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. We review four collections of papers by Donald Knuth. There is a large variety of types of papers in all four collections: long, short, published, unpublished, "serious" and "fun", though the last two categories overlap quite a bit. The titles are self explanatory.

   (a) **Selected Papers on Discrete Mathematics** by Donald E. Knuth. Review by Daniel Apon.

   (b) **Selected Papers on Design of Algorithms** by Donald E. Knuth. Review by Daniel Apon

   (c) **Selected Papers on Fun & Games** by Donald E. Knuth. Review by William Gasarch.

   (d) **Companion to the Papers of Donald Knuth** by Donald E. Knuth. Review by William Gasarch.

2. We review jointly four books from the Bolyai Society of Math Studies. The books in this series are usually collections of article in combinatorics that arise from a conference or workshop. This is the case for the four we review. The articles here vary tremendously in terms of length and if they include proofs. Most of the articles are surveys, not original work. The joint review if by William Gasarch.

   (a) **Horizons of Combinatorics (Conference on Combinatorics** Edited by Ervin Györi, Gyula Katona, László Lovász.

   (b) **Building Bridges (In honor of László Lovász's 60th birthday-Vol 1)** Edited by Martin Grötschel and Gyula Katona.

   (c) **Fete of Combinatorics and Computer Science (In honor of László Lovász's 60th birthday-Vol 2)** Edited by Gyula Katona, Alexander Schrijver, and Tamás.

   (d) **Erdős Centennial (In honor of Paul Erdős's 100th birthday)** Edited by László Lovász, Imre Ruzsa, Vera Sós.

3. **Bayesian Reasoning and Machine Learning** by David Barber Review by Matthias Gallé. This book tries to present a unified view of Machine Learning using as much as possible a Bayesian approach. It uses *graphical models* as an underlying general representation throughout the book.

4. **Integrated Methods for Optimization** by John Hooker. Review by S.V.Nagaraj. This book integrates key concepts from mathematical programming and constraint programming to solve optimization problems.

5. **Programming with Higher-Order Logic** by Dale Miller and Gopalan Nadathur. Review by Vaishak Belle. To summarize this book in 2 sentences I quote the review: *This book presents deep techniques that take the many ideas at the heart of PROLOG, and extend its power and expressivity by elegantly combining these ideas with a simply typed version of higher-order logic. The end result is a rich programming language that benefits from the paradigms of higher-order logic and logic programming.*

6. **People, Problems, and Proofs** by Richard Lipton and Ken Regan. Review by William Gasarch. Lipton and Regan run a computer science theory blog called *Godel's Lost Letter*. This book is a collection of articles that are either blog posts or modified versions of blog posts or inspired by a set of blog posts. As such it has a wide range of topics. The review is itself a fictional blog.

7. **Who's Bigger? Where Historical Figures *Really* Rank** by Steven Skiena and Charles B. Ward. Review by Nicholas Mattei. Say you are in a bar and want to argue who was more famous Charles Darwin or Abraham Lincoln (both born Feb 12, 1809). You could get into a fist fight OR you could figure out a rigorous criteria of fame and measure it. Or you could look at this book.

BOOKS I NEED REVIEWED FOR SIGACT NEWS COLUMN

Algorithms

1. *Greedy Approximation* by Temlyakov
2. *Algorithmics of matching under preferences* By Manlove.
3. *Data clustering: Algorithms and Applications* Edited by Aggarawal and Reddy.
4. *The LLL Algorithm*. Edited by Nguyen and Vallee.
5. *Jewels of Stringology Text Algorithms* by Maxime Crochemor and Wojciech Rytter.
6. *Tractability: Practical approach to Hard Problems* Edited by Bordeaux, Hamadi, Kohli.

Misc Computer Science

1. *Introduction to reversible computing* by Perumalla.
2. *Distributed Computing through combinatorial topology* by Herlihy, Kozlov, Rajsbaum.
3. *Selected Papers on Computer Languages* by Donald Knuth.
4. *Algebraic Geometry Modelling in Information Theory* Edited by Edgar Moro.

Misc Math

1. *The Golden Ratio and Fibonacci Numbers* by Richard Dunlap.
2. *Probability Theory: An Analytic View* by Daniel Stroock.
3. *Proof Analysis: A Contribution to Hilbert's Last Problem* by Negri and Von Plato.
4. *Towers of Hanoi— Myths and Maths* by Hinz, Klavzar, Milutinovic, Petr.
5. *A Mathematical Orchard: Problems and Solutions* by Krusemeyer, Gilbert, Larson.
6. *Mathematics Galore! The first five years of the St. Marks Institue of Mathematics* by Tanton.
7. *Six sources of collapse: A mathematician's perspective on how things can fall apart in the blink of an eye* by Hadlock.
8. *Classic papers in combinatorics* edited by Ira Gessel and Gian-Carlo Rota.
9. *The king of infinite space: Euclid and his elements* by David Erlinski.
10. *Archimedes: What did he do besides cry eureka?* by Sherman Stein.

Review of [2]
Selected Papers on Discrete Mathematics
by Donald E. Knuth
Center for the Study of Language and Information (CSLI), 2003

812 pages, Softcover – New/Used, from \$20.00, Amazon
812 pages, Hardcover – New/Used, from \$33.00, Amazon
812 pages, Cloth – New, \$80.00, University of Chicago Press

Reviewed by Daniel Apon, Dept. of CS, Univ. of MD
dapon@cs.umd.edu

1 Introduction

Selected Papers on Discrete Mathematics is a 2003 collection of *forty-one* of Knuth's papers on discrete mathematics, bringing together "almost everything [he] has written about mathematical topics during the past four decades." This is the sixth entry in a nine-volume series archiving Knuth's published papers. The full series in order is: (i) *Literate Programming*, (ii) *Selected Papers on Computer Science*, (iii) *Digital Typography*, (iv) *Selected Papers on Analysis of Algorithms*, (v) *Selected Papers on Computer Languages*, (vi) the current book, (vii) *Selected Papers on Design of Algorithms*, (viii) *Selected Papers on Fun and Games*, and (ix) *Companion to the Papers of Donald Knuth*.

While not designed as a textbook, this book is *huge* and covers a number of diverse topics in great detail. Measuring nearly 800 pages before the index, the book contains something on almost every fundamental area of discrete mathematics: mathematical notation, permutations, partitions, identities, recurrences, combinatorial designs, matrix theory, number theory, graph theory, probability theory, and a dash of algebra. As if to emphasize this point, the final two papers in the book (both dedicated to Paul Erdős) comprise an intensive, nearly 200-page study of the properties of "randomly-generated (evolving) graphs" first initiated in a classic 1959 paper of Erdős and Rényi.

In the sequel, I give a chapter-by-chapter summary of the book. For lack of space, I can only briefly skim over a few chapters, but I have tried to go into more depth on some of the parts I found most interesting and rewarding. Afterward, I conclude with my opinion of the book.

2 Summary

Chapter 1 begins the book, perhaps symbolically, with a 1965 paper of Knuth and Hall's observing that computers may (in fact!) be useful for solving combinatorial problems by searching through the possible witnesses. The chapter introduces the basic backtracking technique and discusses applications to constructing latin squares and projective planes.

Chapters 2 and 3 discuss aspects of notation that Knuth would like to promote. One of these is *Iverson's convention*, which lets us write any sum as an infinite sum without limits: if $P(k)$ is any Boolean property of the integer k, then we could write $\sum_{P(k)} f(x)$ as $\sum_k f(x)[P(x)]$. For example, the sum-of-squares for all integers k in some set S could be written as $\sum_k k^2[k \in S]$.

Chapters 4 and 5 discuss mathematical results from the early 17th century, due to Johann Faulhaber and Thomas Harriot. (There's even a picture of an engraving of Johann in 1630!) The first of the two proves a result of Johann's on sums of powers (for which Johann did not publish a proof). Namely, the r-times-repeated summation of $1^m, 2^m, ..., n^m$ is a polynomial in $n(n + r)$ times the r-fold sum of $1, 2, ..., n$, when m is a positive odd number. Of particular interest – Knuth's goal was to prove this fact only using techniques available in the early 17th century (and thus, to Johann).

A particular favorite of mine was Chapter 6, which gives a self-contained exposition of G.P. Egorychev's 1980 proof of a conjecture first made by van der Waerden in 1926. VDW's conjecture (now a theorem) states that the permanent of an $n \times n$ doubly stochastic matrix is never less than $n!/n^n$, which implies the matrix where all entries are equal to $1/n$ gives the minimum permanent for this class of matrices. The proof is given from only elementary principles, with the exception of a single, simple fact from analysis in the proof of one lemma, for which Knuth spends a little extra time giving intuition.

Having examined the permanent, Knuth turns to exploring the Pfaffian of an array of numbers in Chapter 7. Let $A = (a_{i,j})$ be a $2n \times 2n$ skew-symmetric matrix. The *Pfaffian of A* is defined as

$$\mathrm{pf}(A) = \frac{1}{2^n n!} \sum_{\sigma \in S_{2n}} \mathrm{sign}(\sigma) \prod_{i=1}^{n} a_{\sigma(2i-1),\sigma(2i)}$$

where S_{2n} is the symmetric group; that is, the set of all permutations on $2n$ elements (whose group operation is permutation composition). For the curious reader: Knuth argues that the Pfaffian is "more fundamental" than the determinant. To quote him, "a determinant is just the bipartite special case of a Pfaffian." In any case, Chapter 7 gives an identity for the product of $\mathrm{pf}(A) \cdot \mathrm{pf}(B)$ where B is a submatrix of A.

Chapter 8 discusses the Sandwich Theorem. It is NP-complete to compute $\omega(G)$, the size of the largest clique in a graph G, and it is NP-complete to compute $\chi(G)$, the minimum number of colors needed to color the vertices of G. But *in polynomial time*, it is nonetheless possible to compute a real number — the Lovász number $\vartheta(G)$— that is "sandwiched" in between these two:

$$\omega(G) \leq \vartheta(G) \leq \chi(G).$$

The chapter is nearly 50 pages long, and written in an expository format, building up numerous aspects of the theory surrounding the Sandwich Theorem and Lovász's function.

Chapters 9 through 14 explore various relationships between matrices, graphs, and trees. In Chapter 13, we consider the following problem: Given a set of vertices v that each been assigned a color $c_v \in [m]$ with n_j vertices of color $j \in [m]$, how many oriented trees on these vertices with designated root exist, subject to constraints of the form "no edge connects vertices colored i and j?" While the general statement requires introducing prohibitively much notation for this review, here is an exact statement for a simple case: Suppose we have $|A|$ vertices of color a, which have directed edges only onto vertices of color b and c, and $|B|$ vertices of color b, which have directed edges only onto vertices of color a and c, and $|C|$ vertices of color c, which have directed edges only onto vertices of color c and the root. How many oriented trees of this type exist? Exactly

$$(|B| + |C|)^{|A|-1}(|A| + |C|)^{|B|-1}(|C| + |1|)^{|C|-1}(|A||C| + |B||C| + |C|^2).$$

Chapters 15 and 16 discuss properties of polynomials. The first considers families of convolution polynomials that arise as coefficients when a power series is raised to the power x. These are families of polynomials $F_0(x), F_1(x), ...$ where $F_n(x)$ has degree $\leq n$ and

$$F_n(x + y) = F_n(x)F_0(y) + F_{n-1}(x)F_1(y) + \cdots + F_0(x)F_n(y)$$

hods for all x and y and for all $n \geq 0$. For example, setting $F_n(x) = x^n/n!$ yields the binomial theorem for integer exponents.

Chapters 17 and 18 are fairly light. Chapter 17 constructs an equidistributed sequence of real numbers, and Chapter 18 presents a construction (from Knuth's college days) of a base-$2i$ number system and implements addition and multiplication.

Chapters 19 through 21 cover finite fields and semifields. Chapters 22 and 23 go further into topics in algebra. An interesting twist here is Knuth's construction of Huffman's algorithm (for finding minimum redundancy codes) from abstract algebra. In particular, he defines a *Huffman algebra* $(A, <, \circ)$, which is a linearly ordered set A with a binary operator $\circ$. Huffman's algorithm, in this context, is given as input elements $a_1, ..., a_n$ of A and builds an expression (over the algebra) out of the elements by repeatedly applying $\circ$ to the smallest and second-smallest elements until only a single expression remains. The expression yields the corresponding Huffman code.

Chapters 24 and 25 discuss how graphs are constructable from various components. An example theorem is that the complement of a transitive closure of the complement of a transitive relation is, itself, transitive.

Chapters 26 and 27 explore the theory of matroids. The first of these, in particular, gives pseudocode from one of Knuth's attempts to generate random matroids for experimental analysis. This is also the only piece of mathematical writing that I've seen use the phrase "homomorphic image of a free erection" with a serious face.

If $x_1, x_2, ..., x_n$ are real numbers whose sum is zero, then we can always find some permutation $p(1)p(2)...p(n)$ such that each of the partial sums $x_{p(1)} + \cdots + x_{p(j)}$ is nonnegative, for $1 \leq j \leq n$. Daniel Kleitman conjectured that the number of such permutations is always at most $2n!/(n+2)$, if the x's are nonzero. Chapter 28 presents a proof of Kleitman's conjecture, but strengthened to be parameterized by the number of positive and negative x_i.

Chapter 29 presents an efficient construction of a balanced code that optimizes serial encoding and decoding. The scheme is extended to optimizations for parallel encoding and decoding.

Chapters 30 through 36 present numerous results on partitions. Here are a few examples:

In Chapter 30, the following problem (known as the Knowlton-Graham partition problem) is considered: Two parties want to communicate over a long cable containing n indistinguishable wires. They want to label the wires consistently so that both ends of each wire receive the same label, perhaps by locally re-wiring connections. How can they do this? (Using a special partition!)

A *plane partition* of n is a two-dimensional array of nonnegative integers $\{n_{i,j}\}$ for $i, j \geq 1$, for which $\sum_{i,j} n_{i,j} = n$ and the rows and columns are in nonincreasing order:

$$n_{i,j} \geq n_{(i+1),j} \text{ and } n_{i,j} \geq n_{i,(j+1)}.$$

Chapter 32 gives generating functions for various classes of plane partitions.

Chapter 34 obtains a generalization of Jacobi's triple product identity, by analogy to a similar generalization for Euler's partition identity, which can written

$$\prod_{j=1}^{\infty}(1 - q^{2j-1}z)(1 - q^{2j-1}z^{-1})(1 - q^{2j}) = \sum_{k=-\infty}^{\infty}(-1)^k q^{k^2} z^k.$$

Chapter 36 studies coefficients that arise in the preceding study of partitions. Interestingly (at least to me), there are connections to cyclotomic polynomials and determinants of semi-lattices from geometry of numbers.

Chapters 37, 38, and 39 discuss recurrence relations. For example, Chapter 38 investigates solutions of the general recurrence

$$M(0) = g(0), \qquad M(n+1) = g(n+1) + \min_{0 \le k \le n} \left(\alpha M(k) + \beta M(n-k)\right),$$

for various choices of α, β, and $g(n)$. In many cases, $M(n)$ is shown to be a convex function (and thus much more efficiently computable).

Finally, there are the mammoth chapters – 40 and 41 – on *evolving graphs*. Nothing I say in a few sentences can quite do two hundred pages of work justice, but here is the general idea: Begin with n disconnected points. Now add edges between the points at random. Then...

How long is the expected length of the first cycle that appears? Asymptotically, what is the average length of the kth cycle that appears? What is the probability that the graph has a component with more than one cycle at the point that the number of edges passes $n/2$? When such a graph has approximately $n/2$ edges, what is the probability that the graph consists entirely of trees, unicyclic components, and bicyclic components as $n \to \infty$?

Knuth (and co-authors) obtain numerous high-quality estimates in a uniform manner, and obtain closed-form expressions of multiple constants of interest.

3 Opinion

As with all of Knuth's books, this book is written authoritatively and eloquently. Knuth's presentation is a testament to clean, formal writing; his standard format is to begin with a few paragraphs describing the topic, and then present a list clear lemmas and their proofs, followed by a main theorem and a proof connecting each of the lemmas together. Interspersed between each of the lemmas are a few paragraphs describing the purpose of each subsequent lemma. The notation and typography are beautiful and succinct. As mentioned above, some of the chapters even explicitly focus on *which notation is the most clear and useful*. I found it a delight to read.

Review of [3]
Selected Papers on Design of Algorithms
by Donald E. Knuth
Center for the Study of Language and Information (CSLI), 2010

453 pages, Softcover – New, $31.11, Amazon Prime
453 pages, Hardcover – New/Used, ~$150.00, Amazon
453 pages, Cloth – New, $75.00, University of Chicago Press

Reviewed by Daniel Apon, Dept. of CS, Univ. of MD
dapon@cs.umd.edu

1 Introduction

Selected Papers on Design of Algorithms is a compilation of twenty-seven of Knuth's technical papers focusing on the design of new algorithms. This is the seventh entry in a nine-volume series archiving Knuth's published papers. The full series in order is: (i) *Literate Programming*, (ii) *Selected Papers on Computer Science*, (iii) *Digital Typography*, (iv) *Selected Papers on Analysis of Algorithms*, (v) *Selected Papers on Computer Languages*, (vi) *Selected Papers on Discrete Mathematics*, (vii) the current book, (viii) *Selected Papers on Fun and Games*, and (ix) *Companion to the Papers of Donald Knuth*.

The papers are revised for errors, and cover a range of algorithmic topics – from combinatorics and optimization, to algebra and theorem proving, to managing error in numerical computations. To quote from the back cover of the book:

> "Nearly thirty of Knuth's classic papers on the subject are collected in this book, brought up to date with extensive revisions and notes on subsequent developments. Many of these algorithms have seen wide use — for example, Knuth's algorithm for optimum search trees, the Faller–Gallager–Knuth algorithm for adaptive Huffman coding, the Knuth–Morris–Pratt algorithm for pattern matching, the Dijkstra–Knuth algorithm for optimum expressions, and the Knuth–Bendix algorithm for deducing the consequences of axioms. Others are pedagogically important, helping students to learn how to design new algorithms for new tasks. One or two are significant historically, as they show how things were done in computing's early days. All are found here, together with more than 40 newly created illustrations."

The book is primarily an "archival" work. That is, the majority of the book proper is a sequence of (significant) papers, though on disparate topics. It is *not* particularly intended to be read from cover-to-cover: An effort was made to group papers into loosely related themes, but the book lacks a "coherent narrative" when read in order. As a consequence, not everyone will like every paper, but on the other hand, most people will find something to enjoy in the 27 papers. Summaries of some of the chapters I enjoyed the most appear below.

2 Summary

The first chapter of the book is distinct; it is not about an algorithm, but rather about the life and work of Knuth's long-time colleague and co-author, Robert W Floyd (1936-2001). If this name doesn't ring a bell, recall at least the Floyd-Warshall algorithm for All-Pairs-Shortest-Path from your undergraduate Algorithms course! This opening chapter chronicles the professional and personal relationship between Knuth and Floyd, originating through correspondence by mail in 1962 (*mail* – not *email!*), and developing further after Knuth and Floyd both joined Stanford's computer science department in 1967 and 1968, respectively.

Chapter 5 is on dynamic Huffman coding. A Huffman tree implements a minimal-weight prefix code, which can be used as the core of a one-pass algorithm for file compression. For a letter a_j in a file, a_j's weight w_j is the number of occurrences of a_j. Huffman coding uses this weight to optimize the construction of the underlying tree. But what if the file is *streamed*? Each time a new character a_j of the file is seen, the corresponding weight w_j is incremented. This necessitates *efficiently updating* the structure of the underlying tree each time a new character arrives. This is accomplished by a clever swapping of sub-trees after each update so that the optimality of the tree's weights are maintained.

Chapter 9 covers the well-known Knuth-Morris-Pratt algorithm for fast pattern matching in strings. Here, we have a text file of n characters and a string of $m < n$ characters. The goal is to find the string within the text file, if it exists. Naïvely, this takes $O(mn)$ time, but the KMP algorithm does it in $O(m + n)$ using clever preprocessing.

Chapter 10 gives algorithms for addition machines – that is, machines that can read/write/copy numbers to and from registers as well as evaluate ≥, +, and -. The chapter begins with a simple $O(\log^2(x/y))$ time algorithm for computing $x \bmod y$. Surprisingly, this can be improved to $O(\log(x/y))$ by using the (unique!) representation of numbers as sums of Fibonacci numbers with pairwise difference ≥ 2, instead of their binary representations. The Fibonacci-based techniques are extended to efficient algorithms for multiplication, division, computing $\gcd(x, y)$, implementing stacks, sorting, and computing $x^y \bmod z$.

Chapter 11, entitled "A simple program who's proof isn't," is about the subroutine for converting between decimal fractions and fixed-point binary in TEX. This is an interesting example of a general principle arising from careful examination of a finite problem. Internally, TEX represents numbers as integer multiples of 2^{-16}, and it must frequently convert a given decimal fraction with k digits of precision, written $.d_1d_2...d_k$, into this format while rounding correctly. The straightforward procedure is to compute

$$N = 10^k \left(2^{16} \sum_{j=1}^{k} d_j/10^j + 1/2 \right)$$

then letting the output be $\lfloor N/10^k \rfloor$. Unfortunately, since k may be arbitrarily large, the values N and 10^k may be too large for the computer's hardware to support. Digging into the details of the computation, however, reveals a procedure for the task that only requires TEX to maintain an array of 17 digits for the input and computes intermediate values no larger than $1,310,720 \ll 10^k$. A generalization to arbitrary combinations of precision is discussed at the end.

Chapter 23 is "Evading the drift in floating point addition" and begins by considering the following simple form of rounding after floating-point arithmetic: Suppose we are storing numbers to three significant digits and that rounding is performed by adding 5 to the fourth significant digit, then truncating to three significant digits. This is rounding to the nearest number, and breaking ties by rounding up. If $x = 400$ and $y = 3.5$, then $x + y = 404$, $(x + y) - y = 401$, $((x + y) - y) + y = 405$, and so on. This *drift* towards $+\infty$ is due to the bias of always rounding up in the case of ties. Ideally we would like to always have

$((\cdots((x+y)-y)\cdots+y)-y) = x$, regardless of how many times we add-then-subtract the same number. The solution is to analyze a list of 30+ different cases covering the possible, arithmetic relationships between the numbers being added or subtracted. Sometimes we round ties up, sometimes we round ties down. On the plus side, the resulting nested depth of **if-else**'s in code is only 5 or 6, so the more complicated rounding scheme can be implemented to run efficiently and avoid the drift.

3 Opinion

Selected Papers on Design of Algorithms bears Knuth's usual eloquence in writing. The algorithms and proofs in each chapter are presented cleanly, and pseudocode for implementing them accompanies most of the algorithms. Part of the real *charm* of this collection comes from the historical notes interspersed throughout the book. These take the form of either additional commentary attached to the end of a paper explaining how the technical topic has progressed since the writing of the paper (explaining which open problems have been solved, which haven't, and how the newer results were obtained), or a story of how the original result was obtained.

A particularly nice example from the book is the sequence of events leading up to the Knuth-Morris-Pratt algorithms for fast pattern matching in strings. One of the authors, J. H. Morris, originally invented the method while implementing a text editor for the CDC 6400 computer during the summer of 1969. His code, however, was rather complicated, and after several months other implementers of the system had "fixed" his routine into a complete mess. Independently a year later, Knuth and Pratt developed a string matching algorithm, working from a theorem of Cook that showed any language recognizable by a two-way deterministic pushdown automata in *any* amount of time, can be recognized on a RAM in $O(n)$ time. Later, Pratt described the algorithm to Morris, who recognized it as the same as his own – and the Knuth-Morris-Pratt algorithm was born.

Gems like these are probably the best reason to own the book. On the other hand, I would *not* recommend this book as a tool for research or as the primary text for a classroom. It could be used as supplemental material for an algorithms course – but in these cases, it is simpler (and cheaper) to find the specific paper you want students to read, and distribute that by itself. Alternatively, it could find some use for self-study – particularly in illuminating not just technical material (some of which is dated at this point), but *how* the human beings who developed the technical material came up with it in the first place. And of course, this book is another must-have for any Knuth-ophile who loves their copy of *The Art of Computer Programming*.

Review [4] of
Selected Papers on Fun & Games
by Donald E. Knuth
Center for the Study of Language and Information
742 pages, SOFTCOVER, \$37.00, 2011
Review by William Gasarch gasarch@cs.umd.edu

1 Introduction

Recently Donald Knuth had his publisher send me five books of his Selected papers to review for my column. Knuth said that this book, *Selected Papers on Fun & Games*, was his favorite. After reading every word of its 700+ pages I can see why. The book format and the topic allows him to put in whatever he finds interesting. By contrast his book *Selected Papers on Discrete Math* has to have all papers on . . . Discrete Math.

In the preface he says *I've never been able to see the boundary between scientific research and game playing.* This book also blurs the distinction between recreational and serious research, between math and computer science, and between light and heavy reading. By the latter I mean that many of the chapters are easy to read— if you skip details.

Many of the topics in the book come from things either he noticed or people told him leading to a math or CS problem of interest. Hence he is in many of the chapters as a character. I will do the same: here is a story that happened to ME but it's the KIND OF THING that happens to him quite often.

I was having dinner with my darling and two other couples. We were sitting at a rectangular table and I noticed that I sat across from my darling while the other two couples sat next to each other. I then thought: If there are n couples sitting down to eat at a rectangular table how many ways can they do it if everyone sits next to or across from their darling. I solved it and I invite my readers to do the same.

There are $7^2 = 49$ papers in this book. Its hard to find a theme that ties them together except *Stuff that Donald Knuth thinks is fun and hopefully you will too!*. That makes the book hard to summarize. Hence I will just comment on some of the chapters. I try to pick a representative sample.

One of the best features: many of the chapters have addenda that were added just recently. This is excellent since an old article needs some commentary about *what happened next?*.

2 Summary

Chapter 1 is a reprint of Donald Knuth's first publication. Was it in American Math Monthly? Mathematics Magazine? Fibonacci Quarterly? No. It was in MAD magazine! Its a satirical (or is it?) way of doing weights and measures. Chapter 2,3,4, and 5 do not contain any real math or computer science. Chapter 6, *The Complexity of Writing Songs* is borderline since it uses O-notation.

Chapter 7, *TPK in INTERCAL* is about a rather odd challenge— writing programs in the language INTERCAL. This language has very few operators and what it does have is not the usual arithmetic operation. In this chapter he discusses writing a particular program in it. Challenging!

Chapter 8, *Math Ace: The Plot Thickens* considers the following. We all know how to graph such equations as $x = y^2$. But what about $|x + y| \leq 1$? This would not be a curve but a region in the plane. Which regions of the plane can you achieve this way? He has some rather complicated equations that yield very interesting shapes.

Chapter 12, *The Gamov-Stern Elevator Problem* takes a real world problem and solves it. Essentially, Gamov and Stern both thought that when they were waiting for an elevator to go down they would always have the next elevator going up, and vice versa. Let

$$p = \frac{\text{distance from our floor to the bottom floor}}{\text{distance from top floor to bottom floor}}$$

Assuming that the elevators start at a random position and cycle up and down continuously what is the probability (in terms of p) that the next elevator that gets to your floor will be going down? Knuth first solves this using hard math (Integral signs! Lots of them!). He notes that the answer is simple and writes … *our formula is so simple it suggests that there must be a much simpler way to derive it*. He then does indeed present a much more elegant solution.

Chapters 13,14, and 15 are on Fibonacci Numbers; however, I suspect they contain much material *you* do not know (they contained much material *I* did not know). Here is a tidbit: (1) Every number can be written as a sum of Fibonacci numbers, (2) If you do not allow the use of two consecutive Fibonacci numbers then this representation is unique.

Chapter 17 is titled *Mathematical Vanity Plates* but it has much information on vanity plates in general (it's 40 pages long!). His interest in vanity plates was sparked by seeing the license plate *H65 537* and wondering if the driver knew that his plate was the fourth Fermat Prime. This chapter is very indicative of Knuth's style: something in the real world inspires him and then he learns *everything* about it.

Chapter 23, *Basketball's Electronic Coach* is about a program he wrote as an undergraduate that helped rate how good the basketball players were and hence gave some idea of who to play when. This is impressive since (1) it was 1956-1960, way before Bill James and the Moneyball revolution, and (2) basketball would seem to be much harder than baseball to analyze this way. The article quotes various news sources about what he did, including Walter Cronkite. Knuth himself is skeptical that it really worked, though the team did improve quite a bit. We also learn that Knuth himself is 6 feet 5 inches, which makes me wonder if he would have done even more good by being on the court. Perhaps his program told him otherwise.

Chapters 40,41,42,43 are all on knight tours. On a chessboard (or an $a \times b$ grid) a knight's tour is a way for a knight to visit every square on the board. What if you insist that the tour not cross itself (Chapter 40)? What if you insist that the tour makes a beautiful Celtic pattern (a what? Chapter 41)? What if you insist the knight's tour be skinny (Chapter 42)? How do you generalize this concept (Chapter 43)?

There are articles on music, math, music&math, word games, other games, a candy bar contest that a young Donald Knuth entered, and … I want to say *etc* but it's not clear that I have established a pattern. This makes it a delight to read but hard to write a coherent review of.

3 Opinion

As noted above, the book is a joy to read. The diversity of topics is overall a good thing, though there may be a topic you do not care for. As an extreme example he has a chapter that has the code and commentary on it for the game *Adventure*. My darling, who is a Software Engineering and has spent (wasted?) many an hour playing Adventure, is very interested in that. Myself… less so.

I predict that

$$(\forall p \in P)(\exists C \subseteq CH)[|C| \geq 30 \land (\forall c \in C)[L(p, c)]]$$

where

- P is the set of people who read this column.
- CH is the set of chapters of the book.
- $L(p, c)$ means that person p liked chapter c.

Review of [5]
Companion to the Papers of Donald Knuth
by Donald E. Knuth
Center for the Study of Language and Information (CSLI), 2010
438 pages

Paperback – New, $30.00
Paperback – Used, $25.00
Hardcover – New $62.00
Hardcover – Used $83.00

Reviewed by William Gasarch, Dept. of CS, Univ. of MD
gasarch@cs.umd.edu

1 Introduction

Companion to the Papers of Donald Knuth is really five books: (1) problems and a few solutions (30 pages), (2) essays by Donald Knuth on some topics (10 pages), (3) conversations with Donald Knuth from 1996 (157 pages) (4) Donald Knuth's list of papers and other information about him (100 pages) (5) an index to this book and to the other books of Selected papers of Donald Knuth (150 pages).

2 Summary

The problems are all problems that he posed in other journals. They vary tremendously in their scope and difficulty.

The essays are interesting and short so they make their point quickly. The essay that really gets to the heart of what makes Knuth a great researcher is *Theory and Practice and Fun* which says in just 2 pages that we should be driven by our curiosity and a sense of fun.

The conversations I will discuss in the next section of this review.

The list of papers is very nicely annotated. Most papers have brief summaries. There is a list available online `http://www-cs-faculty.stanford.edu/~knuth/vita.pdf` but it does not have the summaries. In this modern electronic age it would be good to have the papers themselves all online at some website, as has been done for Paul Erdős (`http://www.renyi.hu/~p_erdos/`) and Ronald Graham. (`http://www.math.ucsd.edu/~ronspubs/`).

The index is useful if you have all of the other books.

3 The Conversations

Kurt Vonnegut once said (I am paraphrasing) that interviews do not work that well since the interviewer does not know what questions will lead to interesting responses. Kurt Vonnegut cut this Gordian knot by interviewing himself. The conversations in *Companion* are between DEK (Donald E Knuth) and DK, so I originally wondered if Donald Knuth did the same. No he did not— DK is allegedly Dikran Karagueuzian, a name so contrived it has to be real. [6]

The conversations are fascinating in that they tell much about Donald Knuth and about how computer science has changed. As I was reading them I noted some interesting tidbits for this review; however, that soon became impossible since there was at least one on every page. I will just mention a few.

The conversations took place shortly after Knuth won the Kyoto Prize, hence some of it is about the prize and what he will do with the money. He gave it all to various charities including helping his church get a new organ. That seems so down-to-earth I find it charming.

I vaguely knew that Donald Knuth was the first person to really apply mathematical analysis to algorithms; however, I didn't know that Knuth himself originally saw math (his major) and computer science (his job) as two distinct subjects with no intersection or interplay. The turning point was when he realized that linear probing should and could be analyzed mathematically.

I knew that Donald Knuth created TEX. I had thought that one of the reasons TEX is so much better than Troff (an older word processing system) is that TEX was written by ONE person while Troff was written by a committee. While this may be true it is also important who the one person was. Donald Knuth learned about *every single aspect of typography* while working on TEX. He consulted many of the worlds experts and hence became an expert himself. His knowledge of typesetting— from soup to nuts— is astounding. Here is a tidbit: typesetting a 4th grade arithmetic primer is *harder* than typesetting (say) Knuth Volume 3, since they often have in such primers really big numerals (I don't mean the numbers are large like 10^{100}, I mean the numerals are many inches tall).

I had often said (partially in jest) that all great scientists and mathematicians come from awful family lives. Donald Knuth is a counterexample to this. His childhood and his own life are very happy. He *was not* bored in school, he *was not* an outcast. He earned several varsity letters in sports for being a scorekeeper (not for playing) and he helped the basketball team at Case (his ugrad school) win by devising a way to tell how good a player was by using statistics. This was actually reported on by Walter Cronkite. He is still on his first marriage and has two children John and Jennifer and four grandchildren. John is high school math teacher and Jennifer is a homemaker. Neither of them have had a CS course for fear of being compared to their father. All very normal and happy. This is all interesting because it's not— that is, it's interesting that there are no hardships or scandals. In addition he is neither boastful nor falsely modest.

A list of tidbits:

1. The term *multiset* is relatively new. Knuth saw the need for a term for a set with repeated elements and he asked Dick de Bruijn about it. De Bruijn suggested *multiset* which became the term. The only competitor was *bag* which is dying out.

2. *The Art of Computer Programming* was supposed to be a book (yes ONE book) on compiler design. But his need to get everything rigorous and right made it become what it is now.

3. Say the variable x is 5 and you do $x := x + 1$, so now x is 6. The fact that it was 5 is gone! This concept, so natural to us now, was troubling to people early on in the field, including von Neumann.

4. In 1967 computer scientists either did *Programming Languages* or *Numerical Analysis* or *Artificial Intelligence*. He had to define the very name of his field: *Analysis of Algorithms*

[6] Donald Knuth told me that in the hardcover version of the book there is a picture on the back cover of Donald Knuth and Dikran Karagueuzian together. Amazing what photoshop can do nowadays.

5. Code Optimization: The conventional wisdom is that compiled code is only 10% slower than if you did it in assembly yourself looking for optimizations. This is already an odd statement since it depends on who YOU are. Donald Knuth states that he's never seen a program that he couldn't make 5 to 10 times faster in assembly; however, he doesn't need that kind of speed.

4 Opinion

I like this book mostly for the conversations. They really give you insight into the Knuth more than a biography or even an autobiography would. Donald Knuth has been called *The father of computer science* and hence his thoughts are worth knowing. The book also tells you how things were in the past which is interesting and good to know as we ponder the future.

Joint Review[7]

Horizons of Combinatorics
(Conference on Combinatorics)
Edited by Ervin Györi, Gyula Katona, László Lovász
Publisher: Springer, 2008, $129.00, 545 pages
Bolyai Society of Math Studies Volume 17

Building Bridges
(In honor of László Lovász's 60th birthday-Vol 1)
Edited by Martin Grötschel and Gyula Katona
Publisher: Springer, 2008, $129.00, 545 pages
Bolyai Society of Math Studies Volume 19

Fete of Combinatorics and Computer Science
(In honor of László Lovász's 60th birthday-Vol 2)
Edited by Gyula Katona, Alexander Schrijver, and Tamás Szoönyi
Publisher: Springer, 2008, $169.00, 365 pages
Bolyai Society of Math Studies Volume 20

Erdős Centennial
(In honor of Paul Erdős's 100th birthday)
Edited by László Lovász, Imre Ruzsa, Vera Sós
Publisher: Springer, 2013, $149.00, 700 pages
Bolyai Society of Math Studies Volume 25

Reviewer: William Gasarch gasarch@cs.umd.edu

1 Overview

The Janos Bolyai Society has published 25 collections of articles from math conferences and I am sure they will publish more. Most are in combinatorics. We review four of them and then in the opinion section we address not only the value of these books but the value of collections like them. The first of the four, *Horizons of Combinatorics* is a combinatorics conference without a more detailed theme. The second and third, *Building Bridges* and *Fete of Combinatorics and Computer Science* were Volumes 1 and 2 of a conference to honor Lázló Lovasz. The papers in them were mostly combinatorics though some have a computer science flavor or application. The fourth one, *Erdős Centennial* is largely articles from people who spoke at the Erdős Centennial which was July 1-5 in 2013 (I was there!). Since Paul Erdős did mathematics across a wide swath or mathematics (not just combinatorics for which he is best known) the articles are in many fields.

I will comment on a subset of articles from these collections. The subset is not random. It reflects my tastes and my conjecture of my readers tastes.

2 Horizons of Combinatorics (Volume 17)

Ballot Theorems, Old and New by Addario-Berry and Reed.

The following theorem was proven in 1887. I use modern notation.

Theorem: Assume that Alice and Bob are the only two candidates in an election. There are $n + m$ voters with $n > m$. Alice will get n votes and be elected. Bob will get m votes. Voters vote one at a time. The probability that Alice is always ahead of Bob (except when nobody has voted) is $(n - m)/(n + m)$.

The proof of this theorem is not hard and is given in this chapter. This lead to many variants of the theorem: (1) what if we want Alice to always have at least k times what Bob has? (2) if $n = m$ then what is the probability that Alice is never behind? They also look at gambling versions and continuous versions of these types of problems. Many proofs are given.

Erdős-Hajnal-Type Results on Intersection Patterns of Geometric Objects by Jacob Fox and Janos Pach.

By Ramsey's Theorem if G is any graph on n vertices then there is either a clique or ind. set of size $\Omega(\log n)$. What if G is of a certain type? Let H be a graph. Let $\mathcal{F}(H)$ be the set of all graphs G that do not have H as an induced subgraph. Then there exists a constant c (which depends only on H) so that every G in $\mathcal{F}(H)$ on n vertices has a clique or independent set of size $e^{c\sqrt{\log n}}$. Note that $\log n \ll e^{c\sqrt{\log n}} \ll n^{\epsilon}$.

This paper is concerned with families of graphs $\mathcal{F}$ where one can obtain a lower bound of n^c for some c. We state one of their theorems: Let $\mathcal{S}$ be a family of vertically convex sets in the plane[8] Let $a_1, \ldots, a_n$ be n elements of $\mathcal{S}$. We phrase the conclusion in two ways

1. There are either $n^{1/5}$ a_i's that are pairwise disjoint or there are $n^{1/5}$ a_i's that all intersect.
2. Consider the graph obtained by connecting i, j iff $a_i \cap a_j \neq \emptyset$. This graph has either a clique or independent set of size $n^{1/5}$.

This chapter states many interesting results but has no proofs.

3 Building Bridges Between Mathematics and Computer Science (Volume 19)

On the Power of Linear Dependency by Imre Barany.

Easy Theorem: If $V = \{v_1, \ldots, v_n\} \subseteq [-1, 1]$ is such that $\sum_{i=0}^{n} v_i = 0$ then there exists a reordering $\{u_1, \ldots, u_n\}$ of V such that $(\forall k)[|\sum_{i=1}^{k} u_i| \leq 1]$.

What happens if instead of reals you have points in R^2? In R^d? We state two results:

Steinitz's Lemma: Let $d \in N$, B^d be the unit ball in R^d. If $V = \{v_1, \ldots, v_n\} \subseteq B$ is such that $\sum_{i=0}^{n} v_i = 0$ then there exists a reordering $\{u_1, \ldots, u_n$ of V such that $(\forall k)[|\sum_{i=1}^{k} u_i| \leq d]$.

The paper proves this nicely by induction and also proves a weaker result which ends $\leq \sqrt{(4^d - 1)/3}$ but with a different (and I think nicer) proof. I worked through the proof for $d = 2$ and will present it in seminar— this is the highest praise I can give to an article.

The article also discusses lower bounds and generalizations. Many proofs are given.

[8] A set is *vertically convex* if every vertical line intersects it in an interval.

These questions and answers look like they are surely from pure math. But wait! There are applications of this sort of thing! And I don't mean applications to other parts of pure math! I don't even mean applications to complexity theory! There are applications to algorithms! Here are three references to applications from Imre Barany's article Only one was online (alas).

1. A vector sum theorem and its application to improving flow shop guarantees. By I. Barany. Math. Oper. Res. Volume 6, 1981, 445-452. `http://www.jstor.org/stable/i288433` or if it goes away or does not work then `http://www.cs.umd.edu/~gasarch/BLOGPAPERS/appsteinitz.pdf`
2. On approximate solutions of a scheduling problem. by S.V. Sevastyanov. Metody Diskr. Analiza. Volume 32, 1978, 66-75. (In Russian).
3. On some geometric methods in scheduling theory: a survey, Discrete Applied Math, Volume 55, 1994, 59-82.

The article is very nice but the title is a bit off since its mostly on Steinitz's lemma. Linear dependencies are used in many proofs, but that does not seem to be the theme.

Surplus of Graphs and the Lovasz Local Lemma by Josef Beck.

We first look at a very concrete case: The *Maker-Breaker Game* on an $n \times n$ board. In this game the players alternate coloring squares of the $n \times n$ grid. Player I, the Maker, RED, is trying to get some row or column with LOTS of RED squares. Player II, the Breaker, BLUE, is trying to just prevent Maker from doing this. Clearly the Maker can get a row or column with $n/2$ RED squares. Informally, the *surplus* is how much MORE than that the Maker can get if he plays optimally. This article shows that the surplus is between $\Omega(\sqrt{n})$ and $O(\sqrt{n}(\log n)^2)$.

Let G be a graph. The *Maker-Breaker Game on G* is as follows. In this game the players alternate coloring edges of G. Player I, the Maker, RED, is trying to get some vertex of high RED degree. Player II, the Breaker, BLUE, is trying to just prevent Maker from doing this. Note that if d is the largest degree of the graph then Maker can clearly get $d/2$. The *Surplus of G* is roughly how much more than $d/2$ can Maker get.

This article proves many theorems about the Surplus of a graph, and about other games. Some of the proves use the Lovasz Local Lemma. Many of the results appear here for the first time.

4 Fete of Combinatorics and Computer Science (Volume 20)

Iterated Triangle Problems by Butler and Graham.

Take a triangle T. Take its incenter p (the intersection of the angle bisectors). Draw lines from each vertex to p. This divides the triangle into three triangles. Repeat this process on all three triangles. Iterate. After n iterations there will be 3^n triangles. What can you say about an nth iteration triangle? The answer has to do with the Sierpinski triangle.

What about other ways to do this (e.g., take the centroid of a triangle)? This paper has very few theorems; however, it does have lots of data and information about this problem. The answers depend on the triangle involved and on their angles (e.g., are they rational multiples of π). The data suggests there may be connections of some of these problems to the Farey numbers.

This paper is one that starts research on a problem. Will there be more? Hopefully yes and before the Lovasz Centennial.

Solution of Peter Winkler's Pizza Problem by Cibulka, Kyncl, Meszaros, Stolar, Valtr.

Peter Winkler posed the following problem: Alice and Bob are going to split a round Pizza. Bob cuts it into (not necc. equal) slices. Then Alice takes a slice. After that they alternate taking slices; however, they must take a slice adjacent to a slice already taken.

Clearly Bob can guarantee himself 1/2 of the pie. Winkler showed that Bob has a strategy that guarantees 5/9. He asked *Does Alice have a strategy that guarantees 4/9?*. This paper shows that yes, she does.

This paper blurs the line between "recreational" and "real" mathematics. Winklers result of 5/9 is recreational in that it is clever and can be described to a laymen. The 4/9 result is very clever but likely too long to be called recreational. The problem itself sounds recreational but that's not really a criteria since many deep branches of math have their start in recreational problems (notably probability).

5 Erdős Centennial (Volume 25)

Paul Erdős and Probabilistic Reasoning by Alon.

As my readers surely know, the Prob. Method is often used to prove results that do not mention probability. We mention two that are presented in this article in the hope that the probability that my readers do not already know them is nonzero.

Theorem: (Erdős and Füredi, 1983). For every $d \geq 1$ there is a set of at least $\left\lfloor \frac{1}{2}(\frac{2}{\sqrt{3}})^d \right\rfloor$ points in R^d such that all angles formed by three of them are strictly less than $\pi/2$.

Theorem: (Erdős 1965). If A is a set of n integers then there is a subset of size at least $n/3$ that has no x, y, z such that $x + y = z$.

This theorems proof is so short and elegant that I sketch it: Let $x \in [0, 1]$. Let $B_x = \{x \in A \mid ax \pmod 1 \in (1/3, 2/3)\}$. One can show easily show that (1) B_x is sum free, and (2) if x is chosen uniformly in $[0, 1]$ then the expected value of the $|B_x|$ is $n/3$. Hence for some x it IS $n/3$.

The Phase Transition in the Erdős-Rényi Random Graph Process by Bollobas and Riordan.

Let n be large. Take a random graph with cn edges. What is the expected size of its largest component? This depends on c. For $c < 1/2$ the max component has expected size $O(\log n)$. If $c > 1/2$ then the max component has expected size $\Omega(n)$. What happens at $c = 1/2$? We need to be more precise. If the number of edges is $n/2 - \omega(n)n^{2/3}$ then there are many large components of about equal size. They are mostly trees. If the number of edges is $n/2 + \omega(n)n^{2/3}$ then ... the situation is complicated (though known). If the number of edges is $n/2 + n^{2/3}$ then there is one giant component that is much bigger than all fo the others.

The History of Degenerate (Bipartite) Extremal Graph Problems by Füredi and Simonovits

What is the max number of edges for a graph on n vertices? We all know its $\binom{n}{2}$. What if we only look at graphs that do not have a certain subgraphs? Let $ex(n, H)$ be the max number of edges in a graph on N vertices that does not have H as a subgraph.

1. $ex(n, C_{2k}) \leq 100kn^{1+(1/k)}$.
2. $ex(n, C_4) \leq \frac{1}{2}q(q + 1)^2$ where $n = q^2 + q + 1$.
3. If n is a power of a prime then $ex(n, C_4) = \frac{1}{2}q(q + 1)^2$ where $n = q^2 + q + 1$.

Erdős and Arithmetic Progressions by Timothy Gowers

Erdős and Turan made the following conjecture: if A is a set and $\sum_{a \in A} \frac{1}{a} = \infty$ then A has arbitrarily long arithmetic sequences. There has not been much progress on this conjecture. This is roughly equivalent to saying that if a set has density $\frac{1}{\log n}$ then it has arbitrarily long arithmetic sequences. This seems hard to prove: it is not even know if there is a 3-AP. The best known result is by Sanders: If a set has density $\Omega((\log\log n)^5/\log n)$ then it has a 3-AP.

A 3-AP can be restated as a triple (x, y, z) such that $x + z = 2y$. Schoen and Shkredov showed that for another equation a very nice result can be proven: If A is a set of density $\exp(-c(\log n)^{1/(6-\epsilon)})$ then there exists $x_1, x_2, x_3, x_4, x_5, y \in A$ such that $x_1 + x_2 + x_3 + x_4 + x_5 = 5y$.

Erdős's Work on Infinite Graphs by Komjath

Let G be any graph that has K_6 as a subgraph By (easy) Ramsey Theory, no matter how you 2-color the edges of G there is a mono K_3. What if G does not have K_6 as a subgraph? More precisely, Erdős and Hajnal asked (in 1967): is there a graph G that does not have K_6 as a subgraph such that no matter how you 2-color the edges of G there is a mono K_3? Such a graph was quickly found by both Cherlin, Graham, van Lint, and Posa. Posa's graph also had no K_5.

Is there a graph G that does not have K_4 as a subgraph such that No matter how you 2-color the edges of G there is a mono K_3? We will return to this question a bit later.

What about the infinite case? Is there an uncountable graph G that does not have K_4 as a subgraph such that no matter how you ω-color the edges of G there is a mono K_3? Shelah showed using forcing that there is a model of set theory where such a graph exists. Let M be the model and G be the graph.

The following much weaker statement is true: no matter how you 2-color the edges of G there is a mono K_3. By compactness there must exist (in the model M) a finite graph G' with no K_4 subgraph, such that no matter how you 2-color the edges of G' there is a mono K_3. But Forcing cannot add finite graphs! Hence there exists (in any model of set theory) a (finite) graph G' with no K_4 subgraph such that no matter how you 2-color the edges of G' there is a monochromatic K_3.

This has got to be (informally) the most nonconstructive proof I have ever seen!

However, and perhaps unfortunately, earlier than that there already was such a finite graph. Folkman had a gigantic example ($10^{10^{\cdot^{\cdot^{\cdot}}}}$ with 7 stacks of 10's). Over time smaller and smaller graphs were found and now the smallest such graph has 786 vertices (see `www.cs.rit.edu/~arl9577/is/folkman/paper/fe334_mc.pdf`).

6 Opinion

The papers in these volumes are not Journal papers. This frees the authors to write whatever types of paper they want: serious math, recreational math (though it may be hard to tell the difference), surveys, original results, include proofs, do not include proofs, starting a topic, ending a topic, giving data but no proofs. Not including proofs of known results is not a problem in the internet age if there is a pointer to a free online manuscript that has a proof. The variety of types of papers in these volumes is a strength. These volumes make me question the value of journal articles which have to be of a certain form (original research).

Another issue is how interesting are the articles. This is more a function of the reader; however, I will say that personally about 1/2 of the articles interested me enough to want to read them. Another 1/4 were interesting enough so I wanted to know the result.

Should YOU buy this series of books? This is a hard and more profound question than it may appear. 20 years ago I would say that you should certainly get your library to buy this series so that students and profs could goto the library, check out the books, and/or photocopy articles from them (you can ask your grandfather what a library is, what checking a book out means, and what a photocopy is). In the internet age where everything is online (many of these articles are online) it is less clear. I admit that having an actual

BOOK on my desk motivates me in a way that just knowing there are articles on the web does not. But books are expensive to buy (and likely expensive to produce). And I suspect that my great nephew will say: *There goes Uncle Bill babbling about those old fashion things called books! You can't do a search on them, you can't make the font size go up up and down, I don't know what the old coot sees in them. Then again, my own grand-daughter doesn't understand why I like old fashion 2-D TV, and not her darn 3-D contraptions that I can never get to work.*

You *could* just goto the Bolyai Society Math Studies website and look at tables of contents and find the articles on the web. But *will you?* The rather innocent question of *Should your have your school library buy this (or any) series of books?* leads to profound questions about how the book world, and the world in general, is changing.

Review of[9]
Bayesian Reasoning and Machine Learning
by David Barber
Cambridge University Press, 2012
697 pages, Hardcover, US$ 90

Review by
Matthias Gallé, matthias.galle@xrce.xerox.com
Xerox Research Centre Europe

1 Introduction

Yet another book about machine learning? you may be thinking if you are not working in this area and are getting tired of the hype in newspaper, companies and conferences. I would like to add two caveats to such a potential exclamation: first, there are not that many textbooks on the area as one would imagine (David Barber lists 10 in his introduction, and I had a hard time to bring this number up to 15). And, second, strictly speaking, this book is not only about machine learning. It is definitively not an applied toolbox of algorithms and recipes on how to process any data you get. *Bayesian Reasoning and Machine Learning (BRML)* tries to present a unified view of Machine Learning , using as much as possible a Bayesian approach. It is therefore fitting that BRML uses *graphical models* as underlying general representation throughout the book. In a nutshell, graphical models permit one to represent random variables (nodes) and their dependencies (edges), and generic algorithms exists for inference and learning. By stating all the dependencies, it suits a Bayesian treatment and at the same time it is general enough so that many algorithms and approaches can be represented as graphical models.

2 Summary

BRML consists of 28 chapters, divided into 5 parts.

Chapter 1 starts gently, explaining – using several examples – notions around probability (conditional probability, independence, Bayes theorem, etc). Chapter 2 gives the necessary background in graphs and matrix representation showing how several complex functions can be cast as matrix operations. Chapter 3 and 4 are about the most important notions of the book, the one of graphical model and belief networks. Here too the author gives several examples and tries to only give the math that is needed. Towards the end of the chapters the explanations become deeper and more technical. Chapter 5 follows this trend with some exhaustive explanations of several algorithms using graphical models, and the entire Chapter 6 is dedicated to the Junction Tree Algorithm. To finish this first part, Chapter 7 englobes other important concepts connected to graphical models (decision trees, Expectation-Maximization (EM) algorithm, etc).

Part 2 gives more background on notions related to learning. It starts with tons of definition (the different probability distributions to be used later on) in Chapter 8 and Chapter 9 considers how to learn the parameter for this distributions and how they relate to the graphical models studied in Part 1. Chapter 10 is a nice wrap-up, using Naïve Bayes as an example to review the concepts seen so far. Chapter 11 is devoted to the concept of hidden (latent) variables and the general two families to learn them (EM and Variational Bayes). No book on learning can be complete without mentioning Occam's' Razor: this is done in Chapter 12 around model selection.

Part 3 starts like a classical book on machine learning: Chapter 13 describes the high-level differences between supervised vs unsupervised learning, Bayes vs frequentist and generative vs discriminative models. The short Chapter 14 talks about the Nearest-Neighbor algorithm and is a nice example of an ongoing pattern in the book: every time a general machine learning algorithm is described, the author then tries to explain it in Bayesian terms clarifying the differences and highlighting advantages and drawbacks. Chapter 15 is all around the most important acronyms in Machine Learning: Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Probabilistic Latent Semantic Analysis (PLSA), Non-negative Matrix Factorization (NMF) and Canonical Correlation Analysis (CCA). Chapter 16 then deep-dives into supervised linear dimensional reduction techniques, notably Fishers' Linear Discriminant Analysis. Chapter 17 explains some of the most well known classification algorithms (logistic regression, perceptron, Supervised Vector Machines (SVM)) as parts of the larger family of linear models. Chapter 18 then puts on the Bayesian lenses and explains Bayesian formalizations of such linear models. Chapter 19 is devoted to non-parametric Bayesian: this is a growing trend in machine learning, and the book explains the basic building block here, namely Gaussian Processes. Chapter 20 is about mixture models and it showcases important examples: Gaussian Mixture Models, k-means and Latent Dirichlet Allocation. The short Chapter 21 just explains probabilistic PCA and this part finishes with a case-study on measuring the ability of players (Chapter 22).

Part 4 finally shows powerful usages of graphical models by the means of Dynamical Models. Chapter 23 is about general Markov models (discrete time), and Chapter 24 about linear dynamic system (continuous time). Chapter 25 is much more technical than the others and shows different ways of combining both previous models by using a discrete number of linear dynamic systems. Chapter 26 is called distributed computing and is about neural networks, which is arguably the most popular topic in machine learning these days.

The last Part of the book deals with approximate inference. Chapter 27 treats the two big approaches for sampling (Gibbs sampling and Markov-Chain Monte-Carlo methods) and Chapter 28 is about deterministic approximate inference (the variational approach).

The Appendix contains basic notions of linear algebra and calculus. Additionally, the author chose to explain here the most common used optimization techniques, so that a reader not familiar with them should maybe look them up before entering the more technical chapters.

3 Opinion

The book promises to "convey the basic computational reasoning and more advanced techniques [..] for students without a firm background in statistics, calculus or linear algebra". Specially in the first chapters, I felt a certain tension between not going too much into mathematical details, but at the same time being formally correct and not hand-waving explanations away. As such BRML is clearly not deep enough for a graduate student or researcher who wants to know everything about a specific technique. At the same time, it is far from being a toolbox description for machine learning application, or a gentle introduction to the field. For both cases (although more so for the latter) there are several books and online courses out there.

Trying to convey more than an overall intuition of machine learning algorithms without assuming a strong mathematical background is not an easy task. The author takes up this challenge by providing regularly lots of examples. If a reader really wants to understand a concept these will certainly help, while they can be ignored if one just wants to get a general idea.

Putting graphical models in the beginning makes for a hard start, but this gets payed back afterwards when several other algorithms derive easily making reference to these initial chapters. I enjoyed the explanation of several topics, like the presentations of Markov models (Chapter 23) and Gibbs sampling (Chapter 27).

So, who should read this book? The author targeted it towards “final year undergraduates and graduates without significant experience in mathematics”. I think he achieved his goal, although the book certainly is formal (specially in the latter chapters), many formulas are completely derived and the occasional curious student who just wants know what machine learning is about or how to use it may get disappointed.

In my opinion the readership who would benefit the most of BRML would mainly be two groups: on one hand graduate students or researches in a scientific field (notably physics, engineering or biology) who are not afraid of a bit of math and want to use **and** understand methods of machine learning. While understanding the whole book will probably not give you enough background to start doing a contribution to hard-core machine learning, it will for sure be more than enough to get a good notion of computational reasoning and let you easily participate at a machine learning cocktail party. The second group which I encourage to have a look at this book are lecturers. Upon contact, the author provides access to a set of slides (604 slides last time I checked) in LaTeX(Beamer package). This is in addition to the public Matlab toolbox to which the book refers to constantly[10], and the whole design of the books (use of examples, large lists of exercises, detailed derivation of important formulas) is very well suited for a course. Those parts that are harder to digest will be eased with a good explanation, while giving enough background to check the necessary details.

[10] `http://www.cs.ucl.ac.uk/staff/d.barber/brml/`

Review of[11]
Integrated Methods for Optimization
Second Edition, 2012
by John Hooker
Vol. 170, International Series in Operations Research and Management Science, Springer
Hardcover

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

1 Introduction

This review concerns the second edition of a book on Integrated Methods for Optimization. The book integrates key concepts from mathematical programming and constraint programming. It has been published as Vol. 170 of the International Series in Operations Research and Management Science by Springer. The first edition of the book was published as Vol. 100 of the same series. The book is available in both hardcover (ISBN 978-1-4614-1899-3, price US $119) and eBook (ISBN 978-1-4614-1900-6, price US $ 89.95) formats. A solution manual for instructors is available. The book has been authored by John Hooker well-known for his contributions to mathematical optimization. The author has been a strong votary for the advocacy of integrated methods for optimization.

2 Summary

The book has eight chapters.

The first chapter is the introductory chapter. It outlines the framework of the author that integrates methods for search, inference, and relaxation. The role of duality is mentioned and the advantages of integrated methods are highlighted. The chapter looks at some applications and software packages for solving problems by employing integrated methods for optimization.

The second chapter contains many examples of practical problems such as those related to freight transfer, production planning, scheduling employees, product configuration, and planning and scheduling.

The third chapter focuses on the basics of optimization. Linear programming, non-linear programming, dynamic programming, and network flows are briefly explained. In linear programming, the Simplex method developed by George Dantzig in 1947 is discussed. This algorithm is known to take exponential time for some inputs. Polynomial time algorithms such as the ellipsoid algorithm of Khachiyan and the interior point algorithm of Karmarkar are not described in the book. Non-linear programming is helpful for solving convex relaxations and reducing variable domains in global optimization problems. Dynamic programming can be helpful for domain reduction in sequencing problems. The book's sixth chapter on inference makes use of dynamic programming. Network flows have a role in filtering methods for several global constraints. The well-known minimum-cost network flow problem is a special case of linear programming. The chapter describes the network simplex method for network flows. Well-known algorithms for network flows such as the Ford-Fulkerson algorithm and other newer algorithms are not mentioned in this chapter. The topics in this chapter are selective and their treatment is not intended to be comprehensive.

The fourth chapter is on duality. Various types of duality are discussed such as inference duality, relaxation duality, linear programming duality, surrogate duality, Lagrangian duality, sub-additive duality, and branching duality. Duality is an important concept that helps to merge optimization techniques. It links search with inference and relaxation. It is possible to solve a problem by searching for the most beneficial solution but we may simultaneously search for a solution of the inference dual and/or the relaxation dual. Many successful optimization methods make use of the concept of duality.

The fifth chapter describes methods for search. Search is the enumeration of problem restrictions. The chief concern in search is deciding where to look next. Exhaustive search methods often take one of the following two forms: branching search and constraint-directed search. This chapter concentrates on branching search, constraint-directed search, and also local search. Local search algorithms proceed from solution to solution in the space of prospective solutions by applying local changes, until a solution viewed as optimal is found or a time bound is passed. The chapter illustrates the application of search techniques by considering problems such as airline crew scheduling, satisfiability of propositions, and single-vehicle routing.

The sixth chapter focuses on the concept of inference. Various kinds of inequalities and constraints are discussed. Inference and relaxation help to make search more intelligent. Inference helps to extract implicit information about where the solution may reside. This helps to reduce the amount of search needed. Inference may be considered as a way of learning more about the search space so that time is not spent in looking at the inappropriate places.

The seventh chapter looks at relaxation methods. It addresses linear inequalities and a diversity of constraints.

The eighth chapter is essentially a dictionary of various kinds of constraints that are often encountered in practical problems.

3 Opinion

The second edition is an important and timely update to the first edition. Its release is fully justified given the numerous developments in the field of integrated methods for optimization since the appearance of the first edition. The second edition includes three more chapters when compared to the first edition. These chapters comprise the chapter containing examples, the chapter on optimization basics, and the chapter on duality. The new edition provides solutions to exercises. It provides useful ideas from mathematical programming and constraint programming. An integrated approach is helpful as software packages now tend to use techniques from both the fields. Many books on operations research do not cover constraint programming but this one does. The detailed treatment of constraint programming in the book is worthy of praise. It is likely that in the future, integrated methods for optimization will be increasingly emphasized in books on mathematical optimization.

This book will be useful for students, practitioners, and researchers. Students studying computer science will find this book particularly useful as there are not many books for them that give an insight into mathematical programming or global optimization. The benefits of combining mathematical programming and constraint programming are significant such as more potential for modeling, faster computation, and shorter computer programs. It also makes it easier to solve real-life problems with the help of software packages and computers. The author introduces a useful framework that integrates search, inference, and relaxation. The book will be very useful for pedagogy as the concepts are explained clearly and there are numerous exercises and bibliographic notes, in addition to a solution manual for instructors. The book is well organized in spite of the diverse topics covered by it.

Mathematical programming and constraint programming are vast fields, however, this book provides useful snapshots from both the fields. In the revised edition, the author has added numerous references to the bibliography. This is very useful for those who wish to know more. The index is quite satisfactory. I strongly recommend this book for those interested in gaining from integrated methods for optimization.

Review of[12]
Programming with Higher-Order Logic
by Dale Miller, Gopalan Nadathur
Cambridge University Press, 2012
320 pages, HARDCOVER

Review by
Vaishak Belle
vaishak@cs.toronto.edu
University of Toronto, Canada

1 Introduction

Formal models of systems in computer science typically involve specifications over syntactic structures, such as formulas in a logic, λ-terms, π-calculus expressions, among others. Logic programming is one way to realize these specifications and studying such models; Prolog, for example, is an important logic programming language heavily used in declarative frameworks in computer science. This book presents deep techniques that take the many ideas at the heart of Prolog, and extend its power and expressivity by elegantly combining these ideas with a simply typed version of higher-order logic. The end result is a rich programming language, called *λProlog*, that benefits from the paradigms of higher-order logic and logic programming.

The technical material in the book is perhaps most readily accessible to readers familiar with Prolog and aspects of functional programming. That is, the book does not deal with introductory material on either of these topics. The proof-theoretic framework used to justify the various derivation rules also require readers to be familiar with logic.

2 Summary and Review

Broadly speaking, this book is about how logic programming can be realized in a higher-order logic setting. There are interesting benefits to such a realization: it provides an extension to the expressivity of a popular logic programming language, such as Prolog, in allowing for λ-abstractions. Most significantly, the endeavor behind the book is to take this realization as a full-fledged programming language, supporting complex data types and other kinds of modularity encountered in conventional programming languages. However, combining logic programming and higher-order logic in this style raises technical concerns, especially in providing a semantics to the programming language. To deal satisfactorily with these issues, the authors appeal to the sequent calculus, among other things, all of which is shown to lead to a computational methodology over formal objects. The resulting programming language is then demonstrated to be expressive enough to represent many applications, such as π-calculus specifications and functional programming. Their ideas are actually implemented in a system called *Teyjus*. However, the book should not be seen as a manual for this implementation, and indeed, the notions discussed in many chapters of the book are at a fairly high level of generality.

In the sequel, I will discuss the essentials of the individual chapters in the book.

- **Introduction.** The authors discuss the nature of the endeavor in this chapter. Roughly speaking, there are two distinct views on how logic is used in computer science. In the first, logic is viewed as a mathematical language to encode truths about a system; for example, in dynamic logic, programs are syntactic structures in a logical language, and logic is used to reason about the system. In the second view, expressions in the logic, such as terms and formulas, are understood as elements of the computation. Here, one might then search for a proof for a goal, as done in logic programming, or sometimes, computation is about reducing goal terms to some normal form, as in functional programming.

  This book is about logic programming, and so it clearly falls within the second view. In general terms, the logic programming paradigm works as follows. Assume a logical language and a particular signature S, consisting of predicate and function symbols. Using the symbols from S, programs or rules δ may be specified. The aim then is to obtain a proof for some goal formula ϕ, denoted $S\,;\delta \rightarrow \phi$.

  The second concern the authors clarify in this chapter is the sense in which higher-order logic is understood in the book. In philosophy, there is often the distinction made between first-order logic and second-order logic; the latter allows us to capture the standard model of arithmetic, but it is not recursively axiomatizable. Second-order logic is sometimes called higher-order logic in this sense. This is not what the authors are after. They are more concerned with quantification over λ-expressions, which allows one to form abstractions over formula expressions and is a powerful programming paradigm. Putting this together, the book is concerned with logic programming, that is, finding proofs for goal formulas ϕ, when programs δ may include both first-order terms and λ-terms.

  To get a flavor of the kind of programs that can be expressed in the language, consider the following example from the book:

  ```
  rel R :- primrel R.
  rel (x\ y\ sigma z\ R x z, S z y) :- primrel R, primrel S.
  ```

  This encodes "relationships" between individuals. Here, `primrel` is a predicate for a "primary relation", *e.g*

  ```
  primrel father & primrel mother & primrel wife & primrel husband.
  ```

  That is, predicates *father*, *mother*, *wife*, and *husband,* all of which are binary, are classified as being primary relationships. From primary relationships, other relationships can be extracted. So the first line of the program is saying that if R is a primary relationship, it is also a relationship. The second sentence is saying that if R and S are primary relationships, then $\lambda x \lambda y \exists z[R(x,z) \wedge S(z,y)]$ embodies a relationship between x and y. That is, there is an individual z such that $R(x,z)$ and $S(z,y)$ hold, which would then mean that x and y are related. If the user were to now include that Alice is Bob's wife, and Mary is Alice's mother, the relationship between Bob and Mary would be implicit in this program.

  Analogously, (complex) goals can be defined. For example, in one of the more expressive programming fragments considered in the book (*cf.* discussion on *First-Order Hereditary Harrop Formulas* below), we may give goals such as $\forall p(p)$ which simply encodes $\perp$, and $\forall p((A \supset p) \supset (B \supset p) \supset p)$ which simply encodes $A \vee B$. For a more interesting (and impressive) example query, consider the Fibonacci numbers. Let us suppose `(fibonacci n m)` indicates that the n^{th} Fibonacci number is m. Then the following query can be used to search for all numbers $0 \leq n \leq 20$ such that the n^{th} Fibonacci number is n^2:

```
fib_memo 20 (fibonacci\ sigma M\ fibonacci N M, M is N * N).
```

Here, `fib_memo` is used to compute and store an initial part of the `fibonacci` relation; its definition appears in the book and will not be reproduced here. Thus, in the above query, we are invoking the predicate `fib_memo` with an input 20, applied to the expression

$$\lambda f \exists m[f(n, m) \wedge m = n \times n],$$

with f denoting the Fibonacci relation.

The reader may verify that $n = 0$ satisfies this query (the 0^{th} Fibonacci number), as does $n = 1$ (the 1^{st} Fibonacci number) and $n = 144$ (the 12^{th} Fibonacci number).

- **First-Order Terms and Data Representation.** This chapter sketches the logical language and how data will be represented, often mixing linguistic and meta-linguistic notions. This has the advantage that a reader intending to delve into programming would know how to begin immediately. The actual variant of higher-order logic that is used in the book is Alonzo Church's *Simple Theory of Types*. Therefore, not surprisingly, types and typing constructors play a major role in the presentation. It is shown, for example, how abstract objects such as binary trees can be represented.

  For this logical language, complex formulas are built in the usual way, using logical connectives such as $\neg, \wedge, \supset$. Logical expressions appearing in the book are assumed to be universally quantified from the outside, following notions from logic programming. They also show how imperative programs, such as *while* loops, can be defined as first-order terms. The main concern with a programming language that uses the expressivity of first-order logic is this: how can we unify over first-order terms with equality? The authors discuss their strategy here which involves the reduction of complex terms, the orientation of variables and finally, the elimination of variables.

- **First-Order Horn Clauses (FOHC).** This chapter presents the first installment of the logic programming methodology in the book, based on Horn logic. Recall that given a program δ written using symbols from some signature S, the task is to find a proof for a goal formula ϕ. In this chapter, δ is built from Horn clauses, and ϕ is allowed to also mention disjunctions and existential quantifiers; so this is a reasonably expressive logic programming fragment. The question, then, is how ϕ is to be proved and the book introduces the search semantics for this task. A detailed programming exercise is provided, where the authors encode reachability in a finite state machine. Other aspects of interacting with the programming system, such as how to check for multiple proofs, are also discussed.

- **First-Order Hereditary Harrop Formulas (FOHH).** This chapter investigates a more expressive fragment than the version of first-order Horn logic considered in the previous chapter. Basically, this fragment allows implications and universal quantifiers in a certain way in the goal formula ϕ. An interesting example illustrating hypothetical reasoning is considered for this new fragment. The authors also provide an illuminating analysis of the symbol-level manipulations, on the one hand, and the knowledge-level semantical considerations, on the other. For example, it is shown that the sequents computed for the FOHH fragment is sound and complete for intuitionistic logic, but not sound for classical logic.

- **Typed λ-Terms and Formulas.** The two programming models above focused only on first-order logic. This is the first chapter where the discussion is taken to a higher-order logical setting. More precisely, as mentioned, using the simply typed λ-calculus the expressivity of the programming language is extended with λ-terms. For example, the extended language allows terms of the form $\lambda f \lambda x(g(f(x)))$. For this extended language, the authors then discuss how these abstractions are to be interpreted by using a kind of formula rewriting. It is then shown that this conversion mechanism is very expressive, capable of defining functions over natural numbers, such as Church numerals. The chapter concludes with many illustrations on unification in higher-order logic.

- **Using Quantification at Higher-Order Types.** This chapter is concerned with the use of predicate quantification in computations: that is, since predicates essentially correspond to procedures, the treatment of predicates as variables would form the basis of higher-order programming. There are number of pragmatic issues to be discussed for doing this, and the authors explore these issues here.

- **Structuring Large Programs.** This chapter presents ideas on how large programs can be handled manageably, essentially by conceptualizing a kind of modular programming. That is, by appropriating scoping of variables, a "module" construct is introduced, somewhat analogous to the "class" abstraction in conventional programming languages. It is worth reiterating here that these constructs basically macro expand into logical formulas; that is, they are defined entirely within the logic. A number of examples are also given on how complex data structures can be formalized using these constructs.

- The next two chapters explicate fully the *expressivity* obtained from λ-abstractions, how they can be used to build higher-order terms, and most significantly, how *unification* can be defined over such terms.

- The last three chapters discuss *applications* of the programming language and techniques discussed in the previous chapters. They show: (a) how proof procedures can be implemented, (b) how functional programming can be interpreted, and (c) how π-calculus can be encoded in their proposal. The actual specifications and encoding seems quite natural, perhaps suggesting a general way to analyze these different computational notions.

3 Opinion

The book studies how logic programming can be extended to express and reason with higher-order features, such as λ-abstractions. Of course, then, the book would certainly be of interest to computer scientists working in the areas of logic programming and functional programming. However, more broadly, I think the book would also be of interest to computer scientists in other areas, to see how computations can be defined over logical specifications, how such specifications can be defined in a modular fashion, the subtleties of abstraction, among other notions. While the presentation of the material is fairly involved, a reader skimming through some of the application-oriented and example-filled chapters might obtain insights on the richness of such syntactic structures. In my view, the book provides an interesting integration of two major computational paradigms founded in logic.

Review of
People, Problems, and Proofs[13]
Author: Richard Lipton and Ken Regan
Publisher: Springer, 2014
ISBN 978-3-642-41421-3, $40.00

Reviewer: William Gasarch `gasarch@cs.umd.edu`

1 Introduction

This is the second book to be written based on the blog *Godel's Lost Letter and P=NP*. The first one was *The P=NP Question and Godel's Lost Letter.* I reviewed that one in the December 2010 issue of SIGACT news.

I write this review in the form of a series of fictional blog posts. If there is a comment by, say, Colbert Nation, then that does not mean that someone named Colbert Nation actually thinks that. It may mean that I think that Colbert Nation thinks that.

Do not confuse this with my real blog or theirs.

INTRODUCING LIPTON-REGAN REVIEW BLOG
April 1, 2014

This blog will be dedicated to reviewing the Lipton-Regan book **People, Problems and Proofs**. The book is based on their blog *Godel's Lost Letter and P=NP*.

I have read the entire book so I will give my general impressions before blogging on particular articles.

When I read their blog I often read it, get interested, but then something comes up so I can't finish it and I promise I'll get back to it *tomorrow*. Like that song: *The blog will get read tomorrow! Bet your bottom dollar that tomorrow, I will read!* Yet having it in book form it seems easier to read. It seems that the chapters in this book are shorter than the blog. If so that's a bit odd since one would think the book version could afford to be longer.

The upshot is positive— I read the entire book (rare for a math book) understood most of it (rarer still) and am inspired to read more about the topics he introduced, and try to prove things for myself. *I'll prove the ther-ems tomorrow! Bet your bottom dollar that tomorrow, I will prove.*

ARE LIPTON AND REGAN REALLY THAT NICE?
April 15, 2014

Richard Lipton and Ken Regan are nice people. Or their blog-selves are nice people. Most chapters have as its title a person and a subtitle that is more about the content. The link between the person and the content varies. His descriptions of the people in the title of the chapters is always quite positive.

In Chapter 34, titled *Sam Buss: Bounded Logic*, Lipton talks about two courses he had in logic. In one the professor was often confused. The other course was taught in a unmotivated way. He said they were both great courses. That's being too nice.

The chapter itself was also nice. It involved ways to write quantifiers and refers to a result (which I will look up tomorrow) about inexpressibility.

Chapter 38 is about definitions. The title is *Alfonso Bedoya: Definitions, Definitions, and Definitions*. You may be wondering *who is Alfonso Bedoya?*. If you are under 35 you've probably already Googled it on some device you carry around and found out that he was the character Gold Hat in *The Treasure of Sierra Madre* who uttered the famous line:

Badges? We ain't got no badges. We don't need no badges.
I don't have to show you any stinking badges!

(Number 36 in the American Film Institutes 100 best movie quotes.)

Their point is that you don't need definitions— they can always be removed. I thought they would say:

Definitions? We ain't got no definitions. We don't need no definitions.
I don't have to show you any stinking definitions!

But they cleaned it up (I didn't even know it was dirty!) to

Definitions, definitions, we don't need no definitions.

I'm surprised they didn't also remove the double negative, in case children are reading, to obtain

Definitions, definitions, we don't need any definitions.

The chapter itself was nice. It was about what makes a nice definition, and it had some nice history. It was material I already knew but nice to have it all laid out.

COMMENTS:

Town Falconer: You forgot to mention the one time they really were nice to a fault. They were nicer to Deolalikar, the guy who rudely wasted their time with a false proof that P≠NP proof.

Glacial Warmish: Gee Town, if we ever turned out blog into a book nobody would accuse you of being to nice. Anyway, they wisely spend was Chapter One, *The Claimant, the readers, and the crowd*, *not* replicating their blog, but telling the whole story about Deolalikar from start to finish. This is really good since now that the end is known its good to see how it all began. However, Town, you are right, Lipton and Regan do not have any unkind words about him at all. Not even a tepid *he should at some point make a formal retraction*.

Deolalikar did not quite intend his proof to go public when it did. That gives him more sympathy.

Ken Regan: Too nice! How nice of you to say! However, note that the first paragraph of Section 1.12 of Chapter One I do note that Deolalikar has not addressed the issues raised. So there!

Sonata Consort: Ken, you call that not-being-nice? You use words like *Unfortunate* and phrases like *we glean that (the revised version) did not increase appreciably in content*. Only nice people write like that. If I had spend a month pouring over an obviously flawed paper I would have written

REST OF THIS COMMENT DELETED BY THE MODERATOR

One-Cat Tree: The chapter was more about crowd sourcing math than about the proof itself. This is an interesting topic; however, I think Terry Tao's polymath problems are a better and more productive example. And I also think that Lipton-Regan are too nice to say they disagree with me.

H. K. Donnut: Wow, it sounds like Chapter One is awesome. Would you say it's worth the price of the book?

Bill G: Well . . . I got my copy for free. However, yes, Chapter 1 was, as the kids say, jawesome!

Not Porn: I find your posts very nice. For something even nicer click HERE for what I promise is NOT a porn site. At least not in most states.

FOURTEEN LIGHT CHAPTERS
April 30, 2014

Not every post can be about an interesting piece of mathematics. It would just be too hard (though Terry Tao seems to manage it). And in fact Lipton-Regan do not attempt this. Of the 63 chapters in the book, 14 of them are more *about math* then have hard math in them. Things like how to guess what a result will be, how to try to prove it, the importance of a good notation, how to write up a result. These chapters were light reading but still informative.

COMMENTS

Colbert nation: HEY, it can either be light reading OR informative, but it can't be both. We're at war here, pick a side!

Bill G: Here is an example. Chapter 2, titled *Kenneth Iverson: Notation and Thinking* didn't have any real math in it but it did tell me the following:

1. Descartes is the first person to use x^4 instead of $xxxx$.
2. Euler is the first person to use $\sum$ for summation and also the first one to use the notation $f(x)$ for a function.
3. There is some debate about whether π was the right number to since 2π come up more often.

Alma Rho-Grand: Hey, YOU blogged on the π thing yourself. So that can't be news to you.

Bill G: Yeah, but I FORGOT! As I was reading it I thought it was a neat issue to raise *before* I saw that I was the one who raised. Awkward! More to the point— this book is good at reminding you of things you once knew.

FORTY NINE HEAVY CHAPTERS
May 1, 2014

If there are 63 chapters and 14 are light then 49 must be heavy. Actually the world is not quite so binary. However, there are 49 chapters that have math in them, much of it new and interesting. Since blogs are themselves summaries if I summarized all of them we may end up with some Quantum effects (Chapter 63 is *Charles Bennett: Quantum Protocols*). Hence I summarize just a few.

Chapter 37 is titled *Thomas Jech: The Axiom of Choice*. Recall the usual axiom of choice:

Let I be a set (it could be of any cardinality). Assume that for every $i \in I$ there is a set X_i. There exists a function f (a choice function) from I to $\bigcup_{i\in I} X_i$ such that $f(i) \in A_i$.

Look at the following restrictions of this. Let C_n be the following statement:

Let I be a set (it could be of any cardinality). Assume that for every $i \in I$ there is a set X_i such that $|X_i| = n$. There exists a function f (a choice function) from I to $\bigcup_{i\in I} X_i$ such that $f(i) \in A_i$.

For which n, m does C_n imply C_m? This chapter gives a clear statement of when this is true and gives some proofs of the easy direction (showing that C_n implies C_m). The hard direction, showing that C_n does not imply C_m, is really hard— it involves constructing models of set theory where C_n is true but C_m is not. These proofs are wisely omitted.

Chapter 43 is titled *Denis Thérien: Solvable Groups*. Recall that the complex numbers are algebraically closed. Let us put that a different way: If you want to solve a polynomial $p(x) = 0$ over the rationals then there will be a large enough field extension of the rationals where it has a root.

What if you are trying to solve a group equation over a group, such as $ax = bx$ over $S_5 \times S_7$ where $a = (1234)$ and $b = (13)(15)$. (I honestly do not know if that has a solution). If there is no solution then is there some group that contains $S_5 \times S_7$ as a subgroup where there is a solution? For this equation I do not know. But more to the point, is there some general theorem that says you can always find a larger group with a solution? NO. In this chapter they give an example where you cannot find a larger group (and they prove it works— it's not that hard) and state some theorems and conjectures about this issue.

Both Chapter 37 and Chapter 43 were excellent since they told me about a problem I had not thought of but once introduced were very interesting. In both cases they gave me a simple proof that I could follow. I plan to read more on these topics. Tomorrow.

COMMENTS:

Tim Andrer Grant: Is the math serious or recreational?

Bill G: To understand the statements of the theorems you need to know some math, say that of a sophomore math major. Even then, some terms would have to be explained to you. So you might say the statements (not the proofs) are recreational to people who read Martin Gardner's or Ian Stewart's columns AND went on to learn some more math.

Ana Writset: Why are the names of the comments so odd? Even mine!

Bill G: Aside from *Bill G*, *Not Porn*, and *Colbert Nation* all of the names are anagrams. Some of the anagrams are from the April 1, 2014 post on the Godel's Lost Letter Blog (Lipton-Regan blog) and some are from an April 1, 2013 post (more properly, a paper pointed to from that post) of computationalcomplexity blog (Fortnow-Gasarch blog).

WHO SHOULD BUY THIS BOOK?

May 15, 2014

If you are an undergraduate in math or CS (and like theory) then this book will tell you some tips on how to get started in research and give you some nice topics to read up on, though some may be hard. As you go from ugrad to grad student to professional the light chapters will get less interesting and the heavy chapters will get more interesting. I think Lipton and Regan have calculated the exact

right balance so the book is good for anyone.

Review of[14]
Who's Bigger? Where Historical Figures *Really* Rank
by Steven Skiena and Charles B. Ward
Cambridge University Press, 2014
379 pages, Hardcover

Review by
Nicholas Mattei
nicholas.mattei@nicta.com.au
Optimization Research Group
NICTA and UNSW

1 Introduction

Steven Skiena and Charles Ward ask an interesting question: can we say whose historical significance through purely statistical means using only public data? The answer is this book and its enjoyable answer, maybe. While at the end of the day this book is more of a playful walk down history the question itself is intriguing. Extremely data driven at its heart, *Who's Bigger* manages to entertain and enlighten throughout. It is an easy, enjoyable read that anyone who loves history and trivia, along with a fondness for arguing over ephemeral measures of "importance," will never want to put down.

2 Summary

The book is divided into two main parts and several appendices. Using Wikipedia (and Google nGrams for additional analysis) the authors rank order every *person* that is significant enough to have a Wikipedia page. This ranking of over 800,000 individuals serves as a jumping off point for the authors to explore a number of topics related to gravitas, celebrity, and the ways in which we are remembered (or not).

The first several chapters provide an overview of the datasets and ranking methodologies that are used throughout the rest of the book. There are also several different visualizations that are explained in the first chapters. These give the reader an understanding of the sheer volume of data that has gone into what is to follow. From the first chapter, Skiena and Ward jump in front of many of the questions of bias and sufficiency that hard core academics would have about this undertaking. Skiena and Ward do a nice job of answering many of these questions and emphasize that the methods employed can, despite the inherent bias of using the English language Wikipedia to compare the significance of topics as diverse as American presidents and Chinese artists, shed new light on some interesting questions. The key point being that the content of the open access Wikipedia does capture an idea of *memes* or *mindshare* — the importance that the users of Wikipedia give to the topics. While a historian or literary critic may question the significant conclusions that can be drawn from such a study, the results are still informative and entertaining in their own right.

I found the "Who Belongs in Bonnie's Textbook?" chapter to be one of the most interesting. The chapter focuses on a 5th grade history textbook and its list of "250 figures highlighted in the text." Skiena and Ward walk through many of the figures in the textbook asking, "why this figure and not that one."

This comparison is investigated not only their significance scores, but the required nature of some of these figures for different state curricula. They identify several important omissions and make some well reasoned, concrete suggestions for changes to the contents of the history text. This type of analysis, a more objective and critical look at what our children are learning, is one of the big messages that the book can lay claim too.

The later chapters of Part I compare the compiled significance rankings to other measures of historical merit, specifically, the Baseball Hall of Fame and the Hall for Great Americans. As the authors propose these two institutions as qualified arbiters of historical importance and compare the results of their method (and hindsight) to the selection processes for the two halls. I found the discussion (repeated in later sections) of the year over year significance of those included in the respective halls to be of great interest. Seeing the relative significance drop over a number of years, "to clear the backlog of qualified candidates," was interesting and raised a point I hadn't thought of before. Namely, it's hard to select a set number of well qualified individuals for an award that is given annually; awardees may just slip in because someone has too. The rest of discussion in these sections is lively and informative. There is an interesting analysis of the ways in which fame decays, which allows the authors to discuss their "decay" model, which allows them to compare historical figures to more modern ones. These chapters lay the remainder of the foundation for may of the interesting patterns that are observed in Part II.

Part II of the book is, for me, the most entertaining. The authors take us on a journey through a dizzying array of domains, ranking figures in each, with chapters on American political figures, world leaders, science, religion, sports, and the arts. Each section contains a breakdown of the significance scores of key figures in the respective area along with a conversation of why they are significant, and some interesting comparisons to other measures of significance, such as Nobel prizes and Oscars. This section is where the book really shines, I can imagine (and experienced) the book facilitating a number of heated discussions about who was left out of the rankings and why. Some of the most interesting points raised in these sections concern the unique perspective of combining the Wikipedia and Google nGrams data sets. By examining when certain figures were active versus when they were written about, it is possible to visually understand how success can emerge (and be maintained or decline) posthumously. These mixed graphs were among the most interesting for me throughout the book.

The more technical details of the ranking methodologies are saved for the first appendix. There is enough detail here for the curious to understand the general methodologies used and agree or disagree with them. The other appendices included are more playful in nature and contain the most famous people to live or die on a particular date, information about the "Whose Bigger" game and webpage, and brief biographies of the 100 most significant figures in the book.

3 Opinion

This is a well written book that I enjoyed reading. While not technically deep it kept me engaged and reading throughout. Skiena and Ward approach the subject with a mix of humor and rigor that is unique and fun. The book can serve as a springboard for anyone interested in rating and ranking. In addition to being a fun popular science book I could see sections of it being successfully integrated into a statistics or data analytics course.

I can't help but feel that there was an opportunity with this type of analysis that was missed. I would have liked to see a longer version of the text, including a historian, or other domain expert author, to comment on the chapters. Mixing these viewpoints to compare and contrast the rankings would have taken the book from good to great. It would have allowed a dialog to form within the text. The authors attempt to facilitate such a dialogues in the book but I feel the addition of a real counterpoint perspective would have made this book truly phenomenal.

However, this criticism does not stop the book from being fantastic for what is there. I highly recommending the book for anyone interested in history, sports, or getting into good arguments about what's important for being remembered.