A Kind of Preface

Oded Goldreich
Department of Computer Science
Weizmann Institute of Science
Rehovot, Israel.
oded.goldreich@weizmann.ac.il

March 12, 2010

Introduction

Property Testing is the study of super-fast (randomized) algorithms for approximate decision making. These 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 approximate decision is made by accessing a small portion of the data set.

Property Testing has been a subject of intensive research in the last couple of decades (see, e.g., recent surveys [13, 14]), with hundreds of studies conducted in it and in closely related areas. Indeed, Property Testing is closely related to Probabilistically Checkable Proofs (PCPs), and is related to Coding Theory, Combinatorics, Statistics, Computational Learning Theory, Computational Geometry, and more.

The mini-workshop, hosted by the Institute for Computer Science (ITCS) at Tsinghua University (Beijing), brought together a couple of dozen of leading international researchers in Property Testing and related areas. At the end of this mini-workshop it was decided to compile a collection of extended abstracts and surveys that reflect the program of the mini-workshop. The result is the current volume.

Property Testing at a Glance

Property testing is a relaxation of decision problems and it focuses on algorithms that can only read parts of the input. Thus, the input is represented as a function (to which the tester has oracle access) and the tester is required to accept functions that have some predetermined property (i.e., reside in some predetermined set) and reject any function that is "far" from the set of functions having the property. Distances between functions are defined as the fraction of the domain on which the functions disagree, and the threshold determining what is considered far is presented as a proximity parameter, which is explicitly given to the tester.

An asymptotic analysis is enabled by considering an infinite sequence of domains, functions, and properties. That is, for any n, we consider functions from D_n to R_n, where $|D_n| = n$. (Often, one just assumes that $D_n = [n] \stackrel{\rm def}{=} \{1, 2, ..., n\}$.) Thus, in addition to the input oracle, representing a function $f : D_n \to R_n$, the tester is explicitly given two parameters: a size parameter, denoted n, and a proximity parameter, denoted ϵ.

Definition 1 *Let* $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$*, where* Π_n *contains functions defined over the domain* D_n*. A* tester for a property Π *is a probabilistic oracle machine* T *that satisfies the following two conditions:*

1. *The tester accepts each* $f \in \Pi$ *with probability at least* $2/3$*; that is, for every* $n \in \mathbb{N}$ *and* $f \in \Pi_n$ (and every $\epsilon > 0$)*, it holds that* $\Pr[T^f(n,\epsilon)=1] \geq 2/3$.

2. *Given* $\epsilon > 0$ *and oracle access to any* f *that is* ϵ*-far from* Π*, the tester rejects with probability at least* $2/3$*; that is, for every* $\epsilon > 0$ *and* $n \in \mathbb{N}$*, if* $f : D_n \to R_n$ *is* ϵ*-far from* Π_n*, then* $\Pr[T^f(n,\epsilon)=0] \geq 2/3$*, where* f *is* ϵ-far from Π_n *if, for every* $g \in \Pi_n$*, it holds that* $|\{e \in D_n : f(e) \neq g(e)\}| > \epsilon \cdot n$.

If the tester accepts every function in Π *with probability 1, then we say that it has* one-sided error*; that is,* T *has one-sided error if for every* $f \in \Pi$ *and every* $\epsilon > 0$*, it holds that* $\Pr[T^f(n,\epsilon)=1] = 1$*. A tester is called* non-adaptive *if it determines all its queries based solely on its internal coin tosses* (and the parameters n and ϵ)*; otherwise it is called* adaptive.

Definition 1 does not specify the query complexity of the tester, and indeed an oracle machine that queries the entire domain of the function qualifies as a tester (with zero error probability...). Needless to say, we are interested in testers that have significantly lower query complexity.

Research in property testing is often categorized according to the type of functions and properties being considered. In particular, algebraic property testing focuses on the case that the domain and range are associated with some algebraic structures (e.g., groups, fields, and vector spaces) and studies algebraic properties such as being a polynomial of low degree (see, e.g., [5, 15]). In the context of testing graph properties (see, e.g., [6]), the functions represent graphs or rather allow certain queries to such graphs (e.g., in the adjacency matrix model, graphs are represented by their adjacency relation and queries correspond to pairs of vertices where the answers indicate whether or not the two vertices are adjacent in the graph).[1]

Ramifications. While most research in property testing refers to distances with respect to the uniform distribution on the function's domain, other distributions and even distribution-free models were also considered. That is, for a (known or unknown) distribution μ on the domain, we say that f is ϵ-far from g (w.r.t μ) if $\Pr_{e\sim\mu}[f(e)\neq g(e)] > \epsilon$. Indeed, Definition 1 refers to the case that μ is uniform over the domain (i.e., D_n).

A somewhat related model is one in which the tester obtains random pairs $(e, f(e))$, where each sample e is drawn (independently) from the aforementioned distribution. Such random (f-labeled) example can be either obtained on top of the queries to f or instead of them. This is also the context of testing distributions, where the examples are actually unlabeled and the aim is testing properties of the underlying distribution (rather than properties of the labeling which is null here).

A third ramification refers to the related notions of *tolerant testing* and *distance approximation* (cf. [12]). In the latter, the algorithm is required to estimate the distance of the input (i.e., f) from the predetermined set of instances having the property (i.e., Π). Tolerant testing usually means only a crude distance approximation that guarantees that inputs close to Π (rather than only inputs in Π) are accepted while inputs that are far from Π are rejected (as usual).

[1] In an alternative model, known as the incidence-list model, graphs are represented by functions that assign to the pair (v, i) the ith neighbor of vertex v.

On the current focus on query complexity. Current research in property testing focuses mainly on query (and/or sample) complexity, while either ignoring time complexity or considering it a secondary issue. The current focus on these information theoretic measures is justified by the fact that even the latter are far from being understood. (Indeed, this stands in contrast to the situation in, say, PAC learning.)

On the importance of representation. The representation of problems' instances is crucial to any study of computation, since the representation determines the type of information that is explicit in the input. This issue becomes much more acute when one is only allowed partial access to the input (i.e., making a number of queries that result in answers that do not fully determine the input). An additional issue, which is unique to property testing, is that the representation may effect the distance measure (i.e., the definition of distances between inputs). This is crucial because property testing problems are defined in terms of this distance measure.

The importance of representation is forcefully demonstrated in the gap between the complexity of testing numerous natural graph properties in two natural representations: the adjacency matrix representation (cf. [6]) and the incidence lists representation (cf. [7]).

Things get to the extreme in the study of locally testable codes, which may be viewed as evolving around testing whether the input is "well formed" with respect to some fixed error correcting code. Interestingly, the general study of locally testable codes seeks an arbitrary succinct representation (i.e., a code of good rate) such that well-formed inputs (i.e., codewords) are far apart and testing well-formness is easy (i.e., there exists a low complexity codeword test).

Caveat. Needless to say, this section falls very short of providing a comprehensive account of research in property testing. Furthermore, the presentation is biased towards the presentation of models and the numerous results obtained in the various models are barely mentioned. The same holds with respect to the next section.

A Brief Historical Perspective

Property testing first appeared as a tool towards program checking (see the linearity tester of [5]) and the construction of PCPs (see the low-degree tests and their relation to locally testable codes, as discussed in [15]). In these settings it was natural to view the tested object as a function, and this convention continued also in [6], which defined property testing in relation to PAC learning. More importantly, in [6] property testing is promoted as a new type of computational problems, which transcends all its natural applications.

While [5, 15] focused on algebraic properties, the focus of [6] was on graph properties. From this perspective the choice of representation became less obvious, and oracle access was viewed as allowing local inspection of the graph rather than being the graph itself.[2] The distinction between objects and their representations became more clear when an alternative representation of graphs was studied in [7, 8]. At this point, query complexity that is polynomially related to the size of the object (e.g., its square root) was no longer considered inhibiting. This shift in scale is discussed next.

Recall that initially property testing was viewed as referring to functions that are implicitly defined by some succinct programs (as in the context of program checking) or by "transcendental"

[2]That is, in this case the starting point is the (unlabeled) graph itself, and its representation as a (labeled) graph by either its adjacency matrix or incidence list is an auxiliary conceptual step.

entities (as in the context of PAC learning). From this perspective the yardstick for efficiency is being polynomial in the length of the query, which means being polylogarithmic in the size of the object. However, when viewing property testing as being applied to (huge) objects that may exist in explicit form in reality, it is evident that any sub-linear complexity may be beneficial.

The realization that property testing may mean any algorithm that does not inspect its entire input seems crucial to the study of testing distributions, which emerged with [3]. In general, property testing became identified as a study of a special type of sublinear-time algorithms.

Another consequence of the aforementioned shift in scale is the decoupling of the representation from the query types. In the context of graph properties, this culminated in the model of [9].

Nevertheless, the study of testing properties within query complexity that only depends on the proximity parameter (and is thus totally independent of the size of the object) remains an appealing and natural direction. A remarkable result in this direction is the characterization of graph properties that are testable within such complexity in the adjacency matrix model [1].

The Program of the Mini-Workshop

Following is the original program, annotated by brief comments.

Session 1 (Friday, January 8, 9:00-10:15) [chair: Oded Goldreich]

- Amy Wang: Welcoming comments.
- Rocco Servedio: Testing by Implicit Learning.
- Kevin Matulef: Testing (subclasses of) Linear Threshold Functions.

Kevin was ill, and so Rocco gave both talks. I am fascinated by the **implicit learning** paradigm, which is pivoted on emulating learning algorithms for k-variable functions by using n-bit long samples with k influential variables. The emulation proceeds by randomly partitioning the variables to $O(k^2)$ sets, identifying k sets that are likely to each contain a single influential variable, and converting the n-bit long samples to corresponding k-bit samples. Specifically, each n-bit sample is converted by determining for each influential set whether the subset of variables labeled 1 or the subset labeled 0 is influential, and setting the corresponding bit accordingly. Once the learning algorithm outputs a hypothesis for the k-bit function, it is tested using adequate queries (which are again emulated correspondingly). The second work shows that, while the class of all halfspaces is testable in query complexity that is independent of n, testing some natural subclasses of halfspaces (i.e., "unate reorientations of majority" where the weights are ± 1) requires query complexity that depends on n.

Session 2 (Friday, January 8, 10:45-12:00) [chair: Avrim Blum]

- Eric Blais: Testing juntas and function isomorphism.
- Michael Krivelevich: Hierarchy Theorems for Property Testing

Eric motivated the problem of optimizing the complexity of testing juntas by referring to the wide applicability of juntas (i.e., functions that depend on few of their inputs). The algorithm presented in Eric's work is very appealing, and has query complexity $\widetilde{O}(k/\epsilon)$, where ϵ is the proximity parameter. In addition, Eric advocated a study of the following class of parameterized problems. For a fixed function g, which is known to the tester, the task is testing whether the given oracle/function is isomorphic to g (i.e., f equals g under a relabeling of the indices of the variables). In a sharp shift from the concrete to the general, Michael presented an overview of results that assert the existence of properties with arbitrary (reasonable) query complexity bound.

Session 3 (Friday, January 8, 14:00-15:30) [chair: Bernard Chazelle]

- Ilan Newman: A Survey on Property Testing in the "Underlying Graph" Model or the "Massively Parameterized" Model.
- Sofya Raskhodnikova: Transitive-Closure Spanners with Applications to Monotonicity Testing.
- Christian Sohler: Testing Euclidean Spanners.

Ilan surveyed works that refer to "massively parameterized" testing problems. The parameters are huge structures (e.g., graphs) and the problem refers to a substructure of it (e.g., an assignment to the graph's edges). In the case of graphs, the assignment may be viewed as an orientation of the fixed graph or a subgraph of it. Sofya discussed the notion of transitive-closure spanners (i.e., spanners of the transitive closure graph), which are implicit in some monotonicity testers. Christian discussed the notion of geometric graphs, which are graphs embedded in a Euclidean space R^d, and the corresponding notion of spanners (which are geometric graphs in which the graphical distance between pairs of vertices is approximated by their Euclidean distance).

Panel 1 (Friday, January 8, 16:00-17:00) *On the Connection of Property Testing to Computational Learning Theory and Computational Geometry.* Panelists: Avrim Blum, Bernard Chazelle, and Rocco Servedio.

Bernard advocated an attempt to provide a sublinear time analysis of dynamic systems, which may consist of selecting few objects and tracing their activity during time. Avrim emphasized the high cost of queries (rather than random examples) in typical learning applications, and I suggested to consider a two-parameter complexity measure that separates the number of random examples from the number of actual queries. Rocco suggested to try to relate property testing to agnostic learning (rather than to standard learning), and highlighted the open problem of tolerant testing of half planes.

Session 4 (Saturday, January 9, 9:00-10:15) [chair: Shafi Goldwasser]

- Madhu Sudan: Invariance in Property Testing.
- Victor Chen: Testing Linear-Invariant Non-Linear Properties.

Madhu mused on what makes problems easily testable and suggested that closure under a rich permutation group plays a major role. This seems supported by the numerous results regarding algebraic functions (which are all invariant under suitable affine transformations) and graph properties (which, by definition, are closed under isomorphism). He surveyed a unified approach that refers to the former, where the affine transformations refer to the identification of the domain with a vector space. Victor demonstrated that the approach extends also to non-linear properties, and Noga commented that an alternative demonstration may be provided by a property that is a union of two disjoint linear spaces.

Session 5 (Saturday, January 9, 10:45-12:00) [chair: Michael Saks]

- Asaf Shapira: Testing Linear Invariant Properties.
- Noga Alon: On Constant-Time Approximation of Invariants of Bounded Degree Graphs.
- Ronitt Rubinfeld: Maintaining a Large Matching or a Small Vertex Cover.
- Krzysztof Onak: External Sampling.

Asaf's study refers to the paramertized problem of whether a given subset of $[n]$ is free from containing any solution to a fixed set of linear equations. This was studied before with respect to a single equation (i.e., $x + y = z$), and Asaf's work treats any linear system. Noga's study refers to approximating quantities such as the independence number of bounded-degree graphs. He shows that constant-time algorithms can almost match the best approximation bounds that are previously known for PTAS, and that better bounds cannot be achieved in constant-time. (This is related to Krzysztof's presentation in Session 6, which follows prior work of Dana Ron *et al.* [11, 10].) Ronitt's work demonstrates that techniques employed in the context of distributed algorithms can be applied to the context of dynamic graph algorithms. This augments prior connections between constant-round distributed algorithms and constant-time approximation algorithms (discovered by Parnas and Ron [11]). Ronitt asked whether a direct relation can be shown between dynamic graph algorithms and constant-time approximation algorithms. Krzysztof advocated applying "external memory cost" measures to property testing algorithms. He showed such a result for the problem of element distinctness, presenting an algorithm that samples $\sqrt{nB/\epsilon}$ random B-bit blocks.

Session 6 (Saturday, January 9, 14:00-15:30) [chair: Noga Alon]

- Krzysztof Onak: Sublinear Graph Approximation Algorithms.
- Michael Saks: Local Monotonicity Reconstruction.
- Ronitt Rubinfeld: Testing Properties of Distributions (a survey).

A natural setting in which sublinear time algorithms may be useful is in approximating the value of various graph theoretic quantities, especially when these quantities are hard to determine (e.g., minVC). Parnas, Ron, and Marko [11, 10] focused on maximal matching and minVC, while observing a correspondence between constant-time approximation algorithms for the size of the maximal independent set and distributed algorithms that find such sets. Krzysztof surveyed further developments along these lines, focusing on the extension of the study to many other problems (e.g., maximum matching, minimum dominating sets, etc) and the introduction of new techniques and improved results. The new techniques are also applicable to various testing problems; see the improvement obtained over [4] in the context of testing minor-free graphs in the bounded-degree model. Michael discussed the local reconstruction problem applied to objects for which there are exponentially many adequate solutions. The problem in this case is to determine some fixed solution as a function of the corrupted oracle and the random bits (used in the reconstruction), where in some applications it is desirable to use few random bits. Ronitt surveyed the study of testing properties of distributions, starting with the problem of testing identity to some known distribution (i.e., a parameterized problem), which is solvable by $\sqrt{n}$ samples when n is the domain's size (and the estimate is via collision probabilities). Testing that two given distributions (i.e., both distributions are "given" by samples) requires 3-way collisions and so $n^{2/3}$ samples. In contrast, approximating the distance to the uniform distribution requires an "almost linear" number of samples.

Panel 2 (Saturday, January 9, 16:00-17:00) *On the Connection of Property Testing to Coding Theory, Combinatorics, and Statistics.* Panelists: Madhu Sudan, Noga Alon, and Ronitt Rubinfeld.

Noga focused on the "global vs local" nature of property testing that is closely related to a main theme in combinatorics initiated by Erdos in the 1950's. He noted that a tester of k-colorability is implicit in work of the 1980's, but the query bound obtained there is a tower of exponents in the proximity parameter. Answering a question, Noga kind of suggested the challenge of testing triangle-freeness in $2^{O(1/\epsilon)^2}$ queries. (Recall that the currently best known bound is a tower of exponents in $\text{poly}(1/\epsilon)$.) Ronitt reported on the awakening of interest of statisticians in the effect of the size of the domain (of values). (Traditionally, Statistics is concerned with the effect of the size of the universe (sample space), whereas we are concerned with the effect of the number of values actually obtained by samples in this space.) Still there are significant cultural differences, since in Statistics one often makes (implicit) assumptions about the distribution and/or assumes that the distribution itself is selected at random among some possibilities (hence "likelihood" is relevant). Ronitt also pointed out possible applications to areas that use Statistics such as natural language processing and data bases. Madhu pointed out that sublinear time algorithms arise naturally in the context of coding theory. Shafi advocated the study of property testing of algebraic (and number theoretic) problems beyond linearity and low-degree properties. A concrete example may be testing that a polynomial (given by its evaluations) is an irreducible polynomial. Can a tester be significantly more efficient than polynomial interpolation?

Session 7 (Sunday, January 10, 9:00-10:15) [chair: Ronitt Rubinfeld]

- Prahladh Harsha: Composition of Low-Error 2-Query PCPs using Decodable PCPs.
- Victor Chen: A Hypergraph Dictatorship Test with Perfect Completeness.
- Michael Krivelevich: Comparing the Strength of Query Types in Property Testing.

Prahladh started by relating two-query PCPs to (highly) robust PCPs, and proceeded in terms of the latter. to two-query PCPs. The new composition of robust PCPs starts by observing that there is no need to keep the outer proof for consistency, since consistency can be tested by comparing two related inner proofs (which refer to the same position in the outer proof). This, however, changes nothing because still consistency can be achieved by modifying less than half of the total length of both scanned parts. The new idea is to check consistency among d such parts, which may allow to reduce robustness error to approximately $1/d$. Indeed, this is doable, and is what the new composition theorem does, while relying on the fact that the inner PCP is decodable. My view is that the resulting two-query low-error PCPs are interesting mainly because decreasing the error to an arbitrarily small constant (and even below) does not result in increasing the proof length. Moving from PCPs to codeword tests, Victor surveyed the state of the art regarding the amortized query complexity of dictatorship testing (with perfect completeness), which currently stands on soundness error of $O(q \cdot 2^{-q})$ per q non-adaptive queries (see [16]). Michael discussed a general framework for testing graph properties, where distances are normalized by the actual number of edges and various types of queries are considered. In the work, "group queries" (akin of "group testing") are shown to be strictly stronger than the combination of the standard vertex-pair and neighbor queries.

Session 8 (Sunday, January 10, 10:45-12:00) [chair: Michael Krivelevich]

- Shubhangi Saraf: Some Recent Results on Testing of Sparse Linear Codes.
- Swastik Kopparty: Optimal Testing of Reed-Muller Codes.

Shubhangi advocated viewing locally testable linear codes as a special case of *tolerant* testing the Hadamard codewords under some adequate distributions (specifically, the distribution that is uniform on the corresponding positions of the Hadamard code). The tolerance condition prevents rejecting a codeword based on positions that are assigned zero probability. She presented two results, relating to two notion of tolerant testing (or rather distance approximation (cf. [12]): the more general result yields a weak notion of approximation and is obtained under the so-called *uniformly correlated condition*, whereas a strong notion of approximation requires a stronger correlated condition. A distribution μ is called k-uniformly correlated if there is a joint distribution of k-tuples such that each element (in the k-tuple) is distributed as μ but their sum is uniformly distributed. The stronger condition requires that this holds when these k elements are independently drawn from μ, which is equivalent to requiring that the code be sparse and have small Fourier coefficients. Swastik presented an improved analysis of the AKKLR linearity test (for degree d multi-linear polynomials over GF(2)) [2], showing that the basic test rejects far away function with constant probability (rather than with probability $\Omega(2^{-d})$).

Session 9 (Sunday, January 10, 14:00-15:30) [chair: Madhu Sudan]

- Eli Ben-Sasson: Limiting the Rate of Locally Testable Codes.
- Tali Kaufman: Symmetric LDPC Codes and Local Testing.
- Artur Czumaj: Testing Monotone Continuous Distributions on High-Dimensional Real Cubes.
- Krzysztof Onak: The Query Complexity of Edit Distance.
- Sofya Raskhodnikova: Testing and Reconstruction of Lipschitz Functions with Applications to Privacy.
- Oded Goldreich: Algorithmic Aspects of Property Testing in the Dense Graphs Model.

Eli surveyed several results that assert that various parameters of LTCs imply an exponentially vanishing rate. One such result refers to LTCs that are tested by a small set of constraints (which is only somewhat larger than the dimension of the dual code). Tali advocated the study of codes that are each characterized by a short constraint and its orbit under a suitable group. Arthur advocated the study of testing continuous distributions, focusing on the case that they are actually discrete (i.e., assume a finite number of possible values). Krzysztof contrasted the query complexity of testing the edit distance of a given string (from a fixed string) to the corresponding problem for Ulam distance. Sofya presented an application of reconstruction procedures (as presented by Michael Saks in Session 2) to data privacy; specifically, if data privacy is maintained for functions of a certain type, then we may want to modify the given function to that class. I tried to call attention to the work "Algorithmic Aspects of Property Testing in the Dense Graphs Model" (co-authored by Dana), while highlighting several perspectives and questions that arise.

Panel 3 (Sunday, January 10, 16:00-17:00)

- Prahladh Harsha on decoding in the low-error regime
- Ron Rivest on possible applications of property testing to the security evaluation of hashing functions.

Prahladh mused on whether decoding in the low-error regime may find additional applications in property testing. Ron asked whether property testing techniques can be employed to the evaluation of the quality of various cryptographic compression functions.

The Contents of this Volume

In addition to extended abstracts of almost all works presented at the workshop, this volume contains a few surveys.

Oded's Note: *The following list is highly tentative.*

- Artur Czumaj and Christian Sohler: Sublinear Time Approximation Algorithms.
- Oded Goldreich: Locally Testing Codes and Proofs.
- Oded Goldreich: Testing Graph Properties.
- Ronitt Rubinfeld and Asaf Shapira: TBA.

References

[1] N. Alon, E. Fischer, I. Newman, and A. Shapira. A Combinatorial Characterization of the Testable Graph Properties: It's All About Regularity. In *38th STOC*, pages 251–260, 2006.

[2] N. Alon, M. Krivelevich, T. Kaufman, S. Litsyn, and D. Ron. Testing Reed-Muller codes. *IEEE Transactions on Information Theory*, Vol. 51 (11), pages 4032–4038, 2005. An extended abstract appeared in the proceedings of *RANDOM'03* (under the title "Testing Low-Degree Polynomials over GF(2)").

[3] T. Batu, L. Fortnow, R. Rubinfeld, W.D. Smith and P. White. Testing that Distributions are Close. In *41st FOCS*, pages 259–269, 2000.

[4] I. Benjamini, O. Schramm, and A. Shapira. Every Minor-Closed Property of Sparse Graphs is Testable. In *40th STOC*, pages 393–402, 2008.

[5] M. Blum, M. Luby and R. Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *JCSS*, Vol. 47, No. 3, pages 549–595, 1993. Extended abstract in *22nd STOC*, 1990.

[6] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, pages 653–750, July 1998. Extended abstract in *37th FOCS*, 1996.

[7] O. Goldreich and D. Ron. Property Testing in Bounded Degree Graphs. *Algorithmica*, Vol. 32 (2), pages 302–343, 2002. Extended abstract in *29th STOC*, 1997.

[8] O. Goldreich and D. Ron. A Sublinear Bipartitness Tester for Bounded Degree Graphs. *Combinatorica*, Vol. 19 (3), pages 335–373, 1999. Extended abstract in *30th STOC*, 1998.

[9] T. Kaufman, M. Krivelevich, and D. Ron. Tight Bounds for Testing Bipartiteness in General Graphs. In *Proc. of RANDOM'03*, pages 341–353, 2003.

[10] S. Marko and D. Ron: Approximating the Distance to Properties in Bounded-Degree and General Sparse Graphs. *ACM Transactions on Algorithms*, Vol. 5 (2), 2009. Extended abstarct in the proceedings of *Random'06*.

[11] M. Parnas and D. Ron: Approximating the Minimum Vertex Cover in Sublinear Time and a Connection to Distributed Algorithms. *TCS*, Vol. 381 (1-3), pages 183–196, 2007. Preliminary version in ECCC, 2005.

[12] M. Parnas, D. Ron, and R. Rubinfeld: Tolerant Property Testing and Distance Approximation. *JCSS*, Vol. 72 (6), pages 1012–1042, 2006. Preliminary version in ECCC, 2004.

[13] D. Ron. Property Testing: A Learning Theory Perspective. *Foundations and Trends in Machine Learning*, Vol. 1 (3), pages 307–402, 2008.

[14] D. Ron. Algorithmic and Analysis Techniques in Property Testing. *Foundations and Trends in TCS*, to appear.

[15] R. Rubinfeld and M. Sudan. Robust Characterization of Polynomials with Applications to Program Testing. *SIAM Journal on Computing*, Vol. 25 (2), pages 252–271, 1996.

[16] S. Tamaki and Y. Yoshida: A Query Efficient Non-Adaptive Long Code Test with Perfect Completeness. ECCC, TR09-074, 2009.