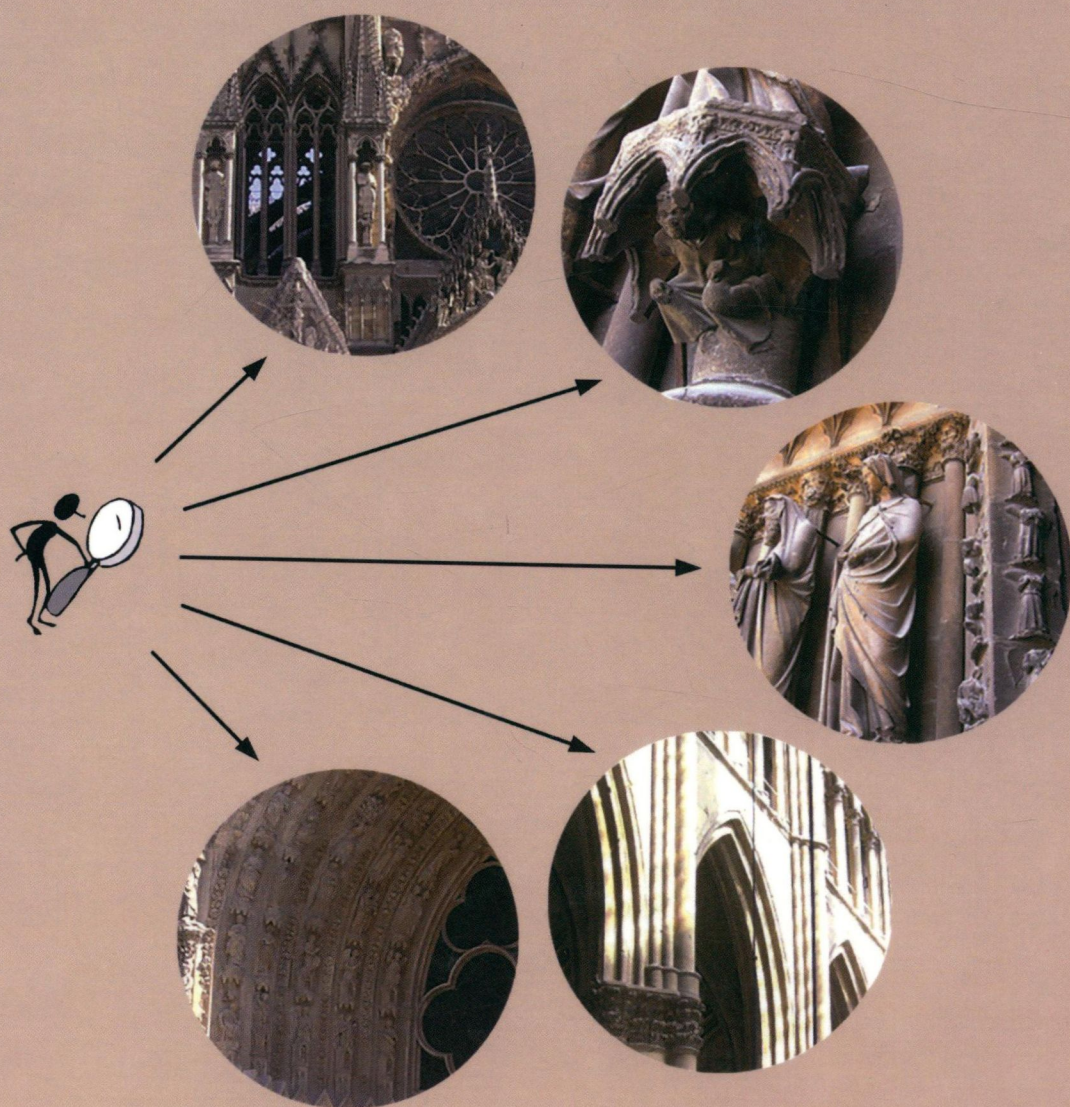


# INTRODUCTION TO **PROPERTY TESTING**



ODED GOLDREICH



**Property testing** is concerned with the design of super-fast algorithms for the structural analysis of large quantities of data. The aim is to unveil global features of the data, such as determining whether the data has a particular property or estimating global parameters. Remarkably, it is possible for decisions to be made by accessing only a small portion of the data. Property testing focuses on properties and parameters that go beyond simple statistics. This book provides an extensive and authoritative introduction to property testing. It provides a wide range of algorithmic techniques for the design and analysis of tests for algebraic properties, properties of Boolean functions, graph properties, and properties of distributions.

**Oded Goldreich** is Professor of Computer Science at the Weizmann Institute of Science and holds the Meyer W. Weisgal Professorial Chair. He has made numerous contributions to property testing and to the theory of computation at large, and was awarded the 2017 Knuth Prize for this work. He is the author of several books, including the two-volume *Foundations of Cryptography* and the single-volume *Computational Complexity: A Conceptual Perspective*. He is an associate editor of the journal *Computational Complexity*, former editor of the *Journal of Cryptology* and *SIAM Journal on Computing*, and has been invited to speak at numerous conferences including the 1994 International Congress of Mathematicians.



GOLDREICH

# Introduction to Property Testing

CAMBRIDGE

# Introduction to Property Testing

**Oded Goldreich**

*Weizmann Institute of Science, Israel*



**CAMBRIDGE**  
UNIVERSITY PRESS



# CAMBRIDGE UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314-321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi - 110025, India

79 Anson Road, #06-04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

[www.cambridge.org](http://www.cambridge.org)

Information on this title: [www.cambridge.org/9781107194052](http://www.cambridge.org/9781107194052)

DOI: 10.1017/9781108135252

© Oded Goldreich 2017

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published 2017

Printed in the United States of America by Sheridan Books, Inc.

*A catalogue record for this publication is available from the British Library.*

*Library of Congress Cataloging-in-Publication Data*

Names: Goldreich, Oded, author.

Title: Introduction to property testing / Oded Goldreich, Weizmann Institute of Science, Israel.

Description: Cambridge, United Kingdom ; New York, NY, USA : Cambridge University Press, 2018. |

Includes bibliographical references and index.

Identifiers: LCCN 2017023051 | ISBN 9781107194052 (hardback : alk. paper)

Subjects: LCSH: Computer algorithms. | Structural analysis (Engineering)

Classification: LCC QA76.9.A43 G655 2018 | DDC 005.1 – dc23

LC record available at <https://lccn.loc.gov/2017023051>

ISBN 978-1-107-19405-2 Hardback

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party Internet Web sites referred to in this publication and does not guarantee that any content on such Web sites is, or will remain, accurate or appropriate.



## Introduction to Property Testing

Property testing is concerned with the design of superfast algorithms for the structural analysis of large quantities of data. The aim is to unveil global features of the data, such as determining whether the data have a particular property or estimating global parameters. Remarkably, it is possible to achieve this aim by accessing only a small portion of the data. Property testing focuses on properties and parameters that go beyond simple statistics.

This book provides an extensive and authoritative introduction to property testing. It provides a wide range of algorithmic techniques for the design and analysis of tests for algebraic properties, properties of Boolean functions, graph properties, and properties of distributions.

Oded Goldreich is Professor of Computer Science at the Weizmann Institute of Science and holds the Meyer W. Weisgal Professorial Chair. He has made numerous contributions to property testing and to the theory of computation at large, and was awarded the 2017 Knuth Prize for this work. He is the author of several books, including the two-volume work *Foundations of Cryptography* and the book *Computational Complexity: A Conceptual Perspective*. He is an associate editor of the journal *Computational Complexity*, former editor of the *Journal of Cryptology* and *SIAM Journal on Computing*, and has been invited to speak at numerous conferences including the 1994 International Congress of Mathematicians.





# Preface

---

Property testing is concerned with the design of superfast algorithms for structural analysis of huge amounts of data, where by structural analysis we mean an analysis aimed at unveiling global features of the data. Examples include determining whether the data as a whole have some property or estimating some global parameter of the data. The focus is on properties and parameters that go beyond simple statistics of the type that refers to the frequency of occurrence of various local patterns. The algorithms are given direct access to items of a huge data set, and determine whether this data set has some predetermined (global) property or is far from having this property. Remarkably, this decision is made by accessing only a small portion of the data set.

In other words, property testing is concerned with the design of *superfast algorithms for approximate decision making*, where the decision refers to properties or parameters of huge objects. In particular, we seek algorithms that inspect only relatively small portions of the huge object. Such algorithms must be randomized and can provide only approximate answers. Indeed, two salient aspects of property testing are that (1) it studies algorithms that can read only parts of the input, and (2) it focuses on algorithms that solve “approximate decision” problems. Both aspects are quite puzzling: What can one do without even reading the entire input? What does approximate decision mean?

The answer is that these two aspects are indeed linked: *Approximate decision* means distinguishing objects that have some predetermined property (i.e., reside in some predetermined set) from objects that are “far” from having the property (i.e., are far from any object having the property), where the notion of distance employed here is the relative number of different symbols in the descriptions of the objects. Such approximate decisions may be valuable in settings in which an exact decision is infeasible or very expensive or just considerably more expensive than obtaining an approximate decision.

The point is that, in many cases, an approximate decision can be achieved by means of *superfast randomized algorithms*. One well-known example is the common practice of estimating various statistics by sampling, which can be cast as a small collection of approximate decision problems (with respect to some threshold values). Research in property testing aims to extend this useful practice to properties that cannot be cast as statistics of values (which are associated with individual members of a large population). Examples in which this goal was achieved include testing properties of functions such as being a low-degree polynomial, being monotone, and depending on a specified number of attributes; testing properties of graphs such as being bipartite and being triangle-free, and testing properties of geometric objects and visual images such as being well clustered and being a convex body.

**Objects as Functions and Their Exploration.** Viewing the input object as a function is natural in the context of algorithms that do not read their entire input. Such algorithms must probe the input at locations of their choice, and such probing can be thought of as querying a function that represents the input. The key point here is that the number of probes (or queries) is smaller than the size of the input, and so decisions are taken after seeing only a small part of the input. However, the inspected positions are not fixed but rather are chosen at random by the algorithm, possibly based on answers obtained to prior queries. Thus, in general, these algorithms may “explore” the input, rather than merely obtain its value at a uniformly selected sample of locations. Such exploration is most appealing when the tested input is a graph, which may be represented by a function (e.g., by its adjacency predicate), but the notion of exploration applies also in other cases.

**Wider Perspective and Connections.** Research in property testing may be both algorithmic and complexity theoretic. This is reflected both in its goals, which may be either the design of better algorithms or the presentation of lower bounds on their complexity, and in its tools and techniques. Such research is related to several areas of computer science and mathematics including combinatorics, statistics, computational learning theory, computational geometry, and coding theory. Historically, property testing was closely associated with the study of Probabilistically Checkable Proofs (PCPs), and some connections do exist between the two, but property testing is not confined to PCPs (and/or to the study of “locally testable codes”).

**This Book.** The current book aims to provide an introduction to property testing, by presenting some of the main themes, results, and techniques that characterize and are used in the area. As usual in such cases, *the choice of material reflects a judgment of what is most adequate for presentation in the context of such an introductory text*, and this selection does not reflect lack of appreciation of the omitted material but rather an opinion that it is less suitable for the intended purpose of the text.

In addition to the selection of material for this book, several choices were made regarding the organization of the material and the amount of interdependencies among its parts.

**Organizational Choices.** We chose to organize the material by the type of objects and the properties being tested. By the “type of object” we refer to the natural perception of the object; for example, whether it is most naturally perceived as a function or as a graph. Within the world of functions, the types correspond to the structure of the domain on which the function is defined (e.g., a group, a vector space, a Boolean hypercube, or a hypergrid). The structure of the domain is often reflected in the invariances that are satisfied by the properties that we consider (e.g., affine invariance, closure under graph isomorphism, etc.). Hence, our organization may be viewed as structurally oriented. (Possible alternatives to our organization include an organization by techniques (as in Ron [242]) or by complexity levels, for example, whether the complexity of testing is independent of the size of the object, is mildly dependent on it, is barely sublinear, or somewhere in between.)<sup>1</sup>

<sup>1</sup> Denoting the size of the object by  $n$ , one may distinguish bounds that are independent of  $n$  from bounds that are polylogarithmic in  $n$ , bounds that are expressed by a constant power of  $n$  (i.e.,  $n^c$  for  $c \in (0, 1)$ ), or are barely sublinear such as  $n/\text{poly}(\log n)$ .



We chose to present the material with as few links between chapters as possible. Of course, all chapters depend on the core notions that are introduced in the first chapter, but additional interdependencies are rare and never heavily relied upon. Hence, the ordering of the other chapters is not very important, although we preferred a specific one (for reasons outlined in the Organization and Chapter Summaries section).

**More Specific Choices.** We chose to use (one-sided error) proximity-oblivious testers (POTs) *whenever possible*. This reflects our belief that when a tester (implicitly or explicitly) consists of repeating a POT for a number of times that depends on the proximity parameter, one should focus on the POT itself and rely on the generic transformation from POTs to standard testers.

For the sake of uniformity,  $n$  always denotes the size of the object in its natural representation (which is not grossly redundant). Hence, objects are typically viewed as functions  $f : [n] \rightarrow R_n$ . Consequently, Boolean functions are presented as  $f : \{0, 1\}^\ell \rightarrow \{0, 1\}$ , where  $n = 2^\ell$  (rather than as having domain  $\{0, 1\}^n$ ).

We made the choice of defining  $\epsilon$ -far as the negation of  $\epsilon$ -close. That is, a string  $x$  is  $\epsilon$ -far from  $S$  if its relative distance from  $S$  is *strictly greater* than  $\epsilon$ , and it is  $\epsilon$ -close otherwise. We warn that in various sources different conventions are used regarding this minor issue.

**The Use of Footnotes.** We use footnotes quite heavily, and definitely much more often than is the norm in textbooks. Although this style is often criticized, it does offer the advantage of focusing on a main thread while deferring relevant elaborations of some related issues to an easy to locate place. We trust the reader to make the choice of whether to continue with the main thread or go for some elaborations of a point. Typical uses of such footnotes fall into two distinct categories. The first use is for the elaboration of technical details, which many readers may be willing to take on faith (and some may even figure out by themselves) but other readers may want to see fully justified before proceeding. The second use of footnotes is for advanced comments, which provide a somewhat wider perspective or refer to sources where such perspectives can be found.

**Technical Style.** At times, justifications for a sequence of (in)equalities appear after the sequence. This is typically done when we believe that these justifications are implicit in the text that preceded the sequence.

**Teaching Note:** The book contains several teaching notes, set as boxed text such as this.

**Required Preliminaries.** There are no required preliminaries for this text, but basic familiarity with some notions and results of the theory of computation and probability theory will be useful. These include

1. The notions of decision, search, and promise problems (see, e.g., [131, Sec. 1.2]);
2. Probabilistic algorithms (see, e.g., [131, Sec. 6.1] or [212]); and
3. Basic notions and facts regarding discrete probability distributions, including probabilistic inequalities such as the Union Bound and Chernoff Bound (see Appendix A,

although this material is covered in many textbooks, including [131, Apdx. D.1] and [212]).

**Website for Notices Regarding This Book.** We intend to maintain a website for this book, listing corrections and updates of various types. The site is located at

<http://www.wisdom.weizmann.ac.il/~oded/pt-intro.html>

# Organization and Chapter Summaries

---

All chapters rely on a few core notions that are introduced in Sections 1.3.1 and 1.3.3. Although these parts of Section 1.3 provide a sufficient basis for reading any of the subsequent chapters, we strongly recommend reading the entire first chapter before proceeding to any other one.

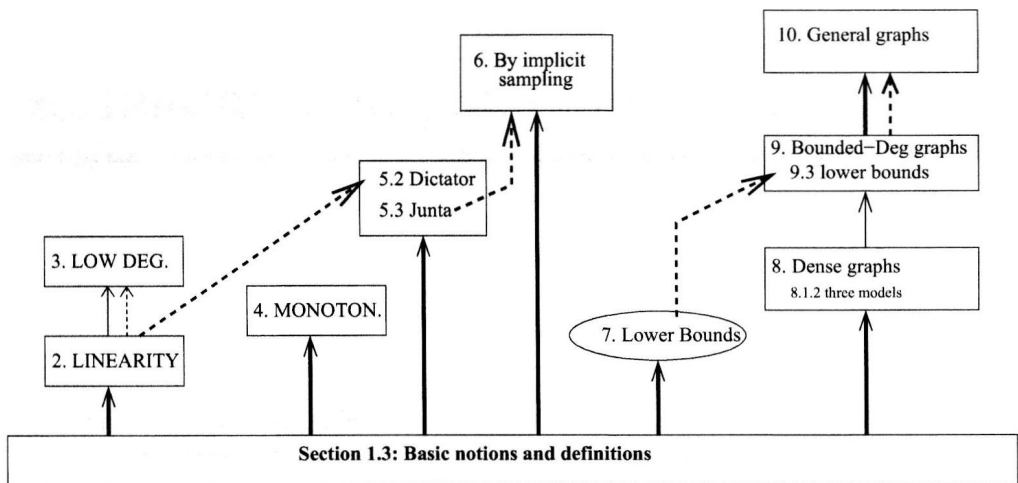
In contrast to the central role of Chapter 1, no other chapter is essential for reading the other chapters. In particular, interdependencies between the other chapters are rare and never heavily relied on. The main dependencies are depicted in Figure 1, where thematic dependencies are marked by solid lines and technical dependencies by dashed lines.

Although the ordering of the chapters that follow Chapter 1 is not very important, a choice had to be made. We chose to start with simple properties of functions such as group homomorphism, low-degree polynomials, monotonicity (with respect to various partial orders), and depending on few variables (i.e., juntas). In all these cases, the correspondence between the object and its representation is transparent: *the function is the object*. In contrast, when moving to graph properties, the question of representation arises in an acute manner, and three different chapters are devoted to three different representations that correspond to three different testing models. Hence, from the perspective of property testing per se, it seems to make sense to start with functions and then move to graphs.

In accordance with the foregoing, the first cluster of chapters (Chapters 2–6) deals with testing properties of functions, whereas a second cluster (Chapters 8–10) deals with testing properties of graphs. A chapter on lower bound techniques (i.e., Chapter 7) is located in between these two clusters, since lower bounds are hardly mentioned in the first cluster, whereas they appear quite prominently in Chapters 9 and 10. The reason for this phenomenon is that these lower bounds are used to justify the significantly higher complexity of some testers that are presented in Chapters 9 and 10. Indeed, in the context of this book, we view lower bounds mainly as a justification for algorithms that may be considered to have a higher than expected complexity; the lower bounds assert that this impression is actually false, and that one cannot do significantly better.

Chapters 11–13 form a third cluster, which is actually a cluster of outliers with respect to the rest of this book. These chapters are indeed related to the previous chapters and yet they have a different flavor: Chapter 11 deals with testing properties of distributions, Chapter 12 explores a few variants of the basic setting (some of which were mentioned in Section 1.3.2), and Chapter 13 reviews locally testable codes and proofs. We stress that in Chapter 11 the tested objects are fundamentally different from those considered in all the other chapters, whereas in Chapter 13 we consider objects that are artificially designed so to offer superfast testing.





**Figure 1:** Dependencies among Chapters 1–10. Thicker lines represent greater dependency.

### Chapter 1: The Main Themes (Approximate Decision and Sublinear Complexity).

This chapter introduces and illustrates the basic notions of property testing, emphasizing the themes of *approximate decision* and *sublinear complexity*. The chapter starts with a discussion of the potential benefits of property testing and culminates with a presentation of the definitions of (standard) testers and of proximity-oblivious testers (POTs). These definitions (and the relationship between them) are used extensively throughout the book. In addition, Chapter 1 discusses the key role of representation, points out the focus on properties that are not fully symmetric, and presents several general observations regarding POTs, testing, and learning.

**Teaching Note:** The conceptual framework put forward in Chapter 1 is pivotal for the rest of the book. It is essential that the main definitions (presented in Sections 1.3.1 and 1.3.3) and the rationales underlying them be internalized.

The following five chapters deal with properties of objects that are most naturally viewed as functions: Chapters 2 and 3 deal with algebraic properties, whereas Chapters 4–6 deal mostly with properties of Boolean functions. This distinction is quite fuzzy, and is reflected in the difference between the invariants that these properties satisfy: Algebraic properties are invariant under general affine transformations of the corresponding vector spaces, whereas the properties of Boolean functions that we consider are invariant only under transformations that permute the basis vectors.

**Chapter 2: Testing Linearity (Group Homomorphism).** This chapter presents an analysis of a linearity tester that, on input a description of two groups  $G, H$  and oracle access to a function  $f : G \rightarrow H$ , queries the function at three points and satisfies the following conditions:

1. If  $f$  is a homomorphism from  $G$  to  $H$ , then the tester accepts with probability 1.
2. If  $f$  is  $\delta$ -far from the set of all homomorphisms from  $G$  to  $H$ , then the tester rejects with probability at least  $\min(0.5\delta, 0.1666)$ .

The three queries are  $x, y, x + y$ , where  $x$  and  $y$  are selected uniformly at random in  $G$ . The archetypical case is that  $H$  is a finite field and  $G$  is a vector space over this field.

**Chapter 3: Low-Degree Tests.** For a finite field of prime cardinality  $\mathcal{F}$ , a degree bound  $d < |\mathcal{F}|/2$ , and number  $m \in \mathbb{N}$ , we consider the problem of testing whether a function  $f : \mathcal{F}^m \rightarrow \mathcal{F}$  is a polynomial of total degree at most  $d$ . We present and analyze a low-degree tester that, given oracle access to  $f : \mathcal{F}^m \rightarrow \mathcal{F}$ , queries it at  $d + 2$  points and satisfies the following conditions:

1. If  $f$  is an  $m$ -variate polynomial of (total) degree  $d$ , then the tester accepts with probability 1.
2. If  $f$  is  $\delta$ -far from the set of  $m$ -variate polynomials of (total) degree  $d$ , then the tester rejects with probability at least  $\min(0.5\delta, \Omega(d^{-2}))$ .

The sequence of queries is generated by selecting at random  $\bar{x}$  and  $\bar{h}$  uniformly in  $\mathcal{F}^m$ , and using  $\bar{x} + i\bar{h}$  as the  $i^{\text{th}}$  query.

**Teaching Note:** The analysis of the low-degree test is quite similar to the analysis of the linearity test, alas it is more complex (let alone that it depends on elementary preliminaries that are presented in Section 3.3). Hence, if short on time, then do consider skipping Chapter 3.

**Chapter 4: Testing Monotonicity.** For each  $n$ , we consider functions from a partially ordered set  $D_n$  to a totally ordered set  $R_n$ . Such a function  $f : D_n \rightarrow R_n$  is called monotone if for every  $x < y$  in  $D_n$  it holds that  $f(x) \leq f(y)$ , where  $<$  denotes the partial order of  $D_n$  and  $\leq$  refers to the total order in  $R_n$ . Two special cases of interest are

1. Boolean functions on the Boolean hypercube: In this case,  $D_n$  is the  $\ell$ -dimensional Boolean hypercube (with the natural partial order), where  $\ell = \log_2 n$ , and  $R_n = \{0, 1\}$ . According to this partial order,  $x_1 \cdots x_\ell \leq y_1 \cdots y_\ell$  if and only if  $x_i \leq y_i$  for every  $i \in [\ell]$ .
2. Real functions on the discrete line: In this case,  $D_n = [n]$  and  $R_n = \mathbb{R}$ , both with the natural total order.

Combining these two extremes, we also consider the case of the hypergrid domain  $D_n = [m]^\ell$ , for any  $m, \ell \in \mathbb{N}$  such that  $m^\ell = n$ , and general ranges  $R_n$ . In all these cases, we present property testers of complexity  $\text{poly}(\epsilon^{-1} \log n)$ . In addition, we briefly survey relatively recent developments regarding the first case as well as known results regarding testing convexity, submodularity, and the Lipschitz property of functions from  $[m]^\ell$  to  $\mathbb{R}$ .

**Teaching Note:** Only parts of Chapter 4 (i.e., Sections 4.2.1 and 4.3.1) are recommended for teaching, and the rest is better left for optional independent reading.

**Chapter 5: Testing Dictatorships, Juntas, and Monomials.** We consider testing three basic properties of Boolean functions of the form  $f : \{0, 1\}^\ell \rightarrow \{0, 1\}$ :

1. Dictatorship: The case where the value of  $f$  depends on a single Boolean variable (i.e.,  $f(x) = x_i \oplus \sigma$  for some  $i \in [\ell]$  and  $\sigma \in \{0, 1\}$ )

2. Junta (of size  $k$ ): The case where the value of  $f$  depends on at most  $k$  Boolean variables (i.e.,  $f(x) = f'(x_I)$  for some  $k$ -subset  $I \subset [\ell]$  and  $f' : \{0, 1\}^k \rightarrow \{0, 1\}$ )
3. Monomial (of size  $k$ ): The case where the value of  $f$  is the conjunction of exactly  $k$  Boolean literals (i.e.,  $f(x) = \bigwedge_{i \in I} (x_i \oplus \sigma_i)$  for some  $k$ -subset  $I \subseteq [\ell]$  and  $\sigma_1, \dots, \sigma_\ell \in \{0, 1\}$ )

We present two different testers for dictatorship, where one generalizes to testing  $k$ -juntas and the other generalizes to testing  $k$ -monomials. (The presentation starts with the latter tester for dictatorship, sketches its generalization to testing  $k$ -monomials, and ends with the tester for  $k$ -juntas.)

**Teaching Note:** We suggest leaving the overview section that discusses testing monomials (i.e., Section 5.2.2) for advanced independent reading.

**Chapter 6: Testing by Implicit Sampling.** Building on the junta tester, we present a general methodology for constructing testers for properties of Boolean functions (of the form  $f : \{0, 1\}^\ell \rightarrow \{0, 1\}$ ) that can be approximated by small juntas. This methodology yields testers of low query complexity for many natural properties, which contain functions that depend on relatively few relevant variables; specifically, the query complexity is related to the size of the junta and is independent of the length of the input to the function (i.e.,  $\ell$ ).

**Chapter 7: Lower Bounds Techniques.** We present and illustrate three techniques for proving lower bounds on the query complexity of property testers.

1. Showing a distribution on instances that have the property and a distribution on instances that are far from the property such that an oracle machine of low query complexity cannot distinguish these two distributions
2. Showing a reduction from communication complexity; that is, showing that a communication complexity problem of high complexity can be solved within communication complexity that is related to the query complexity of the property testing task that we are interested in
3. Showing a reduction from another testing problem; that is, showing a “local” reduction of a hard testing problem to the testing problem that we are interested in

We also present simplifications of these techniques for the cases of one-sided error probability testers and nonadaptive testers.

**Teaching Note:** The first method (i.e., the method of “indistinguishability of distributions”) is used much more often than the other two methods, and studying it should be at the highest priority.

The following three chapters deal with properties of objects that are most naturally viewed as graphs: These chapters consider three models that differ in the way that graphs are represented (and by the definition of relative distance between graphs).



**Teaching Note:** Chapters 8–10 contain material that may occupy half the duration of a course that is based on the current book. Hence, painful choices will have to be made, unless a decision is made to spend this amount of time on studying the topic of testing graph properties, which is not an unreasonable decision in light of the ubiquitous presence of graphs in computer science. Our own (painful) choices regarding the material to be taught appear in the introductory sections of these chapters.

**Chapter 8: Testing Graph Properties in the Dense Graph Model.** Following a general introduction to testing graph properties, this chapter focuses on the dense graph model, where graphs are represented by their adjacency matrix (predicate). The highlights of this chapter include

1. A presentation of a natural class of graph properties that can each be tested within query complexity that is polynomial in the reciprocal of the proximity parameter. This class, called general graph partition problems, contains properties such as  $t$ -colorability (for any  $t \geq 2$ ) and properties that refer to the density of the max-clique and to the density of the max-cut in a graph.
2. An exposition of the connection of testing (in this model) to Szemerédi's Regularity Lemma. The starting point and pivot of this exposition is the existence of constant-query (one-sided error) proximity-oblivious testers for all subgraph freeness properties.

We conclude this chapter with a taxonomy of known testers, organized according to their query complexity.

**Chapter 9: Testing Graph Properties in the Bounded-Degree Graph Model.** This chapter is devoted to testing graph properties in the bounded-degree graph model, where graphs are represented by their incidence lists (lumped together in an incidence function). The highlights of this chapter include

1. Upper and lower bounds on the complexity of testing **Bipartiteness**; specifically, we present a  $\text{poly}(1/\epsilon) \cdot \tilde{O}(\sqrt{k})$ -time tester, and an  $\Omega(\sqrt{k})$  lower bound on the query complexity of any tester for **Bipartiteness**.
2. A quasi- $\text{poly}(1/\epsilon)$ -time tester for **Planarity**. The result extends to testing any minor-closed property (i.e., a graph property that is preserved under the omission of edges and vertices and under edge contraction).

We conclude this chapter with a taxonomy of known testers, organized according to their query complexity.

**Chapter 10: Testing Graph Properties in the General Graph Model.** This chapter is devoted to testing graph properties in the general graph model, where graphs are inspected via incidence and adjacency queries, and distances between graphs are normalized by their actual size (i.e., actual number of edges). The highlights of this chapter include

1. Demonstrating the derivation of testers for this model from testers for the bounded-degree graph model

2. Studying the tasks of estimating the number of edges in a graph and sampling edges uniformly at random

We conclude this chapter with some reflections regarding the three models of testing graph properties.

**Teaching Note:** Although it is possible to study Chapter 10 without first studying Chapter 9, we strongly recommend not doing so. A basic familiarity with the bounded-degree graph model and some of the results regarding it will greatly facilitate the study of the general graph model. See further comments at the beginning of Chapter 10.

The last three chapters explore topics that are related to but significantly different from the topics studied in the previous chapters. Chapter 11 is most different in flavor, since it refers to a totally different type of objects and to a very different model of testing such objects. Chapter 13 seems more in line with the previous chapters, but it differs from them in considering objects that are artificially designed so to offer superfast testing. The topics explored in Chapter 12 are closest in spirit to those explored in previous chapters (and, indeed, some of these topics were mentioned in Section 1.3.2).

**Teaching Note:** Chapter 11 can be read without reading any prior chapter (i.e., not even Sections 1.3.1 and 1.3.3), but some perspectives will be lost when doing so. Given the different flavor of this chapter, we recommend placing it at the end of a course based on the current book.

**Chapter 11: Testing Properties of Distributions.** This chapter provides an introduction to the study of testing properties of distributions, where the tester obtains samples of an unknown distribution (resp., samples from several unknown distributions) and is required to determine whether the distribution (resp., the tuple of distributions) has some predetermined property. We focus on the problems of testing whether an unknown distribution equals a fixed distribution and of testing equality between two unknown distributions. Our presentation is based on reductions from the general cases to some seemingly easier special cases. In addition, we also provide a brief survey of general results.

**Teaching Note:** Chapters 12 and 13 are intended for optional independent reading. They both have more of the flavor of a survey than of a textbook. Chapter 12 follows up on topics that were mentioned briefly in prior chapters (including in Section 1.3.2). Chapter 13 focuses on topics that are somewhat related to property testing, while building on results presented in Chapters 2 and 3 (but doing so in a self-contained manner). Indeed, Chapter 13 can be read independently of the other chapters.

**Chapter 12: Ramifications and Related Topics.** We briefly review a few ramifications of the notion of property testers as well as related topics. The list includes tolerant testing and distance approximation; testing in the presence of additional promises on the input; sample-based testers; testing with respect to other distance measures; local computation algorithms; and noninteractive proofs of proximity (MAPs). The different sections of this chapter can be read independently of one another.