Model Theory* by Libkin.
2. *Mathematical Approaches to Software Quality* by O'Regan.
3. *Software Abstractions: Logic, Language, and Analysis* by Jackson.
4. *Formal Models of Communicating Systems: Languages, Automata, and Monadic Second-Order Logic* by Benedikt Bollig.

Misc Books

1. *Automata Theory with Modern Applications* by James Anderson.
2. *An Introduction to Difference Equations* by Elaydi.

Review[2]

Symbolic Asymptotics
Author of book: John R. Shackell
Publisher: Springer Verlag
243 Pages, $79.95
Author of review: James C. Beaumont

1 Introduction

Everyone who has taken a first year University course of analysis is familiar with the notion of how fast functions of a real variable x 'grow' as we let x tend to infinity. For example, one recalls that, 'the exponential function grows faster than any fixed, power of x'. Loosely speaking, asymptotics can be viewed as the study of questions such as, given two functions of a real variable f and g, how does one measure the speed of growth of each of these functions, and then, how does one measure these two quantities relative to one another? Beyond undergraduate mathematics, such questions have long since been important in the physical sciences, where the variable in question is thought of as time, and one is interested in how a given system will change as a function of it; often for arbitrarily far periods in the future— as in the study of radioactive decay for example. It is therefore unsurprising that today, with the ubiquitous role that computers play in scientific development, that there is now great interest in the problem of how to to automate the processes which can answer questions such as those mentioned above. This is exactly what this book is about: how does one go about doing this within a *symbolic environment*, that is, when using a computer algebra system such as Maple, Mathematica, or one of the many others that are now readily available to people working within Universities or industry alike. It discusses ways of representing asymptotic information, and algorithms to calculate it, for a wide class of functions. Practically all of this book comprises the research of Professor Shackell in this field over the last 20 years; previously this was only available in various conference papers and journals.

[2]

2 Coverage

2.1 Chapter 2

After a brief introduction, the first main chapter concerns itself with the problem of *zero equivalence*. This is in fact a fundamental problem in all areas of symbolic computing, but it is quite a pervasive one in symbolic asymptotics as will become apparent once one has read chapter 4. Essentially one has to rule out that a function is not the zero function before one attempts to compute its asymptotics; further when computing the asymptotics of a non-zero function, say when using power series, one frequently performs operations upon them such as the arithmetic ones, or perhaps exponentiating or taking n^{th} roots. The coefficients, which will be symbolic, must be checked for zero equivalence— otherwise how will one know if he or she has sufficient terms? More seriously, we must guard against, when using division, whether we are dividing by zero...

Zero equivalence is well known to be undecidable in its most general form, as was shown by Richardson[3]. Nevertheless, various algorithms have been developed for testing whether a given element of a wide class of functions is identically zero, modulo the ability to decide whether a given constant is zero or not. The latter unfortunately is a very difficult question. Decision proceedures have been given (see [4] for example) but they rely on the truth of certain number theoretic conjectures whose proofs are generally thought to lie quite far beyond the present knowledge. A basic method of determining whether an exp-log constant, that is, numbers built from finite applications of the exponential, logarithm, arithmetic operations, together with a base field of constants such as Q is zero is given. The intuition is that if the expression is zero, then there will be dependencies between various subexpressions which are defined by the well-known laws of logarithms and exponentials, and these can be found by gcd calculations. A faster method which is still an active point of research is presented— that of using witness conjectures. Essentially, the idea is that only expressions which are complex syntactically can evaluate to numbers which are extremely small. This *potentially* leads to a zero-test where the decision procedure simply looks at the syntactic length of the expression.

The rest of the chapter looks at the problem of zero equivalence of functions. Loosely speaking, it can be summarized as follows. Each algebraically independent function in our given expression are defined in terms of differential equations, from which one looks for dependencies between derivatives. Ulimately, this brings us into the area of differential algebra, and generalisations of Gröbner bases for this setting to this problem are surveyed, such as the Rosenfeld-Gröbner algorithm. The adaptation and cost of such algorithms is discussed and later in the chapter, it is shown how to adapt well-known techniques such as modular methods, that is, working over a base field Z_p for some prime p as opposed to Z say in order to reduce the problem of intermediate expression swell.

2.2 Chapter 3

This chapter provides the tools required in order to answer the question stated above, of how to measure the rate of which a function grows. The important concept of a Hardy field, named in honour of G. H. Hardy, who proved some important results in his 1910 treatise, is given. Essentially, elements of Hardy fields are functions which are ultimately monotonic, and therefore must tend to a limit, which is either finite or infinite. This fact makes them very suitable with regard to calculating their asymptotics, and Hardy field theory forms a central theme throughout this book. Some key theorems about Hardy fields from the literature are then presented, such as the fact that one can

adjoin the exponential, logarithm, and even integral of an element of a Hardy field and the result is also Hardy field. One needs to be able to compare the asymptotic growth of different Hardy field elements, and a number of such measures are then presented which divide up the elements into equivalence classes, and various lemmas are proved. One such measure, the valuation, will be natural to anyone who has met L'Hôpital's rule; it looks at the limiting behaviour of the quotient of two elements f, g and deems them to have the same sort of behaviour if this limit is finite and non-zero. Thus two polynomials $p[x], q[x]$ have equivalent behaviour iff. their degrees are the same. Another measure captures the idea behind examples like the one made in the introduction above, that of *comparability*; here we look at the limit of $\dfrac{\log(f)}{\log(g)}$, and one sees that x^n and $\exp(x)$ are not comparable whilst two polynomials $p[x], q[x]$ are regardless of their degrees.

2.3 Chapter 4

This chapter sets up the machinery for making asymptotics automatic. Three main data structures are introduced: multiseries, nested expansions and star-products. The first of these objects is probably the most familiar to most readers. Essentially they are a generalistion of multivariate Taylor series expansions; the key difference being that each variable is not simply a real indeterminate, but actually a what is termed a *scale element*, each representing a different order of growth. For example, with $x \to \infty$, the function $\exp(x + \exp(-x))$ would have the multiseries expansion,
$(1 + x^{-1} + \dfrac{x^{-2}}{2!} + \cdots) + (1 + x^{-1} + \dfrac{x^{-2}}{2!} + \cdots)\exp(-x) + (1 + x^{-1} + \dfrac{x^{-2}}{2!} + \cdots)\dfrac{\exp(-2x)}{2!}$
which has two scale elements, namely x and $\exp(x)$.

Nested expansions can be viewed as a generalisation of continued fractions, where the order of growth of an exp-log is successively approximated closer and closer by recursive expansion inside iterated exponentials. Unlike multiseries, where scale elements are either iterated logarithms of x, or some suitable exponential function, the equivalent of scale elements in this context can only be logarithms, and exponentiation plays the role here that addition did for multiseries.

Star-products will be the least familiar of all, yet the basic idea of them will be familiar. The usual notation in this area is to use $\exp_n(f)$ and $\log_n(f)$ with $n \in \mathsf{N}$ to mean the n^{th} iterated exponential or logarithm of a function f. For two elements a, b of a field equipped with an exponential and (partial) logarithm operation, one defines the star-product of a and b, $a *_k b$, to be $\exp_{k-1}\{\log_{k-1}(a) \cdot \log_{k-1}(b)\}$. Writing this expression as $\exp_k \log\{\log_{k-1}(a) \cdot \log_{k-1}(b)\}$, this leads us to the identity, $a *_k b = \exp_k(\log_k(a) + \log_k(b))$; thus one can view star-products as a generalistion of the classical idea that one can perform multiplication by adding the logarithms of two numbers and then exponentiating. Again the addition and multiplication operations that play fundamental roles in multiseries do not dominate here, and in fact all operations are given equal weight so the algebraic flavour contrasts strongly.

For all three data structures, the action of each of the exp-log operations on a given object are described. For the case of multiseries, these operations are similar to the well-known methods, say, for multiplying two power series; however the methods here may be unfamiliar to most readers since they include the case where we no longer have Puiseux or Laurent series; algarithms presented in the next section show that such series arise naturally.

2.4 Chapter 5

This chapter strives to develop algorithms for more and more larger domains of functions. The most natural class beyond the class of rational functions (whose asymptotics are easily calculated by familiar means) to start with is that of the exp-log functions, which are functions built from finite applications of the exponential, logarithm, arithmetic operations, together with a field of constants. The major difficulty is to avoid the problem of *indefinate cancellation*. If expanded naively using Taylor series, a non-zero exp-log expression may apparently have an expansion consisting of an infinite series of zero coefficients! What is required first is careful rewriting of the expression in terms of scale elements of differing growth order— in the sense of the notion of comparablity— before the usual Taylor series expansions can be then safely applied. The extension to allow logarithms, exponentials and integrals of elements (mentioned in chapter 3) for a particular type of Hardy field is then considered. Here the problems involved in rewriting the expression mentioned above are much more involved. Nevertheless, a great deal can be done, and the section on meromorphic functions shows that one can also admit functions which can be defined in terms of algebraic differential equations (such as trigonometric functions, and even including special functions such as the Jacobi elliptic function) provided that at their arguments, they are meromorphic at infinity.

2.5 Chapter 6

Given an ordinary algebraic differential equation, $P = P(y, y', \ldots, y^{(n)}) = 0$, the focus of this chapter is to be able to describe the asymptotics of all the solutions which belong to Hardy fields. As the author points out, it is not an attempt to provide an algorithm to describe the behaviour of a *particular solution*; the example given there of $y^2+1-y' = 0$ with solution $y = \tan(x+k)$,with $k \in \mathsf{R}$ shows the difficulty. A major result is proved here regarding the possible *nested form* solutions of P— these objects may be regarded as providing an analogous role in nested expansions that the leading monomial does in a multiseries expansion— using an extensive case analysis. Three examples are then worked to illustrate the theory. The problem which is discussed next occurs when one wishes to compute further terms in a nested expansion solutions. Essentially, the idea is to substitute the unknown part of the nested expansion back into the differential equation recursively, but this generally results in differential equations of higher and higher order. This makes for problems of computational complexity, since the previous result showed (amongst other things) that the number of nested forms generally increases as a function of the order. Using a classical technique from combinatorics, a measure of the growth of the number of nested form solutions as a function of the order of the differential equation being considered is given. Finally there is some discussion on how this measure may be unduly pessimistic in practice, and a theorem for the important case of sparse differential equations is given which reduces the number of cases to be considered.

2.6 Chapter 7

Having calculated the asymptotics of an exp-log function $y(x)$, the question addressed here of how to do this for the *inverse* of y. There are two sections; the first presents an algorithm for inverting a nested expansion, whilst the second concerns inverting a multiseries expansion. Star-products are not treated here.

2.7 Chapter 8

It is natural to enquire what can be said for functions of more than one real variable. The idea is that one has systems of equations each of the form $H(x, y) = 0$ where H is an exp-log function. With x tending to infinity, (say) we are interested in calculating the asymptotics of solutions $y = y(x)$, that is, implicit solutions of this system. Nested forms are used to do this here.

The extension to 3 or more variables is briefly covered— this was not previously found in [5]. Here the open problems are expounded by means of examples: that of detecting degenerate systems of equations, preferably before a good deal of computational effort has been expended, and proving that a possible solution is in fact actually a solution.

2.8 Chapter 9

Star-products, introduced briefly in chapter four, receive the full attention of this chapter. The motivation to use them to represent asymptotic information is discussed, drawing on previous experience with multiseries and nested expansions. Then the machinery to produce an analogue to the exp-log algorithm is developed— that is how to compare growth orders of expressions in this new setting, and how to rewrite the expression into a star-product expansion. A major advantage of star-products over multiseries is their greater generality; and unlike the latter this allows them to express the asymptotics of *transexponential* functions, i.e. functions which grow faster than any iterated exponential. This interesting area is explored in the last section of the chapter.

2.9 Chapter 10

Intuitively, trignometric functions do not belong to Hardy fields so have been excluded in the book up until now. This part of the book brings the reader up to date with the most cutting edge research in this area. The first algorithm provides a way of extending the exp-log algorithm to admit trignometric functions by using an interval calculus; that is, by replacing each occurence by intervals representing the possible range values. The limitations of this algorithm are acknowledged, and the remainder of the book seeks to find more effective methods. The notion of a *wandering function* is given. This captures the idea that a function may oscillate with no limit, and indeed have "no particular asymptotic trend". An algorithm is given to calculate a wandering multiseries expansion for a large class of functions. The basic idea is to partition the real line into intervals on which the functions are 'reasonably behaved' and to try and obtain some information on these. An important difference between these objects and usual multiseries is that the coefficients of the former are no longer uniquely determined by the input function. Naturally there are certain restrictions in place, and as the author notes what the actual definition of a wandering function should be is still a matter of active debate between himself and one of his collaborators, Bruno Salvy.

3 Opinion and Summary

Since previously most of this material was only available in conference papers or journal articles, this book is a welcome and convenient source of this information. Moreover, the material is not merely a rewrite of these earlier works but some results are new and there is a greater exploration of the concepts; particulary in chapters 9 and 10. In this reviewer's own experience, it provides a

new presentation which is easier to understand and overall comes highly recommended. I have no real critiscism as such; just the remarks which follow.

It is not meant to be an encyclopedic treatise on asymptotics, but in the author's own words, the area of the subject which he has developed. Despite this the bibliography is extensive enough that this book makes a very good starting place indeed. The algorithms that are presented do not generally provide the reader with black boxes as such in the sense that some books provide routines given in pseudo-code. For example, how to best implement operations on multiseries is not covered in depth. References to well-known papers including up-to-date algorithms by J. Van der Hoeven for example are given, and it is up to the reader to read these in addition if he or she wishes to actually implement these algorithms. Nested expansions and star-products are the author's own discovery, there is no literature here on how to implement these objects.

Zero equivalence is a difficult area and the literature unavoidably difficult and technical. I would recommend that one saves a detailed reading of this technical chapter until the end, and in the meantime simply takes on board the author's strong message that zero equivalence is indeed something that must be handled, at least in the sense that any possible errors are well reported, before the algorithms in the rest of the book can be realised in practice. The chapters which follow can be readily understood notwithstanding. Such an approach would make the book more accessible to say, beginning graduate students. Despite being at the interface between mathematics and computer science, I would judge the material a little too difficult for anyone without a degree containing a good deal of pure mathematics. Additionally, it would be useful to have read a general introductory textbook on computer algebra, such as [2] for example, before reading this book. A passing appreciation with the problems involved in computing on the other hand is quite sufficient. Examples, are fairly plentiful, and usually very helpful, with a good level of detail in the working. There are only a few exercises throughout, although this is not a criticism as such; perhaps in a future edition this issue could be addressed, if the readership is be extended to say, graduate students. In this regard sample solutions would be useful in helping the reader check their grasp of the more technical side of the material.

References

[1] Du-Bois Reymond, P. Ueber asymptotische Werthe, infinitaere Approximationen und infinitaere Auflosung von Gleichungen. (1875)

[2] Davenport, J.H., Siret, Y. and Tournier, E. Computer Algebra Harcourt Brace Jovanovich (1988)

[3] Richardson, D. Some Undecidable Problems Involving Elementary Functions of a Real Variable. J. Symbolic Logic (1968), 33:514—520

[4] Richardson, D. How to Recognize Zero. In *Journal of Symbolic Computation* (1994), 24(6), 627–646.

[5] Shackell, J.R. Symbolic asymptotics: Functions of two variables, implicit functions. In *Journal of Symbolic Computation* (1998), 25(3), 329–349.

Review of[3]
Complexity and Cryptography. An Introduction
by John Talbot and Dominic Welsh
Cambridge University Press, 2006
292 pages, Softcover

Review by
Jörg Rothe rothe@cs.uni-duesseldorf.de
Heinrich-Heine-Universität Düsseldorf, 40225 Düsseldorf, Germany

1 Introduction

This textbook is written on an introductory level and in an easily accessible style. Three of the eleven chapters are concerned with the basics of complexity theory. To mention just one fundamental, well-known goal of this rich discipline, complexity theory studies how much of a resource is needed to solve a computational problem on a computer, where computation time and memory space are the most typical resources (or complexity measures). The three complexity-related chapters of this book provide an introduction to the field, present the theory of NP-completeness, and introduce some fundamental probabilistic complexity classes.

The remaining eight chapters focus on elementary topics from cryptography, including symmetric and public-key cryptosystems, one-way functions, digital signature schemes, key distribution protocols, pseudorandom generators, zero-knowledge protocols, and identification schemes. One important goal of cryptography is to design protocols that enable two or more parties to securely communicate over an insecure channel so that their communication and their data is protected against eavesdropping, forgery, impersonation, and other types of attack. The security of cryptosystems is often based on the assumption that certain problems are hard to solve, which makes complexity-theoretic considerations an important aspect of cryptography.

There are many books on cryptography, fewer books on complexity theory, and only very few books that cover both of these closely related fields. Recent textbooks on cryptography (that treat also complexity-theoretic issues to some extent) include [Gol01, Sch96, Sti05]; recent textbooks on complexity theory (that treat also cryptographic issues to some extent) include [HO02, Pap94]; and recent textbooks on both cryptography and complexity theory include [Rot05].

The title of "*Complexity and Cryptography. An Introduction*" by Talbot and Welsh might suggest that it falls into the third category—to be a book on complexity theory and cryptography. This review, however, will make the point that this book in fact falls into the first category. As such, it is a recommendable source for teaching an introductory course on cryptography. It is especially recommendable because of its compactness.

2 Summary

Chapter 1: Basics of cryptography. The most important cryptographic scenarios, models, and tasks are introduced, along with the definitions of symmetric and public-key cryptosystems and

the basic types of cryptographic attack. Differences between classical cryptogaphy and modern cryptogaphy are explained.

Chapter 2: Complexity theory. This chapter introduces to the field of complexity theory. Simple examples (e.g., unary integer addition, binary integer addition, and naive primality testing) are given to show how to estimate the runtimes of algorithms. Algorithms are always stated in pseudocode. The model of deterministic Turing machine (DTM) is presented; examples of DTMs are given for some functions defined earlier in this chapter. The notion of computability by Turing machines is defined (even though only for total functions), and the complexity measures time and space are introduced (again for total functions only). Decision problems (such as satisfiability and some graph problems) are considered and fundamental deterministic complexity classes such as P, PSPACE, and EXP are defined.

Chapter 3: Non-deterministic computation.[4] NP is defined, examples of NP problems are given (e.g., SAT, 3-COL, k-CLIQUE, and COMPOSITE), and basic inclusions between complexity classes are proven. The polynomial-time many-one reducibility, the polynomial-time Turing reducibility, and the notion of NP-completeness are defined. Cook's Theorem is proven, and starting from the NP-completeness of SAT a number of further NP-completeness results is established. Then coNP is introduced, and it is shown that the primality problem is both in NP and in coNP. The far stronger result by Agrawal, Kayal, and Saxena that this problem is in P is stated without proof (which makes sense in this kind of introductory textbook). Containments between complexity classes are proven, and the Berman–Hartmanis isomorphism conjecture is discussed. Finally, the model of nondeterministic Turing machine (NTM) is defined formally.

Chapter 4: Probabilistic computation. Probabilistic Turing machines and the probabilistic complexity classes RP, ZPP, and BPP are discussed. The standard class PP (probabilistic polynomial time), which admittedly is less important for cryptographic tasks, is not considered. Well-known randomized primality tests, such as the Fermat test and the Miller–Rabin test, are presented; other randomized primality tests, such as the Solovay–Strassen test, are omitted. It is then shown that BPP allows for probability amplification by which the error can be made exponentially small. Finally, boolean circuits are introduced as a nonuniform model of computation, it is shown that every problem in BPP has a polynomial-size circuit, and circuit complexity lower bounds are demonstrated.

Chapter 5: Symmetric cryptosystems. This chapter presents symmetric cryptosystems such as a simple monoalphabetic substitution cipher, the Vigenère cipher, and Vernam's one-time pad. It is shown that the latter guarantees perfect secrecy (while omitting the related notion of entropy and the statement of Shannon's Theorem, although both are mentioned in the chapter notes). Linear feedback registers are presented, and their insecurity is explained. Then nonlinear combination generators, block ciphers, and in particular the Data Encryption Standard (DES) are briefly described. The Advanced Encryption Standard (AES) is mentioned. Finally, the Pohlig–Hellman cryptosystem is presented.

Chapter 6: One way functions. The notion of strong (cryptographic) one-way function is defined and some candidates for one-way functions, including the modular exponentiation (whose inverse is the discrete logarithm), are given. Then the discrete logarithm assumption is discussed in relation with complexity class separations. A weaker notion of one-way function is also considered and discussed with respect to the factoring assumption.

[4] Chapter titles here are stated as in the book.

Chapter 7: Public key cryptosystems. This chapter introduces public-key cryptosystems, including the Cocks–Ellis, the RSA, and the ElGamal system. The importance of trapdoor one-way functions for public-key cryptosystems is discussed. Potential insecurities of RSA are demonstrated by describing certain possible attacks on RSA and appropriate countermeasures. In particular, the problem of determining the RSA secret key is related to the factoring problem: it is shown that both problems are equivalent with respect to randomized polynomial-time reductions.[5] Next, Rabin's cryptosystem and the Merkle–Hellman cryptosystem, which is based on the NP-complete SUBSET SUM problem, are presented. Finally, some problems with public-key cryptosystems that make use of trapdoor one-way functions are discussed. These problems include attacks that are possible when such one-way functions leak partial information, or when encrypted messages are not random, or when one and the same message is encrypted more than once.

Chapter 8: Digital signatures. The RSA digital signature scheme and the ElGamal digital signature scheme are presented. Both are discussed with respect to security issues. Also, the Digital Signature Algorithm (DSA, now modified to the Digital Signature Standard, DSS) is described. Digital signatures are discussed with respect to privacy requirements. Finally, the importance of hashing for signature schemes is emphasized, and the birthday attack is described.

Chapter 9: Key establishment protocols. This chapter provides secret-key agreement and distribution protocols. After describing the basic problems, a key distribution protocol by Blom (1984) is presented in its simplified version due to Blundo et al. (1993); this protocol makes use of so-called maximum distance separable codes and requires secure channels. Then the Diffie–Hellman secret-key agreement protocol is described, which doesn't require secure channels, and its security (formalized as the Diffie–Hellman problem) is discussed in relation to the difficulty of solving the discrete logarithm problem. The connection to the ElGamal public-key cryptosystem is also mentioned. Next, the issues of authentication and secret sharing are discussed. Finally, Shamir's secret-sharing scheme is presented.

Chapter 10: Secure encryption. The usage of (secure) pseudorandom generators for cryptographic purposes is described, and the next-bit test is presented. It is then explained that one-way functions may have easy and hard bits, and the notion of hard-core predicate is introduced. It is shown how to construct pseudorandom generators from hard-core predicates. Finally, probabilistic encryption methods are discussed, and the Blum–Micali cryptosystem is presented.

Chapter 11: Identification schemes. The last chapter turns to identification schemes. After an example of a classical challenge-and-response identification scheme, the important notion of interactive proof system is introduced, and concrete interactive proof systems are presented for the graph nonisomorphism problem and for the quadratic nonresidue problem. Then the idea of zero-knowledge protocols is explained, and perfect zero-knowledge and computational zero-knowledge is treated in more depth. In particular, a perfect zero-knowledge protocol is given for the quadratic residue problem. Relatedly, bit-commitment and coin-tossing over telephone are briefly discussed. Finally, the Fiat–Shamir (1987) identification scheme, which was modified by Feige, Fiat, and Shamir in 1988, is presented.

Appendices, Bibliography, and Index. In addition, there are six appendices that summarize

[5] May's result [May04] that, under certain assumptions, computing the RSA secret key is equivalent to factoring even under *deterministic* polynomial-time Turing reductions might have been mentioned here, at least in the chapter notes; even more so as randomized reductions, unlike deterministic Turing reductions, are not formally defined in the complexity theory part of the book.

the required mathematical background (e.g., from algebra, number theory, and probability theory), introduce to graph theory, and provide some hints and answers to selected exercises and problems. Finally, the book has a bibliography containing more than 150 references and a six-page index.

3 Opinion

This is a nice introductory textbook that covers the most important areas of cryptography in an easily accessible manner, provides numerous examples and easy exercises, and I really enjoyed reading it. Perhaps due to its compactness (with less than 300 pages), this book in some parts takes an approach that oversimplifies things. For example, the notion of Turing computability and the definition of the complexity measures time and space each are restricted to total functions. This might mislead a casual reader or a reader not familiar with the field into thinking that only functions with algorithms that always halt are computable, and also that only everywhere defined functions can have a computational complexity. In light of the central importance of partial functions in the fields of computability and complexity theory, the reader should at least have been warned about simplifications of this kind.

While the selection of topics and protocols from cryptography appears to be extensive and well-balanced, complexity theory is covered rather sparsely. Given the prominent position of the word "Complexity" in the title of the book, one might have expected a deeper and more comprehensive treatment of complexity-theoretic issues. For example, the Arthur-Merlin hierarchy, which combines the power of nondeterminism with the power of randomization (in the sense of BPP) and which is closely related to the interactive proof systems treated in Chapter 11, is not considered (only mentioned in the chapter notes). Thus, the reader learns nothing about the complexity-theoretic properties (inclusions, collapse, etc.) of this hierarchy.

However, as mentioned in the introduction, "*Complexity and Cryptography. An Introduction*" by Talbot and Welsh is a recommendable source for an introductory course on cryptography, which provides "cryptographically related" complexity theory to some extent.

References

[Gol01] O. Goldreich. *Foundations of Cryptography.* Cambridge University Press, 2001.

[HO02] L. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion.* EATCS Texts in Theoretical Computer Science. Springer-Verlag, Berlin, Heidelberg, New York, 2002.

[May04] A. May. Computing the RSA secret key is deterministic polynomial time equivalent to factoring. In *Advances in Cryptology – CRYPTO '04*, pages 213–219. Springer-Verlag *Lecture Notes in Computer Science #3152*, 2004.

[Pap94] C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[Rot05] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity.* EATCS Texts in Theoretical Computer Science. Springer-Verlag, Berlin, Heidelberg, New York, 2005.

[Sch96] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* John Wiley and Sons, New York, second edition, 1996.

[Sti05] D. Stinson. *Cryptography: Theory and Practice.* CRC Press, Boca Raton, third edition, 2005.

Review of[6]
Complexity Theory and Cryptology: An Introduction to Cryptocomplexity
by Jörg Rothe
Springer, 2005
484 pages, Hardcover

Review by
Piotr Faliszewski pfali@cs.rochester.edu
University of Rochester, Rochester, NY 14627

1 Introduction

Complexity theory is concerned with the study of difficulty of computational problems. Complexity theorists attempt to characterize which problems are easy to solve on a computer, which are hard, and how do particular classes of problems relate to each other. Cryptology is concerned with the design and analysis of secure information exchange protocols. That is, protocols that attempt to ensure that only the intended recipients are capable of decrypting the messages sent.

During the last couple of decades cryptology and complexity theory became inseparably intertwined. Complexity theory provides language using which communication protocols can be analyzed and either proved secure in some sense, e.g., via showing equivalence with some believed-to-be-hard problem, or broken. On the other hand, cryptology motivates and inspires studies within complexity theory. The textbook by Rothe presents this unified approach to the study of complexity theory and cryptology, coining the term *cryptocomplexity*. The book covers classical topics of complexity theory, together with some less known ones (e.g., Boolean Hierarchy), and standard topics in cryptology (e.g., public-key cryptography, digital signatures, etc.). The themes of complexity theory and cryptology are intertwined but each chapter of the book follows only one. As a result, it is easy to use the book as either a text for a purely complexity theoretic course or for a purely cryptology oriented class. Naturally, the text is best suited for a unified cryptocomplexity class.

The book is intended as a textbook for advanced undergraduate and graduate students and is based on author's long experience of teaching a series of interrelated courses on cryptocomplexity.

2 Summary

The book contains eight chapters that can loosely be divided into three parts: Preliminaries, Complexity Theory, and Public-Key Cryptography. (This division occurred to me as I was reading the book; the author did not partition the chapters in any way.)

The first four chapters provide an introduction to the topic of cryptocomplexity and appropriate preliminaries. The book is mostly self-contained and Chapter 2 introduces the preliminary mathematical concepts used throughout the book. Then, Chapters 3 and 4 give detailed introduction to, respectively, complexity theory and cryptology. This part of the book is written in a

somewhat reference-like style; intended readers of this book should have been exposed to the basics of complexity at least once before so this is a good approach.

The second part of the book, comprising Chapters 5 and 6, discusses advanced topics in complexity theory. Chapter 5 discusses hierarchies based on the class NP, e.g., the Polynomial Hierarchy and the Boolean Hierarchy and many others including the high and the low hierarchies. Polynomial Hierarchy is a well known hierarchy of alternating quantifiers. On the other hand, Boolean Hierarchy is not so well known and is not presented in textbooks often. Loosely speaking, k-th level of the Boolean Hierarchy contains languages L that can be expressed as $L = L_1 - L_2 - \cdots - L_k$, where each L_i is in NP and $L_k \subseteq L_{k-1} \subseteq \cdots \subseteq L_1$. The High Hierarchy is a generalization of the concept of NP-completeness, and the Low Hierarchy contains increasingly difficult problems that, nonetheless, are very unlikely to be complete for any level of the Polynomial Hierarchy. Other topics covered in Chapter 5 include query hierarchies (where we study how much more power does a polynomial-time algorithm get if it is allowed to pose an increasing number of queries to an NP oracle), parallel access to NP (where a polynomial-time algorithm is allowed to make a series of non-adaptive queries to an NP-oracle), and alternation (which generalizes existential mode of computation of nondeterministic machines to allow universal steps).

Chapter 6 discusses various modes of randomized computation. The author first describes randomized computations in general terms, and then moves on to discussing various way in which randomized algorithms can operate and accept languages. For example, the author discusses algorithms with one-sided error that for a yes/no problem have the following behavior: If the correct answer is *yes* then they accept with probability at least half, and if the correct answer is *no* then they never accept. The class of polynomial-time randomized algorithms of this type is called RP. Other randomized classes such as PP, ZPP, and BPP are studied too. In addition to that, the chapter covers Arthur-Merlin games, and graph isomorphism problem. Arthur-Merlin games model interactions between some computationally powerful agent (Merlin) and a simple bounded-error probabilistic polynomial-time agent (Arthur). The author goes back to the issue of graph isomorphism and Arthur-Merlin games later in Chapter 8 as they can be useful for authentication.

The last part of the book, comprising of the remaining two chapters, is devoted to classical public-key cryptology. In Chapter 7 the author provides a detailed discussion of the RSA cryptosystem, together witch issues of primality testing and factoring. Multiple algorithms for primality testing and factoring are given and the author discusses how to come up with primes for the RSA cryptosystem in order for those factoring algorithms to be inefficient. This chapter also contains an extended discussion of possible attacks at RSA and countermeasures to those.

The final chapter gives a quick glance at multiple other public-key cryptosystems (e.g., ElGamal's protocol), digital signature systems (again, e.g., a version of ElGamal's system), and key-exchange protocols (e.g., Diffie and Helman's protocol). The author also discusses Arthur-Merlin games with respect to cryptology and so called zero-knowledge proofs, which provide a way for an individual to authenticate him or herself.

Almost all of the chapters (except for the first one, which is an introduction) contain an extensive set of exercises and more involved problems that allow readers to test the skills they acquire. Many exercises either extend the results presented in the book or ask the reader to prove some theorems that appear in the text but for which the proof was omitted. This makes the reading process more interactive and guarantees that anyone who reads the book carefully becomes proficient in the subject.

It is also worth mentioning that the book contains a very well organized index. A very useful

feature of the index is that for each term in the index, the page number that contains its definition is typeset in bold. This greatly helps if someone is reading a chapter out of order and just needs a definition or two from other parts of the book.

3 Opinion

The book is well written and interesting. Some parts of it are quite unique (e.g., large parts of Chapter 5) in that most other textbooks on similar topics do not cover them, or do not cover them in as much detail as the textbook of Rothe. I like the book and I think it is a good textbook for graduate students and for advanced undergraduate students. The book is not, and does not try to be, an easy introduction to complexity theory and cryptology for students that have never been exposed to the material before.

A nice thing about the book is that it is written with a sense of humor. While one might expect that there is no place for laughter when discussing such an important and serious topic, there certainly is. Often, before discussing various theorems or definitions, the author includes a short story that feature either his daughters, or king Arthur, or Saruman the White, that gracefully introduces and explains difficult material that is about to be covered.

Thus, to summarize, I think it is a solid textbook for advanced students and a very good, comprehensive, reference for researchers.

Joint Review of[7] Three Blogs by theorists:
Computational Complexity (weblog.fortnow.com) by Lance Fortnow
Shtetl-Optimized (www.scottaaronson.com/blog/) by Scott Aaronson
in theory (in-theory.blogspot.com) by Luca Trevisan)

1 Overview

Disclosures: I wrote this review of blogs before Lance Fortnow retired. After he retired I took over his blog. But even before that I have guest blogged for both Lance Fortnow and Luca Trevisan, have been on podcasts with Lance Fortnow for his blog.
Note: I have changed a few things in this review because of Lance's retiring, but not much.

Blogs and books are very different in character. To contrast them consider the book *Freakonomics* by Levitt and Dubner, and the blog (www.freakonomics.com/blog/) with the same name and authors. The book had well thought out careful studies of issues such as 'does the crime rate go down because of abortions' and 'why do drug dealers still live with their mothers (that is, why isn't it more lucrative)'. The studies they did were careful and I have a certain confidence in them. If someone disagreed with their conclusions they would have to either do a different study or interpret the studies done. By contrast, consider the following topic from July 21, 2006:

Why are there now adults delivering newspapers rather than kids ?

The article says that it *seems* to him that there are no more kids delivering papers, only adults. He wonders why this is true.

[7] ©2007

Gee, I could speculate like that also! If it had been a topic in his book he would have carefully researched it and then based on the data have some reasonable conclusion.

Books can be a way to develop a well thought out theme (I realize that they often are not, but they can be). Blogs, due to their nature, can rarely do this.

So in reviewing blogs one should keep in mind that they are not really intended for deep thought. This is both their blessing and their curse.

2 Computational Complexity

Lance Fortnow was the first person to run a theory blog. His blog is on called *Computational Complexity* and that is his main topic. He has a new entry most days. When he is on vacation he often has a guest blogger. Hence you can usually go to his blog for a new entry.

He will announce new results of interest and also revisit old results. He has had a series of 'favorite theorems of the decade' postings which are technical but interesting. However, there is only so many new and old results you can post about so he often ventures into related topics such as NSF funding or academia. On a rare occasion he goes completely off topic such as when he boldly predicted the Bears would win the Superbowl. (Whoops.)

The posts are usually interesting and shows that he has a mature outlook on things. The most thought provoking for me was Thursday Jan 4, 2007 which was about how fields of study view themselves as just solving puzzles.

Looking over all of his Jan 2007 posts I was surprised to find that *none* were really on complexity theory proper. But they were almost all interesting and quite readable.

Why is Lance an authority? This is a chicken-and-egg thing- people read him so he has become the authority. However, if he was saying goofy things then perhaps people would read him less. Or more, as Scott Aaronson's blog may prove.

The comments on Lance's post's are of varying quality. One posting (June 28, 2006) that innocently annouced that that the FOCS accepts have been posted got 214 comments debating the merits of the program committee system. Actually there were 20 additional comments on the previous day's blogs on this topic. Comments can be good, but they can also be some combination of repetitive, off topic, or stupid.

I tend to read Lance's blog everyday and read a few of the comments. If the comments are getting repeitive I stop. If I feel I can add to it, I may do that, but often it is too late- a comment posted shortly before the next posting is wasted. This is a bug/feature of blogs- once the next post is up, the last one is forgotten.

3 Shtetl-Optimized

The one word I would use to describe Lance Fortnow Blog is 'maturity'. By contrast, the one word I would use to describe Scott Aaronson's blog is 'wild and crazy'. I know that 'wild and crazy' is three words, but it *feels* more like one. Lance is Walter Cronkite. Scott is Steven Colbert.

The name of the blog, 'Shtetl-Optimzed' does not really say what it is about. With this in mind one can never say Scott has gone 'off topic' since its not clear what the topic should be.

Scott often throws out ideas to provoke a discussion. For example- Scienfific papers are a waste of time-Oct 5, 2005 (I think he believes this) and Praising the president of Iran- Oct 29, 2005 (I

think he does not believe this) Sometimes I can't tell when he is serious. Other times he has posts that are somewhat technical- like a series of lecture notes on Quantum Computing. (I do not have the expertise to judge them). He also has the absolute best exposition of Peter Shor's quantum factoring algorithm I have ever read (Feb 24, 2007). And other random topics like Global Warming (I think he's against it), politics (I think he is not a fan of George W. Bush) and very miscellanous things like what his favorite foods are. I'm tempted to say 'why should I care what Scott's favorite foods are'; however, there were 49 comments on that post. But 14 of them were from Scott himself.

He caused a stir when he agreed to take whatever side of the String Theory debate would pay him the most money. This series of postings (and responses in other blogs) were enlightenting to me in that I found out that *there is* a debate about string theory. However, I couldn't figure out what the debate was really about.

Scotts posts tend to get lots of comments; however, the comments often go away from the topic of the blog. This happens on Lance's blog also, though I think its more common on Scotts. One example: in a post about Paul Cohen's death (March 24, 2007) there seemed to be a debate about whether or not C^* algebras are boring. Paul Cohen never worked in C^* algebras. This thread was started by accident when Scott himself answered a question of Greg Kuperberg.

Scott posts a new entry about twice a week. He also comments on his own blog entries leading to a 'blog within a blog'. I find some of his posts interesting, but many are ignorable. For example, unless he's coming over to my house for dinner, I don't care what his favorite foods are.

4 In Theory

Luca Trevian's blog is entitled 'In Theory' so the topic could be construed as 'Theoretical Computer Science' However, the term 'Theory' is somewhat broad. Hence, like Scott, one cannot really say he's gone 'off-topic'.

Luca Trevisan often posts quite technical material. This is both good and bad. On the one hand, the only readable online account of why Szemeredi's Regularity Lemma implies Szemeredi's theorem for the $k = 3$ case that I know of is Lucas Blog. On the other hand I often just cannot read it, or have to print it out and read it at home very carefully.

His nontechnical posts are about a variety of topics- some politics, some travel. His posts on China were very interesting (April 4,5,6). No real themes have emerged yet, but his is the youngest blog so that might take time.

He posts about twice a week. I try to read them. Often the technical ones I am interested in but don't have the time for. The non-technical ones are a matter of taste, however they are intelligent and well thought out.

5 Personal Comments

Tom Wolfe once said

We all have at least one good novel in us, some version of our autobiography. Thats why so many novelists have a good first novel and then burn out. They've said it all already.

I think we all have about a month of blogs in us. I applaud Lance, Scott, and Luca for keeping it up as long as they have. It helps that none of them blog only on computer science theory. Lance posts about Computer Science and Science in general rather than talk about Complexity theory

all the time. Scott provokes interesting discussions and tries to be funny (sometimes he succeeds). Luca posts deep math results that may not be that well known to the community, and other topics that are usually interesting. For all three this is hard work. I have confidence that they will not run out of things to say. (Lance has recently retired; however, this was not because he ran out of things to say. It became more of a chore than a pleasure.)

Should you read their blogs? Lance's blog is now mine, so I can't comment on that. The other two are good to read and I recommend them, but I warn you that between these blogs, other blogs you may read, other websites you browse, sending email, reading email, and cleaning up your spam, you won't have time to solve P vs NP.

6 Other Blogs

There are several other blogs that have theory as one of their themes. I invite anyone who wants to write a reivew of them for this column. Rather than list them (and hence start a pointless discussion of 'is that really a theory-centered blog') I direct you to my blog (same place Lance's was) where there are pointers to any blog that could remotely be called theory-centered.

John R. Shackell. Symbolic Asymptotics ~. Springer. John R. Shackell Institute of Mathematics, Statistics and Actuarial Science University of Kent Canterbury KentCT2 7NZ United Kingdom e-mail: j.r.shackell@kent.ac.uk. Mathematics Subject Classification (2000): 68W30, 41A60. Library of Congress Cataloging - in - Publication Data Shackell, John R., 1943 Symbolic asymptotics / John R. Shackell. p. cm.Â However there are two which are worth mentioning now. We have seen that we can represent our input functions by expressions using either trees or towers of the form (1.1). Then each expression corresponds to a unique function, but unfortunately the representation is not one-to-one. In this column we review the following books. 1. Symbolic Asymptotics by John R. Shackell. Review by James C. Beaumont. Given two functions f (x) and g(x) , how to you compare their growth as x goesâ€¦ Expand.Â Algorithms were presented in [4, 2] to solve this problem, and it was seen that both methods had their own strengths and weaknesses. Expand. 10. 1. PDF. View on ACM. Symbolic Asymptotics book. Read reviews from worldâ€™s largest community for readers. Symbolic asymptotics has recently undergone considerable theoretical ...Â Â· 0 ratings Â· 0 reviews. Symbolic asymptotics has recently undergone considerable theoretical development, especially in areas where power series are no longer an appropriate tool. Implementation is beginning to follow. The present book, written by one of the leading specialists in the area, is currently the only one to treat this part of symbolic asymptotics.Â Some results appear here for the first time, while others have hitherto only been given in preprints. Due to its clear presentation, this book is interesting for a broad audience of mathematicians and theoretical computer scientists. ...more. Get A Copy. This section reviews the underlying techniques of symbolic execution, which is a preliminary for discussing the chal-lenges and the benchmark approach we proposed in this work. 3.1 Theoretical Basis. The core principle of symbolic execution is symbolic reasoning. Informally, given a sequence of instructions along a control path, a symbolic reasoning engine can extract a constraint model and generates a test case for the path by solving the model. Formally, we can use Hoare Logic [20] to model the sym-bolic reasoning problem.Â The details are discussed as follows: â€¢ Symbolic variable declaration (Svar): In this stage, we have to declare symbolic variables which will be employed in the following symbolic analysis process. Symbolic Modelling in a nutshell Symbolic Modelling is a method for facilitating individuals to become familiar with the symbolic domain of their experience so that they discover new ways of perceiving themselves and their world. It uses Clean Language to facilitate them to attend to their metaphoric expressions so that they create a model of their symbolic mindbody perceptions.Â Our primary focus in this book is psychotherapy. And while we describe a complete process that can be used in its own right, many therapists and counsellors have found ways to combine Symbolic Modelling with their preferred approach.Â Assume the following statements refer to the same experience. Take a moment to say them out loud, and notice your internal response.