The Book Review Column[1]
by Frederic Green

Department of Mathematics and Computer Science
Clark University
Worcester, MA 01610
email: fgreen@clarku.edu

The three books reviewed in this column are about central ideas in algorithms, complexity, and geometry. The third one brings together topics from the first two by applying techniques of both *property testing* (the subject of the first book) and *parameterized complexity* (including its more focused incarnation studied in the second book, *kernelization*) to geometric problems.

1. **Introduction to Property Testing**, by Oded Goldreich. Review by Sarvagya Upadhyay.
2. **Kernelization: Theory of Parameterized Preprocessing**, by Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Review by Tim Jackman and Steve Homer.
3. **Forbidden Configurations in Discrete Geometry**, by David Eppstein. Review by Frederic Green.

Life in the ronaverse continues to be rendered somewhat more tolerable by reading a good book. Choose from among the books listed on the next page, or find a reasonably recent title that interests you. The latter remains preferable in the present moment, as publishers can send books directly to you at my request.

BOOKS THAT NEED REVIEWERS FOR THE SIGACT NEWS COLUMN

Computability, Complexity, Logic

1. *The Foundations of Computability Theory,* by Borut Robič
2. *Applied Logic for Computer Scientists: Computational Deduction and Formal Proofs*, by Mauricio Ayala-Rincón and Flávio L.C. de Moura.
3. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, by Martin Grohe.

Miscellaneous Computer Science

1. *The Age of Algorithms*, by Serge Abiteoul and Gilles Dowek.
2. *Elements of Causal Inference: Foundations and Learning Algorithms*, by Jonas Peters, Dominik Janzing, and Bernhard Schölkopf.
3. *Partially Observed Markov Decision Processes,* by Vikram Krishnamurthy
4. *Statistical Modeling and Machine Learning for Molecular Biology*, by Alan Moses
5. *Language, Cognition, and Computational Models,* Theirry Poibeau and Aline Villavicencio, eds.
6. *Computational Bayesian Statistics, An Introduction,* by M. Antónia Amaral Turkman, Carlos Daniel Paulino, and Peter Müller.
7. *Variational Bayesian Learning Theory,* by Shinichi Nakajima, Kazuho Watanabe, and Masashi Sugiyama.
8. *Programming for the Puzzled: Learn to Program While Solving Puzzles*, by Srini Devadas.
9. *Knowledge Engineering: Building Cognitive Assistants for Evidence-based Reasoning*, by Gheorghe Tecuci, Dorin Marcu, Mihai Boicu, and David A. Schum.

Combinatorics and Graph Theory

1. *Finite Geometry and Combinatorial Applications*, by Simeon Ball
2. *Introduction to Random Graphs*, by Alan Frieze and Michał Karoński
3. *Erdős –Ko–Rado Theorems: Algebraic Approaches*, by Christopher Godsil and Karen Meagher
4. *Combinatorics, Words and Symbolic Dynamics,* Edited by Valérie Berthé and Michel Rigo

Miscellaneous Mathematics

1. *Introduction to Probability*, by David F. Anderson, Timo Seppäläinen, and Benedek Valkó.
2. *Fat Chance! Probability from 0 to 1*, by Benedict Gross, Joe Harris, and Emily Riehl.

Review of[2]
Introduction To Property Testing
by Oded Goldreich
Cambridge University Press
Hardcover, 445 pages, $89.99

Review by
Sarvagya Upadhyay (supadhyay@fujitsu.com)
Fujitsu Laboratories of America
1240 East Arques Avenue, Sunnyvale CA 94085, USA

1 Overview

The area of property testing is concerned with designing methods to decide whether an input object possesses a certain property or not. Usually the problem is described as a promise problem: either the input object has the property or the input object is *far* from possessing the property. Here, the meaning of object being far from possessing the property is based on a specified and meaningful notion of distance.

The main objective of property testing is accomplishing this decision making by developing a super efficient tester. A tester that reads through the entire object can easily determine whether the property is satisfied or not. However, one wishes the tester to probe the input at very few random locations and determine whether the property is satisfied. As such, randomness is a necessary ingredient for testing and having the tester erring on few instances is a necessary price to pay for designing highly efficient methodologies.

Much of the literature on property testing has focused on two types of objects: functions and graphs. Naturally they form the major portion of the book: functions are discussed from Chapters 2 to 6 and graph properties are discussed from Chapters 8 to 10. The final three chapters focus on distribution testing, probabilistically checkable proofs (PCPs) and locally testable codes, and ramifications of property testing on other related topics in Computer Science and Statistics. A separate chapter is devoted to query lower bound techniques.

2 Summary of Contents

The book is divided into 13 chapters. The first chapter is somewhat essential to follow the rest of the chapters. There is little to no dependencies between other chapters. A detailed summary of each chapter is given below. The few mathematical notations that we employ are directly from the book.

Chapter 1: The Main Themes: Approximate Decision and Sub-linear Complexity This chapter focuses on the fundamental concepts of property testing. It starts with motivations for studying property testing, a short overview of algorithmic and complexity-theoretic directions of research and two main themes of property testing – approximate decision and sublinear complexity. The technical part of the chapter is Section 1.3. This particular section begins with the definition of proximity-oblivious testers and a few algebraic properties of property testing. For instance, given two properties that are easier to test, testing their union is easier as well, while testing their intersection can be quite hard. Finally, the chapter culminates with the relationship between learning and testing.

Chapter 2: Testing Linearity (Group Homomorphism) One fundamental property to test is whether a function is linear or not. The literal definition of linearity (homomorphism in groups) forms the testing algorithm – randomly pick two points x and y, and query at three different locations: x, y, and $x + y$. This so called linearity test has a fundamental role in construction of PCPs. The chapter ends with a simple extension of linearity testing to affine testing.

Chapter 3: Low Degree Tests Linearity testing can be thought of as testing whether a univariate polynomial has degree one or not. A natural extension of linearity testing is low-degree testing where one wishes to construct efficient testers to test whether a polynomial over a prime field is of low degree. This chapter explores low-degree testers for multivariate polynomials. The chapter starts with intuition on how to proceed, first with univariate polynomials and then with multivariate polynomials. A detailed background on multivariate polynomials (over prime fields) is given before rigorously analyzing the tester, which is to simply query the polynomial at d+2 carefully chosen points where d is the (total) degree of the polynomial.

Chapter 4: Testing Monotonicity This chapter focuses on monotone functions – functions of the form $f : D \to R$, where D forms a partial order and R forms a total order, such that for all x and y such that $x < y$, it holds that $f(x) \leq f(y)$ (here, "$<$" represents partial order and "$\leq$" represents total order). The two main classes of monotone functions considered in this chapter are Boolean functions over the Boolean hypercube and real-valued functions on the discrete line. For the former, the chapter analyzes two types of testers: (i) edge-based testing that randomly selects an edge of the Boolean hypercube to test for monotonicity; and (ii) path-based testing that queries at two endpoints of a random path and tests for the monotone property. The second class of functions (real functions on the discrete line) follows next and a tester based on querying points based on binary search of the line is discussed. Finally, these two classes of functions are coalesced (hypergrids) and monotonicity testers for such partial orders are analyzed.

Chapter 5: Testing Dictatorships, Juntas, and Monomials This chapter is devoted to three classes of Boolean functions of the form $f : \{0, 1\}^\ell \to \{0, 1\}$: (i) dictatorship functions, where the function value depends on exactly one Boolean variable; (ii) k-monomials where the value of the function is a conjunction of exactly k literals; and (iii) k-juntas where the value of the function depends on at most k Boolean variables. The testers for each of these functions rely on special properties derived from the definition of these functions. As an example, dictatorship testing is based on first reducing the space of Boolean functions to the space of linear functions (via linearity testing) and then utilizes a sufficient characterization of dictatorships for the next steps of testing. Testers for monomials and juntas follow next. Junta testing relies upon the notion of influence of Boolean variables and figuring out an ingenuous way of finding how many indices have positive influence on the function.

Chapter 6: Testing by Implicit Sampling This chapter focuses on Boolean functions that can be approximated by k-juntas and gives a general technique to construct testers for such functions.

Chapter 7: Lower Bounds Techniques This chapter illustrates three different query lower bound methods employed in property testing. The first method relates to indistinguishability of two probability distributions, where one distribution is on instances that possess the property and the other distribution is on instances far away from property. The main idea is to demonstrate that a randomized query algorithm with small query complexity will err with high probability in the task of distinguishing between these two distributions. Much of Section 7.2 is devoted to this and finally the technique is applied to the problem of testing membership in linear codes. The second method (elaborated in Section 7.3) is based on communication complexity lower bounds and showing how these lower bounds can be translated to query complexity for certain property testing problems. A reduction theorem that relates property testing with communication complexity is presented, followed by an application to k-linearity testing. The final method is discussed in Section 7.4 which relies on reduction among property testers. Various reduction methodologies are discussed. The chapter culminates with property testers with restrictions, such as testers with one-sided error probability and testers using non-adaptive queries.

Chapter 8: Testing Graph Properties in the Dense Graph Model This chapter initiates the discussion of graph properties. One of the ways of representing a graph is by its adjacency matrix, which is referred to as the dense graph model in this book. This is discussed in Section 8.2. The subsequent two sections focus on two themes of this chapter: (i) an exposition of graph partition problems from the property testing point of view; and (ii) the connection between property testing and Szemeŕedi's regularity lemma. The final two sections provide a classification of known results in terms of query complexity in this model and detailed final comments.

Chapter 9: Testing Graph Properties in the Bounded-Degree Graph Model Bounded-degree graph models are models where graphs are represented by their incidence lists. The model is discussed in Section 9.1 and ends with comparison with dense graph models with respect to query complexity of various graph properties. The next section introduces testing algorithms for several graph property problems. These problems have somewhat small query complexity compared to testers presented in Section 9.4. Sandwiched between these two sections is a section on lower bounds on query complexity in bounded-degree graph models, which illustrates why the property testers in Section 9.4 have higher query complexity. Section 9.5 focuses on testers that utilize partition oracles, which are used to partition a graph into small connected components such that there are very few edges across different components. As in the previous chapter, the final section classifies known results in this model based on query complexity.

Chapter 10: Testing Graph Properties in the General Graph Model This chapter puts emphasis on general graph models, where graphs can be inspected via queries to both adjacency matrices and incidence lists. This is the crux of the first section of the chapter. This model and the bounded-degree graph model are more similar to each other than to the dense graph model. As such, it is natural to extend the testers that work in the bounded-degree graph model to the general graph model. The next section starts with two main points that need to be addressed for adapting and/or extending these testers. Graph connectivity is given as an example where the tester in the bounded-degree model can be easily adapted to general graph models. The case of testing bipartiteness is an example of easy extension to general graph models. Section 10.3 highlights two property testing tasks and gives algorithmic techniques for them. The final section focuses on how incidence as well as adjacency queries can both be beneficial in testing graph properties.

Chapter 11: Testing Properties of Distributions This chapter focuses on testing properties of discrete probability distributions. This is completely different from what the previous nine chapters have focused on. One can focus on testing properties of a single distribution or testing properties of a pair of distributions. Section 11.2 focuses on a single distribution and gives an efficient tester to test whether an unknown distribution is equal to a fixed known distribution. It culminates with a lower bound result on the number of samples required to test whether a distribution is uniform. The next section extends the problem presented in Section 11.2 and tests whether two unknown distributions are equal or not. The final section presents a brief survey of results in this field.

Chapter 12: Ramifications and Related Topics The notion of property testing has consequences to a few related topics. These consequences form the topic of this chapter. Some of the topics considered in this chapter are tolerant testing, sample-based testing, testing under different distance measures, and non-interactive proofs of proximity.

Chapter 13: Locally Testable Codes and Proofs One can argue that property testing came to prominence due to various constructions of PCPs that included linearity testing, low-degree testing, dictatorship, etc. Two notions are discussed in this chapter: locally testable codes and locally testable proofs. The former is a set of error-correcting codes where a valid codeword can be tested with very few local queries to the code. The latter are also known as PCPs where one wishes to find the validity of proof with very few queries. This chapter gives a detailed account of property testing in locally testable codes and PCPs.

3 Evaluation and Opinion

The author is a well respected researcher in Theoretical Computer Science. While my area of research has been very different from his, I have always enjoyed reading his books. Keeping my personal biases aside, this book is an excellent read. It illustrates his profound knowledge on this subject.

Chapters 1 and 12 are my personal favorites. Going through Chapter 1 is essential because a careful reading of this chapter generates a lot of interest in this topic. I love interconnections between topics and that is beautifully covered in Chapter 12. The chapter notes and historical remarks are quite comprehensive as well.

As discussed by the author, the first cluster of chapters discuss properties of functions. It should be noted that most of the algorithmic testers are simple to explain even though their analysis may be slightly complicated to understand. These clusters of topics start with testing of very simple properties and move on to more complicated properties. For a beginner, this is perhaps the best way to initiate exploring the topic. While the chapters can be studied more or less independently, it is advisable to follow the chapters in the order in which they are presented.

The second cluster of chapters are on testing graph properties in different models. Understanding the subtle and not so subtle differences between models is very important and this has been thoroughly discussed throughout. Personally, this portion of the book was somewhat difficult to read and it is recommended that one pauses for few moments to internalize the topics before proceeding further.

The last three chapters are independent of each other and they provide a different flavor to the book. Chapter 13 is a condensed version of a huge body of research and it can be easily elaborated into a full graduate course.

I would like to state that rigour is very important in this book. The definitions and proofs are presented in full rigour and they can be overwhelming at times for a casual reader. There are parts of the book that I enjoyed reading thoroughly. To me, the chapter on lower bound techniques was quite illuminating. The collection of problems in the exercises are interesting to work on. However, I am not hugely supportive of the idea of providing hints right after posing the problem. The hints could have been a separate appendix in the book.

Regarding the organization of the book, I found Chapter 7 (on lower bounds) somewhat misplaced. I would have preferred moving from function property testing to graph property testing and the lower bound techniques to be discussed after these two topics. Moreover, the initial discussion on three graph models in Chapter 8 could have been discussed before Chapter 8 as an introduction to the second part of the book (which the author refers to as the second cluster of chapters).

Overall, the book is an excellent and comprehensive read. The range of topics discussed in this book is sufficient to attract many students to use it as reference. It can definitely be used as a reference to an advanced undergraduate level course or a beginning graduate level course.

4 Typos

There are few typographical mistakes in the book. Most of them have already been listed on the author's web link to the book. There are couple of them that have not listed on the link. On Page 2, the first line after Section 1.1.1, it should be "superfast approximate decisions" rather than "superfact approximate decisions". On Page 271, when discussing organization of the chapter, "adjacency" is misspelled as "gadjecency".

Review of[3] Kernelization: Theory of Parameterized Preprocessing
by Fedor V. Fomin, Daniel Lokshtanov,
Saket Saurabh, and Meirav Zehavi
Cambridge University Press 2019
$69.99, Hardcover, 528 pages

Review by
Tim Jackman (tjackman@bu.edu) **and Steve Homer** (homer@bu.edu)
Department of Computer Science
Boston University

1 Overview

On the surface, the concept of preprocessing may not seem particularly novel nor insightful. Indeed, the simple idea that putting in computational effort upfront may pay off in dividends of time saved later is not only advice that we have received countless times but it is also advice that we ourselves have given to countless others. It may well seem a cliché, but it turns out to be more powerful than it first appears to be, and in fact plays a rich role in numerous areas of algorithmic research and is a powerful tool for myriad applications. This turns out to be particularly true in Computer Science where pattern repetition within and between algorithms is ubiquitous. The idea of storing results once to avoid repeating them innumerable times can pay huge dividends in computational problems of many kinds. So, while possibly overdone, the advice is still sound and often revelatory.

Kernelization is the idea of taking a parameterized decision problem and first reducing it to a smaller instance which can then be solved in order to solve the original. Instances are reduced by repeatedly applying so-called *reduction rules* which shrink the instance without changing the yes-no answer. This smaller instance of the problem thus captures the difficulty of the problem. It turns out that being able to find a kernel is equivalent to a concept known as fixed parameter tractability—where a problem, parameterized by some variable, could be solved in polynomial time if this given parameter was held constant. Thus kernelization often plays a role in books on the broader topic of parameterized complexity.

Kernelization: Theory of Parameterized Preprocessing, by Fomin et al., is unique in that it is a text focusing solely on the titular topic of kernelization. The book consists of 23 chapters which are divided into four parts. Following an introductory chapter, the first half of the chapters—and indeed much of the book—is dedicated to showcasing the main tools used to prove the existence of and size of polynomial kernels for various problems. This central section of the text, called *Upper Bounds*, consists of kernelization algorithms for many common combinatorial problems. These tools for the most part are fairly problem specific and are followed by the smaller second section of the book, *Meta Theorems*, which analyzes combinatorial structures in order to develop more general and widely applicable upper bound techniques.

Following these discussions on proving upper bounds of kernel sizes is a *Lower Bounds* section which is focused on demonstrating how limits on kernel size are proved. These results depend on expected computational hardness assumptions whose truth implies the kernel size limits. Finally, the book concludes with a brief miscellaneous section, *Beyond Kernelization,* which captures two current areas of study which are outside of the scope of the prior sections.

By being solely focused on kernelization, this text is able to more effectively showcase and teach the tools used in the field than a more traditional text on fixed parameter complexity—which may only spend a chapter or two on the topic. One of the important strengths of this book is the number of detailed examples which make the text's carefully worded definitions and problems even clearer. While in places there may be too many examples, they help the text flow from section to section as problems are often revisited with each new kernel technique and prior kernel methods are improved upon. Among the primary problem examples that are most often used throughout the text are Feedback Vertex Set, Dominating Set, d-Hitting Set, Connected Vertex Cover, Cycle Padding, Independent Set, Longest Path, Odd Cycle Transversal, and Vertex Cover.

This text serves primarily as a reference for researchers in the field. However it can also serve as an introduction for researchers and graduate students who wish to begin their study of this research topic. For researchers, the book's extensive bibliographic notes and collection of problem definitions and current open problems will be an asset. For students and newcomers, the text's well-paced and methodical introduction to kernelization and its numerous examples will help them get up to speed in no time. The main prerequisite to this text is knowledge of algorithms and algorithm design, and as the authors note this text works both as the basis for a course on kernelization algorithms or as a companion text to a course on the broader topic of parameterized complexity.

2 Summary of Contents: Upper Bounds and Meta Theorems

After a brief introductory chapter explaining the concept of kernelization, the book begins with its first long section on Upper Bounds. This section begins in Chapter 2, a very measured and straightforward introduction to different methods, explaining some standard examples of problem kernels that are readily obtained. From there, the section becomes a mixture of different topics and kernelization techniques. Chapter 3 introduces inductive priorities, a systematic method for coming up with reduction rules. From there, the book analyzes a graph structure known as a crown in order to obtain kernels via "crown decomposition" (Chapter 4) and extends this analysis to more general types of bipartite graphs (Chapter 5).

The remaining eight chapters of this section cover a wide range of methods and specific techniques that can be leveraged in order to obtain problem kernels. These include linear programming, hypertrees and their properties, the sunflower lemma, modules, matroids, representative sets, greedy algorithms, and Euler's formula. The first three of these are relatively simple and straightforward. A case in point is Chapter 8 on the Sunflower Lemma introduces the famous Erdős /Rado result and its proof. The lemma is then used to establish some straightforward kernelization results for problems such as d-Hitting Set and d-Set Packing. Some more involved examples include Dominating Set on degenerate graphs follow this. The last five of these chapters are somewhat longer and more advanced.

As might be apparent from how varied the last eight chapters were, these techniques and topics can be narrow and do not necessarily generalize to a large class of problems. Thus the next brief section of the text, Meta Theorems, attempts to rectify this by introducing some more generic kernelization methods. To this end, Chapter 14 introduces the concept of treewidth which is then applied to the concepts of bidimensionality, protrusions, (Chapter 15) and graph surgery (Chapter 16) which provide generic kernelization algorithms for many problems.

3 Summary of Contents: Lower Bounds and Beyond Kernelization

After 16 chapters focusing on ways to create and improve kernels for various problems, the book transitions to proving limits on kernel size in the Lower Bounds section. Chapter 17 introduces the framework for proving these lower bounds using a technique called OR-cross-composition which, it explains, can be used to show a problem does not admit a polynomial sized kernel unless coNP $\subseteq$ NP/poly. Obtaining OR-cross-compositions using a method known as Instance Selector is the focus of Chapter 18. A type of reduction which preserves these kernel lower bounds is then introduced (Chapter 19). Lower bounds for problems which do have polynomial kernels are discussed in Chapter 20. The section concludes with a discussion on how to extend the concept of these Or-cross-decompositions using communication protocols. These methods are used in Chapter 21 to obtain kernelizations of several previously discussed problems based on uniform complexity theoretic hypotheses.

The book ends with a small miscellaneous section: Beyond Kernelization. This section covers two topics that do not fit within the other topics of the book: Turing kernels (Chapter 22) and lossy kernels (Chapter 23). The former is where the kernelization algorithm has access to an oracle which can decide the answer for smaller input instances, and the latter extends to notions of kernels for optimization problems and approximations—rather than the reduced instance, returning the same yes-no answer as the original, the answers to the kernels need to correspond to approximate answers to the original problem which are within a fixed factor of optimal. These two topics do not fit in the scope of the other sections and thus feel out of place in the overall work. However, they do provide two open avenues of research in the field and thus may be more useful as references for researchers rather than students.

4 Highlights

We will now call attention to some of the many strengths of the text. As mentioned earlier the first four chapters of the text (*What is a Kernel* through *Crown Decomposition*), 57 pages in all, form a strong, well-written introduction to the subject. These sections are introduced with care and are kept simple. They focus on a few central examples and provide full explanations and illustrations. The fundamental definitions and key concepts of kernelization are presented intuitively and connected to these examples. These concepts are then extended with slightly harder problems which point the way to some of the more complex kernelizations in the following chapters. Overall these well-motivated introductory chapters set up the many examples in the rest of Part 1 and some of the more advanced topics in the second half of the text. The introduction provides a basic grasp of the flavor of the subject as it can be used for more complex algorithms and more difficult algorithmic techniques.

The same can be said of the introductory two chapters of Part 3 (Chapter 17: *Framework* and Chapter 18: *Instance Selectors*). The first introduces the framework for proving limits on the minimum size of a kernel which can exist for certain problems. This is done by first introducing an intuitive example and then by formalizing and expanding the theory. The central example is then revisited formally, and several more examples are provided. The chapter is well written and easy to follow. The second chapter provides a simple method for using the established framework to provide lower bounds and clearly explains the core ideas. The section then provides three well-explained examples, covering more powerful and difficult kernelization methods.

In sum, the book manages to present an incredible number of techniques, methods, and examples in its 528 pages. Each chapter ends with a bibliographic notes section, which often provides some small historical context for the material covered. It also points to more current results and papers although it does so very briefly. Together, this makes the textbook a valuable resource book to researchers. The authors also provide some suggestions on how to adapt the text to teach a semester class (namely focusing on Parts 1 and 3). The mostly self-contained final eight chapters of the first part can allow instructors to further cut down some of the material by choosing which topics to discuss and which to omit.

5 Criticism, and small corrections and typos

Researchers looking to learn about a specific problem will have to navigate the index and numerous different chapters and sections. The book's organization makes sense for a textbook, but for a reader trying to focus on a single problem rather than a textbook for kernelization techniques, it may be difficult to read these different chapters which have different purposes, approaches, and points of view concerning the problem.

While the introductory chapters are well written and cogent, some of the later sections of the book are not as consistent or well-coordinated with earlier parts as they could be. For example, while the introductory chapters of Part 1 contain many exercises which would be valuable for an instructor or student using the text, exercises are sometimes sparser or less relevant in later chapters and parts. Part 3 suffers from this. While there are a few good exercises in Chapter 17 (*Framework*), Chapter 18 does not appear to feature any problems using the Instance Selectors technique and Chapter 20 does not feature any exercises at all. In some cases it may benefit the text to remove some of the many worked through examples from the main body of the text and instead make them exercises.

Another instance of this can be seen at the end of the text, in section 4 *Beyond Kernelization*, which while containing a short introduction to two interesting topics which extend earlier parts of the text, is quite abbreviated and not very revealing of the goals of the work. Some definitions are repeated from earlier in the text, and the examples are not as well-motivated as prior ones.

Finally we present a short list of small corrections and typos we found in our reading.

Page 371, last line: Should be "All but one of its vertices are in S."
Page 440, line 13: "no more hold" should be "no longer hold".
Page 440, line 14: "setup" should be "set up".

Review of[4]
Forbidden Configurations in Discrete Geometry
by David Eppstein
Cambridge University Press, 2018
238 pages, \$105.00 (Hardcover), \$39.99 (Paperback)

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

1 Introduction

In 1930, the mathematician Esther Klein observed that any five points in the plane in general position (i.e., no three points forming a line) contain four points forming a convex quadrilateral. This innocent-sounding discovery led to major lines of research in discrete geometry. Klein's friends Paul Erdős and George Szekeres generalized this theorem, and also conjectured that $2^{k-2} + 1$ points (again in general position) would be enough to force a convex k-gon to exist. The resolution of this conjecture became known as the "happy ending problem," because Klein and Szekeres ended up getting married. The unhappy side is that it has, to date, not been completely solved, although a recent breakthrough of Suk made significant progress.

This both mathematically and personally charming little story is a great beginning for this elegant book about discrete geometry. It typifies the type of problems that are studied throughout, and also captures the spirit of curiosity that drives such studies. The book covers many problems that lie at the intersection of three fields: discrete geometry, algorithms and computational complexity.

2 Summary of Contents

There are a total of 18 chapters. A brief introduction (Chapter 1, "A Happy Ending") introduces the Happy Marriage Problem and the associated characters mentioned above (Klein, Erdős , Szekeres), followed by (Chapter 2, "Overview") an outline of the structure and aims of the book. The next five chapters present ideas that are crucial to understanding the rest of the book. The central notions of configurations appear in Chapter 3 ("Configurations"), and subconfigurations in Chapter 4 ("Subconfigurations"). A configuration is defined so as not to depend on the absolute coordinates of its points, but rather their relative orientation. The type of *forbidden* properties that are studied here are those that are *monotone*, i.e., if a given configuration has the property, then so do its subconfigurations. More generally, one defines monotone *parameters* as functions over configurations that do not increase when points are removed from a configuration. *Obstacles* to monotone properties are the smallest subconfigurations that do *not* have a certain property (so all proper subconfigurations of an obstacle *do* have the property). Thus the central property in line with the title of the book, denoted $\textsc{forbidden}(C_1, C_2, \ldots; S)$, is defined to be true iff none of $C_1, C_2, \ldots$ are subconfigurations of S. It is useful to investigate other properties, such as the *size of the largest* subconfiguration which contains a set of obstacles $C_1, C_2, \ldots$; this is denoted $\textsc{avoids}(C_1, C_2, \ldots)$. These latter ideas are introduced and relationships between these properties are proved in Chapter 5 ("Properties, Parameters, and Obstacles"). In order to perform computations, it is important to formalize how to represent configurations and understand the basic ideas of algorithm analysis and property testing, topics introduced in Chapter 6 ("Computing with Configurations"). Chapter 7 ("Complexity Theory") is a quick introduction to computational complexity theory, including the ideas of completeness and hardness, both in the context of standard complexity as well as fixed-parameter complexity and kernelization.

Subsequent chapters treat a rich variety of problems about configurations, based on the framework developed in the early chapters. Thus, for example, since a configuration lies on a single line iff it contains no triangle, a triangle constitutes an *obstacle* to collinearity. Chapter 8 ("Collinearity") presents a number of problems relating to collinearity, beginning with the "orchard-planting problem": Given a configuration, what is the maximum number of lines connecting more than two points? One may also ask for the number of points on a single line within a configuration (the "ONLINE" property). This property is efficiently computable. The chapter also looks into different aspects of hardness, e.g., that the minimum "LINE-COVER" problem is NP-complete, while it is fixed-parameter tractable when the number of lines is small.

A configuration is in *general position* if no three points lie along a line. Thus, a three-point line is an obstacle to general position. While determining if a configuration is in general position can be solved in quadratic time, determining a maximum subconfiguration in general position is NP-hard (and even hard to approximate); nevertheless, the problem is fixed-parameter tractable. These and some related notions and results are the subject of Chapter 9 ("General Position"). It is also of interest to determine the smallest size of a partition into subconfigurations each in general position. This is the topic of Chapter 10 ("General-Position Partitions"), where it is proved that although the problem is NP-complete even if we only want to know if the number of such partitions is < 3, one does obtain an algorithm that gives a square-root approximation ratio.

Proceeding along similar lines, a configuration is *convex* if it forbids 3 points in a line and a *tetrad* (a triangle with an interior point). Chapters 11 and 12 study convexity. The Erdős and Szekeres construction (already referred to way back in Chapter 1), in connection with the happy ending problem, is presented here via an ingenious geometric analog of Pascal's triangle. Other properties involving convex subconfigurations and convex partitions (among others), as well as algorithmic and complexity-theoretic aspects of these problems, are studied in Chapter 11 ("Convexity"). Chapter 12 ("More on Convexity") looks at other aspects of convexity, for example *weak convexity* (regarding configurations in which points may lie on the convex hull without being vertices of it). Weak convexity is characterized by forbidding the tetrad and the *quincunx* (google it, if you're curious...much better, read the book!). The idea of "onion layers" (nested convex layers) is also studied here, in addition to other types of configurations.

Some configurations can be represented in the plane with integer coordinates (for example, any one that is in general position). This may not be possible for certain types of configurations (for example, points on a line). After introducing the Perles Configuration, which does not admit rational coordinates, Chapter 13 ("Integer Realizations") consists mostly of a collection of intriguing open problems along related lines.

Permutations have a simple geometric analogy (think of an array representing a permutation, arranged on a 2D grid). Chapter 14 ("The Stretched Geometry of Permutations") studies the important family of *stretched configurations*, all of which can be derived from these permutations. These ideas lead to a significant strengthening of the happy ending theorem. Configurations can also be derived from graphs, via the process of "convex embedding," as described in Chapter 15 ("Configurations from Graphs"). This technique is the basis of some results quoted in earlier chapters, e.g., the NP-completeness and fixed-parameter hardness of determining if one configuration is a subconfiguration of another (stated in Chapter 7), and some ordering properties of configurations in general position (stated in Chapter 9). Graphs are further pursued in Chapter 16 ("Universality"), in which straight-line drawings of planar graphs are understood in terms of the underlying ("supporting") configurations. In fact, some configurations of n points can support all planar graphs with n vertices, and these are called *universal* for such graphs. The chapter explores various aspects of universality in this sense.

A fundamental property of a set of line segments is the maximum number of those segments intersected ("stabbed") by another line at single points. This is called the *stabbing number*. This and related ideas are the basis of a number of properties of configurations, which can be used to prove several positive theorems about quasi-well-ordering, stated in earlier chapters, e.g., from Chapter 11, the fixed-parameter tractability of monotone parameters of configurations obeying a certain convexity property. This is presented in the penultimate Chapter 17 ("Stabbing").

Finally the results are summarized in Chapter 18 ("The Big Picture").

3 Opinion

This book is distinguished by a number of attractive features. Perhaps most prominent is its strong unity of approach. The first 7 chapters establish a coherent foundation and language for expressing and investigating the subjects studied in the remaining 10.

Secondly, while the geometry studied in the book pretty much confines itself to two dimensions, the mathematical ideas synthesized here are multi-dimensional: Geometry, algorithms, complexity, parameterized complexity, property testing, and combinatorics all play key roles (and this is only a partial list). It is very enlightening to see some of these less "geometrical" topics (such as complexity) manifest themselves in pictorial ways.

Another is its clarity of presentation and reader-friendliness. In most chapters the author adopts the strategy of introducing the topic in terms of an easily-understood problem that is accessible to virtually any reader. These are often in the form of simply-stated geometric puzzles. For example, we learn in Chapter 9 that in 1917 Dudeney asked how one might place 16 pawns on a chessboard without allowing any lines of three pawns. This generalizes to the "no-three-in-a-line" problem: In the language of the book, this asks a question about general position, namely for the value of the parameter MAX-GENERAL(GRID(n, n)) for each n. Thus the earlier parts of each chapter can reach a wide readership. That being said, it is not an easy book. Any reader who desires a deeper understanding of the material would require a significant mathematical background. The later parts of the chapters are more technical, and would be appreciated by specialists in discrete geometry, and/or advanced students or experts in other fields interested in learning about the subject.

The book is certainly suitable for a seminar-style course, at the advanced graduate or graduate level. While it may not be immediately suitable for a conventional course, this is mainly due to the lack of exercises, which doesn't seem like that difficult a problem to overcome.

Although not as important as the advantages noted above, the book's very elegant production is both striking and appealing. I've even shown it to some artist friends, who very much appreciated the numerous beautiful (color) figures.

If you have any interest in learning about this field, I highly recommend this book.