On the strength of comparisons in property testing

Eldar Fischer *

Abstract

An ϵ-test for a property P of functions from $\mathcal{D} = \{1, \ldots, d\}$ to the positive integers is a randomized algorithm, which makes queries on the value of an input function at specified locations, and distinguishes with high probability between the case of the function satisfying P, and the case that it has to be modified in more than ϵd places to make it satisfy P.

We prove that an ϵ-test for any property defined in terms of the order relation, such as the property of being a monotone nondecreasing sequence, cannot perform less queries (in the worst case) than the best ϵ-test which uses only comparisons between the queried values. In particular, we show that an adaptive algorithm for testing that a sequence is monotone nondecreasing performs no better than the best non-adaptive one, with respect to query complexity. From this follows a tight lower bound on tests for this property.

1 Introduction

In the following we consider inputs given as finite sequences of positive integers. Given a property P of the possible inputs, we say that a sequence of length d is *ϵ-far from satisfying* P if it cannot be modified in ϵd or less places to make it satisfy P. An *ϵ-test* for P is a randomized algorithm which makes queries about an input sequence of length d, each query consisting of finding the value of the sequence at a specified location, and distinguishes with probability at least $\frac{2}{3}$ between the case that the given input satisfies P and the case that it is ϵ-far from satisfying P. We make no assumption on the computational complexity of deciding which queries the algorithm makes.

The object of investigation here is the minimum number of queries that an ϵ-test needs to make; for example, it is proven in [7] (see also [3] for related applications) that the property of being a

*NEC Research Institute, 4 Independence Way, Princeton NJ 08540, USA; and DIMACS. Email address: fischer@research.nj.nec.com

monotone nondecreasing sequence requires no more than $O(\log d)$ queries for any fixed ϵ; it was also proven there that this bound is tight for non-adaptive testing algorithms based on comparisons.

Several notions of testing were investigated in various works. The general notion of property testing was first formulated by Rubinfeld and Sudan [11], who were motivated mainly by its connection to the study of program checking. In [8] the notion of testability in the context of graphs was introduced, and investigated further in [1]. In [2] the notion of testability in the context of regular languages was investigated. In [7], [3] and [6] properties defined in terms of monotonicity and order relations were investigated.

In the following we concern ourselves with properties of sequences of positive integers, defined only in terms of the order relations between the values of their members. More formally, we say that a property P of such sequences is *order based* if for any two sequences $u_1, \ldots, u_d$ and $v_1, \ldots, v_d$ such that $u_i \leq u_j$ if and only if $v_i \leq v_j$ for every $1 \leq i, j \leq d$, either both satisfy P or both do not; we say that a property P is *strongly order based* if in addition, two sequences as above are either both ϵ-far from satisfying P or both are not (for every ϵ). In particular, properties defined as the sequence satisfying certain weak inequalities between its members are strongly order based.

The main result proven in the following is that for any strongly order based property, such as the property of being a monotone nondecreasing sequence, and for any ϵ, there exists an ϵ-test with an optimal number of queries which makes its queries based only on the locations of the previous queries and the order relations between the values of the input in these locations, and makes no other use of the values. In particular, for the property of being a monotone nondecreasing sequence, it is shown that there exists an optimal ϵ-test which is non-adaptive; together with the results from [7] this gives the tight bound of such a test requiring $\Theta(\log d)$ queries for any fixed small enough ϵ.

The proof combines a method from [5] with additional arguments. Section 2 presents the general tools used in the following, including Ramsey's Theorem, and a characterization of testing algorithms by functions, which we develop here for the purpose of formalizing and proving correctness of certain manipulations of the algorithms. Section 3 presents the proof of the main result and of the aforementioned tight bound on monotonicity testing. Section 4 is about the possible difference between adaptive and non-adaptive testing of order based properties, and the final Section 5 contains some additional concluding comments.

2 Preliminaries

Since we are only interested in the number of queries a testing algorithm makes, we can make a separate analysis for every possible input size. We also assume without loss of generality that the algorithm always makes the same number of queries for all inputs of the same size, and then makes the decision whether to accept or reject based on these. In this spirit we formulate the following definition.

A $(\mathcal{D}, t)$-*tester* is a randomized algorithm whose input is given as a function from a fixed domain $\mathcal{D}$ to the set of positive integers, which makes t queries; a query of the algorithm consists of finding out the value of this function on a specified member of $\mathcal{D}$. For simplicity, we assume that the domain $\mathcal{D}$ is the set $\{1, \ldots, d\}$ and denote the values of the input function by $v_1, \ldots, v_d$ respectively.

Algorithms by functions

A formal way to characterize a given $(\mathcal{D}, t)$-tester $\mathcal{A}$, is by defining for every $1 \leq i \leq d$ and $1 \leq k \leq t$ the function $p_i^{(k)}(i_1, w_1, \ldots, i_{k-1}, w_{k-1})$ as the probability that, given that the first $k-1$ queries of the algorithm were $i_1, \ldots, i_{k-1}$ and that the corresponding answers for these queries were $w_1 = v_{i_1}, \ldots, w_{k-1} = v_{i_{k-1}}$, the k'th query of the algorithm will be $i_k = i$ (in particular $p_i^{(0)}$ is just the probability that the algorithm makes i its first query); the probability that the algorithm accepts after the above queries are made is denoted by $p_a(i_1, w_1, \ldots, i_t, w_t)$. This family of functions, which we call in the following the *p-functions corresponding to* $\mathcal{A}$, characterizes it but is not always unique, as it might be the case that for a certain input and a certain sequence $i_1, \ldots, i_{k-1}$ the algorithm never makes this sequence of queries. It is also worth noting that for every family of non-negative functions which satisfy in the above notation

$$\sum_{i=1}^{d} p_i^{(k)}(i_1, w_1, \ldots, i_{k-1}, w_{k-1}) = 1$$

and

$$p_a(i_1, w_1, \ldots, i_t, w_t) \leq 1$$

for every $1 \leq k \leq t$, $1 \leq i_j \leq d$ and w_j, there exists a corresponding $(\mathcal{D}, t)$-tester.

The functions defined by

$$q^{(0)}(i_1) = p_{i_1}^{(0)},$$

$$q^{(k)}(i_1, w_1, \ldots, i_{k-1}, w_{k-1}, i_k) = p_{i_k}^{(k)}(i_1, w_1, \ldots, i_{k-1}, w_{k-1}) q^{(k-1)}(i_1, w_1, \ldots, i_{k-2}, w_{k-2}, i_{k-1})$$

and

$$q_a(i_1, w_1, \ldots, i_t, w_t) = q^{(t)}(i_1, w_1, \ldots, i_{t-1}, w_{t-1}, i_t) p_a(i_1, w_1, \ldots, i_t, w_t),$$

which we call the *q-functions corresponding to* $\mathcal{A}$, are unique to $\mathcal{A}$ and characterize it. Moreover, for any family $\mathcal{Q}$ of non-negative functions satisfying in the above notation

$$\sum_{i=1}^{d} q^{(0)}(i) = 1,$$

$$\sum_{i=1}^{d} q^{(k)}(i_1, w_1, \ldots, i_{k-1}, w_{k-1}, i) = q^{(k-1)}(i_1, w_1, \ldots, i_{k-2}, w_{k-2}, i_{k-1})$$

and

$$q_a(i_1, w_1, \ldots, i_t, w_t) \leq q^{(t)}(i_1, w_1, \ldots, i_{t-1}, w_{t-1}, i_t)$$

for all k, i_j and w_j, there exists a corresponding $(\mathcal{D}, t)$-tester.

We say that a sequence $\{\mathcal{A}_j | 1 \leq j\}$ of $(\mathcal{D}, t)$-testers *(pointwise) converges to* $\mathcal{A}$ if the sequence of their corresponding q-function families $\mathcal{Q}_j$ pointwise converges to the family $\mathcal{Q}$ of the q-functions of $\mathcal{A}$. This is equivalent to saying that $\mathcal{A}_j$ converge to $\mathcal{A}$ if for any fixed input $v_1, \ldots, v_d$ and any fixed query sequence $i_1, \ldots, i_t$, the probability that $\mathcal{A}_j$ performs this query sequence and accepts, as well as the probability that $\mathcal{A}_j$ performs it and rejects, converge to the corresponding probabilities of $\mathcal{A}$.

We say that a function is *order based* in some of its variables if it depends only on the order relations between these variables and the values of the other variables. For example, the function $\text{pos}(u_1, \ldots, u_k, i)$ that gives the index of the i'th largest value among $u_1, \ldots, u_k$ (the first if there are several such u_i) is order based in $u_1, \ldots, u_k$.

We say that a $(\mathcal{D}, t)$-tester $\mathcal{A}$ is *order based* if all its q-functions are order based in their input variables (w_j in the above notation). A proposition stating that a $(\mathcal{D}, t)$-tester for monotonicity which is order based can also be considered to be non-adaptive (i.e. one all of whose q-functions but q_a do not depend on the input variables at all) is proven in Section 3.

The following lemma, based on compactness (remember that all variables of the q-functions of a $(\mathcal{D}, t)$-tester but the input variables are restricted to $\{1, \ldots, d\}$), is immediate.

Lemma 2.1 *Every sequence of order based* $(\mathcal{D}, t)$*-testers has a converging subsequence.* □

Ramsey's theorem

As with the results of [5] regarding the strength of comparison based algorithms in other contexts, Ramsey's Theorem plays a major role in the proofs here.

Lemma 2.2 (Ramsey's Theorem, see e.g. [10] or [4]) *If $\mathcal{F}$ is any finite family of functions from the subsets of size k of the positive integers to a finite range, then there exists an infinite subset E of the positive integers, such that the restriction of the members of $\mathcal{F}$ to the subsets of size k of the members of E are all constant functions.*

In order to deal with functions with k variables in general, the following simple and well known corollary is used.

Corollary 2.3 (See e.g. [10]) *If $\mathcal{F}$ is a finite family of functions with k variables from the positive integers to a finite range, then there exists an infinite subset E of the integers such that the restriction of the members of $\mathcal{F}$ to E are all order based in their variables.*

3 Proof of the main results

We say that two $(\mathcal{D}, t)$-testers $\mathcal{A}$ and $\mathcal{B}$ are δ *similarly behaved* if for every possible input over the domain $\mathcal{D}$, the probability that $\mathcal{A}$ accepts it and the probability that $\mathcal{B}$ accepts it differ by no more than δ.

We say that $\mathcal{B}$ *behaves δ as well as $\mathcal{A}$ with respect to a property P* (of the possible inputs over $\mathcal{D}$), if for every input satisfying P the probability that $\mathcal{B}$ accepts is no more than δ less from the infimum probability that $\mathcal{A}$ accepts any input satisfying P, and for any fixed ϵ and every input which is ϵ-far from satisfying P, the probability that $\mathcal{B}$ rejects is no more than δ less from the infimum probability that $\mathcal{A}$ rejects any input which is ϵ-far from satisfying P. In particular, if $\mathcal{B}$ is δ similarly behaved as $\mathcal{A}$ then it also behaves δ as well as $\mathcal{A}$ with respect to any property. We say that $\mathcal{B}$ *behaves as well as $\mathcal{A}$ with respect to P* if it behaves δ as well as $\mathcal{A}$ with respect to P for $\delta = 0$.

The strength of comparisons

For a $(\mathcal{D}, t)$-tester $\mathcal{A}$ and a monotone increasing function over the positive integers f, we define the $(\mathcal{D}, t)$-tester $\mathcal{A}_f$ as simulating $\mathcal{A}$ over $f(v_1), \ldots, f(v_d)$ (where $v_1, \ldots, v_d$ is the original input).

In other words, the family $\mathcal{Q}'$ of the q-functions of $\mathcal{A}_f$ is defined in terms of the family $\mathcal{Q}$ of the q-functions of $\mathcal{A}$ as follows.

$$q'^{(k)}(i_1, w_1, \ldots, i_{k-1}, w_{k-1}, i_k) = q^{(k)}(i_1, f(w_1), \ldots, i_{k-1}, f(w_{k-1}), i_k),$$

$$q'_a(i_1, w_1, \ldots, i_t, w_t) = q_a(i_1, f(w_1), \ldots, i_t, f(w_t)).$$

$\mathcal{A}_f$ clearly behaves as well as $\mathcal{A}$ for any strongly order based property P. This observation when used in conjunction with Ramsey's Theorem brings us to the following key lemma.

Lemma 3.1 *For every $(\mathcal{D}, t)$-tester $\mathcal{A}$, every strongly order based property P and every positive integer r, there exists an order based $(\mathcal{D}, t)$-tester $\mathcal{B}_r$ which behaves $\frac{1}{r}$ as well as $\mathcal{A}$ with respect to P.*

Proof: In this case it is useful to look at a possible family $\mathcal{P}$ of p-functions associated with the tester $\mathcal{A}$. We then let $\mathcal{P}'$ denote a family of functions such that their range is the (finite) set $\{0, \frac{1}{2rt}, \frac{2}{2rt}, \ldots, \frac{2rt-1}{2rt}, 1\}$, satisfying also

$$|p_i'^{(k)}(i_1, w_1, \ldots, i_{k-1}, w_{k-1}) - p_i^{(k)}(i_1, w_1, \ldots, i_{k-1}, w_{k-1})| \leq \frac{1}{2rt},$$

$$\sum_{i=1}^{d} p_i'^{(k)}(i_1, w_1, \ldots, i_{k-1}, w_{k-1}) = 1,$$

and

$$|p'_a(i_1, w_1, \ldots, i_t, w_t) - p_a(i_1, w_1, \ldots, i_t, w_t)| \leq \frac{1}{2r}.$$

It is not hard to see that such a family exists. Letting $\mathcal{A}'$ denote the $(\mathcal{D}, t)$-tester corresponding to $\mathcal{P}'$, it is also not hard to see that $\mathcal{A}'$ is $\frac{1}{r}$ similarly behaved as $\mathcal{A}$.

Using Corollary 2.3, we now find a subset E of the integers such that the restriction of all members of $\mathcal{P}'$ to E are order based (since all members of $\mathcal{P}'$ now have a finite range, and all their variables apart from the input variables are restricted to $\{1, \ldots, d\}$, we can reduce $\mathcal{P}'$ to a family of functions as in the formulation of the corollary). We define the monotone function f so that $f(i)$ is the i'th smallest member of E for every i. The $(\mathcal{D}, t)$-tester $\mathcal{B}_r = \mathcal{A}'_f$ is clearly order based. It also behaves as well as $\mathcal{A}'$, and thus behaves $\frac{1}{r}$ as well as $\mathcal{A}$. □

The following is the main result. It implies that for any strongly order based property P and any ϵ, there exists an optimal ϵ-test (which distinguishes with probability at least $\frac{2}{3}$ between the case that the input satisfies P and the case that it is ϵ-far from satisfying P), which is order based.

Theorem 3.2 *For every $(\mathcal{D}, t)$-tester $\mathcal{A}$ and strongly order based property P there exists an order based $(\mathcal{D}, t)$-tester $\mathcal{B}$ which behaves as well as $\mathcal{A}$ with respect to P.*

Proof: For every r, let $\mathcal{B}_r$ be the order based $(\mathcal{D}, t)$-tester which behaves $\frac{1}{r}$ as well as $\mathcal{A}$ respect to P, which is provided by Lemma 3.1.

By the previous discussion about order based testers, there exists a subsequence $\{\mathcal{B}_{r_t} | 1 \leq t\}$ of $\{\mathcal{B}_r | 1 \leq r\}$ which converges; let us denote the limit $(\mathcal{D}, t)$-tester by $\mathcal{B}$. We now prove that $\mathcal{B}$ behaves δ as well as $\mathcal{A}$ for any $\delta > 0$; this implies that it is the required algorithm.

Given δ and an input $v_1, \ldots, v_d$ which satisfies P, let t be an integer satisfying $r_t > 2\delta^{-1}$ and also that the probability that $\mathcal{B}$ accepts $v_1, \ldots, v_d$ differs by no more than $\frac{1}{2}\delta$ from the probability that $\mathcal{B}_{r_t}$ accepts it (such a t exists by the pointwise convergence to $\mathcal{B}$). Since $\mathcal{B}_{r_t}$ also behaves $\frac{1}{2}\delta$ as well as $\mathcal{A}$, this implies in particular that the probability that $\mathcal{B}$ accepts this input is no more than δ less from the infimum probability that $\mathcal{A}$ accepts any input in P. The case for inputs which are ϵ-far from satisfying P is proven in an analogue fashion. □

Testing for monotonicity

The following simple proposition states that an order based algorithm for testing whether a sequence is monotone nondecreasing can be considered to be a non-adaptive one.

Proposition 3.3 *For every order based $(\mathcal{D}, t)$-tester $\mathcal{A}$ there exists a non-adaptive order based $(\mathcal{D}, t)$-tester $\mathcal{B}$ which behaves as well as $\mathcal{A}$ with respect to the property of the input being a monotone nondecreasing sequence.*

Proof: We may safely assume that $\mathcal{A}$ always rejects the input if in the end all queried values do not represent a monotone nondecreasing subsequence of the input.

We let $\mathcal{A}'$ denote the $(\mathcal{D}, t)$-tester which is obtained by simulating $\mathcal{A}$ over the input sequence $dv_1 + 1, \ldots, dv_d + d$, where $v_1, \ldots, v_d$ represent the input given to $\mathcal{A}'$. In other words, the family $\mathcal{Q}'$ of q-functions of $\mathcal{A}'$ is defined in terms of the family $\mathcal{Q}$ of q-functions of $\mathcal{A}$ as follows.

$$q'^{(k)}(i_1, w_1, \ldots, i_{k-1}, w_{k-1}, i_k) = q^{(k)}(i_1, dw_1 + i_1, \ldots, i_{k-1}, dw_{k-1} + i_{k-1}, i_k),$$

$$q'_a(i_1, w_1, \ldots, i_t, w_t) = q_a(i_1, dw_1 + i_1, \ldots, i_t, dw_t + i_t).$$

It is clear that $dv_1 + 1, \ldots, dv_d + d$ is a (strictly) monotone increasing sequence if and only if $v_1, \ldots, v_d$ is a monotone nondecreasing sequence, and that $dv_1 + 1, \ldots, dv_d + d$ is ϵ-far from being monotone nondecreasing if and only if so is $v_1, \ldots, v_d$. It is also clear that if for some k the values of the input at $i_1, \ldots, i_k$ do not form a monotone increasing subsequence, the tester will reject regardless of what queries are made after the k'th query.

Thus we can define $\mathcal{B}$ as the tester which simulates the queries of $\mathcal{A}'$ (or $\mathcal{A}$ for that matter) over, say, the input $1, \ldots, d$ (we use here the assumption that $\mathcal{A}$ is order based), then checks whether the values of the actual input at $i_1, \ldots, i_t$ indeed form a monotone subsequence, and accepts (with the probability that $\mathcal{A}$ accepts in this case) or rejects accordingly. $\Box$

When used together with Theorem 3.2 and the results from [7], the following tight bound on the query complexity of testing a sequence for monotonicity is achieved.

Corollary 3.4 *For every fixed small enough ϵ, ϵ-testing the property of $v_1, \ldots, v_d$ being a monotone nondecreasing sequence requires $\Theta(\log d)$ queries.*

Proof: From Theorem 3.2 and Proposition 3.3 it follows that no ϵ-test for this property can perform better with regards to the query complexity than the best order based non-adaptive one. In [7] it was shown that for a fixed small enough ϵ there exists no such ϵ-test for this property which makes $o(\log d)$ queries, and hence the lower bound.

A particular ϵ-test (for any fixed ϵ) which makes $O(\log d)$ queries on the input sequence was provided in [7], so this bound is tight. $\Box$

4 On the strength of adaptivity in property testing

A gap between adaptive and non-adaptive testing

There exist order based properties where, unlike monotonicity, there is a large gap between the query complexity of the best adaptive test and that of the best non-adaptive one. To show this, note first that properties of sequences of *bits* (with a corresponding definition of testing) which have such a gap can be converted into strongly order based properties of sequences of positive integers with a similar gap by replacing each bit with two positive integers, considering this bit to be "1" if and only if these two integers are equal.

A property of sequences of bits with an exponential gap between its adaptive and non-adaptive tests can be constructed from the context free language proven in [2] not to be testable with a constant number of queries (see also [9] for another model with a large gap between adaptive and non-adaptive testing of some properties). Here we sketch another such property. We consider the input to encode adjacency matrices of two graphs with an identical number of vertices, a function from the vertices of the first graph to the vertices of the second one, and a function from the vertices of the second to the vertices of the first. The property is defined as that of the two functions being inverses of each other as well as isomorphisms between the two graphs.

Denoting the size of the input by m, for every ϵ there exists an adaptive ϵ-test which makes $O(\log m)$ queries – it picks randomly, uniformly and independently a constant number of vertices of the first graph, queries the values of the first function at these locations, the values of the second function at the target locations specified by the first function to test that the functions are inverses (see [7] or [3] where it is explained in a different context why such a procedure tests that the functions are close to being inverses), and finally checks by querying on the graphs that the functions satisfy the isomorphism conditions in these locations. The querying of the values of the functions takes $O(\log m)$ bit queries per location, and the querying on the graphs adds a constant number to these.

To show that non-adaptive testing for this property is hard, we use the explanation in [1] (some of it is in the concluding comments) that for some ϵ it is hard to ϵ-test with $o(\sqrt{n})$ queries for the property of two graphs with n vertices having *any* isomorphism between them, and give an $\Omega(m^{1/4})$ lower bound on the query complexity here.

We construct the following two inputs. The first input consists of a random graph (with each edge taken independently with probability $\frac{1}{2}$), a second graph constructed by randomly permuting the vertices of the first, and the two functions corresponding to this isomorphism between them. The second input is constructed from the first one by replacing the second graph with another, independently random, graph. The first input clearly satisfies the property, while with high probability the second input is far from satisfying the property and at the same time cannot be distinguished from the first one by any non-adaptive algorithm which makes $o(m^{1/4})$ queries.

We should also note that with a slight modification of the above construction one can construct (non-strongly) order based properties (of sequences of integers) with a factorial gap between the best adaptive and the best non-adaptive tests, and properties of integer sequences which are not order based and for which there is an arbitrarily large gap between adaptive and non-adaptive testing.

Non-adaptive testing of strongly order based properties

Theorem 3.2 implies (by going over the decision tree of the order based algorithm) that for strongly order based properties there exists no more than a factorial gap between their adaptive and non-adaptive testing.

Corollary 4.1 *For every $(\mathcal{D}, t)$-tester $\mathcal{A}$ and every strongly order based property P there exists a non-adaptive $(\mathcal{D}, (2t)!)$-tester $\mathcal{B}$ which behaves as well as $\mathcal{A}$ with respect to P.*

Proof sketch: Using Lemma 3.2 we construct an order based $(\mathcal{D}, t)$-tester $\mathcal{A}'$ which behaves as well as $\mathcal{A}$. To construct $\mathcal{B}$, for every k instead of making one query using the value of the p-functions of $\mathcal{A}'$ for the k'th query of $\mathcal{A}'$, we make (less than $(2t-1)!$) queries according to the values of these p-functions for each possible set of order relations between $w_1, \ldots, w_{k-1}$. Note that this procedure is independent of the actual input.

After all queries have been made, for every k choose i_k to be the query made according to the values of the p-functions for the order relations between the values of the input in locations $i_1, \ldots, i_{k-1}$. To decide whether to accept or reject, use the value of the acceptance function of $\mathcal{A}'$ for $i_1, \ldots, i_k$ and the values of the input in these locations. □

5 Concluding comments

Non-strongly order based properties

When considering properties of integers, Theorem 3.2 does not always hold for non-strongly order based properties. For example, the property of being a (strictly) monotone **increasing** sequence still requires $\Theta(\log d)$ queries to test, but an order based test would need much more, as shown by considering inputs of the form $1, 2, \ldots, i-1, i, i, i+1, \ldots, d-2, d-1$ (these look almost increasing, but they are far from being so because there are not enough integers between 1 and $d-1$). Results similar to Theorem 3.2 can be extended however to many non-strongly order based properties.

The above unwelcome peculiarity does not hold for properties of sequences of rational numbers (or other dense number sets). For them a counterpart of Theorem 3.2 can be proven, using the above methods together with the fact that an order based test that performs well for the integer inputs will perform well for other inputs too. Moreover, since there is no difference between being order based and being strongly order based in this context, the related results regarding non-adaptive testing become tighter.

Preserving computational complexity

The results above hold for a model which concerns itself only with the number of queries required for ϵ-testing a property for every fixed domain, disregarding the computational complexity of calculating the probabilities for each query when the domain size is given as an input (though as is the case with Corollary 3.4, this in many cases is sufficient for providing also a tight lower bound on the running time). It would be interesting to develop a method for obtaining results which preserve also some of the computational complexity of the original (non order-based) algorithms.

Algorithmical calculus

It would be interesting to find non-obvious applications of "algorithmical calculus" using notions of convergence similar to the one defined above for testers. For example, if $\mathcal{M}$ is a (not necessarily discrete) probability space, and for every $M \in \mathcal{M}$ there exists a specified $(\mathcal{D}, t)$-tester $\mathcal{A}_M$, then one could sometimes "integrate" this family of algorithms over $\mathcal{M}$, to give a rigorous definition of the $(\mathcal{D}, t)$-tester whose informal definition is "Pick randomly a member of $\mathcal{M}$ and perform $\mathcal{A}_M$". Actually, some preliminary versions of the results here were proven using such a technique.

References

[1] N. Alon, E. Fischer, M. Krivelevich and M. Szegedy, Efficient testing of large graphs, *Proceedings of the* 40^{th} *IEEE FOCS* (1999), 656–666.

[2] N. Alon, M. Krivelevich, I. Newman and M. Szegedy, Regular languages are testable with a constant number of queries, *Proceedings of the* 40^{th} *IEEE FOCS* (1999), 645–655.

[3] T. Batu, R. Rubinfeld and P. White, Fast approximation PCPs for multidimensional bin-packing problems, *Proceedings of the third international workshop on randomization and approximation techniques in computer science* (D. Hochbaum, K. Jansen, J. P. D. Rolim and A. Sinclair eds), pp. 245–256, Springer-Verlag, Berkeley CA (1999).

[4] B. Bollobás, *Extremal Graph Theory*, Academic Press, New York (1978).

[5] D. Breslauer, A. Czumaj, D. P. Dubhashi and F. Meyer auf der Heide, Transforming comparison model lower bounds to the parallel-random-access-machine, *Information Processing Letters* 62 (1997), 103–110.

[6] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron and A. Samorodnitsky, Improved testing algorithms for monotonicity, *Proceedings of the third international workshop on randomization and approximation techniques in computer science* (D. Hochbaum, K. Jansen, J. P. D. Rolim and A. Sinclair eds), pp. 97–108, Springer-Verlag, Berkeley CA (1999).

[7] F. Ergün, S. Kannan, S. R. Kumar, R. Rubinfeld and M. Viswanathan, Spot checkers, *Proceedings of the* 30^{th} *ACM STOC*, 1998. Also: *Journal of computer and system science*, to appear.

[8] O. Goldreich, S. Goldwasser and D. Ron, Property testing and its connection to learning and approximation, *Proceedings of the* 37^{th} *IEEE FOCS* (1996), 339–348. Also: Journal of the ACM, to appear.

[9] O. Goldreich and D. Ron, Property testing in bounded degree graphs, *Proceedings of the* 29^{th} *ACM STOC* (1997), 406–415.

[10] R. L. Graham, B. L. Rothschild and J. H. Spencer, *Ramsey Theory*, Joe Wiley and Sons, New York, (Second ed. 1990).

[11] R. Rubinfeld and M. Sudan, Robust characterization of polynomials with applications to program testing, *SIAM Journal of Computing* 25 (1996), 252–271.