The Book Review Column[1]

by Frederic Green
Department of Mathematics and Computer Science
Clark University
Worcester, MA 02465
email: fgreen@clarku.edu

In this column, five books are reviewed, this time across six reviews.

1. **The Nature of Computation**, Cristopher Moore and Stephan Mertens. Reviewed by Haris Aziz and, in a separate review, by Frederic Green. This is a major recent textbook exploring in depth many aspects of the theory of computing, especially computational complexity. Haris and I agreed that it would be worthwhile to give this book two separate, hopefully complementary points of view.

2. **ReCombinatorics: The algorithmics of ancestral recombination graphs and explicit phylogenetic networks**, by Dan Gusfield. Review by Steven Kelk. Evolution does not just entail mutation and selection but also events such as recombination. This leads to the mathematical and algorithmic study of DAG-like structures, rather than more traditional trees, to model phylogeny. These include "phylogenetic networks" and "ancestral recombination graphs," the main subjects of this book.

3. **What is College For? The Public Purpose of Higher Education**, Ellen Condliffe Lagemann and Harry Lewis, editors. A collection of essays about this important issue, one of the editors (Harry Lewis) being a well-known theoretical computer scientist. Reviewed by William Gasarch.

4. **Slicing the Truth: On the Computability Theoretic and Reverse Mathematical Analysis of Combinatorial Principles**, by Denis Hirschfeldt. Reviewed by William Gasarch. A book focusing on the relationship between Reverse Mathematics and Ramsey Theory.

5. **The Scholar and the State: In Search of Van der Waerden**, by Alexander Soifer. A biography of the great mathematician, especially during his time working in Germany when the Nazis were in power, before and during World War II. Reviewed by William Gasarch.

BOOKS THAT NEED REVIEWERS FOR THE SIGACT NEWS COLUMN

Algorithms

1. *Distributed Systems: An algorithmic approach (second edition)* by Ghosh.
2. *Tractability: Practical approach to Hard Problems* Edited by Bordeaux, Hamadi, Kohli.
3. *Recent progress in the Boolean Domain* Edited by Bernd Steinbach
4. *The Art of Computer Programming, Volume 4, Fascicle 6: Satisfiability* by Donald Knuth

Programming Languages

1. *Selected Papers on Computer Languages* by Donald Knuth.

Miscellaneous Computer Science

1. *Algebraic Geometry Modeling in Information Theory* Edited by Edgar Moro.
2. *Digital Logic Design: A Rigorous Approach* by Even and Medina.
3. *Communication Networks: An Optimization, Control, and Stochastic Networks Perspective* by Srikant and Ying.
4. *CoCo: The colorful history of Tandy's Underdog Computer* by Boisy Pitre and Bill Loguidice.
5. *Introduction to Reversible Computing*, by Kalyan S. Perumalla

Cryptography

1. *The Mathematics of Encryption: An Elementary Introduction*, by Margaret Cozzens and Steven J. Miller.

Miscellaneous Mathematics

1. *The Magic of Math*, by Arthur Benjamin.

Mathematics and History

1. *Professor Stewart's Casebook of Mathematical Mysteries* by Ian Stewart.
2. *An Episodic History of Mathematics: Mathematical Culture Through Problem Solving* by Krantz.
3. *Proof Analysis: A Contribution to Hilbert's Last Problem* by Negri and Von Plato.

Review of [2]
The Nature of Computation
by Cristopher Moore and Stephan Mertens
Publisher: Oxford University Press, 2011
985 pages, $100 Hardcover, $64.99 Kindle

Review by
Haris Aziz (haris.aziz@nicta.com.au)
Data61 and UNSW, Sydney, Australia

1 Introduction

The Nature of Computation (TNoC) is a comprehensive, accessible, and highly enjoyable book that conveys the key intellectual contributions of the theory of computing. The project took off as an effort to present theoretical computer science to physicists, but it is equally suitable for any science graduate who is curious to explore beautiful and deep ideas related to the mathematical structure of problems. Moore and Mertens explain the essence of the book as follows:

> "*It is a question about the mathematical structures of problems, and how these structures help us solve problems or frustrate our attempts to do so. This leads us, in turn, to questions about the nature of mathematical proof, and even of intelligence and creativity.*"

TNoC provides not just a window through which people from other disciplines can get glimpses of the interesting nuggets from computer science, but also provides an entertaining open house session where a visitor can meet various deep ideas and understand the core arguments behind key results.

2 Summary

TNoC is divided into fifteen chapters.

1. Prologue
2. The Basics
3. Insights and Algorithms
4. Needles in a Haystack: the Class NP
5. Who is the Hardest One of All? NP-Completeness
6. The Deep Question: P vs. NP
7. The Grand Unified Theory of Computation
8. Memory, Paths, and Games

9. Optimization and Approximation
10. Randomized Algorithms
11. Interaction and Pseudorandomness
12. Random Walks and Rapid Mixing
13. Counting, Sampling, and Statistical Physics
14. When Formulas Freeze: Phase Transitions in Computation
15. Quantum Computation

    [Appendix] Mathematical Tools

The book starts with a beautiful narrative on how the problems of finding an Eulerian cycle and a Hamiltonian cycle are apparently similar but have different complexity. It discusses various complexity classes, and, in later chapters, approximation algorithms, complexity of counting and sampling, phase transitions, etc.

Chapters 1 to 5 consider more traditional topics in algorithms and complexity. The focus of the earlier chapters is on running time analysis, classical algorithmic techniques, and time complexity classes. In particular, Chapter 3 is a one-stop algorithms primer with explanations of dynamic programming and network flows. Chapters 4, 5 and 6 cover NP-completeness, provide intuition for why NP-complete problems are hard, and discusses importance of the 'P vs NP question' in detail.

Chapter 7 covers topics including the halting problem, decidability, and the Church-Turing Thesis. Chapter 8 focuses on issues around space complexity with explanations of complexity classes such as L, NL, and PSPACE. Chapter 9 is a very readable and insightful chapter capturing fundamental topics regarding optimization and approximation, including mathematical programming, inapproximability, duality, interior point methods, and semidefinite programming.

There is great coverage of the role of randomization in computation, especially in Chapters 10, 11, 12, and 13. Chapter 10 provides a taste of randomized algorithms with a deeper study of the nature of primes and randomized algorithms for testing whether a number is a prime. Chapter 11 discusses zero-knowledge proofs and presents the PCP theorem that essentially shows that problems in NP have proofs that can be checked by looking at a few bits. Derandomization is also discussed in Chapter 11. Chapter 12 takes the reader on a wonderful random walk tour. The chapter builds the groundwork for ideas about rapidly mixing Markov chains, which are applied to counting problems in Chapter 13.

Chapter 14 covers the intriguing phenomenon of phase transitions: Problems that almost always have yes instances almost always end up having no instances when the number of constraints per variable crosses a critical threshold. The authors' background in physics is valuable in giving further insights. The book concludes with Chapter 15 — one of the most elegant introductions on quantum computation.

Any reader with some college mathematics background will be able to appreciate the manner in which fundamental algorithmic techniques such as recursion, divide and conquer, dynamic programming and network flows are introduced and explained. Various other ideas on complexity classes are presented in a similar vein.

3 Opinion

It is difficult to understand what category one can put TNoC in. The first impression when one picks up the book is how hefty it is. Built like a handbook or encyclopedia, TNoC is indeed a labor of love of Mertens and Moore. Unike a typical encyclopedia, TNoC focusses on the big ideas and covers them in a gentle and passionate manner. The book is also far removed from a typical textbook which would normally have a series of definitions and theorems. In fact, there is negligible formalism or formal notation. Despite the lack of formalism, TNoC is also not a fuzzy popular science book that is so generic that deep ideas cannot be conveyed. On the contrary, Moore and Mertens communicate ideas in a highly accessible way. The style of writing is very conversational. Without sounding overly serious, the authors manage to convey the beauty of very deep ideas and concepts from computer science.

Since TNoC is written in such a reader-friendly manner, it is a valuable resource for any educator. For so many ideas, my reaction was "this is a great way to explain this idea to a peer or a student." The text is generously complemented with suitable figures and tables. The main arguments of numerous interesting results are provided. For example, I enjoyed the explanation of the connection between complexity of counting and sampling. There are exercises in the main text of each chapter and well-chosen exercise questions at the end of each chapter.

Each chapter includes historical notes that help give a sense of how important ideas were developed. The notes make the book a pleasure to read. For example, in the chapter on Counting, Sampling, and Statistical Physics, one finds the following note:

> *The class #P and the original proof that PERMANENT is #P-complete are due to Leslie Valiant [786, 787], who won the 2010 Turing Award for this and other work. The proof of PERMANENT's #P-completeness that we give here is by Ben-Dor and Halevi [98]. Another simple proof, which reduces from #VERTEX COVERS, appears in Kozen [493]. Parsimonious reductions between NP-complete problems were also considered by Simon [739].*

The text is interspersed with quote gems by people like George Polya, Stephen Kleene, Alan Turing, Jack Edmonds, Donald Knuth, Richard Bellman, Richard Feynman, David Hilbert and Richard Hamming. Philosophers make an entry as well with quotes by Plato and Nietzsche. Both the historical notes and quotes provide a nice context for the big ideas in theoretical computer science.

TNoC is refreshingly modern. Apart from ideas that are communicated in various classical texts, there are nice descriptions of relatively newer ideas such as the PPAD complexity class and the unique games conjecture.

The text is laced with charm and humor. In a section on hash functions, it was amusing to find a quote from a book by the delightful P. G. Wodehouse: "*Heavens, Ginger! There must be something in this world that you wouldn't make a hash of.*" The authors are self-deprecating at various places. For example, in the section on quantum computation, they make fun of physicists' sense of humour at their naming of the "bra-ket" notation for describing quantum states $\langle \phi \mid \psi \rangle$.

The authors also write that "*Overall, we have endeavoured to write our book with the accessibility of Martin Gardner, the playfulness of Douglas Hofstadter, and the lyricism of Vladimir Nabokov. We have almost certainly failed on all three counts.*" I beg to disagree!

Review of[3]

The Nature of Computation
by Cristopher Moore and Stephan Mertens
Publisher: Oxford University Press, 2011
985 pages, $100 Hardcover, $64.99 Kindle

Review by
Frederic Green (`fgreen@clarku.edu`)
Department of Mathematics and Computer Science
Clark University, Worcester, MA

When I read mathematical papers or textbooks, much of the time is spent trying to "reverse engineer" cryptic formalism in an attempt to understand the hidden intuition. What with unraveling all the technical minutiae, and figuring out why they have to be as they are, it can easily take days or weeks to understand a complicated proof. In a traditional mathematical text, the reader is left to his or her own devices with little or no help from the author. But, to our good fortune, recently authors have increasingly made a concerted effort to provide the reader with some conceptual guidance (Sipser takes the time to do this in his admirable textbook on Theory of Computing, for example), and thereby unveil the ideas lurking in the formalism.

The Nature of Computation (following Aziz's review, "TNoC") does a truly outstanding job of putting the underlying ideas in plain sight of the reader. It goes beyond mere "conceptual" guidance, bordering on the *inspirational*. I mean, where you are more likely to read the following sentence, in a self-help book, or in a textbook on computational complexity:

> Wouldn't it be wonderful if you could tell whether your current solution to life's problems is the best possible?

While the book might not abound with queries such as the above, it is certainly infused with an unusually engaging attitude towards and respect for the reader. A more representative example concerns the notion of reducibility. In my experience, despite my best efforts, students always seem to be thrown by what can be interpreted as deceptive terminology. Maybe it *is* deceptive? Moore and Mertens are refreshingly candid on this:

> If the reader finds the word "reduction" confusing, we sympathize. Saying that A can be reduced to B makes it sound as if B is smaller or simpler than A. In fact, it usually means the reverse...

This kind of concern and attention to the reader's potential confusion is sustained for the entire book.

As far as I could see, the authors let no opportunity slip by to provide all-important *motivation*, through substantial, concrete examples *and non-examples*. Concepts emerge through generalizations of these examples. This (in my opinion) is how mathematics should be taught, with one idea naturally flowing into the next. The result is a mathematical exposition that is readable as a narrative, without sacrificing rigor. For example, the book pretty much begins by building intuition about the power of various algorithmic techniques (divide and conquer, dynamic programming, greediness, etc.). None of these algorithms are "pulled out of a hat." The authors motivate working algorithms by first suggesting a natural, perhaps naive technique that doesn't completely work, but leads to an insight for something that does. We learn *how to do it* by first learning *how not to do it*. Just like real life!

After looking at problem-solving techniques that yield efficient solutions, we see that, curiously, certain problems do not lend themselves to these techniques. Why does graph search yield such a nice solution to Eulerian Path, but fails so utterly with the (similar, or so it seems at first) Hamiltonian Path problem? We are led inexorably to the big question, why some problems do, and others do not, yield to such attacks. An entire chapter is devoted just to the definition and nature of NP and the rich set of problems that are to be found there. This is also a natural context in which to introduce reducibility, in order to demonstrate the mutability of these problems between themselves; this includes reductions for proving efficient solutions (e.g. for 2-SAT) as well as reductions that are ultimately used to prove hardness. *Only then*, in another chapter, are we shown an unusually diverse set of NP-completeness results, ranging from old standards (satisfiability, independent set, Hamiltonian path, etc.) to more far-flung and intriguing problems (quadratic diophantine equations, tiling with trominoes, integrating trigonometric functions, cellular automata). Again, as with the algorithms, reductions are well-motivated, and never pulled out of a hat.

Moore and Mertens carry this progression of ideas through the entire book. Furthermore, as illustrated above, they take pains not to lose track of the *two* key questions of computational complexity: not only what it is that makes certain problems *hard*, but also what it is that makes other problems *easy*.

The book is unconventional in many other ways. For example, the development does not rely at all on automata theory or Turing machines (although Turing machines are certainly discussed in the chapter on computability). I regard this as a significant strength, since the material does not depend on a prior course on models of computation, and modularity is a good thing. Most noticeably, it emphasizes physics more than your typical complexity text. This is fitting, since in the past couple of decades there has been a fascinating cross-fertilization between physics and theoretical computer science. Prominent topics in this intersection include the connection between the algorithmics of counting and statistical physics (e.g., in the solution of the 2D Ising Model), critical phenomena in NP-complete problems (the phase transitions between between easy and hard), and of course quantum computing; all of these are treated in TNoC. Topics that ordinarily come to mind in computational complexity are also covered in great detail: You will also find excellent and lucid expositions of complexity classes, interactive proofs, pseudorandomness, hardness of approximation, etc. (for more detail, refer to Aziz's review).

Connections are drawn, and the unity of the field illustrated, wherever possible. For example, the Fast Fourier Transform is introduced as an example of divide and conquer in Chapter 3. It resurfaces a couple of times, most prominently at the heart of Shor's algorithm in Chapter 15. Space complexity in Chapter 8 is described in terms of the "state space" (the space of configurations of memory of a finite computer). Again, we re-encounter this idea many times subsequently, and revisit it in the introduction to quantum computing. The "vector of probabilities" (probability as a function of the state of the machine) of a classical probabilistic computation leads naturally to the complex state vector of QC.

I can't help but remark that (at least in my opinion), the book itself is physically beautiful. The authors and Oxford University Press put a great deal of thought and effort into the design, which is integral to how the concepts of the book are conveyed. This is but one facet of TNoC's overall spirit. There is an ever-present playfulness in the text that keeps the reader going. You can see this just from a random sampling of the titles of some of the sections (and subsections), for example:

- *Games People Play*: About PSPACE-complete problems.
- *Hunting with Eggshells*: About the ellipsoid algorithm for linear programming.
- *In Which Arthur Scrambles a Graph*: About the interactive protocol for graph isomorphism.

- *Magic Moments*: The first and second moment methods for determining critical densities (of clauses to variables).

(You will also find more sober section headings like "Natural Proofs," or "The Minimax Theorem and Yao's Principle.") Even the chapter-end problems have clever and helpfully descriptive names. Or consider one of the opening paragraphs to the chapter on random walks and rapid mixing, closing laconically with four words that ought to propel the reader to the sequel:

> The number of steps that it takes for a Markov chain to approach equilibrium, and thus provide a good random sample of the state space, is called its *mixing time*. As we will see in this chapter, calculating the mixing time of a Markov chain requires us to think about how quickly its choices overwhelm the system's memory of its initial state, how much one part of a system influences another, and how smoothly probability flows from one part of the state space to another. To grapple with these issues, we will find ourselves applying a panoply of mathematical ideas, from combinatorics, probability, group theory, and Fourier analysis.
>
> We begin by considering a classic example from physics: a block of iron.

And I hope that the above demonstrates that not taking oneself too seriously is not to say that the basic intent (and content) is ever anything less than serious!

Now can one say anything at all negative about this book? It is indeed a challenge to criticize something as good as this. The downsides are, at worst, minimal. While informality (done right) usually evades confusion successfully, there are occasional exceptions. The less experienced or inattentive reader can be lulled into complacency by the *seemingly* facile manner in which certain proofs are presented. Many of the proofs essentially consist of an example of a certain construction (e.g., in the proof that Hamiltonian Path is NP-complete). Of course, the details can always be filled in, and the examples are carefully chosen, so this is really more of a cautionary note to the reader that mathematical maturity (and/or a willingness to cultivate it) is absolutely essential for achieving a rigorous understanding of the material. To take another (more technical and punctilious) example, the same notation, "$A \leq B$" is used to express that problem A is reducible to problem B for polynomial-time many-one reductions, polynomial-time Turing reductions and computable reductions (again both many-one and Turing). The type of reduction is always specified in the text (or in the problems/exercises in which the ideas are applied), but the actual names of the reductions are relegated to a chapter endnote in Chapter 5. It would help to distinguish between them notationally in the early chapters. Among other things, that would serve to shed further light on notations such as P^{NP}. In later chapters, special-purpose notations are introduced after all for logspace and randomized reductions, so it seems to me it would not be obtrusive to differentiate the reductions, in the main text, as soon as it is appropriate.

However, the strengths of the TNoC approach so far outweigh the downsides, the risks are well worth taking. I am reaching in any attempt to come up with shortcomings. The book is highly recommended for all interested readers: in or out of courses, students undergraduate or graduate, researchers in other fields eager to learn the subject, or scholars already in the field who wish to enrich their current understanding. It makes for a great textbook in a conventional theory of computing course, as I can testify from recent personal experience (I used it once; I'll use it again!). With its broad and deep wealth of information, it would be a top contender for one of my "desert island" books. TNoC speaks directly, clearly, convincingly, and entertainingly, but also goes much further: it inspires.

Review of[4]

ReCombinatorics: The algorithmics of ancestral recombination graphs and explicit phylogenetic networks
Author of Book: *Dan Gusfield*
Published by MIT Press
Hardcover, $60**, 600 pages**

Review by
Steven Kelk (`steven.kelk@maastrichtuniversity.nl`**)**
Department of Knowledge Engineering (DKE)
Maastricht University, The Netherlands

1 Overview

One of the central mathematical abstractions in the study of evolution is the *phylogenetic tree*. Essentially, this is a rooted tree in which the leaves are bijectively labelled by a set of contemporary species X, and the internal nodes of the tree model the biological events (such as speciation) that caused the root of the tree - representing a hypothetical common ancestor - to diversify into the set X. From a mathematical perspective, the central challenge in phylogenetics is to infer the topology of the tree given only measurements obtained from the species X at its leaves, such as DNA sequence data. There are many different optimality criteria for this tree-inference problem, most of them NP-hard, and the literature is vast. In recent years there has been growing attention to the fact that evolution is not always tree-like. In particular, due to "fusing" biological phenomena such as hybridization, recombination and lateral gene transfer, evolution is sometimes better modeled as a rooted, leaf-labelled directed acyclic graph (DAG), where nodes of indegree two or higher are used to model the fusion phenomenon in question. Such DAGs directly generalise phylogenetic trees and are known by various different names in the literature, the two most commonplace being *phylogenetic network* and *Ancestral Recombination Graph (ARG)*. In this book Gusfield mainly uses the term ARG, and indegree two nodes are called recombination nodes, reflecting the population-genetic origin of his work. Unlike the literature on tree-inference, the literature on ARG-inference is comparatively new and small, and in this book Gusfield gives an overview of some of the main algorithmic results in this area from the last twenty years.

The core of the book concerns the problem of inferring an ARG with a *minimum* number of recombination nodes. More formally, the input is a binary matrix M with n rows and m columns, where the n rows can be viewed as length-m binary strings that label the n leaves of the ARG we are trying to infer. Specifically, we wish to infer an ARG and a labelling of its internal nodes with length-m binary strings, such that (i) per column there is a mutation (i.e. a transition from 0 to 1 or vice-versa) on at most one edge of the ARG and (ii) the sequences labelling recombination nodes are formed by concatenating a prefix of the string labelling one parent, with a suffix of the string labelling the other. An ARG with a minimum number of recombination nodes is called a $MinARG$, and $Rmin(M)$ is used to denote the minimum value itself. Computing $Rmin(M)$ is, inevitably, NP-hard, and notoriously intractable. The majority of the chapters of this book are devoted to the algorithm engineering challenge of computing (bounds on) $Rmin(M)$ in practice. Remaining chapters discuss related but somewhat different problems (such as the problem of "phasing" genotypes into haplotypes) and biological applications, but computation of $Rmin(M)$ is certainly the dominant theme of this book.

2 Summary of Contents

The book has fourteen chapters and pleasingly, it is completely self-contained. Although the book addresses a biologically-motivated problem it is primarily an algorithms book. Anybody with a mathematics or computer science background, and a familiarity with the basics of algorithm design (e.g. big-O analysis) will comfortably be able to read this book. There are frequent references to the biological underpinnings and applications but Gusfield presents them in a very computer-science friendly way which contextualises the algorithmic results, rather than distracts from them. Indeed, Gusfield is a professor of computer science and the whole book has a “journey of discovery” feel, documenting how an algorithms expert (armed with the traditional array of weapons against NP-hard problems) fared as he dug deeper into the world of ARGs.

The first two chapters, encompassing sixty pages, introduce the biological context, give the necessary definitions (including basic graph theory terminology) and present a number of classical results from the literature on constructing phylogenetic trees. The book very quickly settles into a pattern which is repeated in almost all chapters: biological/historical context, definition, theorem, proof, extensions. The MinARG / $Rmin(M)$ problem is defined and motivated in the third chapter. The fourth chapter is much less mathematical than surrounding chapters, discussing three applications of recombination analysis in practice. Gusfield himself notes that this chapter can be skipped by those readers only interested in the algorithms/mathematics.

The algorithm engineering begins proper in the fifth chapter, covering sixty pages, in which at least four different lower bounds on $Rmin(M)$ are analysed in detail. This focus on algorithmic bounds (and computing $Rmin(M)$ in practice by sharpening lower and upper bounds until they meet) is characteristic of the algorithmic ARG literature, contrasting with other parts of the literature where authors have focussed more on approximation algorithms and fixed parameter tractability. Some of the lower bounds presented can be computed in polynomial time, some are composites of other bounds, and some are themselves NP-hard to compute. As an illustration, one bound is based on the observation that each pair of columns inducing a certain forbidden submatrix must be separated by at least one recombination node. Leveraging the linearly ordered nature of the data, this naturally leads to a “Hitting Set on intervals of the line” formulation that can be solved in polynomial time with a greedy algorithm. The bound can subsequently be improved by relaxing “intervals of the line” to “subsets of the line” but this makes the problem NP-hard. At this point Gusfield deploys one of his favourite tools: Integer Linear Programming (ILP). He is a strong proponent of using ILP in computational biology and here, as in many other parts of the book, it turns out to work very well when computing bounds, or solving NP-hard subproblems related to computation of $Rmin(M)$, on realistic-sized datasets. (The book includes an appendix which is essentially a crash-course in modelling with ILP). ILP cannot, however, be used to compute $Rmin(M)$ directly, so it is emphatically *not* the case that computation of $Rmin(M)$ can be solved in practice simply by unleashing industrial-strength solvers such as CPLEX or Gurobi. Indeed, this explains the - at first glance - rather exotic choice to develop lower bounds that are themselves NP-hard to compute: unlike $Rmin(M)$ itself they have natural static formulations which are amenable to formulation as ILPs. It is also worth noting that ILP is by no means the only response to NP-hardness encountered in the book: in various places other options such as exponential-time dynamic programming, and combinatorial branch and bound, are explored.

The sixth and seventh chapters concern a natural decomposition theorem which sometimes allows computation of $Rmin(M)$ to be significantly simplified (and can also be used for developing lower bounds). The core idea is very natural. Specifically, the forbidden-submatrix obstruction described earlier naturally leads to the *conflict graph*, in which each column of M corresponds to a node and there is an edge between two nodes if they form an obstruction. It turns out that, in some circumstances (but by no means all) the connected components of this graph can be processed independently of each other.

Building on this idea, the next chapters see the book switch to upper bounds, i.e. algorithms that actually construct ARGs. The eighth chapter takes a restricted approach, demonstrating that $Rmin(M)$ can be computed in polynomial time if at least one MinARG has a highly restricted topology. Namely, this holds if every biconnected component of the underlying undirected graph, contains at most one recombination node. Such ARGs are called *galled trees* - elsewhere in the literature they are called *level-1 networks* - and in many ways they are the first, most natural step up from trees. Due to their very simple structure they behave very well with respect to the decomposition theorem presented in chapters six and seven, quickly leading to a divide-and-conquer approach. Chapter nine, however, addresses the much more challenging problem of computing $Rmin(M)$ when no assumptions are made about the output topology of the MinARG (or the input matrix M). This is the chapter where the intractability really begins to bite. The issue, as observed elsewhere in the literature, is that computer science as a whole has little experience with optimization problems in which graphs (in this case, DAGs) are the output, rather than the input. The algorithms in this section are thus tantamount to (intelligent) exhaustive search of the space of all ARGs, sometimes accelerated with branch-and-bound style techniques, and this is computationally formidable. Although it does not improve the running time, this chapter also introduces an idea which researchers working elsewhere in the literature will immediately recognise: that the columns of M can naturally be partitioned into intervals which have a common, tree-like history, and that partial information about the topology of these trees (known as *marginal trees* here) can be extracted directly from M. This information is useful, because we know that the MinARG must somehow simultaneously topologically embed all these marginal trees. Hence, at least to a certain extent, construction of MinARGs and computation of $Rmin(M)$ can be re-formulated as a "tree-packing" problem.

Chapter ten discusses two further lower bounds, one of which can be viewed as a matrix partitioning problem, and the other as a type of minimum-length elimination ordering problem. In terms of flow these two bounds would ideally have found a place in chapter four, together with the other lower bounds, but as Gusfield explains they both require ideas developed in the intervening constructive chapters. Chapter eleven is very short and develops some (not entirely surprising, and mostly descriptive rather than constructive) necessary and sufficient conditions for a MinARG to be amenable to the divide-and-conquer approach that was earlier demonstrated to be correct for galled trees.

At this point the book makes a rather sharp jump. In chapter twelve it (temporarily) moves away from computation of $Rmin(M)$, to focus on the so-called Haplotype Inference problem. Here the input is a matrix G over the alphabet $\{0, 1, 2\}$, which represents a set of *genotypes*, and the goal is to identify a binary matrix M (where each row represents a *haplotype*) such that every row of G can be expressed as the "sum" of some two rows of M, where "summation" is defined as: $0 + 0 = 0$, $1 + 1 = 1$, $0 + 1 = 2$ and $1 + 0 = 2$. Combinatorially this is a very interesting problem, with very many variations. The most basal variant - does such a matrix M exist for a given input matrix G? - is shown to be reducible in polynomial-time to the classical and well-understood Graph Realization problem. More sophisticated variants of the problem again bring ARGs back into the equation. For example, for a given G, can we identify an M such that $Rmin(M)$ is minimized? Chapter thirteen also has a rather different flavor than the rest of the book, discussing at a comparatively high-level the use of ARGs in *(Genome Wide) Association Studies* which is a technique for identifying combinatorial patterns in the genome that seem to be causal for diseases. This chapter is much less algorithmic than the others, briefly giving an overview of various different models and techniques, and will be one of the more accessible chapters for biologists.

The final chapter of the book is, from the perspective of unifying the literature in this area (where mathematically isomorphic models often have multiple different terminologies) very important. It emphasizes that $Rmin(M)$ and some of its lower bounds can be re-formulated within models that attempt to quantify the topological discordance between a given set of phylogenetic trees (i.e. where the input is a set of trees, rather than a binary matrix M). These models, and related optimization problems such as the Hybridization Number and Maximum (Acyclic) Agreement Forest problems, have been very well-studied outside the ARG literature. It is commendable that Gusfield emphasizes these commonalities, and this also explains the elaborate title of the book: it is an attempt to cover two separate nomenclatures. The final chapter also touches on an elegant link between multi-state problems in phylogenetics (i.e. where the input matrix is over a larger alphabet) and chordal graph theory.

3 Overall

This is a very well-written and self-contained book which gives a comprehensive overview of algorthmic results concerning ARGs, and everything is referenced rigorously. As stated at the beginning of the review, it is certainly not a book for biologists - it is full of proofs - although they will certainly appreciate the way Gusfield ties the algorithmic results to the applications context. For researchers already working in the area the algorithmic results will not be so surprising, but this is hardly to be expected given that the book primarily serves to integrate and summarize seminal results from the last twenty years. I classify myself in this last group, but I nevertheless greatly enjoyed the exposition, and particularly appreciate the effort Gusfield takes to point to further reading on the statistical and probabilistic side of the story. This is important because, although the book is not statistical at all, a great deal of biomathematics certainly is, and understanding that this dimension exists is critical to understanding the role of combinatorial optimization in this area.

I do have some negative points, but they are mainly stylistic. In a few places the book goes into detail which, for an algorithmic audience, is superfluous. I particularly felt this was the case in the chapters about the decomposition theorem and, related to this, galled trees. These results are not so surprising yet they are presented with a little too much swagger. Also, the book sometimes leans a little bit too much in the direction of algorithm engineering. That is, it occasionally devotes attention to the details of optimizing the running times of already competitive polynomial-time subroutines when the great challenges in this area lie at the other end of the spectrum i.e., dealing with the severe intractability of the core NP-hard problems. Personally I would also have attempted to dissect this NP-hardness a little more, pushing the analysis more towards (fixed) parameterized complexity, and exploring the polyhedral dimension when appropriate, e.g., some of the polynomial-time algorithms presented could equivalently be formulated as totally unimodular linear programs (which are guaranteed to have integral solutions). These points, however, are all a question of taste: this is unquestionably a good book.

Let me conclude by recommending this book to three groups in particular. First: researchers already working in the area who are looking for a reference text on algorithms for ARGs, complete with biological motivation. Secondly, traditional algorithms researchers who are looking for a combinatorially clean entrance-point to computational biology, explained by somebody with a computer science background. Thirdly, this book could quite easily be used as the scaffolding for an entire MSc (or advanced BSc) algorithms course. All the standard tools for polynomial-time algorithm design, and dealing with NP-hardness, are evident here, reinforced by the motivation that people really want - and need - to solve these problems in practice!

Review of

What is College For?

The Public Purpose of Higher Education[5]

Editors: Ellen Condliffe Lagemann and Harry Lewis

Publisher: Teachers College Press

$55.00 hardcopy, $28.20 paperback, $17.02 Kindle

Review by

William Gasarch (`gasarch@cs.umd.edu`)

Department of Computer Science

University of Maryland, College Park, College Park, MD

1 Disclosure

I was a Harvard graduate student and Harry Lewis was my advisor. Note that he is a co-editor of the book and also a a co-author of one of the chapter.

2 Overview

What is college for? Let's consider some answers you may have heard.

1. **Vocational:** College trains us for the workplace and gives us a certificate that we can show people to prove we are trained. (Perhaps the certificate is of polynomial length and the employer is a polynomial time verifier.) Many Computer Science and Engineering majors may be in college for this reason.

2. **Vocational but indirect:** College prepares you to go to a professional school, perhaps in law, business, or medicine. The most common pre-law majors are (according to Wikipedia) Political Science, History, English, Psychology, and Criminal Justice. The most common pre-med major (according to a guy named Joe at Yahoo Answers) are Biology, Chemistry, and Biochemistry. I was unable to find out what the most common pre-business school majors are.

3. **Better citizenship:** You go to college to learn things that will help you be a better citizen. If you understand economics and politics then you are a more informed citizen. Courses in civics (how the US government works and its origins) and ethics are appropriate here as well. Or perhaps such concepts should be embedded in many courses.

4. **Explore your creativity:** You go to college to hone yours skills as an artist. I suspect many English and Art majors feel this way. From talking to professors in English I have been surprised to find out, anecdotally, how many English majors are *not* passionate about their field, considering it is not lucrative.

5. **Leaving home:** The best way to move out of your parents house is to go to college.

6. **Staying home:** The best way to keep living with your parents is to go to college.

This book is a collection of essays written to address what college is for. The main focus is the tension between vocational and better citizenship (they do not address the leaving home/staying home divide).

3 Summary of Contents

Renewing the Civic Mission of American Higher Education by Ellen Condliffe Lagemann and Harry Lewis.

If I were to lament to my students that civics is no longer taught they would ask *what is civics?* It begins with the study of the Declaration of Independence and the Constitution, but it also is about fairness and justice in today's society. Such a course would encourage students to think critically about issues armed with the knowledge of history.

Why is it important? As citizens we are asked to decide on important issue and it seems that today's politics is mostly yelling without facts or content. A more knowledgeable electorate would help. The web (which is not mentioned) is double-edged; people can find out more but people can also get stuck in an echo chamber of their own viewpoints unchallenged.

This chapter gives a short history of the decline of the teaching of civics. The following quote shows how important it was in an earlier time. The quote is about the early 1800's.

> For those relatively few Americans who continued their education in college, civic learning was encompassed within the subject of moral philosophy, a capstone course, usually taught by the college president, and required of all graduating seniors.

Some might say: *The college president? Isn't he the guy who begs corporations to give the school money?* Alas, at one time people would care what college presidents had to say on political and moral issues. Now college presidents are fund raisers.

What happened to civics? As more and more sciences got developed and taught, civics got crowded out. Also faculty get more and more specialized and had more of an allegiance to their field than to their college. So it was harder to find someone to teach it or other core courses. In fact, it was hard to have a core (this was discussed more in Harry Lewis's prior book *Excellence Without a Soul*). The chapter does talk about the attempts at Columbia, Chicago, and Harvard to have a core that would include civics and values. The results were complicated and mixed, but, if anything, negative.

So what to do about this? Lagemann and Lewis recommend that civics be put back into the curriculum, not as a one-course-to-check-off, but as a part of many courses. In private email with Harry Lewis he has suggested that when I teach *Theory of Computation* I should bring up the fact that some women who did excellent work in theory were denied tenure solely *because they were women*. When I did this, one of my students insisted that I misspoke, and then asked, "*Don't you mean that they got tenure because they were women?*"

Today's students have no sense of this kind of recent history. Other biographical information could be embedded in courses to illustrate past discrimination against women, African Americans, and LBGT. For other courses there may be other things one can do. They are (wisely) not that specific, but recommend that we consider these issues.

They also recommend that colleges and universities themselves become exemplars of moral behavior to set a good example. Good luck with that.

Appealing as this may be, they also say what the problems would be. Professors are still over-specialized and hard to dictate to. Lewis and Lagemann have boxed themselves in – they do not want civics just to be a one-course-check-box, however, they also realize how hard it is to get professors to do anything to add civics to their courses. And it would be hard to monitor.

This chapter was fascinating for its history and idealism and I am glad they have brought up these issues; however, they simultaneously seem to say how hopeless it is to fix.

Science, Enlightenment, and Intellectual Tensions in Higher Education by Douglas Taylor.

It seems to be a given that contemporary America is anti-intellectual. The "Intelligent Design" movement, the climate change deniers, and the anti-vaccine movement are examples of this. I feel an obligation to give an example *on the left*[6] so here is one: the movement against genetically modified food. Is my feeling that I must give an example on the left similar to when an evolutionary biologist feels he must mention intelligent design for balance? I honestly do not know.

Who is to blame for this? Dr. Taylor points out that colleges are usually not blamed as this viewpoint is thought to have originated in people before college. He disagrees and points out where colleges have outright promoted anti-intellectualism. There are two such places:

1. Some academics have promoted a relativism where we don't know anything except from personal experience. This has lead to absurdities like *Evolution is true for you but not for me.*

2. When colleges recruit people they sometimes do it with the same mentality as a late night TV commercial. This involves outright lying and hence damages credibility and the whole notion of truth.

What to do about this? For the first point he does not suggest much beyond *we should stop doing that*. For the second point he notes that the top colleges could all agree to have more honest and less aggressive marketing. In addition the ridiculous ranking system of US News and World report should be replaced by a more intelligent set of rankings. He didn't mention Goodhart's law, so I will:

When a measure becomes a target it ceases to be a measure.

The notion of a better ranking system or several of them, better in tune with what students need and want, is of course an excellent idea. Could it work? The top schools could make them work and could, in effect, call some sort of truce on aggressive marketing. But would it filter down to less elite schools? Of this I am skeptical.

Liberated Consumers and the Liberal Arts College by Elaine Tuttle Hansen.

Given the last two chapters one wonders if there is *any* college that has clarity of purpose and a reasonable consensus of what excellence in achieving that purpose looks like[7]. There are! The Liberal Arts Colleges!

This chapter praises liberal arts colleges for what they do. To give two key examples: (1) they call on students to think about complex ideas slowly rather than succumb to, what Lisa Simpson calls, our *instant oatmeal society*, and (2) Learning in close-knit communities based on friendship.

The article then goes on to what may be a problem: cost, rankings, narrowness-of-professors (a problem everywhere) and a lack of diversity. But the article is mostly upbeat.

The article does not mention the issue of college-as-job-training and how liberal arts colleges fit into that. Nor do they mention social media which is quite relevant to point (2) above. Nevertheless, the chapter does present us with a model which might be worth aspiring to.

The other 75%: College Education Beyond the Elite by Paul Attewell and David E. Lavin.

The first three chapters of this book, and especially the third chapter, talk about full time residential students at 4-year colleges who have no financial problems (either their parents or financial aid is paying for them). How many students actually fit this model? The title of the chapter might make you think that only 25% are of that type; however, the number is closer to 14% depending on how you count.

[6] The anti-vax movement is actually on the far left and the far right.

[7] That last sentence fragment was copied from this chapter.

Many of the "75%" students are at 2-year colleges trying to get a professional degree (e.g., Nursing) so that they can get a job. Many of them cannot afford even that so they have to work while in school. Some alternate work and school. This chapter's main point is to not leave them out of the discussion. Also note that this chapter relies on hard data which is explained more at length in books the authors have written.

The authors take on some myths about such students. The statistics would seem to indicate that many of the other 75% don't graduate at high rates. But this is a fallacy based on 2-year or 4-year rates (for 2-year and 4-year colleges). Since many are part time or alternating they naturally take longer *through no fault of their own*. If you look at those who graduate in 6 years the statistics look much better. Alas decisions on how much money to spend on financial aid are often made by looking at the incorrect and misleading statistics.

Another myth is that such students go for vocational majors at a higher rate then the traditional student. This is false since even traditional students are also going for vocational majors at a high rate. This is interesting since the myth feeds into some people looking down on community colleges because they are vocational. This is an idiotic reason to look down on them; however, it's still good to debunk the myth.

The authors are strong on more financial aid and on having remedial courses at community colleges. They argue their case well. The trend in the country now is for the government to spend less money; however, that could change. And teaching remedial courses is already happening; the authors defend the practice. So it seems plausible to actually take up the advice given here.

In the last part of this chapter they talk about a civics education for these students and are for it. They do not really discuss why they are for it, though perhaps that is supposed to be obvious – it's good for everyone. This point may be discussed more in their books.

Professional Education: Aligning Knowledge, Expertise, and Public Purpose by William Sullivan.

Law School, Medical School, Business School, Seminaries, are all schools which train people directly for jobs. Or do they? In this chapter the author breaks down types of learning into three categories: (1) Academic, for example, someone in Medical School learning biology, (2) Practical, e.g., someone in Law looking at actual cases, and (3) Professionalism (ethics, common practice of the field), e.g., if you are in a business and you can do something which is good for your company but absolutely awful for society, do you do it?

This chapter looks at how these three aspects have competed over the years and how it looks now. As you may have guessed, academic and practical have eclipsed ethical over the years. Even in Seminary!

What to do about this? At the end of the chapter he gives a description of an excellent course required of seniors at Stern School of Business in New York titled *Professional Responsibility and Leadership*. In this course the students are forced to consider conflicts where they must balance what is good for the firm, for yourself, for your bosses, for your employees, for your community.

Should this course, or something like it, be instituted at other schools? He clearly thinks yes and I agree; however, I wonder if one course is enough. Taking a tip from the first chapter, some of these issues should be embedded in other courses as well.

Could this course, or something like it, be instituted at other schools? He does not address this so I will. If professional schools were not quite so hung up on research (which is the problem universities have) then yes, this kind of course could be developed and work elsewhere. That may be a big *If*.

Graduate Education: The Nerve Center of Higher Education by Catharine Stimpson.

If I want to find out which school any of the authors of the chapters are at, and what positions they hold, I could go to the back of the book. For some of them there is a mild mention of this in their chapter (e.g., Douglas Taylor mentions teaching Evolutionary Biology). This chapter is more personal. Catharine

Stimpson was a Dean at the NYU Graduate School, and also at Rutgers Graduate School, and she uses this personal experience in her chapter.

She first points out that graduate school is not well understood. The general population knows about community colleges, 4-year colleges, professional schools, but not much about graduate school. This is dangerous since we need graduate schools.

She then points out some statistics about graduate school that are interesting. I will share one of them: In 1989 women got 29% of all doctorates in science and enginnering, but in 2009 they got 42%.

Her main point is to look at some of the tradeoffs that Graduate Education has to deal with. One is depth versus breadth. She doesn't restate the famous quote on depth so I will:

Getting a PhD is learning more and more about less and less
until you know everything about nothing.

This is an issue on all levels. How much should a PhD in physics know about the history and sociology of physics and its affect on society? How much should a PhD in string theory know about thermodynamics? How much should a PhD about protons know about electrons? One can get very narrow and academic incentives encourage that.

Another issue is cooperation. A university is supposed to be competitive and strive to be better than other schools; yet a university is also supposed to cooperate with other schools. This issue is brought up in the context of cooperating with other countries since America may be losing its edge.

She ends on an optimistic note which I quote:

> To be sure, more and more wonderful inventions, discoveries, and ideas will emanate from research universities outside of the Unites States. But, to speak NewYorkese, the Unites States is not yet chopped liver. Moreover, at their strongest, our advanced communities of inquiry are morally charged. They can embody what I call *Humane Excellence*. Our morality, when in action, is a global magnet even in the most competitive of conditions.

4 Opinion

This book raises many questions of interest. It should start many discussions and a few bar fights. But the authors, without really intending it, seem also to say that the situation is hopeless. It's not clear where to go from there.

There is one aspect of modern society that is conspicuously absent from this book: technology. With the web students can look up much more than they used to be able to. MOOCS may make the cost of going to college drop drastically. Social media connects up students to a phenomenal degree. All of this must have some affect on the issues being discussed; however, none of this is mentioned. Given that Harry Lewis is a computer scientist who co-authored (with Hal Abelson, Ken Ledeen) an excellent book about the affects of computers on society, (*Blown to Bits*), this omission is surprising.

Nevertheless, this book is an excellent way to start a debate. Here is hoping it inspires not just debate, but action.

Review of[8]

Slicing the Truth:
On the Computability Theoretic
and Reverse Mathematical Analysis
of Combinatorial Principles
by Denis Hirschfeldt
Publisher: World Science Publishing
$37.00 hardcover, 232 pages, Year: 2014

Review by
William Gasarch (`gasarch@cs.umd.edu`)
Department of Computer Science
University of Maryland, College Park, College Park, MD

1 Introduction

When teaching Discrete Math I may pose the students the following problem:

From the theorem that every number factors uniquely into primes, prove that $\sqrt{2}$ is irrational.

A student submitted the following:

1. Every number factors uniquely into primes.
2. It is well known that if p is prime then $\sqrt{p}$ is irrational.
3. 2 is a prime.
4. Hence $\sqrt{2}$ is irrational.

What's wrong with the above proof, given what I intended to ask, is that the basic assumptions that it uses are too strong.

Episodes like the one sketched above are very rare. The class does have the (correct) sense that when I say *use A to prove B*, what I really mean is that the proof should (1) use A, and (2) only use easy math steps.

The program of reverse mathematics formalizes this notion and tries to unify all of mathematics into equivalent theorems. One goal is to examine which theorems *require* nonconstructive proofs and, in a sense, how nonconstructive. We give one example. Let WKL be the weak Konig's lemma: every infinite binary tree has an infinite branch. The following are equivalent: (1) WKL, and (2) $[0, 1]$ is compact (henceforth $COMPACT$).

2 Summary of Contents

The first few chapters of the book discuss the reverse mathematics program due to Steve Simpson and Harvey Friedman. There is a base-system of axioms called RCA_0 and all equivalences are proved there.

For example, one can show $WKL \implies COMPACT$ and $COMPACT \implies WKL$ with all reasoning in RCA_0. From a proof theory prospective RCA_0 is weak, though much of mathematics can be done in it, including elementary number theory and most theorems in finite combinatorics. There are four other systems which form a hierarchy: WKL_0 (RCA_0 plus the weak Konig's lemma), ACA_0, ATR_0, and Π^1_1-CA_0. The R in RCA_0 stands for recursive (computable) – all of the objects whose existence you can prove are computable. WKL_0 lets you do a few other things than what is computable; however, there is a model of WKL_0 where all of the sets are low. The A in ACA_0 stands for Arithmetic. The set of all arithmetic sets is a model for ACA_0. Virtually all of mathematics can be done in ACA_0. ATR_0 and Π^1_1 are not discussed much. The five classes are called *the big five*.

The author gives some examples of theorems in math that fit exactly into one of these five classes and points us to Steve Simpson's book *Subsystems of Second Order Arithmetic* where even more theorems are so classified. It is hard to measure how many or what percent of theorems fit exactly into one of the big five; however, enough do to make the classification interesting.

However, this author goes in an entirely different direction. Ramsey Theory for pairs *is not* equivalent to *any* of these classes. Ramsey Theory for triples and beyond is equivalent to ACA_0.

Chapter 6 is the real heart of the book. In order to explain its contents, we use the following notation:

- RT^n_c is Ramsey theory for n-tuples and c colors.
- RT^∞_c is $(\forall n)[RT^n_c]$.
- RT^n_∞ is $(\forall c)[RT^n_c]$.
- RT is $(\forall c)(\forall n)[RT^n_c]$.

In this chapter the author classifies many RT^n_c in terms of the big five.

1. For all $n \geq 2$ there is a computable 2-coloring of $\binom{\mathsf{N}}{n}$ with no Σ_n homogenous set. Hence for all $n \geq 1$, RT^n_2 is not equivalent to RCA_0 (all we need is that there is no computable homogenous set). The same holds for c-colorings if $c \geq 2$.

2. The usual proofs of Ramsey's theorem show that for all $n \geq 1$, $c \geq 2$, RT^n_c can be proven in ACA_0. This does not preclude the possibility of being provable in a lower system.

3. Every computable 2-coloring of $\binom{\mathsf{N}}{2}$ has a low_2 homogenous set. This is the key ingredient in the proof that RT^2_2 does not imply ACA_0. (The result that RT^2_2 does not imply ACA_0 was first proven by Seetapun; however, the proof in this book using low_2 sets is a newer easier proof.)

4. There exists a computable 2-coloring of $\binom{\mathsf{N}}{3}$ such that every homogenous set computes HALT. Together with point 2 this implies (with some work) that ACA_0 and RT^3_2 are equivalent. This extends to RT^n_c for all $c \geq 2$ and $n \geq 3$.

5. RT is not in ACA_0 but it is in an extension called ACA_0'.

Chapter 7 is about theories being conservative. For example, if ϕ is a sentence in the language of PA then $PA \vdash \phi$ iff $ACA_0 \vdash \phi$. Chapter 8 has nice diagrams summarizing the results in Chapter 6.

Chapter 9 is about weaker versions of Ramsey Theory (there was also some of this in Chapter 6) and how they relate to each other and to RCA_0. It would have been helpful to have a list of all of the variants of Ramsey Theory; hence I provide one here.

1. RT^n_c: For all c-colorings of $\binom{\mathsf{N}}{n}$ there is a homogenous set.

2. $RT^n_{<\infty c}$: $(\forall c)[RT^n_c]$.

3. $RT^{<\infty}_c$: $(\forall n)[RT^n_c]$.

4. RT : $(\forall c)(\forall n)[RT^n_c]$.

5. SRT^2_c: For all stable c-colorings of $\binom{\mathsf{N}}{2}$ there is a homogenous set. A stable coloring $COL : \binom{\mathsf{N}}{2} \rightarrow [c]$ is one such that, for all x, $\lim_{y\rightarrow\infty} COL(x,y)$ exists.

6. COH : For all countable sequences of sets of naturals $R_1, R_2, R_3, \ldots$ there exists an infinite set C (called a *cohesive set*) such that, for all i, $C \subseteq^* R_i$ or $R_i \subseteq^* C$. This follows from RT^2_2.

7. ADS : Every infinite linear ordering has either an infinite ascending subsequence or an infinite descending subsequence.

8. $SADS$: Every stable infinite linear ordering has either an infinite ascending subsequence or an infinite descending subsequence. An ordering is *stable* if it is discrete and every element has either a finite number of elements less than it or greater than it. A nontrivial example is $\omega + \omega^*$ where ω^* is the naturals in reverse order.

9. $CADS$: Every infinite linear ordering has a stable suborder.

10. CAC : Every infinite partial order has either an infinite chain or an infinite anti-chain.

11. $SCAC$: Every stable infinite partial order has either an infinite chain or an infinite anti-chain.

12. $CCAC$: Every infinite stable partial order has an infinite stable suborder.

13. EM (Erdös-Moser): If T is a tournament on N then there is an infinite $A \subseteq \mathsf{N}$ on which T is transitive. A tournament is a directed graph where, for all x, y, exactly one of $R(x,y)$ or $R(y,x)$ holds.

14. $FS(n)$ (Free Set): For all $f : \binom{\mathsf{N}}{n} \rightarrow \mathsf{N}$ there is an infinite $A \subseteq \mathsf{N}$ such that for all $s \in \binom{A}{n}$ either $f(s) \in A$ or $f(s) \in s$.

15. $TS(n)$ (Thin Set): For all $f : \binom{\mathsf{N}}{n} \rightarrow \mathsf{N}$ there is an infinite $A \subseteq \mathsf{N}$ such that $f(\binom{A}{n}) \neq \mathsf{N}$.

16. FIP (Finite Intersection Principle): Every nontrivial family of sets has a maximal subfamily with the finite intersection property. A family of sets satisfies the *finite intersection property* if every finite subfamily has a nonempty intersection.

Chapter 10 is about theorems that are beyond ACA_0. We give two examples:

1. A *well partial order (wpo)* is a partial order that has neither infinite descending sequences or infinite antichains. J. Kruskal showed that the set of trees under embeddability (or under minor) form a wpo. We denote this KTT. H. Friedeman showed that $ATR_0 \not\vdash KTT$. Hence KTT is a natural theorem which requires a rather strong proof system.

2. Laver showed that the set of all countable linear orderings under embedding is a wpo. This is called FRA since it was original Fraisee's conjecture. Shore showed that FRA implies ATR_0, hence it also requires a rather strong proof system.

3 Opinion of the Book

Who can read this book? To read this book you need to already know some computability theory and some Ramsey Theory. Knowing some Reverse Mathematics would also be good; however, that is less necessary. Many theorems are left for the exercises so readers have to do some work themselves.

Who should read this book? The book gathers together in one place most of the theorems known about where Ramsey Theory and some variants of it fit into the Reverse Mathematics framework. The book also discusses many combinatorial principles that the reader may not realize are really Ramsey Theory, but they are!

There are two theorems that I was surprised were not discussed. (1) Mileti has done work on the reverse mathematics of the *Canonical Ramsey Theorem* that does not seem to have been discussed, and (2) Schmerl has done work on the reverse mathematics of the chromatic number of a graph.

However, if you care about the proof strength of Ramsey Theory, this is THE book for you!

4 Opinion of the Field

(Keep in mind that THIS section really is just MY opinion.)

When is asking where theorems fit into the Reverse Math framework interesting?

1. When it leads to new proofs of old theorems. Jockusch's proof that every computable coloring of $\binom{\mathsf{N}}{2}$ has a Π_2-homogenous set can be presented as a different proof of Ramsey Theory without even mentioning Π_2, but noting that the proof is vaguely more constructive.

2. When it leads to interesting computability theory. The construction of a computable coloring that has no Σ_2 homogenous sets is interesting.

3. When you tie together many different theorems as being equivalent. This is similar to NP-completeness where you need to think of SAT and HAM CYCLE as being *the same problem*.

But I do have one criticism. One of the main results in this book is that RT_2^2 is definitely weaker in proof strength than RT_2^3. But the usual proofs of RT_2^2 and RT_2^3 really don't seem any different from a constructive point of view. AH-HA: hence there should be a new proof of RT_2^2 that is more constructive. Or more something. Alas, I've asked people in the field and they can't really point me to this better proof.

Added later: I've discussed this point with Denis Hirschfeldt when I gave him a copy of the review and we may soon have a different proof of Ramsey Theory inspired by these results.

Review[9] of
The Scholar and the State:
In Search of Van der Waerden
by Alexander Soifer
Springer, 2015 $149.00 Hardcover, $119.00 Kindle
approx 450 pages

Review by
William Gasarch (gasarch@cs.umd.edu)
Department of Computer Science
University of Maryland, College Park, College Park, MD

1 Introduction

Alexander Soifer had previously written the book *The Mathematical Coloring Book: Mathematics of Coloring and the Colorful Life of its Creators* which was part math, part history, part memoir (it was written in the first person). I quote my review of that book:

Ordinary math books are not written in the first person; however, this is no ordinary math book! I pity the Library of Congress person who has to classify it. This book contains much math of interest and pointers to more math of interest. All of it has to do with coloring: Coloring the plane (Alexander Soifer's favorite problem), coloring a graph (e.g., the four color theorem), and of course Ramsey Theory. However, the book also has biographies of the people involved and scholarly discussions of who-conjectured-what-when and who-proved-what-when. When I took Calculus the textbook had a 120-word passage about the life of Newton. This book has a 120-page passage about the life of van der Waerden.

Saying that the prior book contained a 120-page passage about the life of van der Waerden (henceforth VDW) was an exaggeration; however, the book under review is a 450-page biography. . . that does not seem quite right. The book does mainly focus on VDW's life, but there are so many profound issues that arise, that I am reluctant to call it a biography.

Why is VDW's life worth writing about? While his contributions to Algebraic Geometry, and the theorem in combinatorics that bears his name, are quite impressive, these are not the reasons. Most of the book is about the time VDW lived in Germany and the time after that. Why is that remarkable? Because he was a Dutch citizen living in Germany from 1933-1945, under the Nazi regime.

2 Summary

The book has 43 chapters. The order is roughly chronological. Here is a rough breakdown, noting that not every chapter is that well defined as to what it's about: there are 5 chapters on VDW before moving to Germany, 3 chapters on his famous book *Algebra*, 3 chapters on his theorem about arithmetic progressions, 14 chapters on his time in Germany, 7 chapters on his time after leaving Germany (in 1945), 4 chapters on Heisenberg, and the rest are on a smattering of topics.

The real heart of the book is VDW's behavior under the Nazi regime and the questions that arise about what a scientist is supposed to do when working in a brutal dictatorship. VDW chose to stay in Germany, as a college professor at a very prestigious university, despite many opportunities to leave. The questions that come up again and again in the book are *why did he stay?*, and *was it the right thing to do?* On the second question there is a resounding answer of NO, though we will discuss that in the next paragraph. As to why he stayed, a variety of reasons are given by both VDW and others. Was he naive about the Nazis? (No.) Did he think Germany would be the best place to do mathematics? (Yes.) Did he want to be well positioned no matter who won the war? (I see his delaying deciding on an appointment in Holland in the 1940's indicative of this view.) VDW claims there are two Germanys and that the Nazis are an aberration of the real Germany which he hoped would return after Germany lost the war. It was that Germany he was serving. Frankly, this seems like a hard argument to make rigorous.

Why was it wrong to be in Nazi Germany? Note that he was working in pure math so his research did not contribute to the war effort. But also note that he taught students who did contribute. What may be a more important issue: he gave that regime an air of legitimacy.

Did it affect his later career? There are a few different issues about this. When asked to defend his actions his answers were self-serving, insensitive, incomplete, and sometimes fictional. He also never seemed to say his actions were wrong. I would like to say *if he had come clean and apologized from the very beginning he would have been better off* but this is not clear. The notion of *admit what you did early and control the narrative* only works if you really do have a viable defense, which I do not think VDW did. In the end VDW did get a job in Zurich which he kept for the rest of his life, so, sad to say, his approach may have worked.

What are we to think of him? While he was clearly not a Nazi (and early on he objected to Jews being dismissed from the university), he was comfortably employed and respected in Germany and had the support of Nazis (he stopped criticizing the regime very early on). If Germany had won the war he surely would have been a professor there, and that bothers me.

The chapters on Heisenberg pose a different question. Heisenberg worked on the atom bomb for Nazi Germany. He claims he wasn't working that hard on it. There is also the question of whether he was on the right track. His actions, and his defenses, are both even worse than VDW's. But again, he got a job and a life, so, sad to say, his approach may have worked.

The book also covers the following:

1. The history of VDW's Algebra book. My impression from the book under review is that Artin should have been a co-author. Also, it seems as thought VDW blocked competing books from being published. While this is hardly comparable to lending the Nazi regime credibility it may give insight into the man's moral character.

2. The history of VDW's theorem. This raises questions about who should get credit. VDW's behavior here is fine. While it looks like (again!) Artin should be a co-author, when it was published VDW did not think much of the theorem so this is not really a slight. However, Soifer thinks the poser of the question (credited to Baudet, but Soifer gives more evidence then I wanted to read for Schur) should also get credit and maybe be co-authors. While I see his point of view, and it might double my paper count (I am much more of a poser of problems than a solver), this system sounds complicated. Another issue: the theorem got popularized from a book *Three Pearls of Number Theory* by Khinchin. If Khinchen's book was not published then VDW's theorem might still be relatively unknown. Unlike Ramsey's Theorem, VDW's theorem is not used that much to prove other things, so it might not have been rediscovered. It's important to realize that when pure math is not tied to any application it may be somewhat arbitrary what gets out there.

3 Other Questions the Book Raises

The book made me think of the following questions.

1. What should you do if you are stuck in a brutal regime but you yourself are safe?

2. What should you do if you are stuck in a brutal regime but you yourself are safe, and you are asked to help the regime (e.g., teach mathematics at the university, or build an atom bomb)?

3. Why does Nazi Germany get all the attention of being a bad regime when others were also bad? Speculation:

   (a) The holocaust was a genocide that killed 6,000,000 Jews (and 4,000,000 other people for a variety of other reasons). A *genocide* is an intentional killing of a people for no other reason than they are of that people. Stalin killed more people, but it was not a genocide (though it was close to one). Other nations had genocides, but they didn't kill anywhere near 6,000,000. Nazi Germany carried out the largest genocide ever.

   (b) They started WW II and (worst of all for their history) they lost. The winners get to write the textbooks and decide what is and is not a war crime.

   (c) Germany was seen as being part of the first world. The soft bigotry of low expectations[10] makes us yawn when we hear that some country or tribe in Africa is committing genocide on another. Afterwards we say *never again* again. Being part of the so called first world Germany is judged on a high standard and did far worse than anything Africa's ever seen.

4. If Professor X is really good at his job and you want to hire him, do you care about his past? Realize that nobody thought VDW was a Nazi, so the issue is not that his beliefs may infiltrate the students. Can the past be put in the past? For Americans absolutely yes— America hired actual Nazi rocket scientists like Wernher von Braun (the book includes the lyrics to the Tom Lehrer song about him). People who I've talked to about the book are sometimes baffled – if VDW wants a job at your school, then the fact that he happened to be in Germany during the Nazi era is regrettable but he never had those views then and doesn't have them now, so of course you hire him!

5. With regard to the last point, VDW was of course brilliant. For someone much less brilliant would a school not hire them and then feel good about themselves? There are two variables here – how brilliant is the job candidate, and how bad is their past. There may be some brilliant vs. bad-past tradeoff. This may also depend on who else you've hired in the recent past, so it may be a time-dependent tradeoff, maybe a stochastic process.

6. The Nazis judged science partially on whether or not Jews worked on it. The Nazis attitude caused some Jewish scientists to leave, and directly killed others. The death camps were a drain on resources. All of this contributed to them losing the war. Had they only wanted to conquer territories in Europe and used all the people they had towards this goal, and had absolutly no distinction between Jews and others, would they have won the war? More generally, when a society bans certain people from certain jobs this seems to always be a bad idea. (A more recent example was, in America, firing Arabic translators because they were gay.) Why do countries (or people or sports teams or . . .) do this when it is clearly against their interests?

[10] This phrase was due to Michael Gerson, a speechwriter for George W Bush. Since the book under review is about moral choices one makes I want to make sure I credit people fairly.

4 Opinion

For most books in this column the question arises *Who can read this book?* For example, not everyone can read *Canonical Ramsey Theory on Polish Spaces* which is a real book! Honest! For the book under review there is very little barrier to entry. There is very little math in it and the math in it is not the point anyway.

Who should read this book? Anyone who is interested in history and the profound moral questions that arise from its study. I would like to think that means everyone who is reading this review.