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. **Theoretical Computer Science: Introduction to Automata, Computability, Complexity, Algorithmics, Randomization, Communication, and Cryptography** by Juraj Hromkovic. Review by Richard Jankowski. I can do no better than to paraphrase the first line of the review: *This book is an undergraduate-level textbook covering the foundational topics in theoretical computer science. Assuming a background in logic, the reader is taken through a high-level introduction to the underlying theories of computer science. While a relatively short book, Hromkovic provides a great overview to anyone interested in not only the major foundational aspects of computer science, but also how they interrelate and work together.*

2. **The Mathematics of Life** by Ian Stewart. Review by Aaron Sterling. I can do no better than to quote the review: *In an easy, flowing read, with dozens of diagrams and scholarly footnotes—but without a single formula—he introduces the reader to a wide range of interactions between mathematicians and biologists.*

3. **Universal Semantic Communication** by B. Juba. Reviewed by Wesley Calvert. This is a monograph based on the author's PhD thesis. The origins of the problem it addresses are daunting: *Is meaningful communication possible between two intelligent parties who share no common language or background?* The book itself focuses more on how network protocols can be designed.

4. **Graph Algorithms (2nd edition)** by Shimon Even, edited by Guy Even. Review by Anthony Labarre. This is a revised edition of Shimon Evens classic introductory book on algorithmic graph theory. Its focus is mostly on algorithmic graph theory.

5. **How to Fold It** by J. O'Rourke. Review by Brittany Terese Fasy and David L. Millman. Origami can be viewed in many different ways. Most people view it as a fun diversion. There are incredibly diverse and complex shapes one can make; however most people do not think in terms of algorithms and complexity when doing origami. One can also view the various ways to do origami in terms of algorithms and complexity. O'Rourke and Demaine wrote *Geometric Folding Algorithms*, a book aimed at a graduate students and researchers. (And reviewed in SIGACT NEWS Vol 42, No. 1, 2011) This current book is a midground. It can be seen as a lite version of *Geometric Folding Algorithms.*

6. **Bioinformatics for Biologists** Edited by Pavel Pevzner and Ron Shamir. Review by Dimitris Papamichail. This is a collection of articles meant to help get a biologist up to speed in bioinformatics. They are grouped by category and really do stick to their mandate of being introductory.

7. **Extremal Combinatorics with Applications to Computer Science (2nd edition)** by Stasys Jukna. Review by Miklós Bóna. To quote the review: *The book is a collection of very short chapters, each of which is focused on one proof technique. It is a very hands-on book; everything is taught by examples. The 27 chapters are grouped into five parts, proceeding from the ones that are probably interesting for the widest audiences towards the more specialized ones.*

8. **Enumerative Combinatorics, Volume 1, Second Edition** by Richard P. Stanley. Review by Angèle M. Hamel. Normally combinatorics is about counting. And that is still true here. But Enumerative Combinatorics wants to do more. It wants to actually list out (or give an algorithm for it) the objects. This book is the classic in the area.

9. **Combinatorial Optimization** by B. Korte and J. Vygen. Review by Akash. The very title of this book conjures up a vast amount of material. Combinatorial Optimization is about optimizing discrete problems. This includes Linear Programming, other problems in P, Integer Programming, other problems thought to not be in P. And the people in the field want ANSWERS- not this *Sorry, the problem is NP-complete we can't help you* bullsh*t that one often hears. Suffice to say its a heavy book both literally and intellectually.

10. **The Golden Ticket: P, NP, and the Search for the Impossible** by Lance Fortnow. Review by Cynthia DiPaula and Andrew Wonnacott. This is a book on P vs NP for the layperson. The ideal readers are be high school students who are interested in computer science and some math. The reviewers are be high school students who are interested in computer science and some math.

11. **Probably Approximately Correct** by Leslie Valiant. Review by Joshua Brulé. This is a book on Probably approximately correct (PAC) learning for the layperson. As such its focus on how this concept can be used in other domains (e.g., evolution).

BOOKS I NEED REVIEWED FOR SIGACT NEWS COLUMN

Algorithms

1. *Algorithms Unplugged.* Edited by Vocking et. al.
2. *The LLL Algorithm.* Edited by Nguyen and Vallee.
3. *Networked Life: 20 questions and answers* by Mung Chiang
4. *Greedy Approximation* by Vladmir Temlyakov.
5. *Basic Phylogenetic Combinatorics* by Dress, Huber, Koolen, Mouton, Spillner
6. *Integrated Methods for Optimization (second edition)* by John Hooke
7. *Proofs and Algorithms* by Dowek.
8. *A guide to experimental algorithmics* by Mcgeoch.

Logic

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

Cryptography and Coding Theory

1. *Understanding Cryptography: A Textbook for Students and Practitioners* by Paar and Pelzl.
2. *The Block Cipher Companion* by Knudsen and Robshaw.
3. *A Student's Guide to Coding and Information Theory* by Moser and Chen.
4. *Algebraic Shift Register Sequences* by Goresky and Klapper.
5. *Variable-length codes for data compression* by Salomon.

Misc

1. *Visions of Infinity* By Ian Stewart.
2. *Selected Papers on Computer Languages* by Donald Knuth.
3. *Analytic Combinatorics in Several Variables* by Pemantle and Wilson.
4. *Handbook of finite fields* by Mullen and Panario.
5. *Digital Dice* by Paul Nahin.

Review of[2]
Theoretical Computer Science: Introduction to Automata, Computability, Complexity, Algorithmics, Randomization, Communication, and Cryptography
by Juraj Hromkovic
Springer 2003, 323 pp., $89.95, Hardcover.
Review by
Richard Jankowski rjankowski@acm.org

1 Introduction

Theoretical Computer Science: Introduction to Automata, Computability, Complexity, Algorithmics, Randomization, Communication, and Cryptography by Juraj Hromkovic is an undergraduate-level textbook covering the foundational topics in theoretical computer science.

Assuming a background in logic, the reader is taken through a high-level introduction to the underlying theories of computer science. While a relatively short book, Hromkovic provides a great overview to anyone interested in not only the major foundational aspects of computer science, but also how they interrelate and work together.

2 Summary

Chapter 1 covers a high-level introduction to computer science, and the importance of studying the theoretical topics.

Chapter 2, *Alphabets, Words, Languages, and Algorithmic Problems* serves as a background and introduces the formal language of computer science and the ways of representing problems and objects. The fundamental terms of alphabets, words, and language are used to define the high-level notion of an algorithm as expressed by a program, and used to help describe formal language theory.

Chapter 3, *Finite Automata* introduces the reader to simple computing models using finite automation. Topics such as computation, non/determinism, and simulation are covered. The material is used to bring the reader to the first steps of defining and modeling algorithms.

Chapter 4, *Turing Machines* is formally introduced, including multi-tape machines and the comparison of programming languages to the Turing machine models. The chapter builds the foundation upon which later chapters on computability and complexity rely.

Chapter 5, *Computability* is an introduction to the theory and shows the reader that some languages cannot be accepted by any Turing machines.Topics such as the diagonalization and reduction methods, Rice's Theorem, and the Post Correspondence Problem are covered.

Chapter 6, *Complexity Theory* continues the ideas of the previous chapter by showing that even with computational problems, there may not be efficient solutions to solve them. This chapter introduces the reader to the NP-completeness concept, where tractable and intractable problems are defined.

Chapter 7, *Algorithmics for Hard Problems* covers topics such as approximation algorithms and heuristics, in an effort to reduce the complexity of algorithms to make them much more computationally practical.

Chapter 8, *Randomization* is an extension of the material found in Chapter 7, dedicating an entire chapter to the topic. The reader is taken through probability theory, and then the development of a randomized protocol to synchronize two systems along a communications link.

Chapter 9, *Communications and Cryptography* is dedicated to covering issues with communications. Topics such as public-key cryptosystems and zero-knowledge proof systems are introduced, and the chapter closes with a coverage of issues with interconnected networks.

3 Opinion

Theoretical Computer Science is a well-written introductory book covering the most important concepts. Within the confines of a relatively small book, the author did a tremendous job at introducing the reader to the high-level concepts.

The book requires a grasp of logic on the part of the reader, but motivated undergraduate computer science students should be able to work their way through the material without undue difficulty.

I found the material to be presented in elegant terms, making the book a pleasure to read. Its light weight and breadth made this book a very nice resource. As a lot of material is covered in a short amount of space, the author did a great job at citing more resources if the reader wishes to explore each topic further.

Review of[3]
The Mathematics of Life
by Ian Stewart
Basic Books/Profile Books, 2011
358 pages, Hardcover
US$16.68 on Amazon.com

Review by Aaron Sterling `sterling@iastate.edu`

1 Introduction

Ian Stewart is one of the premier popularizers of mathematics. He has written over twenty books about math for lay audiences, including the well-received *Flatterland* and *Professor Stewart's Cabinet of Mathematical Curiosities*. He has also co-authored science fiction, and books on the science of science fiction (three books on "the science of discworld"). In his newest effort, *The Mathematics of Life*, Stewart focuses his talents on the mathematics of biology, and the result is superb. In an easy, flowing read, with dozens of diagrams and scholarly footnotes—but without a single formula—he introduces the reader to a wide range of interactions between mathematicians and biologists. I heartily recommend this book.

2 Summary

The Mathematics of Life contains 19 chapters. Chapter 8, "The Book of Life," focuses on the Human Genome Project, and algorithmic challenges of DNA sequencing. However, as this possibly the area most familiar to SIGACT News readers, I will only mention it briefly, and, instead, focus on chapters that introduced me to areas of mathematical biology I had not previously encountered.

Perhaps the most direct connection to (the roots of) theoretical computer science comes in Chapter 13, "Spots and Stripes," where Stewart considers Alan Turing's famous paper, *The Chemical Basis of Morphogenesis*, and sketches the development of biological thought about animal markings since Turing's groundbreaking proposal. As Stewart says:

> For half a century, mathematical biologists have built on Turing's ideas. His specific model, and the biological theory of pattern-formation that motivated it, turns out to be too simple to explain many details of animal markings, but it captures many important features in a simple context, and points the way to models that are biologically realistic.

Turing proposed"reaction-diffusion" equations to model the creation of patterns on animals during embryonic development. As noted by Stewart, Hans Meinhardt, in *The Algorithmic Beauty of Seashells*, has shown that the patterns on many seashells match the predictions of variations of Turing's equations. The mathematician James Murray extended Turing's ideas with wave systems, and proved the following theorem: a spotted animal can have a striped tail, but a striped animal cannot have a spotted tail. Intuitively, this is because "the smaller diameter of the tail leaves less room for stripes to become unstable, whereas this instability is more likely on the larger-diameter body."

In the very next chapter, "Lizard Games," algorithmic game theory makes an appearance. Stewart introduces us to Barry Siverno's studies of side-blotched lizards. These lizards come in three morphs: orange-throated, blue-throated and yellow-throated. The orange males are the strongest, while the yellow males are the smallest and most female-colored. The blue males are the best at pair-bonding. So the oranges win when they fight the blues, the blues are preferable to the yellows, and—the kicker—the yellow males sneak away with the females while the oranges fight the blues. This suggests an evolutionary game where

- orange beats blue
- blue beats yellow
- yellow beats orange.

Stewart introduces von Neumann's minimax theorem, Smith's definition of evolutionarily stable strategies, and other game-theoretic concepts. He then discusses what "survival of the fittest" means in a context where there is no one clear winner. (One of Stewart's themes throughout the book is an attempt to give the reader a more sophisticated understanding of Darwinian evolution.)

Also in this chapter, Stewart gives many examples of evidence for evolution, and ways in which evolutionary theory has developed since the time of Darwin. For example, the conventional biological wisdom at one time was that *sympiatric speciation* was impossible. Roughly, sympiatric speciation occurs when one species develops into two distinct species in a single geographic area. For many years, it was believed that groups of animals had to be geographically separated for speciation to occur. Nevertheless, this conventional wisdom appears to be false, both because of empirical evidence, and more sophisticated mathematical models. In Stewart's words:

> There are two main forces that act on populations. Gene flow from interbreeding tends to keep them together as a single species. Natural selection, in contrast, is double edged. Sometimes it keeps the species together, because collectively they adapt better to their environment if they all use the same strategy. But sometimes it levers them apart, because several distinct survival strategies can exploit the environment more effectively than one. In the second case, the fate of the organisms depends on which force wins. If gene flow wins, we get one species. If natural selection *against* a uniform strategy wins, we get two. A changing environment can change the balance of these forces, with dramatics results.

There are other computer-science-related chapters: "Networking Opportunities," which introduces graph theory; "What is Life?" which introduces cellular automata and von Neumann's replicating automaton. However, I will use the remaining space in this review to discuss a chapter that relates more to pure mathematics.

Chapter 10, "Virus from the Fourth Dimension," tells the story of scientists identifying viruses with X-ray diffraction and similar methods. In several chapters, Stewart discusses the mathematical importance of symmetry and symmetry-breaking, and this chapter is no exception: the herpes simplex virus is mirror-symmetric and has 120 symmetries. It turns out that many viruses are coated with chemicals in the shape of an icosahedron. These *icosahedral virus coats* are made of triangular arrays 9of capsomers, small self-assembling proteins.

Unfortunately, the mathematics of pure icosahedra did not quite match what was empirically observed. Inspired by the geodesic domes of Buckminster Fuller, Donald Caspar and Aaron Klug in 1962 proposed a theory of *pseudo-icosahedra* to model virus coats. (Expert geometers were already familiar with pseudo-icosahedra, but most mathematicians were not.) While this provided an excellent model of many viruses, over the next forty years, research teams found structures that could not be explained using the Caspar-Klug theory. Finally, starting in the year 2000, the mathematician Reidun Twarock and co-authors proposed a unifying framework, using higher-dimensional geometry.

Twarock introduced a *viral tiling theory* that uses the six-dimensional icosahedral symmetry group, then takes a cut from that 6D lattice and projects it into three dimensions. This approach accurately "predicts" both the pseudo-icosahedral virus coats, and also the exceptional virus coats that were observed after 1962.

3 Opinion

This book is full of great material I did not mention at all. Most of the early chapters are short, and introductory, which is why I focused on the later chapters in this review. The prose style is friendly and clear throughout, without talking down to the reader.

I consider this to be an excellent introduction to the mathematics of biology, for both amateurs and professionals. Seasoned researchers are likely to learn "teasers" about areas unfamiliar to them, and smart people "afraid of math" can read the book and enjoy the material. Highly recommended. I will conclude this review with the same words Stewart used to conclude the book:

> Instead of isolated clusters of scientists, obsessed with their own narrow specialty, today's scientific frontiers increasingly require teams of people with diverse, complementary interests. Science is changing from a collection of villages to a worldwide community. And if the story of mathematical biology shows anything, it is that interconnected communities can achieve things that are impossible for their individual members.
>
> Welcome to the global ecosystem of tomorrow's science.

Review of [4]
Universal Semantic Communication
Author: B. Juba
Publisher: Springer, 2011
ISBN 978-3-642-23296-1, $129, 397pp.

Reviewer: Wesley Calvert (wcalvert@siu.edu)

1 Overview

Consider the problem of networking protocols. In the most naive method, to release a new protocol for some common transaction in a large-scale network would require a deployment of new software for every application that uses the protocol on every device on the network. This is clearly impractical. The second approach is to ensure that the new protocol is backward-compatible: it tolerates the use of the old protocol by other devices on the network.

The question posed in the monograph under review is whether a third way is possible: to have the protocol negotiated at the time of use, in a context where the devices on the network can assume nothing about the meaning of signals coming from another device. This is universal compatibility. Juba expresses the problem in the following way: Is meaningful communication possible between two intelligent parties who share no common language or background?

Of course, this question is a good deal broader than that of network protocols. In fact, some of the pre-existing literature on the question came from the context of communicating with extra-terrestrials.

The analysis of the question calls for a very different model of communication than the traditional Shannon model [4]. Shannon posits two users with a physical channel, and asks about the capacity of the users to exchange data over the channel. The content and meaning of that data is external to the model. Even in the context of cryptanalysis [5], Shannon's fundamental question is what the analyst can do (if anything) to share, at the end, the same string of data the sender had in the beginning. It is assumed that the eavesdropper, in possession of that string, would be able to interpret it.

This approach is insufficient for the present work. If I send you, by a noiseless channel, a binary string representing a document, you have no way of knowing whether to read it as ASCII text, as a bitmap of the image (or in what order the pixels would be presented), or in some other form. If I transmit a voice signal in the Navajo language, receipt of the sound data, however exact, tells you nothing important about the content of my message unless you understand Navajo.

Juba solves this issue by formalizing a different aspect of communication: Communication always takes place in order to achieve some goal. That its use is the essence of language and is definitive of the "meaning" of a speech act is consonant with many important insights of post-1920 philosophy of language [3, 2, 6].

Armed with this insight, Juba assumes a context of n parties, of which the "user" u, the "server" s, and the "environment" e are distinguished. Each party i has state space $\Omega^{(i)}$, and between each pair (i, j) of parties, there is a channel with state space $\Omega^{(i,j)}$. The entire system has for its state space Ω, the product of all the state spaces of the parties and the channels. A strategy for party i

is a map taking an element of $\Omega^{(i)}$ and the states of all channels directed into i, and producing an element of $\Omega^{(i)}$ and a state for each outgoing channel from i. A *goal* is determined by a subset of $\Omega^{(e)}$ to be attained and a range of strategies the environment may possibly adopt. Communication is said to be successful if the process terminates with the environment in one of the goal states irrespective of which of the goal strategies the environment adopts.

2 Summary of Contents

The book under review is a research monograph arising as a revision of the author's Ph.D. thesis. The imprint of this history remains with the first-chapter literature review and informal distinction of the work of the book from the work of the prior literature. Aside from this, and a concluding chapter on future research directions, the book splits neatly into two parts: the first part (Chapters 2-5) covers the theory for terminating systems, and the second part (Chapters 6-9) for reactive systems.

Chapters 2 and 6, respectively, give the formal definitions (the terminating case is given above; the reactive case is the natural adaptation). In each case, the author proves, up to some obviously necessary conditions, that there are "universal users," who can successfully communicate anything, i.e. achieve any goal, that could possibly be communicated under these circumstances.

The complexity cost, as one would expect, is heavy. An important obstruction is the so-called "password-protected" servers, who cheerfully achieve the goal as soon as the user submits the right password as a prefix of the transmission. For a universal user to communicate with such a server, it would have to be prepared to scan all possible passwords, at exponential time cost. Chapters 4 and 8 fill out the details of the complexity situation, and address natural fixes to it: In the terminating case (Chapter 4), the author proves that one can achieve a graceful decline in performance. In Chapter 8, for reactive systems, the author shows a more efficient solution for a special case in which less universality is demanded.

While chapters 4 and 8 examine, from an external perspective, the complexity of universal computation, Chapters 5 and 7 explore the internal complexity theory of the model of computation constituted by agents cooperating to achieve goals in the universal communication model. The main message here is that a meaningful complexity theory is possible in this context.

Chapters 3 and 9 work out sample applications. In chapter 3, the motivating problem is essentially the Turing test, or the CAPTCHA problem: the user wants to determine whether the server is a computer or a human. In Chapter 9, which runs fully a fifth of the length of the main text of the book, the problem is communication under variable TCP-like protocols.

3 The Successes of the Book

Two tests measure the success of the present book:

1. Are the formalizations convincingly the right formalizations of the central problem?

2. Do the solutions to the formalized problem address the real concerns which gave rise to the problem?

The second question is the easier to answer. The analyzes of CAPTCHA and network communication protocols are at least persuasive initial evidence in favor of the practical worth of the theory.

The complexity lower-bounds given, and the existence of universal users are clearly interesting points, and responsive to the design issues.

The first question is a good deal harder, and essentially must be. On the one hand, the fact that the author's formalization of the universal communication problem admits the proof of the existence of universal users, the lower bounds, and the complexity mitigators shows that it has considerable value. It is also an intuitively compelling formulation; it has a feel which is (intentionally) familiar from interactive proofs and from game theory, both of which are very reasonable models for the interaction.

Granting that all models must have some limitations, this model is not without them. One issue is that all unsuccessful attempts at communication are treated alike. I am indebted to communication scholar Jason Jarvis for the following example. If I tell a joke, the goal may be that my audience laughs. However, I would certainly want to distinguish between the following unsuccessful cases:

1. My audience simply finds my statement incomprehensible.
2. My audience understands the joke, but doesn't find it funny, so doesn't respond at all.
3. My audience is offended at the joke, and becomes angry.

We could as easily talk about a printer that misunderstands by doing nothing, or misunderstands by printing three hundred pages of gibberish on expensive rag bond paper.

The book is certainly readable. The initial informal discussion of the theme certainly makes one yearn for the precise definitions, but the author gives fair warning of this feature, and tells the reader where to flip ahead to see them. The reader is assumed to be familiar with very basic complexity theory (PSPACE, for instance). Knowledge of interactive proofs is helpful, but certainly not necessary. The lack of an index is more than a little off-putting. On the whole, though, I would have no reservations about recommending this book to a mathematically mature graduate student with access to Arora-Barak [1], but with little specific background. The book, as a research monograph, is suitable for those interested in its techniques or problems.

Discussion of the present book achieving its goals feels like it should flirt with self-reference, but the book is, on the whole, successful. A reasonable model for an important problem is presented and defended, and some significant cases of that problem are solved.

References

[1] S. Arora and B. Barak, *Computational Complexity*, Cambridge, 2009.

[2] K. Burke, *A Rhetoric of Motives*, University of California Press, 1969, first published 1950.

[3] J. Dewey, *Experience and Nature*, Norton, 1929, first published 1925

[4] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal* 27 (1948) 623–656.

[5] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal* 28 (1949) 656–715.

[6] L. Wittgenstein, *Philosophical Investigations*, Basil Blackwell, 2001, first published 1953.

[7] E. Zegura, et al., “Network Science and Engineering (NetSE) Research Agenda,” A report of the network Science and Engineering Council, Release Version 1.0,” 2009.

Review of[5]

Graph Algorithms (2nd edition)
by Shimon Even, edited by Guy Even
Cambridge University Press, 2011
202 pages, Paperback, $25.00, ISBN: 9780521736534
(also available as HardcoveR, $85.00, ISBN: 9780521517188

Review by Anthony Labarre, Laboratoire d'informatique Gaspard-Monge, (`anthony.labarre@univ-mlv.fr`)

1 Introduction

Shimon Even's book is an introductory book to algorithmic graph theory. The author intended to publish this new edition of this book himself, but passed away before he could finish this task, which was eventually completed by his son, Guy Even.

Not owning the original edition, which was apparently a classic book at the time of its publication (1979), I cannot recount how the book evolved, but Guy Even gives sufficient details in the preface. Both editions mainly differ by the omission of two chapters in the second edition, which were about NP-completeness; this is understandable, since the subject was new back at the time of the book's first publication, but is well-covered nowadays in many textbooks.

2 Summary

Chapter 1 introduces the basics of graph theory, as well as the standard means of representing graphs (adjacency matrices and incidence lists). It then moves on to Euler graphs and de Bruijn graphs and sequences, and concludes with the well-known shortest path algorithms (Dijkstra's, Ford's and Floyd's), carefully distinguishing between the different situations in which each of them should be used.

Chapter 2 is devoted to trees, and begins with various theorems that characterize them. It also discusses Prim's algorithm for finding a minimum spanning tree, and features a section on the problem of enumerating undirected trees on n vertices, where Cayley's theorem is proved using Prüfer sequences. The chapter ends with a section on directed trees and one on König's infinity lemma.

Depth-first search is the subject of chapter 3, where Trémeaux's algorithm as well as Hopcroft and Tarjan's algorithm are presented. The last section of the chapter illustrates how depth-first search can be used to find the strongly connected components of a directed graph in linear time.

Chapter 4 is about ordered trees and *codes*, i.e. sets of distinct words over a given alphabet, and presents an algorithm for testing *unique decipherability*, which answers the question of deciding whether any message encoded using a given code can be recovered in a unique way. It then explains Huffman coding, which provides a way of finding a code that is optimal with respect to a given alphabet and the probabilities of occurrence associated to each character, and illustrates an application of the associated Huffman tree to sorting by merges. The chapter ends with Catalan numbers and a proof of the formula for enumerating well-formed sequences of parentheses.

Chapter 5 tells us about the well-studied subject of finding maximum flows in networks, and its equivalence with finding a minimum cut. The author explains two algorithms for solving this problem: Ford and Fulkerson's, and Dinitz's. The subsequent section is devoted to the variant where the flow associated to each edge is bounded (from above and from below).

Chapter 6 then turns to applications of flows in networks and starts with a complexity analysis of Dinitz's algorithm in the special case where the capacity of each edge is 1, after which the author discusses how to use network flow techniques to answer (edge- or vertex-) connectivity questions in (directed or undirected) graphs. He then moves on to the subject of finding maximum cardinality matchings in bipartite graphs, and concludes the chapter with the critical path method in PERT charts.

Chapter 7 is about planar graphs, and begins their study with a proof of Kuratowski's well-known characterization of planar graphs in terms of forbidden minors. A short subsection follows on the Euler characteristic relating the vertices, edges and faces of a plane graph. The chapter ends with a discussion on duals of connected graphs.

Chapter 8 is about actually testing graph planarity, and presents two algorithms for achieving this: Hopcroft and Tarjan's path addition algorithm, and Lempel, Even and Cederbaum's vertex addition algorithm.

3 Opinion

The book is an excellent introduction to (algorithmic) graph theory, and seems to be a good choice for a class on the topic, or for self-study. Each chapter comes with its own selected bibliography, and ends with a collection of problems to help the reader check his or her understanding of the material presented in that chapter. Proofs are always provided and are also the topic of a few selected exercises. Algorithms are presented in pseudocode only; the reader who is more interested in implementing graph algorithms can complement this book with Sedgewick's [1].

The range of subjects covered in this introductory book might appear a little restricted; additional books might be required depending on the goal an instructor wants to reach when teaching a class on the subject of graph algorithms (for example, graph parameters that relate to traditional or parametrized computational complexity). I also find the choice of some of the topics odd: although they are interesting, I find it surprising to read about de Bruijn graphs or Huffman coding in an introductory book to graph algorithms. These minor issues should however not overshadow the quality of this book, which presents all concepts, algorithms and applications in a very clear way.

References

[1] Robert Sedgewick. *Algorithms in C++ – part 5: graph algorithms.* Addison-Wesley-Longman, 3rd edition, 2002.

Review[6] of
How to Fold It
Author: J. O'Rourke
Cambridge Univ. Press, 2011
171 pages, Softcover

Review by
Brittany Terese Fasy bfasy@cs.cmu.edu
and David L. Millman dave@cs.unc.edu

1 Introduction

In *How to Fold It: The Mathematics of Linkages, Origami, and Polyhedra*, Joseph O'Rourke explains the mathematics behind folding low-dimensional objects. This book is written for a general audience.

O'Rourke, along with Erik Demaine, also wrote *Geometric Folding Algorithms*, a book aimed at a graduate students and researchers. *How to Fold It* can be seen as a lite version of *Geometric Folding Algorithms*. Both books are carefully designed, well written, visually appealing and fun to read. Tailored to assume only a high-school math background, *How to Fold it* leads the reader to develop a mathematically sophisticated perspective of geometric problems.

2 Summary

This book deals with folding (and unfolding) in one, two, and three dimensions–Parts I, II, and III, respectively.

Part I: Linkages. (Chapters 1-3) Using the desk lamp as an example of a one-dimensional linkage, this book begins strong by appealing to the intuition of the reader. The first part of the book deals with one-dimensional folding problems, or linkages. The three topics highlighted are: robot arms, pantographs, and fixed angle chains.

The model for a robot arm is very simple at first: each arm is a segment and the arms connect at joints. For the early analysis, intersecting segments and the feasibility of the joints actually lying in a plane are ignored. When mathematical concepts are needed (for example, the distinctions among theorems, lemmas, propositions, and corollaries), well-articulated definitions are given. The first theorem of the book states that the reachability region of an n-link robot arm is an annulus. Although this theorem is more advanced than any theorem that would be introduced in a high school geometry class, the statement (and the proof by induction) of this theorem is understandable to the novice geometer.

In the context of this book, a pantograph is a linkage that can be used to duplicate drawings at different scales than the original. For example, the engraving in a wedding band is done with a pantograph. (We ignore the second definition of a pantograph that the author mentioned in passing, but did not emphasize in this book). In addition to pantographs, Chapter 2 also discusses the Peaucellier linkage that can draw a straight line and Kempe's Universality Theorem, which states that any algebraic curve can be traced out by a vertex of some linkage, i.e., there exists a linkage that can sign your name. No details of the proof of Kempe's Universality Theorem are given, but an example of a linkage that traces out a crude approximation of the letter J from *John Hancock* alludes to the difficulty of such a proof. This chapter ends with an open problem which asks if it is possible to create a planar linkage that signs your name. A simpler sub-problem is also proposed, which is: Are there planar lineages for the numbers 0, 1, . . . , 9? The only digit for which a planar linkage is known is the number 1, which is solved by the Peaucellier link.

The last chapter of Part I considers two seemingly unrelated topics: protein folding and pop-up spinners. After a brief discussion of proteins, a simplified protein model is presented. The simplified model, called a 90 deg-chain, is an open chain where each link is unit length, angles are fixed at 90 degrees, and only dihedral motions are allowed. The author discusses finding a configuration of a 90 deg-chain that achieves the maximum end-to-end distance, called the maxspan. The chapter concludes by explaining pop-up spinners, guiding the reader though a construction, and showing a connection between pop-up spinners and maxspan.

Part II: Origami. (Chapters 4-6) Although origami started over 1000 years ago in Asia, the computational aspects of origami, often called *computational origami*, started in the 1990's. Part II highlights some of the advances in computational origami, emphasizing how accessible the problems are to both the expert and the novice. In particular, the following three problems/theorems are discussed: single vertex flat folds, the fold and one-cut theorem, and the shopping bag theorem.

Chapter 4 discusses single-vertex flat foldings. A vertex is any point not on the paper boundary where two creases meet. A single-vertex flat folding is when the crease patter has only one vertex and folds flat without cutting along any crease or introducing new creases. The author gradually builds up to the necessary and sufficient conditions for the existence of a flat folding, called the Kawasaki-Justin theorem. The theorem says that the difference between the number of valley folds and mountain folds at a vertex is exactly two. Additionally, map foldings (a folding such that adjacent crease lines that meet at a vertex are orthogonal) are discussed. The author challenges the reader to fold the given folding pattern in Figure 4.14. (We tested this ourselves and verified that it is indeed difficult to do!)

Chapter 5 reveals one of the secrets of Harry Houdini's magic: for any straight line drawing, it is possible to fold the paper and make one straight-line cut in order to cut along all edges of the drawing and nothing else (even if the drawing is not connected). This is called the "fold and one-cut theorem". Even after learning the mathematics behind this magic trick, it is still surprising that it can be done.

Rigid origami deals with stiff paper, where the faces cannot bend in order to go from one configuration to another. Perhaps it is a surprising fact that the brown paper shopping bag cannot be folded/unfolded using rigid origami. The fact that there are only two possible rigid origami configurations (open or closed) of the tall paper bag is proven in Chapter 6.

Part III: Polyhedra. (Chapters 7-9) Whereas Part II deals with folding origami, Part III works with the opposite problem: unfolding polyhedra (a piecewise-linear surface in 3D). First, we define a net: a drawing of the faces of a surface to a planar layout such that no two faces share more than one edge and each face shares at least one edge with another face. Dürer's problem asks if every convex polyhedron has a net. This problem is solved (positively) for the case of a tetrahedra, and it is also known that there exist non-convex (triangulated) polyhedron without an net. It still remains an open problem, however, as to whether or not every convex polyhedron has a net.

The end of this book is filled with (perhaps surprising) theorems interspersed with many open problems. For example, given a planar polygon, there is not a unique way to fold it to obtain a convex polyhedra. However, if one was to assign gluing rules (in other words, to impose a convex polyhedral metric), then the convex polyhedral surface obtained (if one can be obtained) is unique. This is known as Alexandrov's theorem. It is known that every polygon folds to some polyhedra, but the question remains as to how one can characterize polygons that fold to convex polyhedra.

3 Opinion

How to Fold It is aimed at a novice audience as it appeals to intuition and explicitly defines all mathematical terminology, without sacrificing quality or completeness. In fact, the only prerequisite is a high school geometry class. Many exercises of varying difficulty levels are given throughout the book. In addition, solutions are given for all of the problems. (Well, except for the open problems!)

In the author's own words, this book provides "tangible constructions with rich mathematical content" and that is what is, in fact, delivered via easy to digest language. This book covers the basics that are expected, such as the platonic solids and convexity, but also gives the reader many open problems to ponder. We especially like how this book emphasizes the approachability of research in computational geometry. For example, the *Two Kinks* theorem (found on page 18) was solved by John Kutcher when he was a college student. In addition, history is given throughout, giving the reader an appreciation of the advances made.

One of the most fascinating parts of this book is that it presents open problems to a non-expert reader. In a way, the author is teasing the reader into thinking more deeply about these problems. However, publishing open problems in a text book is dangerous, as they can be solved and make the book dated. O'Rourke avoids this potential problem by using the website www.howtofoldit.org in order to keep the reader up to date about the status of the open problems mentioned in the book. In addition, the website provides additional open problems, videos and illustrations of concepts described in the book, and templates that the reader can print and physically manipulate. If the reader does choose to test out the folding on his/her own, then there are hints given in the text to assist in the more difficult maneuvers (for example; see page 58). Upon trying to fold one of the examples ourselves – the one given in Figure 4.14 – we determined that the exercise is not as easy as one would think!

As for the negatives for the book, there is only one in our opinion. The colors used to depict valleys and mountains for origami folding are red and green, two colors which are very difficult for individuals with color vision deficiency. It would be easier to differentiate the two types of folds if the type of line were also different (or if different colors were used). That being said, the author explained on page 57 that he chose to use color over $- - -$ and $- \cdot - \cdot -$ since these two patterns can be easily confused.

In summary, *How to Fold It* is a great book for someone who wants to learn about the mathematics behind origami without being overwhelmed by the mathematics itself. This is a great book for a high school or undergraduate student to get introduced to the open problems in computational origami.

Review of[7]
Bioinformatics for Biologists
Edited by Pavel Pevzner and Ron Shamir
Cambridge University Press, 2011
362 pages
Softcover $55.00, Hardcover $110.00

Review by Dimitris Papamichail, University of Miami, dimitris@cs.miami.edu

1 Introduction

The field of bioinformatics has grown to encompass a wide range of research areas involving the analysis, storage and retrieval of biological data. The inherent complexity of cells and organisms, combined with the wealth of life science data produced by modern high throughput technologies, make the use of computers to analyze and process this data critical, and often imperative. Data analysis is routinely performed by specialists in the fields of computer science, statistics, biochemistry, and several others, making bioinformatics highly interdisciplinary. There exists nevertheless a growing need for biologists themselves to perform complex analysis on their data and comprehend the analytical techniques involved, even when utilizing existing tools. To enable collaborations, biologists may want to comprehend at least the basic principles of some of the most popular computational techniques, for the sake of meaningful interaction with their interdisciplinary peers. Just as computational bioinformaticians often study the principles behind PCR, sequencing, microarrays, and other biotechnologies, biologists may want to learn the algorithmic ideas behind approximate string matching, phylogenetic tree construction, multiple alignment, and other important computational methods.

The *Bioinformatics for Biologists* edited volume represents a meaningful step towards aiding biologists in their quest to introduce themselves to computational and statistical thinking in bioinformatic topics, or enhance their knowledge in the field. This review examines the contents of the book and evaluates its contributions towards those goals.

2 Summary

This book consists of 16 chapters in 5 parts, preceded by introductory material. The themes of the five parts are the following:

1. Genomes, spanning chapters 1 through 5,
2. Gene Transcription and Regulation, chapters 6 to 8,
3. Evolution, chapters 9 to 11,
4. Phylogeny, chapters 12 to 14, and
5. Regulatory Networks, chapters 15 and 16.

Most of the chapters, as expected in an edited volume, are relatively distinct and autonomous, with a couple of exceptions. The parts' structure and sequence is relatively typical for a bioinformatics text. The contents of each chapter are outlined below.

Introduction The first twenty pages contain a preface, acknowledgments, editor and contributor credentials, and a computational micro primer. In the latter the editors introduce concepts such as algorithms, computational complexity, big-O notation, and NP-completeness, in roughly four pages.

Chapter 1 The first chapter provides a short introduction to the large field of statistical genetics, establishing some of the methods and are used to model and analyze the correlation between genetic variation and phenotypic variation, such as susceptibility to diseases. Using simplified evolutionary models, the author explains how mutation and recombination events shape the genetic code of populations, and then proceeds to explain simple statistical tests that can help determine associations which are not expected to be random.

Chapter 2 This chapter examines the biological problem of determining patterns of variation among populations, when variations occur in blocks of single nucleotide polymorphisms. This problem is reduced to the set-cover and integer-programming problems, for which approximate and heuristic solutions are introduced.

Chapter 3 The third chapter discusses genome assembly. Co-written by Pavel Pevzner, who has extensively worked on the topic, the chapter explores the graph theoretical concepts that underlie most common methods used in genome assembly, such as Hamiltonian cycles, Eulerian cycles and De Bruijn graphs. The chapter skillfully delves into historical motivations, DNA sequencing techniques, and theorem proofs. Numerous useful illustrations and thoughtful insights into the ever relevant assembly problem and the algorithmics behind it make this chapter particularly educational and interesting.

Chapter 4 Dynamic programming is an algorithmic technique that utilizes the storage of intermediate results to efficiently navigate towards the optimal solution of an optimization problem, thus exchanging space for time. It can be successfully applied to a specific set of optimization problems, but many interesting biological problems do fall in that category. Mikhail Gelfand explains the principles of dynamic programming and introduces a general framework that is modularly extended to handle the shortest path, string alignment, and gene recognition problems.

Chapter 5 The fifth chapter uses paternity testing as a motivating example to explain statistical inference using Bayes' law. A few other applications, such as estimation of disease risk, are employed to explain different types of inferences. The chapter material builds gradually and the author accompanies all derived formulas with careful explanations, assuming minimal prerequisites.

Chapter 6 This chapter deals with the effects of basic processes, such as DNA replication and transcription, in shaping features of the DNA itself. The author presents his own work on applying and visualizing simple statistics to identify features of organisms that are shaped by the organisms' own processes, generating distinct genomic biases.

Chapter 7 Motif finding is a challenging but critical problem in identifying genomic areas of interest, particularly related to transcription. The seventh chapter explores transcription factor binding site properties, their computational representations, and methods to identify and predict them.

Chapter 8 The eight chapter delivers a short introduction to glycobiology, the study of structure and biology of sugar chains. It uses the ever popular influenza virus and its ability to jump from animals to humans as a motivation to explore molecular mechanisms of RNA viruses, but more so to describe glycan structures and experimental techniques to analyze them and determine structural motifs of importance.

Chapter 9 In chapter nine the authors examine genome rearrangements, which are large scale changes that occur in chromosomes when they break and the fragments reassemble. Identification of rearrangement events can be critical for the successful reconstruction of evolutionary relationships among genomes. Rearrangements have also motivated the study of a classical algorithmic problem in bioinformatics, sorting by reversals. The authors present variations of that problem, most of which are NP-hard, and describe approximation and other algorithms to solve them.

Chapter 10 An evolution part of a book would not be complete without a discussion of phylogenetic tree construction. The authors of the tenth chapter argue that forests and networks may be more appropriate structures to describe species relationships in the presence of horizontal gene transfer among prokaryotes. Nevertheless they describe a common bioinformatics pipeline that is used to construct phylogenetic trees from common proteins among given genomes, and examine some of the details involved in a few steps of the pipeline.

Chapter 11 Reconstructing the history of genomic changes and discovering variation within species can aid the understanding of both how genomes evolved and how changes affect them. Jian Ma offers a short introduction to the problem of reconstructing evolutionary history, and examines algorithms for the study of chromosomal rearrangements and phylogenetic analysis using gene order and orientation. The sorting by reversals problem is revisited.

Chapter 12 In a highly entertaining chapter, Ran Liberskind-Hadas introduces the concept of species co-evolution and its study through the reconstruction of cophylogeny, which involves mapping phylogenetic trees of co-evolving species onto each other. This problem has been shown to be NP-hard and different heuristics are examined, including genetic algorithms, which are introduced at a high level of detail. A large part of the chapter is devoted to the software that the author has created for studying cophylogeny, called Jane.

Chapter 13 Continuing on phylogenies, the thirteenth chapter examines the problem of building consensus trees, by merging groups of trees in meaningful ways. The authors demonstrate the use of several computational techniques and algorithms such as sorting, hashing, and tree traversal. They use the lineage of big cats (pantherine) as an example, the phylogeny of which is debated.

Chapter 14 Tandy Warnow provides a very readable introduction to polynomially solvable and NP-hard problems, using popular examples. She then presents the maximum parsimony problem in phylogenetic tree creation, along with heuristics that can be effectively used to solve this NP-hard problem.

Chapter 15 The regulatory network part of the book starts very appropriately with a chapter that introduces many of the graph theoretical concepts and methods used to analyze biological networks. Once the biological motivation is presented, the author proceeds to describe how to represent the biological data with graphs, how to model their properties, and how to use the models to infer new results of biological significance.

Chapter 16 The last chapter examines the inference of regulatory networks from RNA expression data. It first introduces the problem in simple terms, by describing regulatory networks and the interactions that define them, as well as the problem input in the form of expression data from microarray experiments. The author proceeds with modeling the problem probabilistically, starting with one gene and working his way to calculating maximum likelihood models for multiple genes, incorporating prior knowledge, and extending the discrete model to a continuous one.

3 Opinion

The need for diverse introductory texts in the field of bioinformatics has risen significantly in recent years, due to the increased use of computations for the analysis of biological data, the evolution of current biotechnologies, expansion of their throughput, and introduction of new methodologies. Pavel Pevzner and Ron Shamir are both computer scientists by training and by trade, both well respected experts in algorithms, and both invested for quite some time in the field of computational biology, where the majority of their contributions has been focused. Given their background, this edited volume is expected to have an algorithmic bias, and that expectation is realized to a large extent. But this volume introduces a variety of statistical methods and machine learning techniques as well, both of which are important components of modern bioinformatics research.

The editors made a serious effort not only to assemble a series of interesting current topics in bioinformatics research, but also to carefully consider important topic clusters, unify them into meaningful parts, and promise to further evolve them into a developing platform. As expected from an edited book, there is a variety of writing styles and illustrations, but more importantly depth of analyzes and prerequisites required. The reader can enjoy superb introductory chapters such as the 3rd (genome reconstruction), co-written by Pavel Pevzner, which do not lead to detailed theoretical analysis, but verbosely explain and put in context important principles of genome assembly, or the very entertaining 12th chapter on cophylogeny. Several chapters, such as the 4th (dynamic programming), are lacking the illustration richness of other chapters, making the comprehension of algorithmic concepts somewhat challenging for life scientists with limited previous exposure.

All authors have contributed in topics where their expertise and previous work lie, but in few occasions the chapters ended up being mash ups of past publications. Fortunately this is more the exception than the rule, with most chapters offering genuine general purpose introductions to important concepts and techniques, where only relevant examples were drawn from the authors' work. An obvious effort was made throughout the book to engage the reader with interesting motivational problems, and that works brilliantly. The chapters in the phylogeny part could possibly use re-ordering, since techniques to merge and compare phylogenetic trees are presented before one learns how these trees are created. Since the latter topic is addressed in a later chapter, it would make more sense to place it earlier.

Comprehension of the material requires a solid background in statistics and probability, some introductory biology, and possibly an introductory algorithms course. A short introduction to basic computing concepts is included, but the four pages devoted to it are definitely not enough, and biologists who have not encountered the concepts of algorithm complexity and NP hardness may wonder about the purpose of many analysis and discussions throughout the book. NP hardness is introduced and explained in simple terms in a couple of different places in the book, which makes one wonder whether the editors intended the chapters to be autonomous, or considered undesirable the inclusion of a comprehensive introduction, where shared concepts among chapters would be gathered in a single place.

Overall this volume represents an excellent effort towards creating an interesting and useful introductory bioinformatics text. In its current form it may benefit computational scientists more than biologists, but has the potential to evolve into an invaluable resource for all bioinformaticists, independent of their primary field of study.

Review of[8]

Extremal Combinatorics with Applications to Computer Science (2nd edition)
by Stasys Jukna
Springer, Hardcover, \$60.00, 340 pages

Review by Miklós Bóna

The book is a collection of very short chapters, each of which is focused on one proof technique. It is a very hands-on book; everything is taught by examples. The 27 chapters are grouped into five parts, proceeding from the ones that are probably interesting for the widest audiences towards the more specialized ones. In this review, we give a detailed account of the first two, general interest parts, and briefly survey the other three parts.

1 The Classics

The first chapter is mostly a review of undergraduate counting arguments, with sometimes more difficult exercises. One exception is Jensen's inequality, which states that if $0 \leq \lambda_i \leq 1$, while $\sum_{i=1}^{n} \lambda_i = 1$, and f is a convex function, then

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i).$$

The second chapter is devoted to the idea of double counting, that is, the principle that if we count the same objects in two different ways, then we should get the same results. A classic example for this is presented by István Reiman's upper bound for the number E of edges of a simple graph on n vertices that contains no cycles of length four. Reiman's elementary argument shows that $E \leq \frac{n}{4}\left(1+\sqrt{4n-3}\right)$.

Chapter 3 raises the level a little bit by introducing the notion of probability. In particular, we see a few examples of *existence proofs*. The author shows that structures with certain properties must exist, because the probability that a random structure does not have those properties is less than 1. For instance, it is shown using this technique that if $n > k^2 2^{k+1}$, then there exists a tournament on n vertices in which for every k players, there exists a player who beats all those k players.

Another major technique is based on the principle that in any finite set of real numbers, at least one number is at least as large as the average all the numbers in that set. The author shows several examples of the use of this technique, one of which is another result of Erdős showing that any k-uniform hypergraph with fewer than 2^{k-1} edges is 2-colorable. (That means that the vertices of such a hypergraph can be colored red or blue so that each edge has at least one vertex of each color.)

Chapter 4 is about that most basic tool, the pigeon-hole principle. As most combinatorialists know, the simplicity of the tool does not mean that the results obtained by the tool are all straightforward. This is true for this text as well. After warming up with easy, classic facts like the one that in every finite simple graph there are two vertices that have the same degree, the author shows a few more sophisticated examples, some of which are not very well-known. One of them is a theorem of Graham and Kleitman from 1973 theorem that says that if we bijectively label the edges of K_n with the numbers $1, 2, \cdots, \binom{n}{2}$, then there will always be a trail of length $n-1$ with increasing edges. The well-known examples of the chapter include Turán's theorem, various results from Ramsey theory, and Mantel's theorem stating that if a simple graph on n vertices contains more than $n^2/4$ edges, then it contains a triangle. (We do not see a proof of the much stronger statement that with those conditions, the graph will contain at least $n/2$ triangles.)

Part I of the book ends with a very short Chapter 5, which is about systems of distinct representatives. The classic result of Philip Hall is that it is possible to choose k distinct elements $i_1, i_2, \cdots, i_k$ from the collection $S_1, S_2, \cdots, S_k$ so that for all j, we have $i_j \in S_j$ if and only if the union of any t sets S_a has at least t elements. The chapter consists of applications of this theorem. One of them is the fact that every doubly stochastic matrix (a matrix with non-negative real entries in which each row and column sums to 1) is a convex combination of permutation matrices.

2 Extremal Set Theory

We start with a classic structure investigated by Erdős and Rado. A *sunflower* is a collection of sets $X_1, X_2, \cdots, X_k$, so that if $i \neq j$, then $X_i \cap X_j = Y$. Here Y is a fixed set that we call the *core* of the sunflower, while the sets $X_i - Y$ are the *petals*, which are not allowed to be empty.

Sunflowers are important because they occur in every set system that is sufficiently large. In particular, if $\mathcal{F}$ is a collection of sets, each of which has s elements, and $\mathcal{F}$ consists of more than $s!(k-1)^s$ sets, then $\mathcal{F}$ contains a sunflower with k petals. This theorem is relatively easy to prove by induction on s, but it is not known whether the result is the best possible.

Chapter 7 extends the focus of investigation, but still considers intersecting sets. Two classic results are discussed. The Erdős-Ko-Rado theorem shows that if $2k \leq n$, and $\mathcal{F}$ is a collection of sets so that every two sets in $\mathcal{F}$ have a non-empty intersection, then $\mathcal{F}$ can consist of at most $\binom{n-1}{k-1}$ sets. This bound is obviously optimal. Not surprisingly, the author shows G. O. H. Katona's beautiful proof from 1972. The other classic result is Fisher's inequality from design theory. If $A_1, A_2, \cdots, A_m$ are subsets of $[n] = \{1, 2, \cdots, n\}$ so that the intersection of every two distinct A_i is of size k for some fixed $k > 0$, then $m \geq n$. The historical interest of this result is that this was the first combinatorial result that was proved by an argument from linear algebra (by R. C. Bose, in 1949).

The next, very short chapter is about chains and antichains in partially ordered sets. We see three classic results, namely Dilworth's theorem, its dual, and Sperner's theorem. Dilworth's theorem says that in a finite partially ordered set, the size of the largest antichain is equal to the smallest number of chains whose union is the poset itself. The dual of the theorem is obtained by interchanging the words "chain" and "antichain". Sperner's theorem says that in the Boolean algebra of order n, there is no larger antichain than the middle level, that is, no antichain has more than $\binom{n}{\lfloor n/2 \rfloor}$ elements. A strong, and a bit less well-known result here is Bollobás theorem. Let $A_1, A_2, \cdots, A_m$ and $B_1, B_2, \cdots B_m$ be two sequences of sets so that A_i and B_j are disjoint if and only if $i = j$. Then

$$\sum_{i=1}^{m} \binom{a_i + b_i}{a_i}^{-1} \leq 1.$$

We see two proofs of this result, which generalizes Sperner's theorem.

Chapter 9 is about the interesting concepts of blocking and duality of families of sets. If $\mathcal{F}$ is a family of sets, then T is a blocking set of $\mathcal{F}$ if T intersects each element of $\mathcal{F}$. The family of all minimal blocking sets of $\mathcal{F}$ is called the dual of $\mathcal{F}$.

The minimal number of elements in a blocking set of $\mathcal{F}$ is called the blocking number of $\mathcal{F}$, while the size of the largest set in $\mathcal{F}$ is the rank of $\mathcal{F}$. We are shown a result of Gyárfás from 1987 proving that if $\mathcal{F}$ has rank r and blocking number b, then the number of blocking sets of $\mathcal{F}$ with b elements is at most r^b. This then easily implies that if $\mathcal{F}$ is an intersecting, r-uniform family whose blocking number is Also r, then $\mathcal{F}$ consists of at most r^r sets.

In Chapter 10, we hear about downward closed families of sets, which are really just ideals in Boolean algebras. An example is a result of Kleitman from 1966 showing that if A and B are such families over $[n]$, then

$$|A \cap B| \geq \frac{|A| \cdot |B|}{2^n}.$$

The chapter is full of results, the most famous of which is probably the Kruskal-Katona theorem. For a set families A of subsets of $[n]$, let ∂A denote the families of subsets of $[n]$ that can be obtained by taking a set in A, and removing one of its elements. Now let us assume that A is k-regular, and that for some real number $x \geq k$, the inequality

$$|A| \geq \binom{x}{k} = \frac{x(x-1)\cdots(x-k+1)}{k!}$$

holds. Then the Kruskal-Katona theorem shows that

$$|\partial A| \geq \binom{x}{k-1}.$$

Chapter 11 is about Isolation and Witnesses. The following is an interesting example. Let us assume that a society consists of n people, each of whom is asked about his or her preferences among a set X of options, for example by a referendum. Each of the n people makes up his or her mind by forming a total order on X. A *social choice function* then takes the individual total orders as input an outputs some total order $<$ on X. For a given social choice function f, a person i is called a *dictator* if for all possible referendums, f outputs a total order $<$ that coincides with the total order of preferences of i. The surprising result is that if $|X| > 3$, and three natural *democracy axioms* hold, then a dictator exists.

The inclusion of Designs, which are the subject of Chapter 12, is surprising. Most of the chapter as an introduction to design theory, and has no inequalities. The only result that belongs to extremal combinatorics here is Bruen's theorem. In a projective plane, a blocking set is a set of points that intersects every line. Bruen's theorem says that if B is a non-trivial blocking set (that is, one that does not contain a line) in a projective plane of order q, then $|B| \geq q + \sqrt{q} + 1$.

3 The Linear Algebra Method

This part starts with two chapters of general interest which use basic linear algebraic notions, such as rank of matrices and orthogonality. A classic result is that a if a complete graph on n vertices is decomposed into the edge-disjoint union of k complete bipartite graphs, then $k \geq n-1$. No combinatorial proof of this fact is known. The next two chapters are more specialized. The first one is about spectral graph theory and the difficult topic of expander graphs. The eigenvalues of the adjacency matrix of a graph contain a lot of information about the graph. For example a d-regular graph with second largest eigenvalue λ_2 is connected if and only if $\lambda_2 < d$. The next chapter shows a few applications of the idea that polynomials cannot have too many roots. Finally, for the last chapter of the part, the author covers Coding Theory, in particular bounds on code sizes.

4 The Probabilistic Method

The long introductory chapter of this part is about the Linearity of Expectation, a method that we visited in simpler terms (method of averages) in Chapter 1. Here we consider the method, and its modification, the deletion method, in great detail. We mention two of the more advanced methods of the part. The *Lovász Local Lemma* provides a tool to find a lower bound for the probability that none of k bad events will occur if those k events do not have too many dependency relations among themselves. The *second moment method* is applicable for probability variables whose standard deviation is much less than their expectation. As Tschebyshev's theorem shows that it is very unlikely that a variable differs from its expectation by a large multiple of its standard deviation, it follows that variables with the property described in the previous sentence are tightly concentrated around their expectation.

5 Ramsey Theory

In this part, we hear about the Van der Waerden numbers, which are the smallest integers $W(n)$ so that if we 2-color the elements of $[n]$, we always get a monochromatic arithmetic progression of n terms. A strong, multidimensional generalization of this theory, (and the existence of such numbers in higher dimensions), is discussed in a chapter devoted to the Hales-Jewett theorem. Finally, the book ends in a chapter that applies these results of Ramsey theory in Communication Complexity.

6 Conclusion

As mentioned in the introduction, the book is organized around *proof techniques*. Therefore, the book will be most useful for readers who want to acquire skills by learning various methods to prove combinatorial inequalities. It follows that the book is not the right choice for someone who wants to learn everything about a certain *topic*, such as Matching Theory or Random Walks, since some facts will be in the book, some others will not, based on the method that is necessary to prove them. If used in the right way, the book is a helpful, impressive, and unpretentious tool in the hands of a researcher in combinatorics.

Review[9] of

Enumerative Combinatorics, Volume 1, Second Edition
by Richard P. Stanley
Cambridge University Press, 2012
626 pages, softcover
Paperback $43.00, Hardcover $102.00

Review by Angèle M. Hamel,Wilfrid Laurier Univ, ahamel@wlu.ca

1 Introduction

Enumerative combinatorics is the science of counting. It sits at the intersection of algebra and combinatorics, employing techniques from both. At the most basic level it is all about generating functions, binomial coefficients, and recurrence relations; at the most advanced level is it about symmetric functions, posets, and polytopes, with application to problems in physics, probability, and other areas of mathematics (representation theory, analytic number theory, etc.)

Computer scientists will be familiar with enumerative combinatorics, or enumeration, as half of the typical discrete mathematics course, and, perhaps also familiar with it as the first section of Chapter 5 of Knuth's *The Art of Computer Programming, Vol III.* And, at the most obvious level, enumeration is important for the analysis of algorithms: if you can count how many times various portions of the program execute, you can begin to consider complexity. However, this view just scratches the surface. Counting intersects many of the areas of computer science, for example, linear probing in hashing is related to parking functions in enumeration, sorting is related to increasing subsequences, DNA strings in bioinformatics are related to string counting problems, and anonymization in security is related to permutations and shuffles. In many ways—and for many reasons—enumerative combinatorics is a topic that should be better known in computer science, and the textbook by Richard Stanley is a wonderful place to start.

2 Summary

Chapter 1 develops a deep theory about counting those most basic of combinatorial objects: permutations. The geometrical or graphical representation of permutations is given a prominent place, and with that there is opportunity to explore combinatorial arguments for these objects. Chapter 1 begins, appropriately enough, with a section entitled "How to Count." It goes through some elementary counting problems, as one might see in an introductory course in discrete mathematics, and then introduces generating functions, the fundamental tool of enumerative combinatorics. Although generating functions are used throughout the chapter—and indeed throughout the book—they are not the *raison d'être* of the text, unlike in the case of some other books on the topic such as Goulden and Jackson's *Combinatorial Enumeration* and Wilf's *Generatingfunctionology.* This introductory section also introduces the notion of a bijective or combinatorial proof and demonstrates the cleanness of the approach through examples that show the kind of intuitive understanding that a combinatorial proof can provide.

Section 1.2 explores sets and multisets and gives the reader a chance to exercise, in a familiar setting on a familiar object, what he or she just learned about generating functions and combinatorial proofs. Section 1.3 begins the discussion of permutations and reviews the algebra that underpins them (i.e. the cycle structure) before showing how to count them and, in particular, how to flag or mark their characteristics or attributes and count with respect to these (e.g. the number of permutations with k cycles). Section 1.4 further develops counting with respect to permutation statistics (e.g. descents, inversions). Section 1.5 provides geometric interpretations of permutations, and the number and versatility of these representations emphasizes the combinatorial approach.

Section 1.6 goes deeper into the topic of enumerating particular kinds of permutations, and Section 1.7 generalizes much of the preceding work on permutations to multisets. To enter the multiset world we generalize a number of the combinatorial concepts by using the idea introduced in Section 1.3 of a q–analogue of $n!$, and a number of the results go through seamlessly.

Section 1.8 moves on to the topic of partition identities. Partitions were introduced in the preceding section; however, they have a number of important properties in their own right and exhibit a number of interesting relationships that can be proved very naturally combinatorially.

In Section 1.9 the author introduces "The Twelvefold Way" which he credits to G.-C. Rota. The Twelvefold Way is a method for assembling a table that displays the different types of mappings (injective, surjective, no restriction) on sets where the domain and range consist either of distinguishable or indistinguishable elements. Section 1.10 explores the q–analogues of some of the material that has been covered earlier in the chapter.

Chapter 2 is largely unchanged from the first edition. Its focus is sieve methods, which the author defines loosely as "a method for determining the cardinality of a set S that begins with a larger set and somehow subtracts off or cancels out unwanted elements." As the author also points out, such procedures take two basic forms. The first is called the Principle of Inclusion–Exclusion, and in this situation there is an overcounting, then a removal of the extra, then an adding back in of some of what was removed, etc. The second is called the Principle of Involution and involves a weighting of a set of combinatorial objects, and then a cancellation of the undesirable objects.

Section 2.1 is an introduction to the Principle of Inclusion–Exclusion, and Sections 2.2 to 2.4 follow it up with a number of classical examples, many involving permutations: the derangement problem, the number of permutations with a given descent set, the problème de ménages. These sections also include a discussion of rook polynomials. Section 2.5 introduces the concept of unimodality, which refers to a sequence that is initially increasing then changes to become decreasing—sequences with a hump shape. Section 2.5 is also a taster of unimodal sequences in the form of V–partitions.

Sections 2.6 and 2.7 deal with involutions. An involution is a mapping τ such that $\tau^2 =$ identity, and in this context it is used to show that a set of negatively weighted combinatorial objects are equinumerous, and are indeed in one-to-one correspondence, with a set of positively weighted combinatorial objects. This is exemplified by a discussion of determinants and lattice paths: each term in the determinant is mapped to a particular set of lattice paths and an involution is displayed on the set of intersecting paths.

Chapter 3 is an introduction to the theory of partially ordered sets, or *posets*. There is a substantial theory that has been built around these, and they, and the related lattice theory, have been the subject of many classic texts, e.g. by Birkhoff. The notes at the end of this chapter discuss at length the highlights of both the historical and modern contributions to the topic (in fact every chapter in the book has such a Notes section with well-researched and detailed commentary).

Posets can be viewed from many angles. In this book the author develops them naturally as a generalization of the Principle of Inclusion-Exclusion into a technique known as Möbius inversion. A poset P is a set along with a binary relation $\leq$ such that

1. $\forall t \in P, t \leq t$
2. if $s \leq t$ and $t \leq s$, then $s = t$
3. if $s \leq t$ and $t \leq u$, then $s \leq u$.

Not that it is entirely possible that there is no relation between two given elements of the set, i.e. they are not comparable. If every pair of elements is comparable, the poset is a *chain.* If the poset is such that every pair of elements has a supremum and an infimum, then it is called a *lattice.* Section 3.1 to 3.6 are highly algebraic sections that develop the rigorous aspects of posets and lattices. There are, however, some combinatorial examples to reinforce the definitions and theory. In Section 3.7 the Möbius inversion formula makes its formal debut and the connection to the Principle of Inclusion-Exclusion is demonstrated.

Section 3.8 develops the theory of Möbius functions by considering various methods for computing them, while Sections 3.9 and 3.10 do the same for the special cases of Möbius functions for lattices and semi-lattices.

Section 3.11 is one of the sections that is new in this edition, and it concerns the geometric object known as the hyperplane arrangement. This object has been the subject of many recent papers in enumerative combinatorics, and the theory developed has shown deep connections to other combinatorial objects. Thus this chapter is an accessible introduction to an important topic.

Section 3.12 defines the zeta polynomial, which is an expression for the number of multichains in a poset, and derives several properties for it. Section 3.13 also looks at general properties of posets, those related to rank, and provides a combinatorial interpretation involving Möbius functions that relates to descent sets of permutations.

The last sections of Chapter 3 concern various special cases of poset: R-labelings, Eulerian posets, binomial posets, and differential posets—as well as combinatorial objects and techniques that can be defined using posets: (P, ω)-partitions, a method for the enumerations of partitions, and the combinatorial bijections of *promotion* and *evacuation.* Throughout this chapter connections between posets and other areas of mathematics are made crystal clear.

Chapter 4, like Chapter 2, is largely unchanged from the first edition. It concerns the topic of rational generating functions; that is, generating functions, $F(x) = \sum_{n\geq 0} f(n)x^n$, that are rational functions in the ring $K[[x]]$ for K a field, which means that $F(x)$ can be expressed as $P(x)/Q(x)$ for $P(x),\ Q(x) \in K[x]$.

Section 4.1 introduces the theory behind rational generating functions and proves a fundamental theorem that shows how to go from a recurrence relation to a rational generating function. The next three sections, 4.2 to 4.4, further develop this algebraic theory, exploring a number of special cases. Section 4.5 looks at the application to linear homogeneous Diophantine equations. this theory is developed geometrically in terms of objects such as cones and hyperplanes. This theory is applied in the subsequent Section 4.6 to magic squares and to the Ehrhart quasi polynomial of a rational polytope.

Section 4.7 introduces a new technique, the transfer-matrix method, which uses the adjacency matrix of a weighted digraph, and associated techniques from linear algebra, to define a rational generating function. Enumeration problems that can be interpreted as walks on a digraph framework can thus be solved. The rest of the section is rounded out by a number of examples.

3 Opinion

This book is already a classic. Or, rather, the first edition, which appeared in 1986, is already a classic. The second edition has more than doubled the size of the first edition, going from 300 to 625 pages. Over 350 new problems have been added, including some without solution, in order that they can be assigned as homework problems.

The book is both a textbook and a research level resource. It is also the first of a two volume set that surveys all of enumerative combinatorics. The second volume, which appeared in 1999, covers the following topics: trees and the composition of generating functions; algebraic, D-finite and noncommutative generating functions; and symmetric functions. The question forming in the reader's mind will be, "And when will the second edition of Volume II appear?" However, in the Preface to this edition of Volume I, Stanley declares that there will not be a second edition of Volume II, nor a Volume III. So the reader will have to content himself or herself with the second edition of Volume I for their enumerative combinatorics fix for a while!

The text is suitable for an undergraduate or graduate course in enumerative combinatorics, perhaps as a second course in discrete mathematics, after the usual introduction to binomial coefficients, generating functions, and recurrence relations. A familiarity with geometry is also useful. A grounding in abstract algebra is fundamental, particularly in rings, polynomial rings, and fields. Much of enumerative combinatorics belongs to the much larger field of algebraic combinatorics and algebra is a necessity, not a luxury.

The book is also suitable for a computer scientist or mathematician who wants to learn more about the field, perhaps to satisfy a research curiosity, or simply to learn new techniques. The exposition is terrifically clear, and the wealth of problems makes it easy to read and retain the material. The book was a joy to read and review, and will be a joy to use. It is the definitive work; we shall not see its like again for many years.

Review of[10] of
Combinatorial Optimization
by B. Korte and J. Vygen
Algorithms and Combinatorics 21, Springer Verlag, 2012
678 pages, HARDCOVER
Paperback \$45.00, Hardcover \$55.00

Review by Akash Kumar akashkum@yahoo-inc.com

1 Introduction

From the preface of the book, "Combinatorial optimization is one of the youngest and most active areas of discrete mathematics and is probably its driving force today." This field is a subset of mathematical optimization that is related to operations research, combinatorics and theoretical computer science. Authors note that most combinatorial optimization problems can be formulated naturally in terms of graphs and as linear programs. Therefore, this book begins with a review of relevant literature in graph theory and integer and linear programming. Thereafter, the text studies in great depth the classical topics in Combinatorial Optimization including – Minimum Spanning trees, shortest paths, network flows, matchings and matroids, Knapsack and bin-packing problem, Steiner trees, traveling salesman problem, multi commodity max flow min cut problem and the facility location problem. Now, we move on to a chapter by chapter summary of the text.

2 Summary

Chapters 1 through 5 set the basic tone of the book and give a rigorous and still quick introduction to some concepts in algorithms theory, graph theory and the theory of linear and integer programs. Getting more concrete, the 1^{st} chapter defines polynomial time algorithms as those which consume *rational* input and run in time polynomial in input size. On the other hand we have strongly polynomial time algorithms which consume arbitrary (real valued) inputs and run in time polynomial in the number of these real valued inputs (as stipulating run time to be polynomial in real valued input size does not make much sense). In the chapter on graphs, authors rush through a bunch of formal definitions and theorems with their proofs. Their graphs always allow parallel edges and avoid self loops unless authors note otherwise. The definition section is followed with a few lemmas that establish submodularity of δ^+, δ^- and δ (the directed out neighbors, in-neighbors and undirected neighbors respectively). The chapter goes on to discuss Jordan's curve theorem, planar graphs and gives a quick introduction to planar duality. There are several other theorems that later chapters make use of.

In the chapter on linear programs, authors begin by proving that every linear program that is bounded and is feasible admits optimal solutions. This is followed by a discussion of polyhedra, polyhedral cones and polytopes. Next in line is a description of the simplex algorithm to solve linear programs after which authors present LP duality. The next chapter introduces the Ellipsoid

Method to solve Linear programs with rational input in polynomial time (Simplex algorithm and no variants thereof are known to run in polynomial time). Authors caution that we really want LPs with rational inputs as we do not know any strongly polynomial algorithm to solve LPs. The presentation of Ellipsoid in the text is great and authors do a very nice job conveying the high level intuition behind why Ellipsoid algorithm works. In closing, authors note that to run Ellipsoid Algorithm what we need, given $x \in \mathbb{R}^n$, is to decide whether x falls inside the feasible region or find a violated inequality. They emphasize that it is not crucial to have a complete and explicit description of the polytope. This way they motivate *separation oracle* – which even helps solve linear programs with exponentially many constraints. The rest of the chapter details how to make ellipsoid algorithm work with the separation oracle.

This thread of basics stops in the next chapter which discusses Integer Programming. The chapter continues to take a polyhedral viewpoint of the set of feasible solutions. Let P be the polyhedra which is the set of feasible solutions of a LP relaxation of an Integer Program. Let $P_{\mathcal{I}}$ be the integer hull of P – that is, it is the convex hull of integer points inside P. Authors study the conditions under which $P = P_{\mathcal{I}}$. To this end, several equivalent conditions are shown. The discussion motivates total dual integrality in a very natural way and then totally unimodular matrices are introduced. These are matrices A for which $Ax \leq b,\ x \geq 0$ is TDI for each integral vector b. Authors present the cutting plane method to find the integer hull by cutting off parts of $P \setminus P_{\mathcal{I}}$ – which authors observe is not polynomial time as Integer Programming is NP-Complete.

Chapters 6 through 12 introduce several important optimization problems. This is continued later after a discussion of matroids in chapters 13 and 14. The problems described in these chapters (6 to 12) include spanning trees, shortest paths, network flows, minimum cost flows and maximum matchings. In more detail – the chapter on spanning trees looks at the classic algorithms of Kruskal and Prim on finding MST of a given graph. It also contains the classic theorem due to Edmonds on finding a min weight *branching* (a digraph is a branching if the underlying undirected graph is a forest and every vertex v has at most one entering edge). This is continued with a discussion of MST in polyhedral language. The authors introduce the spanning tree polytope and show that the system of inequalities this polytope arises from is TDI. In closing, authors present the results of Tutte and Nash-Williams for packing multiple edge disjoint spanning trees (or arborescences) in undirected graphs (or digraphs).

The next chapter discusses shortest path problem (mostly on digraphs). Algorithms to solve shortest paths are given on a digraph which allows *conservative* weights (those weight functions which do not create any negative weight cycle). Authors present algorithms for all pairs shortest path and also present an algorithm to find minimum mean weight cycle in both undirected and directed graphs. The next chapter discusses network flows and presents the max flow min cut theorem and Menger's theorem. It also discusses other algorithms for maximum flow problem including *push-relabel* and *Edmonds Karp*. The chapter closes with a discussion of Gomory Hu trees which provide a way to store the minimum capacity of a s-t cut for all pairs of vertices s and t. Next comes a follow up chapter that presents algorithms for min cost flows. Authors begin by generalizing maximum flow problem by adding multiple sources and sinks. To this end, they define a balance function $b\colon V(G) \to \mathbb{R}$ with $\sum_v b(v) = 0$ which simulates demands and supply at the nodes of the network. We want a min cost flow by charging flows pushed along any edge that satisfies in addition to flow constraints a balance constraint (which forces $b(v) = \sum_{e \in \delta^+} f(e) - \sum_{e \in \delta^-} f(e)$).

The observation is that a flow network can admit a b-flow *iff* $\forall X \subseteq V(G)$ the overall capacity of the edges leaving X is no less than the overall balance stored at vertices of X. Soon, authors give one optimality criterion for determining whether a b-flow is optimal. Then they present several algorithms to find min cost b-flow which use this criteria as a certificate to check optimality of the output which is iteratively refined.

In the chapter on matchings, authors discuss fundamental results concerning matchings on bipartite graphs including Konig's minmax theorem, Hall's theorem, Frobenius marriage theorem and introduce Dilworth's theorem in the exercises. Thereafter, fundamental results concerning non bipartite graphs are discussed. These include Tutte's theorem and Berge's formula. The chapter focuses on introducing Edmond's matching algorithm which is a combinatorial method to compute maximum matchings in general graphs. Next chapter extends these results to weighted graphs.

Chapters 13 and 14 discuss matroids and generalizes results discussed in earlier chapters. A simple greedy algorithm can be used to optimize over matroids is presented. Thereafter, authors present Edmond's classic algorithm for matroid intersection.

Chapters 15 and 16 briefly review literature on complexity theory and approximation algorithms. Authors introduce a few concepts like strongly NP-Hard problems, L-reductions and *asymptotic* approximation schemes. The authors continue their discussion of important combinatorial optimization problems in Chapter 17 through 22.

Chapter 17 looks at the Knapsack problem. Authors give a polynomial time greedy algorithm for fractional Knapsack. They add that the Knapsack problem is NP-Complete and give the standard dynamic programming based pseudo polynomial time algorithm for the problem. In order to get a good approximation algorithm, they round down the item weights so that overall weight is bounded by a polynomial in the number of items n and they solve the new Knapsack instance. This gives a $(1 + \epsilon)$-approximation algorithm to the optimal and thus we observe that Knapsack admits a *FPTAS*. Finally, authors present the multi dimensional variant of the Knapsack problem and observe that although it admits a pseudo polynomial time algorithm, it does not lead to any FPTAS. However, the problem does admit a *PTAS*. The next chapter introduces the bin packing problem and observes that bin packing is NP-Hard to approximate within a factor of $(\frac{3}{2} - \epsilon)$ unless $\mathsf{P} = \mathsf{NP}$. This rules out any *PTAS* for bin packing. However, authors present an *asymptotic* approximation scheme for this problem.

Chapter 19 reviews the vast literature on multicommodity flows and edge disjoint paths problem. While multicommodity flow problem admits a polynomial time algorithm, the edge disjoint paths problem is NP-Hard. The authors observe that multicommodity flow is a relaxation of the edge disjoint paths problem. They present a LP for multicommodity flow problem that is quite large but which can be used to obtain a FPTAS for multicommodity flow using a primal-dual algorithm. Thereafter, they introduce the sparsest cut problem. This problem admits a LP relaxation which can be seen as the dual LP to Concurrent flow problem (a variant of multicommodity flow where we want to maximize fraction of demands satisfied). As a clincher, authors present the Leighton

Rao theorem that gives a $O(\log n)$ approximation algorithm for sparsest cut. The next chapter looks at the problem of survivable network design. Authors begin by considering the rather restricted Steiner tree problem, show that it is NP-Hard and present an exponential time dynamic programming based algorithm for this problem. Thereafter, they provide approximation algorithm for the Steiner tree problem and then turn again to Survivable Network Design and give a primal dual approximation algorithm for this problem.

Chapter 21 looks at the Traveling Salesman problem. Authors begin by ruling out any k-approx algorithm for TSP with $k \geq 1$. This is followed with a description of metric TSP problem for which Christofides 3/2 approximation algorithm is presented. The authors note that Metric TSP is strongly NP-Hard even with edge weights 1 and 2. This is followed with a description of Euclidean TSP and a discussion of Arora's algorithm for this problem. The chapter goes on to present *Lin Kernighan* algorithm and the *Held Karp* lower bound on the cost of the optimal tour. Finally, authors discuss the branch and bound scheme combined with a cutting plane method that can be used to solve TSP instances with several thousand cities optimally. This involves solving an integer program which has perfect 2-matchings as its feasible solution set. Then, linear programming methods can be used to both establish a lower bound on solutions and get an initial solution which can often be modified to obtain a low cost tour. We iteratively refine this initial solution by adding a series of conditions till the solution output becomes a valid tour. The last chapter presents the facility location problem. The authors describe deterministic algorithms for the uncapacitated facility location problem via LP rounding algorithms. Thereafter, they provide a primal dual based algorithm due to Jain and Vazirani for this problem.

3 Opinion

The text has been designed as an advanced graduate book which also aims at being useful as a reference for research. It clearly succeeds in these goals – it has been used as one of the recommended texts for in several optimization courses at various universities multiple times. That said, it is also somewhat dense as maybe expected of a text which is designed to be useful for research. On occasions I felt that an accompanying diagram would have been helpful, but at the end it was a wonderful experience reviewing this book. I thank the authors for this incredible achievement and especially I thank Jens Vygen for answering some questions I had while going through the text.

Review[11] of

The Golden Ticket: P, NP, and the Search for the Impossible
Author: Lance Fortnow
Publisher: Princeton University Press 2013, $26.95, 188 pages

Reviewers: Cynthia DiPaula cynlioness@gmail.com and
Andrew Wonnacott tricorder.andrewjw@gmail.com

1 Overview

The field of computer science known as computational complexity deals, among other things, with writing algorithms that scale efficiently. Within this, "P" refers to the set of problems which can be solved efficiently by computers (in polynomial time), while "NP," nondeterministic polynomial, refers to problems whose solutions can be easily checked - if your computer could think about all the solutions at the same time, it could carry out the same checking algorithm on all of them in polynomial time. **The Golden Ticket** explains the origins of, efforts to prove, and ramifications of proof of the conjecture that every element of the second set an element of the first. It is aimed at an audience with little previous knowledge on the problem or of computational complexity, but with an inclination towards math and computers and an appreciation for puzzles.

2 Summary

The Golden Ticket opens with a description of the P vs NP problem's nature at a slightly simpler level than just stated, and with a summary, but soon switches to discuss the problem's applications in the real world. It does this with a story: it supposes someone has discovered that P=NP and proceeds to detail how the world would be different. For instance, media and advertisements would be produced by computer and tailored automatically to every consumer. This "beautiful world" as Fortnow names it also has its downsides: current computer security would be immediately bypassed and wide groups of jobs would be replaced by computer programs. However, he clearly feels the upside outweighs the downside. He also, alas, clearly feels that this beautiful world will not come to pass.

After this introduction, the book becomes more technical, but keeps with the story aspect that so far is its greatest strength, introducing a fictional world of Frenemy where everyone is a friend, or an enemy. This allows easy explanation of a variety of problems: first of cliques, then of several graph theory problems including map colorings. At the end of this, they are categorized into P and NP. Also, it becomes much more clear what problems are in NP and what are outside even that - the previous problems all had solutions that could be easily checked once they were read; some problems for which this is false are brought up. The idea of a problem being NP-complete, or being among the most complicated problems in NP, can now be introduced with examples the reader already is familiar with.

Fortnow then turns to history, starting off with the origin of our thinking about P and NP. He narrates how Steve Cook described that if it was possible to solve satisfiability in P, all NP problems would be in P, Karp showed that another 20 problems were computationally equivalent, and Knuth came up with the name NP-complete for those problems. This is followed by a bunch of puzzles, some of which, like the Sudoku (which is NP-complete), are puzzles real people (including the authors of this review) really solve, and some of which are more abstract, like graph isomorphism (which is not NP-complete). Travelling further back, to Alan Turing, the book then describes what it means for something to be computationally complex and talks about what developments led to Cook's paper. Then the book goes to Russia and examines the work done there on avoiding *perebor*, or brute force search, and how the politics of the Soviet Union kept Leonid Levin from publishing there until after Cook's publication in the States.

In the real world, we still need answers to problems in NP, so people have constructed useful reactions to problems not in P. **The Golden Ticket** explains this by returning to Frenemy to show heuristics and approximations that can be used to get a good-enough solution quickly, and applies them to real-world problems like a coloring a map of Armenia and Switzerland, or a minimizing the cost of a salesman traveling China in a grid. It shows how problems can be changed from NP complete ones to similarly useful problems that are in fact mathematically different. Some problems, however, will still need perebor.

The conclusion of the book deals with what we will be able to do regardless of a proof either way. It describes how quantum computers handle data differently from traditional binary-state computers and how this bypasses NP problems, then concludes with a synopsis of issues we as a population will face whether P=NP or not: issues with magnitude and quality of computable data in the near future.

3 Opinion

The Golden Ticket contains an impressive amount of material regarding the $P = ?NP$ question, at a level which is easy for most people to understand. It is clearly aimed at an audience with some math background, about a high school level. Most of the topics explained do not require familiarity with computer science, a huge advantage over many textbooks: nearly every computer history element, phrase, or problem is explained in detail. Even if you already know the basics of computer science, the book is engaging enough that it's a fun read even if then not as challenging.

It is important that books which can clearly explain computer science topics like the P vs NP issue exist. The introduction of such a topic must be done well to interest potential great minds, and this book certainly succeeds. It captures readers' attention with captivating stories, humor, puzzles, and diverse information. New topics are nearly always introduced as examples instead of bland definitions, and vivid examples and stories convince even the most reluctant readers to continue. Puzzles for the reader to complete include friendship diagrams, color mapping, Sudoku, isomorphism, and several others. In addition to being examples of NP complete problems necessary to the book, the way they are presented lets them become a fun way to understand the information.

Each individual topic presented in this book, if addressed alone, could be considered overly complicated for an inexperienced audience. However, Fortnow successfully connects every new piece of information with something relevant to the reader, including past information. The book flows fantastically from the simple and mundane to quite complex, yet one never quite realizes the jump they have made. For example, many readers would be quite intimidated by the prospect of learning about "quantum computing", thinking it incredibly difficult to comprehend, yet a succinct and well-written explanation of qubits and entanglement gives as much information as is necessary in context.

It is also valuable to stress the ramifications of events such as a proof of P vs. NP, which **The**

Golden Ticket does quite well. Within the first chapter, it shows that this is one of the most important problems in computer science by citing the millennium prize, then goes on in the next chapter to talk about all the ways the world could be improved or hurt by a proof in either way.

With just an understanding of high school-level mathematics, an interested reader can easily get into this book - it is an engaging read independent of your discipline. In other words, **The Golden Ticket** serves as an excellent introductory explanation of a complex problem to an audience who might not have an interest, prior knowledge, or time necessary to devote oneself to a detailed study of the P vs. NP problem.

Review of[12]
Probably Approximately Correct
by Leslie Valiant
Basic Books, 2013
195 pages, Hardcover, $25.00

Review by
Joshua Brulé `jtcbrule@gmail.com`
University of Maryland, College Park

1 Introduction

Probably approximately correct (PAC) learning is a mathematical framework for analyzing the limits of what is 'learnable', not unlike how computational complexity theory acts as a mathematical framework for analyzing the ideas of 'computable' or 'tractable'. PAC learning can also be compared to the (perhaps better known) Bayesian framework which uses a prior distribution on hypotheses and updates via Bayes rule to represent learning.

Remarkably, PAC learning allows learning *without* assuming a prior probability distribution over hypotheses (one of the common criticisms of Bayesian inference). Instead, all that is required is the existence of distribution from which data is drawn, and that the sample data that the learner sees is drawn from the same distribution. Learning a hypothesis that is correct *all* the time on *all* the data is, of course, impossible. But given a polynomial number of examples to learn from, a PAC learner can find a hypothesis that is usually (probably) correct over most of (approximately) all the future data.

For inventing PAC (and other significant contributions to theory of computation), Leslie Valiant was awarded the 2010 Turing Award.

2 Summary

Probably Approximately Correct (the book) is Valiant's exposition on PAC learning written for a general (i.e. non-computer scientist) audience. However, *Probably Approximately Correct* stands out from most other 'popular science' books by explaining not only the theory itself, but the remarkable advances made in the rest of computer science theory that were required to make PAC possible. Valiant also discusses some of more interesting applications of PAC, specifically, how evolution can be viewed as a PAC learning algorithm and how human level AI is most likely not outside of our reach, seeing as we have no reason to believe that human beings learn in some computationally special way.

Chapters 1 and 2 discuss some of the inherent difficulties in machine learning. Valiant coins the term 'ecorithm' to describe an algorithm designed to solve a problem that we lack a quantitative theory for (the 'theoryless'). This acts as a nice distinction from traditional algorithms (e.g. running an n-body simulation) which can designed around a more traditional scientific theory such as Newtonian mechanics (which Valiant refers to as the 'theoryful').

Chapter 3 is a very accessible discussion computational complexity theory, including Turing machines, computability and classic complexity classes such as P, BPP, BQP and NP. Valiant also provides an excellent example of a learning algorithm in his description of the classic perceptron algorithm. Unlike most popular science books (whose publishers seem to be afraid of introducing too many numbers or equations), Valiant uses just enough mathematics to accurately describe the algorithm, while still keeping the description accessible to anyone with a high school background.

Chapter 4 is a brief argument regarding the importance of studying how information is processed in biological systems rather than studying just the substrate of life, similar to how computer science focuses on studying algorithms independently of hardware.

Chapter 5 is a description of the actual PAC learning framework in which Valiant again does a remarkable job of using just enough math to explain the basic ideas while remaining accessible to most readers. Special mention should go to using cryptography as an intuitive example of how not all of P is learnable (or rather, as an example of why we should not *believe* all of P is learnable). For if we could learn the details behind any cryptosystem constructed with polynomial resources in polynomial time, there would be no such thing as a secure cryptosystem.

Chapter 6 is an argument that evolution might be better viewed as an instantiation of a PAC learning process and that perhaps evolution is not as widely accepted as other physical theories (e.g. gravity) because it lacks more quantitative predictions. This foray into analyzing evolution from a PAC standpoint is intended to act as a starting place for more research on evolution with the additional goal of gaining wider acceptance of the theory. (I don't know how convincing this will be to an intelligent design advocate, but it's a worthy effort.)

Chapter 7 is a discussion of classic first order logic's brittleness in building intelligent systems. The first half of the chapter is quite valuable in explaining to a general audience why AI has appeared to fail in delivering on it's early promises and sets up for the next chapter in discussing why we should still have hope. Valiant also argues that an adequate robust logic can be derived from PAC learning. The discussion of robust logic feels less clear (compared to the earlier explanation of PAC learning), but should still be accessible to a general audience.

Finally, Chapter 8, 9 and 10 discuss viewing humans and machines as 'ecorithms'. The overall view is quite optimistic: that we should have strong hopes for advancement in AI since we have no reason to believe that humans have access to a form of inference more powerful than PAC learning. Valiant quite convincing argues that the main difficulties seem to come from the fact that AI is 'competing' against billions of years of evolution. And considering that computer science as a field of study has been around for less then a century, the field has made remarkable progress in such a short time.

3 Opinion

For the mathematically literate, *Probably Approximately Correct* is likely not the best introduction to PAC learning, especially considering that a large part of the book is spent discussing concepts that virtually all computer scientists will be strongly familiar with, only in less detail than a more technical introduction.

However, the book really shines as an introduction to computer science theory to the general public, providing a compact and accessible description of basic, *important* results that are sadly not widely known outside the field. This is a book that should be on every computer scientist's shelf so that when someone asks, “Why is computer science theory important?” the three word response can be, “Read this book”.