Electronic Colloquium on Computational Complexity, Report No. 17 (1999)

Improved Testing Algorithms for Monotonicity

Yevgeniy Dodis* Oded Goldreich† Eric Lehman‡ Sofya Raskhodnikova§
Dana Ron¶ Alex Samorodnitsky‖

June 4, 1999

Abstract

We present improved algorithms for testing monotonicity of functions. Namely, given the ability to query an unknown function $f : \Sigma^n \mapsto \Xi$, where Σ and Ξ are finite ordered sets, the test always accepts a monotone f, and rejects f with high probability if it is ϵ-far from being monotone (i.e., every monotone function differs from f on more than an ϵ fraction of the domain). For any $\epsilon > 0$, the query complexity of the test is $O((n/\epsilon) \cdot \log|\Sigma| \cdot \log|\Xi|)$. The previous best known bound was $\tilde{O}((n^2/\epsilon) \cdot |\Sigma|^2 \cdot |\Xi|)$.

We also present an alternative test for the boolean range $\Xi = \{0, 1\}$ whose query complexity $O(n^2/\epsilon^2)$ is independent of alphabet size $|\Sigma|$.

*Lab for Computer Science, MIT, 545 Technology Sq. Cambridge, MA 02139. email: yevgen@theory.lcs.mit.edu.

†Dept. of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel. e-mail: oded@wisdom.weizmann.ac.il.

‡Lab for Computer Science, MIT, 545 Technology Sq. Cambridge, MA 02139. email: e_lehman@theory.lcs.mit.edu.

§Lab for Computer Science, MIT, 545 Technology Sq. Cambridge, MA 02139. email: sofya@theory.lcs.mit.edu.

¶Dept. of EE – Systems, Tel Aviv University, Ramat Aviv, Israel. e-mail: danar@eng.tau.ac.il.

‖DIMACS Center, Rutgers University, Piscataway, NJ 08854. email: salex@av.rutgers.edu.

ISSN 1433-8092

1 Introduction

Property Testing (cf., [13, 9]) is a general formulation of computational tasks in which one is to determine whether a given object has a predetermined property or is "far" from any object having the property. Thus, property testing captures a natural notion of approximation, where the measure approximated is the object's "distance" to having the property. Typically one aims at performing this task within complexity smaller than the size of the object, while employing a randomized algorithm and given oracle access to a natural encoding of the object (as a function). Thus, we are talking of determining with high probability whether a function, to which we have oracle access, belongs to some class or is "far" from this class (i.e., one needs to modify the function value at many places so to obtain a function in the class).

Much work in this area was devoted to testing algebraic properties of functions such as linearity (e.g., [5, 1, 4, 3]) and low-degree properties (e.g., [5, 7, 13, 12, 2]). Recently, some attention was given to testing combinatorial properties of functions; firstly, for functions representing graphs [9, 10, 11], and more recently for functions per se [6, 8]. The most natural combinatorial property of functions is monotonicity, and indeed [8] focuses on testing monotonicity. The basic problem studied there is the following. Given a distance parameter ϵ and oracle access to a function $f : \{0,1\}^n \mapsto \{0,1\}$, determine whether f is monotone or is "ϵ-far" from being monotone. Monotonicity is defined in the natural manner: One considers the standard partial order $\prec$ on binary strings (i.e., $x_1x_2\cdots x_n \prec y_1y_2\cdots y_n$ iff $x_i \leq y_i$ for every i and $x_i < y_i$ for some i), and f is said to be monotone if $f(x) \leq f(y)$ for every $x \prec y$. The definition extends naturally to functions defined on the standard partial order of strings over an arbitrary alphabet, Σ, and having an arbitrary range Ξ. That is,

Definition 1 (monotone functions and testing): *Let Σ and Ξ be sets with total order $\leq_\Sigma$ and $\leq_\Xi$, respectively. We consider the partial order, $\prec$, defined on equal-length strings over Σ by $x_1x_2\cdots x_n \prec y_1y_2\cdots y_n$ iff $x_i \leq_\Sigma y_i$ for every i and $x_i \neq y_i$ for some i.*

- *A function $f : \Sigma^n \mapsto \Xi$ is* monotone *if $f(x) \leq_\Xi f(y)$ holds for every $x \prec y$.*
- *A* relative distance of $f : \Sigma^n \mapsto \Xi$ from the class of monotone functions, *$\epsilon_{\mathrm{M}}(f)$, is the minimum over all monotone functions $g : \Sigma^n \mapsto \Xi$ of $\mathrm{dist}(f,g) \stackrel{\mathrm{def}}{=} |\{x \in \Sigma^n : f(x) \neq g(x)\}| \;/\; |\Sigma|^n$.*
- *A function $f : \Sigma^n \mapsto \Xi$ is* ϵ-far from monotone *if $\epsilon_{\mathrm{M}}(f) \geq \epsilon$.*
- *A probabilistic oracle machine M is said to be a* tester of monotonicity *if*

$$\mathrm{Prob}[M^f(\epsilon, n) = 1] \;\geq\; \frac{2}{3} \qquad \text{for any monotone function } f, \tag{1}$$

$$\mathrm{Prob}[M^f(\epsilon, n) = 0] \;\geq\; \frac{2}{3} \qquad \text{for } f \text{ which is } \epsilon\text{-far from monotone.} \tag{2}$$

Note that all notions are defined w.r.t. Σ and Ξ, and so at times we prefer terms which explicitly mention this dependence.

The main result of [8] is a tester of monotonicity for the case $\Sigma = \Xi = \{0,1\}$ having query and time complexities of the form $\mathrm{poly}(n)/\epsilon$. Specifically, the analysis of the query complexity in [8] yields a bound of $\tilde{O}(n^2/\epsilon)$, and it was also shown that $\Omega(n/\epsilon)$ is a lower bound on the query complexity of their algorithm. For general Σ and Ξ, the bounds obtained in [8] were proportional to $|\Sigma|^2 \cdot |\Xi|$. Here we improve both the algorithm and the analysis in [8] to obtain the following.

Theorem 1 (main result): *There exists a tester of monotonicity with query complexity*

$$q(\epsilon, n) \stackrel{\mathrm{def}}{=} O\left(\frac{n \cdot (\log|\Sigma|) \cdot (\log|\Xi|)}{\epsilon}\right).$$

The tester works by selecting independently $q(\epsilon, n)/2$ pairs of n-long strings over Σ, and comparing the two f-values obtained for the elements of each pair.[1]

Thus, the global feature of being monotone or far from it, is determined by a sequence of many independent random local checks. Each local check consists of selecting a pair, (x, y), so that (w.l.o.g) $x \prec y$, according to some fixed distribution and checking whether $f(x) \leq_\Xi f(y)$. If we ever find a pair for which this does not hold (i.e., local violation of monotonicity), then we reject. Otherwise we accept. Thus, we never reject a monotone function, and the challenge is to analyze the dependence of rejection probability on the distance of the given function from being monotone.

The only thing left unspecified in the above description of the testing algorithm is the distribution by which the pairs are selected. In case $\Sigma = \{0, 1\}$ there seems to be a very natural choice. Uniformly select $i \in [n] \stackrel{\mathrm{def}}{=} \{1, ..., n\}$, independently and uniformly select $z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n \in \{0, 1\}$, and set $x = z_1 \cdots z_{i-1} 0 z_{i+1} \cdots z_n$ and $y = z_1 \cdots z_{i-1} 1 z_{i+1} \cdots z_n$. Our improvement over [8], in this case (where $\Sigma = \Xi = \{0, 1\}$), comes from a better (and in fact tight for $\Xi = \{0, 1\}$) analysis of this test: Let $\delta_{\mathrm{M}}(f)$ denote the fraction of pairs (x, y) as above for which $f(x) > f(y)$. We show that $\delta_{\mathrm{M}}(f) \geq \epsilon_{\mathrm{M}}(f)/n$ improving on the bound $\delta_{\mathrm{M}}(f) \geq \epsilon_{\mathrm{M}}(f)/n^2 \log(1/\epsilon_{\mathrm{M}}(f))$ in [8] (whereas by [8] there exist functions f for which $\delta_{\mathrm{M}}(f) = 2\epsilon_{\mathrm{M}}(f)/n$).

In case of non-binary $\Sigma = \{1, \ldots, d\}$ there seem to be several natural possibilities: Even if we restrict ourselves (as above) to select only pairs of strings which differ on a single coordinate i, there is still the question of how to select the corresponding pair of symbols. We study several natural possibilities of randomly selecting pairs $(k, \ell) \in \Sigma \times \Sigma$.

1. Distribution p_0: Select uniformly a pair $(k, k+1)$ with $k \in \{1, \ldots, d-1\}$.
2. Distribution p_1: Select uniformly a pair (k, ℓ) from a particular subset of size $O(d \log d)$ of the set of all pairs.
3. Distribution p_2: Select uniformly a pair (k, ℓ) with $k < \ell$.

A key result of this work is the reduction of the analysis of testing algorithms as above for any n and Σ, and for $\Xi = \{0, 1\}$, to their behavior in the special case of $n = 1$ (where we simply select pairs (k, ℓ) according to one of the above distributions and check the order between $f(k)$ and $f(\ell)$). Using this reduction we derive the following theorem.

Theorem 2 (Monotonicity Testing of Boolean functions): *There exist efficiently samplable distributions on pairs $(x, y) \in \Sigma^n \times \Sigma^n$ with $x \prec y$ so that for every function $f : \Sigma^n \mapsto \{0, 1\}$ the following holds:*

1. *If (x, y) is drawn according to one distribution (derived from p_1) then*

$$\mathrm{Prob}[f(x) > f(y)] \; = \; \Omega\left(\frac{\epsilon_{\mathrm{M}}(f)}{n \cdot (\log|\Sigma|)}\right).$$

[1] Since the algorithm is comparison-based, its complexity depends only on the size of the image of the function. Thus, one may replace Ξ in the above bound by $\Xi_f = \{f(x) : x \in \Sigma^n\}$. In particular, $\log|\Xi_f| \leq n \cdot \log|\Sigma|$, so our bound is never worse than $O(n^2 \cdot (\log|\Sigma|)^2/\epsilon)$.

2. *If (x,y) is drawn according to another distribution (derived from p_2) then*

$$\mathrm{Prob}[f(x) > f(y)] \;=\; \Omega\left(\frac{\epsilon_{\mathrm{M}}(f)^2}{n^2}\right).$$

We note that the first item of the theorem can also be derived by applying our reduction and using an alternative distribution on pairs in Σ^2 which was previously suggested in [6], and analyzed for the case $n = 1$. The second item leads to an algorithm having query complexity $O(n^2/\epsilon^2)$. It is possible to obtain an algorithm having complexity $O((n/\epsilon)\log^2(n/\epsilon))$ if the algorithm is not required to select independent pairs of strings. The alternative algorithm, suggested by Noga Alon, works by picking $i \in \{1,\ldots,n\}$ uniformly, and then querying f on $O(1/\epsilon)$ strings that differ only on the i^{th} coordinate. The analysis of this algorithm can also be shown to reduce to the $n = 1$ case (using analogous claims to those presented here).

The reader may be tempted to say that since our algorithm is comparison-based, the analysis should also hold for non-boolean functions. However, this is false. For example, by Item (2) above, boolean functions over Σ may be tested for monotonicity within complexity independent of $|\Sigma|$. In contrast, a lower bound in [6] asserts that arbitrary functions over Σ (e.g., with $\Xi = \Sigma$) cannot be tested for monotonicity within complexity independent of $|\Sigma|$ (but rather require complexity $\Omega(\log|\Sigma|)$ for some fixed distance parameter $\epsilon > 0$). Thus, a natural question arises: *Under what conditions and at what cost can results regarding testing of monotonicity of boolean functions be transformed to results for testing monotonicity of arbitrary functions*? Our most general result is the following.

Theorem 3 (Monotonicity Testing – Range Reduction): *Consider the task of testing monotonicity of functions defined over any partially ordered set S (with p.o. $\prec_S$). Suppose that for some distribution on pairs $(x,y) \in S \times S$ with $x \prec_S y$ and for every function $f : S \mapsto \{0,1\}$*

$$\mathrm{Prob}[f(x) > f(y)] \;\geq\; \frac{\epsilon_{\mathrm{M}}(f)}{C}\;,$$

where C depends on S only. Then, for every Ξ and every function $f : S \mapsto \Xi$ for pairs selected according to the same distribution

$$\mathrm{Prob}[f(x) > f(y)] \;\geq\; \frac{\epsilon_{\mathrm{M}}(f)}{C \cdot \log_2|\Xi|}\;.$$

Theorem 1 follows by combining Part 1 of Theorem 2 and Theorem 3 with $C = O(n \cdot \log|\Sigma|)$.

Organization:

We start with some preliminaries in Section 2. In Section 3 we show how the analysis of our algorithm in the boolean-range case for arbitrary n and Σ, reduces to the case $n = 1$. The algorithms for the case $n = 1$ (each corresponding to a different distribution on pairs in $\Sigma \times \Sigma$), are provided in Subsection 3.3, and the proof of Theorem 2, in Subsection 3.4. Finally, in Section 4 we prove a general reduction from an arbitrary range to the boolean range, and derive Theorem 3.

2 Preliminaries

Let Σ and Ξ be sets with total order $\leq_\Sigma$ and $\leq_\Xi$, respectively. We consider the partial order, $\prec$, defined on equal-length strings over Σ as in the introduction, and shorthand $\leq_\Xi$ by $\leq$.

For any pair of functions $f, g : \Sigma^n \mapsto \Xi$, we define the *distance* between f and g, denoted $\mathrm{dist}(f, g)$, to be the fraction of instances $x \in \Sigma^n$ on which $f(x) \neq g(x)$. As in the introduction, we let $\epsilon_{\mathrm{M}}(f)$ denote the minimum distance between f and any monotone function $g : \Sigma^n \mapsto \Xi$. Let us formally define the algorithmic schema studied in this paper. The schema uses an arbitrary probability distribution $p : \Sigma \times \Sigma \mapsto [0, 1]$. Without loss of generality, we assume that the support of p is restricted to pairs (k, ℓ) with $k < \ell$. The function t referred to below, depends on p.

ALGORITHMIC SCHEMA: Given parameters ϵ, n, Σ, Ξ, and oracle access to an arbitrary function $f : \Sigma^n \mapsto \Xi$, repeat the following steps up to $t(\epsilon, n, |\Sigma|, |\Xi|)$ times:

1. Uniformly select dimension $i \in [n]$, prefix $\alpha \in \Sigma^{i-1}$, and suffix $\beta \in \Sigma^{n-i}$.
2. Select (k, ℓ) according to p. Let $x = \alpha\, k\, \beta$, $y = \alpha\, \ell\, \beta$.
3. If $f(x) > f(y)$ (i.e., (x, y) witnesses that f is not monotone), then reject.

If all iterations were completed without rejecting then accept.

We focus on the analysis of a single iteration of the above test. Such an iteration is fully specified by the distribution, denoted $D_p^n : \Sigma^n \times \Sigma^n \mapsto [0, 1]$, by which pairs (x, y) are selected. That is, $D_p^n(x, y) = \frac{p(k,\ell)}{n \cdot |\Sigma|^{n-1}}$ if $x = \alpha\, k\, \beta$ and $y = \alpha\, \ell\, \beta$, for some α, β, and $D_p^n(x, y) = 0$ otherwise. Observe that $D_p^n(x, y) > 0$ only if $x \prec y$. Let $\mathrm{DETECT}(f, D_p^n)$ be the probability that a pair (x, y) selected according to D_p^n witnesses that f is not monotone; that is,

$$\mathrm{DETECT}(f, D_p^n) \stackrel{\mathrm{def}}{=} \mathrm{Prob}_{(x,y)\sim D_p^n}[f(x) > f(y)] \tag{3}$$

(where the above definition can of course be applied to any distribution D on pairs $x \prec y$). Our goal is to find distributions D_p^n (determined by the distributions p) for which $\mathrm{DETECT}(f, D_p^n)$ is "well" lower-bounded as a function of $\epsilon_{\mathrm{M}}(f)$. If D_p^n is such that $\mathrm{DETECT}(f, D_p^n) \geq \delta(\epsilon, n, |\Sigma|, |\Xi|)$ *for any* $f : \Sigma^n \mapsto \Xi$ with $\epsilon_{\mathrm{M}}(f) \geq \epsilon$, then setting $t(\epsilon, n, |\Sigma|, |\Xi|) = \Theta(1/\delta(\epsilon, n, |\Sigma|, |\Xi|))$ yields a tester for monotonicity.

THE PARTIAL ORDER GRAPH: It will be convenient to view the partial order over Σ^n as a directed (acyclic) graph, denoted G_Σ^n. The vertices of G_Σ^n are the strings in Σ^n and directed edges correspond to comparable pairs (i.e. (x, y) is an edge iff $x \prec y$). An edge (x, y) is said to be violated by f if $f(x) > f(y)$. We denote by $\mathrm{VIOL}(f)$ the set of violated edges of f. We remark that most of the definitions in this section naturally extend to any partially ordered set S in place of Σ^n.

3 Dimension Reduction for Boolean Functions

In this section we restrict our attention to boolean functions $f : \Sigma^n \mapsto \{0, 1\}$. Without loss of generality assume $\Sigma = \{1 \ldots d\}$, so $|\Sigma| = d$. In what follows we reduce the analysis of the performance of our algorithmic schema for any n and Σ (and $\Xi = \{0, 1\}$) to its performance for the case $n = 1$ (the "line"). In Subsection 3.3 we describe and analyze several algorithms for the line. Recall that by our algorithmic schema any such algorithm is determined by a probability distribution p on pairs $(k, \ell) \in \Sigma \times \Sigma$. We conclude this section by combining the reduction with these algorithms to derive Theorem 2.

3.1 A Sorting Operator

We begin with a few definitions. For any $i \in [n]$, we say that a function f is *monotone in dimension* i, if for every $\alpha \in \Sigma^{i-1}$, $\beta \in \Sigma^{n-i}$, and $k, \ell \in \Sigma$ such that $k < \ell$, $f(\alpha\, k\, \beta) \leq f(\alpha\, \ell\, \beta)$. For a set of indices $T \subseteq [n]$, we say that f is *monotone in dimensions* T, if for every $i \in T$, the function f is monotone in dimension i. In what follows we describe *sort* operators which can transform any boolean function over Σ^n into a monotone function (as we prove below).

Definition 2 *For every $i \in [n]$, the function $S_i[f] : \Sigma^n \mapsto \{0,1\}$ is defined as follows: For every $\alpha \in \Sigma^{i-1}$ and every $\beta \in \Sigma^{n-i}$, we let $S_i[f](\alpha\, 1\, \beta), \ldots, S_i[f](\alpha\, d\, \beta)$ be assigned the values of $f(\alpha\, 1\, \beta), \ldots, f(\alpha\, d\, \beta)$, in* sorted *order.*

For every $i \in [n]$ and every pair $(k, \ell) \in \Sigma^2$ so that $k < \ell$, let

$$\Delta_{i,(k,\ell)}(f) \stackrel{\text{def}}{=} \{(x,y) \in \textsc{Viol}(f) \,:\, \exists\, \alpha \in \Sigma^{i-1}, \beta \in \Sigma^{n-i} \text{ s.t. } x = \alpha\, k\, \beta\,,\, y = \alpha\, \ell\, \beta\}. \qquad (4)$$

Thus, $\bigcup_{i,(k,\ell)} \Delta_{i,(k,\ell)}(f)$ is the set of all violated edges of f that differ in a single coordinate. These are the only violated edges of f that can be potentially detected by our algorithmic schema. In the next lemma we show that by sorting in one dimension we do not increase the number of violations in any other dimension.

Lemma 4 *For every $f : \Sigma^n \mapsto \{0,1\}$ and $j \in [n]$, we have:*

1. *If f is monotone in dimensions $T \subseteq [n]$ then $S_j[f]$ is monotone in dimensions $T \cup \{j\}$;*
2. *For every $i \in [n] \setminus \{j\}$, and for every $1 \leq k < \ell \leq d$*

$$|\Delta_{i,(k,\ell)}(S_j[f])| \;\leq\; |\Delta_{i,(k,\ell)}(f)|.$$

Proof: The important observation is that in order to prove both items we may consider the function f restricted at all dimensions but the two in question. Furthermore, proofs of both items boil down to asserting claims about sorting zero-one matrices.

Item 1. Let i be some index in T, and assume without loss of generality that $i < j$. We fix any $\alpha \in \Sigma^{i-1}$, $\beta \in \Sigma^{j-i-1}$ and $\gamma \in \Sigma^{n-j}$, and consider the function $f' : \Sigma^2 \mapsto \{0,1\}$ defined by $f'(\sigma\tau) \stackrel{\text{def}}{=} f(\alpha\, \sigma\, \beta\, \tau\, \gamma)$. Clearly, f' is monotone in its first dimension (as f is monotone in dimension i). We need to show that so is $S_2[f']$ (as by definition of S_2 we have that $S_2[f']$ is monotone in dimension 2). Our claim thus amounts to saying that if one sorts the rows of a d-by-d zero-one matrix which is column-sorted then the columns remain sorted (the matrix we consider has its (σ, τ)-entry equal to $f'(\sigma\, \tau)$).

Let M denote a (d-by-d zero-one) matrix in which each column is sorted. We observe that the number of 1's in the rows of M is monotonically non-decreasing (as each column contributes a unit to the 1-count of row k only if it contributes a unit to the 1-count of row $k+1$). That is, if we let o_k denote the number of 1's in the k^{th} row then $o_k \leq o_{k+1}$ for $k = 1, \ldots, d-1$. Now suppose we sort each row of M resulting in a matrix M'. Then the k^{th} row of M' is $0^{d-o_k}1^{o_k}$, and it follows that the columns of M' remain sorted (as the $k+1^{\text{st}}$ row of M' is $0^{d-o_{k+1}}1^{o_{k+1}}$ and $o_k \leq o_{k+1}$).

Item 2. Fixing $i, j, \alpha, \beta, \gamma$ and defining f' as above, here we need to show that $|\Delta_{1,(k,\ell)}(S_2[f'])| \leq |\Delta_{1,(k,\ell)}(f')|$. The current claim amounts to saying that for any $2 \times d$ zero-one matrix, if we sort the rows of the matrix, then the number of unsorted columns cannot increase. Note that the claim refers only to rows k and ℓ in the d-by-d matrix considered in Item 1, and that $\Delta_{2,(1,\ell)}$ is the set of unsorted columns.

Let Q be any 2-by-d zero-one matrix, and let o_1 (resp., o_2) denote the number of ones in the first (resp., second) row of Q, and let Q' be the matrix resulting from sorting the rows of Q. If $o_1 \le o_2$, then Q' has no unsorted columns and we are done. Otherwise, Q' has exactly $(o_1 - o_2)$ unsorted columns. But then if we look at the o_1 columns of Q that have 1 in the first row, at most o_2 of them can be sorted, so Q should have at least $(o_1 - o_2)$ unsorted columns as well. ■

3.2 Dimension Reduction

With Lemma 4 at our disposal, we are ready to state and prove that the analysis of the algorithmic schema (for any n) reduces to its analysis for the special case $n = 1$. Let A denote one iteration of the algorithmic schema, p be any distribution on pairs $(k, \ell) \in \Sigma \times \Sigma$ such that $k < \ell$, and D_p^n be the corresponding distribution induced on edges of G_Σ^n. The dimension reduction lemma upper bounds $\epsilon_{\mathrm{M}}(f)$ and lower bounds $\textsc{Detect}(f, D_p^n)$ by the corresponding quantities for $n = 1$.

Lemma 5 (Dimension Reduction for Boolean Range) *Let $f : \Sigma^n \mapsto \{0,1\}$. Then there exist functions $f_{i,\alpha,\beta} : \Sigma \mapsto \{0,1\}$, for $i \in [n]$, $\alpha \in \{0,1\}^{i-1}$ and $\beta \in \{0,1\}^{n-i}$, so that the following holds (all expectations below are taken uniformly over $i \in [n]$, $\alpha \in \{0,1\}^{i-1}$ and $\beta \in \{0,1\}^{n-i}$):*

1. $\epsilon_{\mathrm{M}}(f) \le 2n \cdot \mathbf{E}_{i,\alpha,\beta}(\epsilon_{\mathrm{M}}(f_{i,\alpha,\beta}))$.
2. $\textsc{Detect}(f, D_p^n) \ge \mathbf{E}_{i,\alpha,\beta}(\textsc{Detect}(f_{i,\alpha,\beta},\ p))$.

We prove the lemma momentarily, but first consider its implication on the relatively simple case of $\Sigma = \{0,1\}$. First observe that in this case there is only one possible distribution p – the one assigning all weight to the pair $(0,1)$. Also, for any $f' : \{0,1\} \mapsto \{0,1\}$, Algorithm A rejects with probability exactly $2\epsilon_{\mathrm{M}}(f')$. Thus, the lemma implies that in the binary (domain and range) case, for any $f : \{0,1\}^n \mapsto \{0,1\}$,

$$\begin{aligned}
\mathbf{E}_{i,\alpha,\beta}(\textsc{Detect}(f_{i,\alpha,\beta},\ p)) &= \mathbf{E}_{i,\alpha,\beta}(2 \cdot \epsilon_{\mathrm{M}}(f_{i,\alpha,\beta})) \\
&= \frac{2}{n} \cdot \sum_i \mathbf{E}_{\alpha,\beta}(\epsilon_{\mathrm{M}}(f_{i,\alpha,\beta})) \\
&\ge \frac{1}{n} \cdot \epsilon_{\mathrm{M}}(f)
\end{aligned}$$

and we obtain a testing algorithm whose query complexity is $O(n/\epsilon)$. Note that the algorithm is very simple – it uniformly picks an edge whose endpoints differ in exactly one coordinate. Let us now prove the lemma.

Proof: For $i = 1, \ldots, n+1$, we define $f_i \stackrel{\mathrm{def}}{=} S_{i-1} \cdots S_1[f]$. Thus, $f_1 \equiv f$, and by Item 1 of Lemma 4, we have that f_{n+1} is monotone. It follows that

$$\epsilon_{\mathrm{M}}(f) \ \le\ \mathrm{dist}(f, f_{n+1}) \ \le\ \sum_{i=1}^{n} \mathrm{dist}(f_i, f_{i+1}). \tag{5}$$

Next, for $i = 1 \ldots n$, $\alpha \in \{0,1\}^{i-1}$ and $\beta \in \{0,1\}^{n-i}$, define the function $f_{i,\alpha,\beta} : \Sigma \mapsto \{0,1\}$, by $f_{i,\alpha,\beta}(x) = f_i(\alpha\, x\, \beta)$, for $x \in \Sigma$. Throughout the proof, $\sum_{\alpha,\beta}$ refers to summing over all (α, β)'s in

$\Sigma^{i-1} \times \Sigma^{n-i}$, and $\mathbf{E}_{\alpha,\beta}$ refers to expectation over uniformly distributed $(\alpha, \beta) \in \Sigma^{i-1} \times \Sigma^{n-i}$. We claim that

$$\mathrm{dist}(f_i, f_{i+1}) \;\leq\; 2 \cdot \mathbf{E}_{\alpha,\beta}(\epsilon_{\mathrm{M}}(f_{i,\alpha,\beta})). \tag{6}$$

This inequality is proven (below) by observing that f_{i+1} is obtained from f_i by sorting, separately, the elements in each $f_{i,\alpha,\beta}$. (The factor of 2 is due to the relationship between the distance of a vector to its sorted form and its distance to monotone.) We have,

$$\begin{aligned}
d^n \cdot \mathrm{dist}(f_i, f_{i+1}) &= \sum_{\alpha,\beta} |\{x \in \Sigma : f_i(\alpha\, x\, \beta) \neq f_{i+1}(\alpha\, x\, \beta)\}| \\
&= \sum_{\alpha,\beta} |\{x \in \Sigma : f_{i,\alpha,\beta}(x) \neq f_{i+1,\alpha,\beta}(x)\}| \\
&= \sum_{\alpha,\beta} |\{x \in \Sigma : f_{i,\alpha,\beta}(x) \neq S_i[f_{i,\alpha,\beta}](x)\}| \\
&\leq \sum_{\alpha,\beta} 2d \cdot \epsilon_{\mathrm{M}}(f_{i,\alpha,\beta}) = 2d^n \cdot \mathbf{E}_{\alpha,\beta}(\epsilon_{\mathrm{M}}(f_{i,\alpha,\beta}))
\end{aligned}$$

where the inequality is justified as follows. Consider a vector $v \in \{0,1\}^d$, and let $S(v)$ denote its sorted version. Then $S(v) = 0^z 1^{d-z}$, where z denotes the number of zeros in v. Thus, for some $e \geq 0$, the vector v has e 1-entries within its z-prefix and e 0-entries in its $(d-z)$-suffix. So the number of locations on which v and $S(v)$ disagree is exactly $2e$. On the other hand, consider an arbitrary perfect matching of the e 1-entries in the prefix and the e 0-entries in the suffix. To make v monotone one must alter at least one entry in each matched pair; thus, $\epsilon_{\mathrm{M}}(v) \geq e/d$.

Combining Eq. (5) and (6), the first item of the lemma follows:

$$\epsilon_{\mathrm{M}}(f) \;\leq\; \sum_{i=1}^{n} \mathrm{dist}(f_i, f_{i+1}) \leq 2 \cdot \sum_{i=1}^{n} \mathbf{E}_{\alpha,\beta}(\epsilon_{\mathrm{M}}(f_{i,\alpha,\beta})) = 2n \cdot \mathbf{E}_{i,\alpha,\beta}(\epsilon_{\mathrm{M}}(f_{i,\alpha,\beta})).$$

In order to prove the second item, we use the definition of algorithm A, and for any event E, let $\chi(E) = 1$ if E holds and $\chi(E) = 0$ otherwise.

$$\begin{aligned}
\textsc{Detect}(f, D_p^n) &= \frac{1}{n \cdot d^{n-1}} \sum_{i=1}^{n} \sum_{\alpha\,,\,\beta} \mathrm{Prob}_{(k,\ell)\sim p}[f(\alpha\, k\, \beta) > f(\alpha\, \ell\, \beta)] \\
&= \frac{1}{n \cdot d^{n-1}} \sum_{i=1}^{n} \sum_{\alpha\,,\,\beta} \sum_{(k,\ell)} p(k,\ell) \cdot \chi[f(\alpha\, k\, \beta) > f(\alpha\, \ell\, \beta)] \\
&= \frac{1}{n \cdot d^{n-1}} \sum_{i=1}^{n} \sum_{(k,\ell)} p(k,\ell) \cdot \sum_{\alpha\,,\,\beta} \chi[f(\alpha\, k\, \beta) > f(\alpha\, \ell\, \beta)] \\
&= \frac{1}{n \cdot d^{n-1}} \sum_{i=1}^{n} \sum_{(k,\ell)} p(k,\ell) \cdot |\Delta_{i,(k,\ell)}(f)|
\end{aligned}$$

Using Item 2 of Lemma 4, we have

$$|\Delta_{i,(k,\ell)}(f)| \;\geq\; |\Delta_{i,(k,\ell)}(S_1[f])| \;\geq\; \cdots \;\geq\; |\Delta_{i,(k,\ell)}(S_{i-1} \cdots S_1[f])| = |\Delta_{i,(k,\ell)}(f_i)|$$

Unwinding steps (7)-(7) with f_i in place of f and recalling the definition of $f_{i,\alpha,\beta}$, we get

$$\begin{aligned}\textsc{Detect}(f, D_p^n) &\geq \frac{1}{n \cdot d^{n-1}} \sum_{i=1}^{n} \sum_{\alpha\,,\,\beta} \mathrm{Prob}_{(k,\ell)\sim p}[f_i(\alpha\, k\, \beta) > f_i(\alpha\, \ell\, \beta)] \\ &= \mathbf{E}_{i,\alpha,\beta}(\textsc{Detect}(f_{i,\alpha,\beta},\ p)),\end{aligned}$$

and the lemma follows. ∎

3.3 Testing Monotonicity on a Line (the $n = 1$ case)

In this section we design algorithms for the case $n = 1$, for any Σ and Ξ. In accordance with our algorithmic schema, the design of such algorithms amounts to the design of a probability distribution $p : \Sigma^2 \mapsto [0, 1]$ (with support only on pairs (k, ℓ) with $k < \ell$).

Note that for $n = 1$, we have $D_p^n \equiv p$. We present three such distributions, denoted p_0, p_1, and p_2, and provide bounds on $\textsc{Detect}(f, p_j)$, for each j.

The following lemma[2], which relates $\epsilon_{\mathrm{M}}(f)$ to $\textsc{Viol}(f)$, will be used in our analysis of various algorithms. Recall that a *matching* of a graph is a collection of edges that share no common endpoint.

Lemma 6 *For any $f : \Sigma \mapsto \Xi$ the graph $G' = (\Sigma, \textsc{Viol}(f))$ has a matching of size $\epsilon_{\mathrm{M}}(f) \cdot |\Sigma|/2$.*

Proof: Recall that a *vertex cover* of a graph is a subset of vertices such that every edge of the graph has at least one of its endpoints in the subset. We claim that a minimum vertex cover of G' has size at least $\epsilon_{\mathrm{M}}(f) \cdot |\Sigma|$. The lemma directly follows as the size of a maximum matching is at least $1/2$ of the size of the minimum vertex cover. Let $U \subseteq \Sigma$ be any vertex cover of G'. We next show that by modifying the value of f only on points in U, we obtain a monotone function, implying that $|U| \geq \epsilon_{\mathrm{M}}(f) \cdot |\Sigma|$, as claimed.

Let $T = \Sigma \backslash U$. By definition of U, there are no violated edges between pairs of vertices in T. Consider the following iterative process, where in each step we modify the value of f on a single $y \in U$, remove y form U and add it to T. We maintain the property (which holds initially) that following each step there are no violated edges between vertices in T. The process ends when $U = \emptyset$ and $T = \Sigma$, so that the final function is monotone. To redefine the value of f on y, we consider the following two subsets of T: $T_1 = \{x \in T :\ (x, y) \in \textsc{Viol}(f)\}$ and $T_2 = \{z \in T :\ (y, z) \in \textsc{Viol}(f)\}$. By transitivity of the partial order, and the fact that there are no violated edges (x, z), for $x, z \in T$, at most one of these subsets is non-empty. If T_1 is non-empty then we let $f(y) = \max_{x \in T_1}\{f(x)\}$, and if T_2 is non-empty, then $f(y) = \min_{z \in T_2}\{f(y)\}$. In case both are empty (all violated edges incident to y have an end-point in U), the value of y may remain unchanged.

We note that the size of the minimum vertex cover actually equals $\epsilon_{\mathrm{M}}(f) \cdot |\Sigma|$. Consider any set U such that by modifying the value of f only on strings in U we can obtain a monotone function g. Then U must be a vertex cover of G', as otherwise there remain violated edges with respect to g. ∎

DISTRIBUTION p_0: This distribution is uniform over pairs $(k, k+1)$. That is, $p_0(k, k+1) = 1/(d-1)$, for $k = 1, \ldots, d-1$.

Proposition 7 *For any Ξ and $f : \Sigma \mapsto \Xi$, $\textsc{Detect}(f, p_0) \geq \frac{2}{d-1} \cdot \epsilon_{\mathrm{M}}(f)$.*

[2] While stated for a totally ordered set Σ, the result and the same proof hold for any partially ordered set S.

The lower bound can be shown to be tight even for $\Xi = \{0,1\}$ (by considering the function f defined by $f(x) = 1$ if $x < d/2$ and $f(x) = 0$ otherwise).

Proof: If $\epsilon_{\mathrm{M}}(f) > 0$, then there exists some $k \in \{1, \ldots, d-1\}$ so that $f(k) > f(k+1)$. If there are at least two such k's, then we reject with probability at least $2/(d-1) \geq 2\epsilon_{\mathrm{M}}(f)/(d-1)$ as $\epsilon_{\mathrm{M}}(f) \leq 1$. Otherwise, there is a unique k that causes us to reject. In this case $\epsilon_{\mathrm{M}}(f) \leq 1/2$ since we can change either all $f(i)$ to $f(k+1)$ for $i \leq k$, or all $f(i)$ to $f(k)$ for $i > k$ in order to make f monotone. Thus, we reject with probability $1/(d-1) \geq 2\epsilon_{\mathrm{M}}(f)/(d-1)$ in this case as well. ■

We see that the above test is too "short-sighted" since it only looks at the neighboring pairs of vertices. We now describe a distribution that spots the violated edges much better. As noted before, an alternative distribution which meets the same bound was previously suggested and analyzed in [6].

Distribution p_1: This distribution is uniform on a set $P \subset \Sigma \times \Sigma$ which is defined as follows. The set P consists of pairs (k, ℓ), where $0 < \ell - k \leq 2^t$ and 2^t is the largest power of 2 which divides either k or ℓ. That is, let $\mathrm{power}_2(i) \in \{0, 1..., \log_2 i\}$ denote the largest power of 2 which divides i. Then,

$$P \stackrel{\mathrm{def}}{=} \{(k,\ell) \in \Sigma \times \Sigma : 0 < \ell - k \leq 2^{\max(\mathrm{power}_2(k), \mathrm{power}_2(\ell))}\} \tag{7}$$

and $p_1(k, \ell) = \frac{1}{|P|}$ for every $(k, \ell) \in P$, and is 0 otherwise.

Proposition 8 *For any Ξ and $f : \Sigma \mapsto \Xi$,* $\mathrm{Detect}(f, p_1) \geq \frac{1}{O(\log d)} \cdot \epsilon_{\mathrm{M}}(f)$.

Proof: We first show that $|P| = O(d \log d)$. This can be shown by charging each pair $(k, \ell) \in P$ to the element divisible by the larger power of 2 (i.e., to k if $\mathrm{power}_2(k) > \mathrm{power}_2(\ell)$ and to ℓ otherwise), and noting that the charge incurred on each i is at most $2 \cdot 2^{\mathrm{power}_2(i)}$. It follows that the total charge is at most $\sum_{i=1}^{d} 2^{\mathrm{power}_2(i)+1} = \sum_{j=0}^{\log_2 d} \frac{d}{2^j} \cdot 2^{j+1} = O(d \log d)$.

Since p_1 is uniform over P, the value of $\mathrm{Detect}(f, p_1)$ is the ratio between the number of violated edges of f in P and the size of P. Thus, it remains to show that the former is $\Omega(\epsilon_{\mathrm{M}}(f) \cdot d)$. In the following argument it will be convenient to view the indices $1, \ldots, d$ as vertices of a graph and the pairs $(k, \ell) \in P$ as directed edges. We refer to this graph as G_P, and note that it is a subgraph if G_Σ^1.

Claim 8.1: For every two vertices k and ℓ in G_P with $k < \ell$, there is a directed path of length at most 2 from k to ℓ in G_P.

Proof of Claim: Let $r = \lceil \log d \rceil$, and consider the binary strings of length r representing k and ℓ. Let $k = (x_{r-1}, \ldots, x_0)$ and $\ell = (y_{r-1}, \ldots, y_0)$. Let j be the highest index such that $x_j = 0$ and $y_j = 1$. Note that $x_i = y_i$ for $j < i < r$. We claim that the vertex $s = (x_{r-1}, \ldots, x_{t+1}, 1, 0, \ldots 0)$ is on a path of length 2 from k to ℓ. This follows from the definition of P, since s is divided by 2^j, while both $s - k = 2^j - \sum_{i=0}^{j-1} x_i 2^i \leq 2^j$ and $\ell - s = \sum_{i=0}^{j-1} y_i 2^i < 2^j$. □

We now apply Lemma 6 to obtain a matching M of size $m \geq (\epsilon_{\mathrm{M}}(f) \cdot d/2)$ consisting of violated edges of f. By the above claim, there is path of length at most 2 in G_P between every matched pair. Each edge e of G_P belongs to at most 2 such paths: on at most one path it is the first edge, and on at most one it is the second edge (or otherwise M is not a matching). Since for every $(x, y) \in M$ we have $f(x) > f(y)$ (while $x \prec y$), the length-2 path between x and y must contain a violated edge. Thus, we obtain at least $m/2 \geq (\epsilon_{\mathrm{M}}(f) \cdot d/4)$ violated edges in G_P, and the proposition follows. ■

ON THE OPTIMALITY OF DISTRIBUTION p_1. We show that the result of Proposition 8 is optimal (up to a constant factor), even for $\Xi = \{0,1\}$. The following argument is due to Michael Krivelevich.

Proposition 9 *For any distribution $p : \Sigma \times \Sigma \mapsto [0,1]$, with support only on pairs (k,ℓ) such that $k < \ell$, there exists a non-monotone $f : \Sigma \mapsto \{0,1\}$ so that*

$$\text{DETECT}(f,p) \;\leq\; \frac{2}{\log_2 d} \cdot \epsilon_{\mathrm{M}}(f)$$

Proof: Let p be a distribution on pairs as above. We define

$$\rho \stackrel{\text{def}}{=} \max_{f:\Sigma\mapsto\{0,1\} \text{ s.t. } \epsilon_{\mathrm{M}}(f)>0} \left\{ \frac{\text{DETECT}(f,p)}{\epsilon_{\mathrm{M}}(f)} \right\}$$

Our aim is to show that $\rho \leq 2/\log_2 d$. The key observation is that for any consecutive $2a$ indices, p has to assign a probability mass of at least $\rho \cdot a/d$ to pairs (k,ℓ) where k is among the lowest a indices and ℓ is among the higher a such indices. This observation is proven as follows. Let L, H be the low and high parts of the interval in question; that is, $L = \{s+1, ..., s+a\}$ and $H = \{s+a+1, ..., s+2s\}$, for some $s \in \{0, ..., d-2a\}$. Consider the function f defined by $f(i) = 1$ if $i \in L \cup \{s+2a+1, ..., d\}$ and $f(i) = 0$ otherwise. Then $\epsilon_{\mathrm{M}}(f) = a/d$. On the other hand, the only pairs (k,ℓ) with $f(k) > f(\ell)$, are those satisfying $k \in L$ and $\ell \in H$. Thus, by definition of ρ, it must hold that $\rho \leq \Pr_{(k,\ell)\sim p}[k \in L \;\&\; \ell \in H]/(a/d)$, and the observation follows.

The rest of the argument is quite straightforward: Consider $\log_2 d$ partitions of the interval $[1,d]$, so that the i^{th} partition is into consecutive segments of length 2^i. For each segment in the i^{th} partition, probability p assign a probability mass of at least $2^{i-1}\rho/d$ to pairs where one element is in the low part of the segment and the other element is in the high part. Since these segments are disjoint and their number is $d/2^i$, it follows that p assigns a probability mass of at least $\rho/2$ to pairs among halves of segments in the i^{th} partition. These pairs are disjoint from pairs considered in the other partitions and so we conclude that $(\log_2 d) \cdot \frac{\rho}{2} \leq 1$. The proposition follows. ■

We now describe a distribution that works well for the boolean range only.

DISTRIBUTION p_2: This distribution is uniform over all pairs (k,ℓ) such that $k < \ell$. That is, $p_2(k,\ell) = 2/((d-1)d)$ for $1 \leq k < \ell \leq d$.

Proposition 10 *For any $f : \Sigma \mapsto \{0,1\}$, $\text{DETECT}(f,p_2) \geq \epsilon_{\mathrm{M}}(f)^2/2$.*

A slightly more careful analysis (which we omit) can extend the bound to $\epsilon_{\mathrm{M}}(f)^2$, which is tight. For any integer $e < d/2$, consider the function $f(x) = 0$ if $x \in \{2,4,6\ldots 2e\}$ and $f(x) = 1$ otherwise. Then $\epsilon_{\mathrm{M}}(f) = e/d$ and $|\text{VIOL}(f)| = 1 + \ldots + e \approx e^2/2$. Thus, $\text{DETECT}(f,p_2) \approx (e^2/2)/(d^2/2) = (e/d)^2 = \epsilon_{\mathrm{M}}(f)^2$.

Proof: Let z be the number of zeroes in f and let $2e$ be the number of mismatches between f and its sorted form. Then $\epsilon_{\mathrm{M}}(f) \leq 2e/d$ as by swapping the $2e$ mismatches we make f monotone. On the other hand, considering the e 1-entries in the z-prefix of f and the e 0-entries in the $(d-z)$-suffix, we lower bound the rejection probability by $e^2/((d-1)d/2) > 2(e/d)^2$. Combining the two, we conclude that $\text{DETECT}(f,p_2) \geq 2 \cdot (\epsilon_{\mathrm{M}}(f)/2)^2$. ■

We remark that the restriction to boolean range in Proposition 10 is important. For any integer $e \leq d/2$, define $f : \Sigma \mapsto \Sigma$ by $f(2i) = 2i-1$, $f(2i-1) = 2i$ for $i = 1\ldots e$, and $f(i) = i$ for $i > 2e$. Clearly, $\epsilon_{\mathrm{M}}(f) = e/d$, while f has only e violated edges: $(2i-1, 2i)$, $i = 1\ldots e$. Thus, $\text{DETECT}(f,p_2) = e/(d(d-1)/2) = 2\epsilon_{\mathrm{M}}(f)/(d-1)$, which is much less than $\epsilon_{\mathrm{M}}(f)^2$ if e is large.

3.4 Proof of Theorem 2

In this subsection we combine Lemma 5 with the results for the case $n = 1$ provided in Subsection 3.3, and derive Theorem 2

Combining Lemma 5 and Proposition 8 (applied only to $\Xi = \{0, 1\}$), we have

$$\begin{array}{rcll} \textsc{Detect}(f, D^n_{p_1})) & \geq & \mathbf{E}_{i,\alpha,\beta}(\textsc{Detect}(f_{i,\alpha,\beta},\ p_1)) & \text{[By Part 2 of the lemma]} \\ & \geq & \mathbf{E}_{i,\alpha,\beta}(\epsilon_{\mathrm{M}}(f_{i,\alpha,\beta})/O(\log d)) & \text{[By the proposition]} \\ & \geq & \frac{\epsilon_{\mathrm{M}}(f)}{2n \cdot O(\log d)} = \Omega(\frac{\epsilon_{\mathrm{M}}(f)}{n \log d}) & \text{[By Part 1 of the lemma]} \end{array}$$

which establishes the the first item in the theorem.

Combining Lemma 5 and Proposition 10, we have

$$\begin{array}{rcll} \textsc{Detect}(f, D^n_{p_2})) & \geq & \mathbf{E}_{i,\alpha,\beta}(\textsc{Detect}(f_{i,\alpha,\beta},\ p_2)) & \text{[By Part 2 of the lemma]} \\ & \geq & \mathbf{E}_{i,\alpha,\beta}(\epsilon_{\mathrm{M}}(f_{i,\alpha,\beta})^2/2) & \text{[By the proposition]} \\ & \geq & [\mathbf{E}_{i,\alpha,\beta}(\epsilon_{\mathrm{M}}(f_{i,\alpha,\beta}))]^2/2 & \text{[as } \mathbf{E}(X^2) \geq [\mathbf{E}(X)]^2\text{]} \\ & \geq & (\epsilon_{\mathrm{M}}(f)/2n)^2/2 = \Omega(\epsilon_{\mathrm{M}}(f)^2/n^2) & \text{[By Part 1 of the lemma]} \end{array}$$

which establishes the second item in the theorem.

As noted in the introduction, by slightly modifying our algorithmic schema, it is possible to improve upon the algorithm implied by the second item of the theorem. Similarly to our original algorithm, the modified algorithm reduces the the general question to the $n = 1$ case, where it does the following. It selects $O(1/\epsilon)$ values $k \in [d]$, and queries the function on all these values. If a violation of monotonicity is detected (i.e., at least one of $O(1/\epsilon^2)$ possible pairs are out of order), the algorithm rejects. It can be shown that this algorithm for $n = 1$ will reject a function that is ϵ far from monotone with probability at least 2/3.

For general n the algorithm performs the following procedure several times. It uniformly selects a dimension $i \in [n]$, a prefix $\alpha \in \Sigma^{i-1}$ and a suffix $\beta \in \Sigma^{n-i}$. Then it performs the $n = 1$ test (with a possibly different value of ϵ) on $f(\alpha, \cdot, \beta)$. It can be shown that with appropriate choice of parameters, the complexity of this test can be made $O((n/\epsilon)\log^2(n/\epsilon))$. We note, however, that this test actually uses the *values* of f that it gets, while our test only needs to know if a given edge (x, y) is violated, ignoring the other information it gets from $f(x)$ and $f(y)$.

4 Testing Monotonicity over General Ranges

We now reduce the problem of testing arbitrary-range functions to the simpler problem of testing boolean functions, which was considered in the preceding section. This reduction works not only for functions with domain Σ^n, but more generally when the domain is any partially ordered set S. The reduction is characterized by Theorem 3, which states that a certain type of monotonicity test for functions of the form $f : S \mapsto \{0, 1\}$ also works well for functions of the form $f : S \mapsto \Xi$. Here Ξ is a finite totally ordered set of size r, which we can regard as the integers in the interval $[0, r-1]$. Furthermore, for simplicity, we assume that $r = 2^s$ for some integer s. All references to "edges" are references to edges of the partial order graph, whose vertices are strings in the domain S and directed edges correspond to ordered comparable pairs (i.e. (x, y) is an edge iff $x \prec y$).

To ensure that a function far from monotone can be readily detected by our test, we lower bound $\textsc{Detect}(f, D)$ in terms of $\epsilon_{\mathrm{M}}(f)$. Equivalently, we are looking for a good upper bound on $\epsilon_{\mathrm{M}}(f)$ in terms of $\textsc{Detect}(f, D)$. We reduce the task of obtaining an upper bound for functions with an arbitrary range to that of obtaining such an upper bound for functions with binary range.

The general idea of the reduction is to incrementally transform a function f into a monotone function, while ensuring that for each repaired violated edge, the value of the function is changed at only a few points. This transformation allows us to find a monotone function close to f and to upper bound $\epsilon_{\mathrm{M}}(f)$ by the distance from f to that function. The transformation produces the following chain of functions: $f \mapsto f_1 \mapsto f_2 \mapsto f_3$, where f_3 is monotone. The distance between any two consecutive functions in the chain is equal to the distance to monotone of some auxiliary function with a smaller range. Thus, we obtain an upper bound on $\epsilon_{\mathrm{M}}(f)$ in terms of the distance to monotone of smaller-range functions. In addition, edges violated by the auxiliary functions are also violated by f, and we can obtain a lower bound on $\mathrm{DETECT}(f, D)$ in terms of the corresponding probability for the smaller-range auxiliary functions. Using the inductive assumption for smaller-range functions and the two claims above, we finally obtain the needed upper bound on $\epsilon_{\mathrm{M}}(f)$ in terms of $\mathrm{DETECT}(f, D)$.

Subsection 4.1 describes and analyzes operators SQUASH, MON, and CLEAR later used to define functions f_1, f_2, and f_3 described above. Subsection 4.2 proves the range reduction lemma which upper bounds $\epsilon_{\mathrm{M}}(f)$ and lower bounds $\mathrm{DETECT}(f, D)$ by the corresponding quantities for smaller range functions. This section is concluded by the proof of Theorem 3 in Subsection 4.3.

4.1 Operators SQUASH, MON, and CLEAR

First, we introduce operators, later used for obtaining functions f_1, f_2, and f_3 related to f.

Definition 3 *The operators* SQUASH, MON, *and* CLEAR *each map a function $f : S \mapsto [0, r-1]$ to a related function with the same domain and the same or smaller range. In particular,* MON$[f]$ *is some arbitrary monotone function at distance $\epsilon_{\mathrm{M}}(f)$ from the function f. The operators* SQUASH *and* CLEAR *are defined below; in these definitions a and b are elements of $[0, r-1]$ and $a < b$.*

$$\mathrm{SQUASH}[f,a,b](x) = \begin{cases} a & \textit{if } f(x) \leq a \\ b & \textit{if } f(x) \geq b \\ f(x) & \textit{otherwise} \end{cases}$$

$$\mathrm{CLEAR}[f,a,b](x) = \begin{cases} \mathrm{MON}[\mathrm{SQUASH}[f,a,b]](x) & \\ \qquad \textit{if } \mathrm{MON}[\mathrm{SQUASH}[f,a,b]](x) \neq \mathrm{SQUASH}[f,a,b](x) & \\ f(x) \quad \textit{otherwise} & \end{cases}$$

SQUASH operator simply "squashes" the range of f to $[a, b]$. Notice that if an edge is not violated by f, it is not violated by SQUASH$[f, a, b]$.

Claim 11 *For all $f : S \mapsto [0, r-1]$ and all $a, b \in [0, r-1]$ such that $a < b$, the operator* SQUASH *does not introduce any new violated edges, i.e.* $\mathrm{VIOL}(\mathrm{SQUASH}[f, a, b]) \subseteq \mathrm{VIOL}(f)$. ■

CLEAR operator first "squashes" the range to $[a, b]$, then alters the resulting smaller-range function at some points to obtain the closest monotone function, and finally "unsquashes" the function at unaltered points to the original values. This leads to the following simple claim:

Claim 12 *For all $f : S \mapsto [0, r-1]$ and all $a, b \in [0, r-1]$ such that $a < b$,*

$$\mathrm{dist}(f, \mathrm{CLEAR}[f, a, b]) = \epsilon_{\mathrm{M}}(\mathrm{SQUASH}[f, a, b]).$$

Proof: By definitions of the CLEAR and MON operators:

$$\mathrm{dist}(f, \mathrm{CLEAR}[f, a, b]) = \mathrm{dist}(\mathrm{MON}[\mathrm{SQUASH}[f, a, b]], \mathrm{SQUASH}[f, a, b]) = \epsilon_{\mathrm{M}}(\mathrm{SQUASH}[f, a, b]).$$

■

Define the *interval of a violated edge* (x, y) *with respect to function* f to be the interval $[f(y), f(x)]$ (since the edge is violated by f, $f(x) > f(y)$). We say that two intervals *cross* if they intersect in more than one point. Intuitively, if we consider $f(x) - f(y)$ as a measure of how violated an edge (x, y) is, then we can say that $\text{CLEAR}[f, a, b]$ partially repairs violated edges of f whose intervals cross $[a, b]$ without worsening other violated edges. The following lemma formalizes important properties of CLEAR.

Lemma 13 *The function* $\text{CLEAR}[f, a, b]$ *has the following properties for all* $f : S \mapsto [0, r-1]$ *and all* $a, b \in [0, r-1]$ *such that* $a < b$*:*

1. $\text{VIOL}(\text{CLEAR}[f, a, b]) \subseteq \text{VIOL}(f)$*, i.e.* CLEAR *does not introduce any new violated edges.*
2. $\text{CLEAR}[f, a, b]$ *has no violated edges whose intervals cross* $[a, b]$*.*
3. *The interval of a violated edge with respect to* $\text{CLEAR}[f, a, b]$ *is contained in the interval of this edge with respect to* f*.*

Proof: For brevity, define $g = \text{MON}[\text{SQUASH}[f, a, b]]$ and $h = \text{CLEAR}[f, a, b]$. Let (x, y) be an edge violated by h; that is, $h(x) > h(y)$. By its definition, g is monotone and takes values in $[a, b]$. Also notice that $h(x) = f(x)$ if $h(x) \notin [a, b]$, and $h(x) = g(x)$ if $h(x) \in [a, b]$. We consider four cases where each of $h(x)$ and $h(y)$ is either inside or outside the interval $[a, b]$.

- Case 1: $h(x), h(y) \in [a, b]$. This case cannot occur: (x, y) cannot be violated by h because $h(x) = g(x), h(y) = g(y)$ and g is monotone.
- Case 2: $h(x), h(y) \notin [a, b]$. Since f and h agree on both x and y, it follows that (x, y) is violated by f and $[h(y), h(x)] = [f(y), f(x)]$. This proves parts 1 and 3. To show that $[h(y), h(x)]$ does not cross $[a, b]$, it remains to prove that the case when $h(x) > b$ and $h(y) < a$ cannot happen. But in such a case we must have $g(x) = b$ and $g(y) = a$ and that contradicts the monotonicity of g.
- Case 3: $h(x) \notin [a, b]$, $h(y) \in [a, b]$. Since (x, y) is violated, $h(x) > b$. Consequently, $f(x) = h(x) > b$ and, thus, $g(x) = b$. Since g is monotone, $g(y) \geq g(x) = b$, and hence $h(y) = g(y) = b$. This proves that $[h(y), h(x)]$ intersects $[a, b]$ in at most one point (b), establishing part 2. If $f(y) = h(y) = b$, then f and h agree on both x and y, and parts 1 and (3) follow. If not, then $b = g(y) \neq \text{SQUASH}[f, a, b](y)$. Thus, $\text{SQUASH}[f, a, b](y) < b$, and hence $f(y) < b = h(y)$. Since $f(x) = h(x) > b$, parts 1 and 3 follow.
- Case 4: $h(x) \in [a, b], h(y) \notin [a, b]$. This case is symmetrical to Case 3.

■

4.2 Range Reduction

We are now ready to define functions in the chain $f \mapsto f_1 \mapsto f_2 \mapsto f_3$, as well as auxiliary smaller-range functions f_1', f_2', and f_3'. Lemma 14 defines these functions and summarizes their properties. The transition from f to f_1 transforms violated edges with one endpoint in the lower half of the range and the other endpoint in the upper half into edges with both endpoints in the same half of the range. Then we repair violated edges with both endpoints in the lower half of the range to obtain f_2 and finally, upper half of the range to obtain f_3.

Lemma 14 (Range Reduction) *Given* $f : S \mapsto [0, r-1]$*, define*

$$\begin{array}{lll} f' = \text{SQUASH}[f, \frac{r}{2}-1, \frac{r}{2}], & f_1' = \text{SQUASH}[f_1, 0, \frac{r}{2}-1], & f_2' = \text{SQUASH}[f_2, \frac{r}{2}, r-1], \\ f_1 = \text{CLEAR}[f, \frac{r}{2}-1, \frac{r}{2}], & f_2 = \text{CLEAR}[f_1, 0, \frac{r}{2}-1], & f_3 = \text{CLEAR}[f_2, \frac{r}{2}, r-1]. \end{array}$$

These functions have the following properties, for any probability distribution D*.*

1. $\text{DETECT}(f, D) \geq \text{DETECT}(f', D)$
2. $\text{DETECT}(f, D) \geq \text{DETECT}(f_1', D) + \text{DETECT}(f_2', D)$
3. $\epsilon_{\text{M}}(f) \leq \epsilon_{\text{M}}(f') + \epsilon_{\text{M}}(f_1') + \epsilon_{\text{M}}(f_2')$

Proof: All references to "parts" are references to parts of Lemma 13.

(*1*) The SQUASH operator never adds new violated edges by Claim 11. Therefore, $\text{VIOL}(f') \subseteq \text{VIOL}(f)$, and the claim follows.

(*2*) It is enough to show that $\text{VIOL}(f_1')$ and $\text{VIOL}(f_2')$ are *disjoint* subsets of $\text{VIOL}(f)$. First, note that $\text{VIOL}(f_1')$ and $\text{VIOL}(f_2')$ are subsets of $\text{VIOL}(f)$ because f_1' and f_2' are constructed from f using a sequence of CLEAR and SQUASH operators, which never add new violated edges by Claim 11 and part 1.

It remains to prove that $\text{VIOL}(f_1')$ and $\text{VIOL}(f_2')$ are disjoint. By part 2, there is no edge violated by f_1 whose interval crosses $[\frac{r}{2}-1, \frac{r}{2}]$. Hence, the edges violated by f_1 are partitioned into two disjoint subsets: "low" edges with intervals contained in $[0, \frac{r}{2}-1]$ and "high" edges with intervals contained in $[\frac{r}{2}, r-1]$. The edges violated by f_1' are a subset of the low edges, since the SQUASH operator repairs all high violated edges and adds no new violated edges by Claim 11. The edges violated by f_2' are a subset of the high edges, since the CLEAR operator used to form f_2 repairs all low violated edges by parts 2 and 3, and no new violated edges are added by Claim 11 and part 1.

(*3*) First, we show that f_3 is monotone. Since the function f_3 is constructed from f using a sequence of three CLEAR operators, parts 2 and 3 imply that there is no edge violated by f_3 whose interval crosses any of the intervals $[\frac{r}{2}-1, \frac{r}{2}]$, $[0, \frac{r}{2}-1]$, or $[\frac{r}{2}, r-1]$. Therefore, f_3 violates no edges at all and is monotone.

Now the distance from f to the set of monotone functions is at most the distance from f to the particular monotone function f_3, and we get:

$$\epsilon_{\text{M}}(f) \leq \text{dist}(f, f_3) \leq \text{dist}(f, f_1) + \text{dist}(f_1, f_2) + \text{dist}(f_2, f_3) = \epsilon_{\text{M}}(f') + \epsilon_{\text{M}}(f_1') + \epsilon_{\text{M}}(f_2').$$

The last step uses Claim 12. ■

4.3 Proof of Theorem 3

In this subsection we use the results of the preceding lemma to prove Theorem 3. The proof is by induction on s with the inductive hypothesis that for every function $f : S \mapsto \Xi$ where $|\Xi| = 2^s$,

$$\epsilon_{\text{M}}(f) \leq C \cdot \text{DETECT}(f, D) \cdot s.$$

In the base case where $s = 1$, the hypothesis holds by the assumption stated in the theorem. Now assume that the hypothesis holds for $s-1$ to prove that it holds for s. We can reason as follows:

$$\begin{aligned}
\epsilon_{\mathrm{M}}(f) &\leq \epsilon_{\mathrm{M}}(f') + \epsilon_{\mathrm{M}}(f'_1) + \epsilon_{\mathrm{M}}(f'_2) \\
&\leq C \cdot \textsc{Detect}(f', D) + C \cdot \textsc{Detect}(f'_1, D) \cdot (s-1) + C \cdot \textsc{Detect}(f'_2, D) \cdot (s-1) \\
&\leq C \cdot (\textsc{Detect}(f, D) + \textsc{Detect}(f, D)(s-1)) \\
&= C \cdot \textsc{Detect}(f, D) \cdot s
\end{aligned}$$

The first inequality was proved in part 3 of Lemma 14. The second inequality uses the induction hypothesis; recall that the range of f' has size 2^1, and the ranges of f'_1 and f'_2 have size $r/2 = 2^{s-1}$. The third step uses parts 1 and 2 of Lemma 14, and the final step is simplification. This completes the proof.

Acknowledgments

We would like to thank Michael Krivelevich, Michael Sipser and Madhu Sudan for helpful discussions. In particular, Proposition 9 is due to Michael Krivelevich.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. *JACM*, 45(3):501–555, 1998.

[2] S. Arora and S. Sudan. Improved low degree testing and its applications. In *Proceedings of STOC97*, pages 485–495, 1997.

[3] M. Bellare, D. Coppersmith, J. Håstad, M. Kiwi, and M. Sudan. Linearity testing in characteristic two. In *Proceedings of FOCS95*, pages 432–441, 1995.

[4] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximation. In *Proceedings of STOC93*, pages 294–304, 1993.

[5] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *JACM*, 47:549–595, 1993.

[6] F. Ergun, S. Kannan, S. R. Kumar, R. Rubinfeld, and M. Viswanathan. Spot-checkers. In *Proceedings of STOC98*, pages 259–268, 1998.

[7] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, and A. Wigderson. Self-testing/correcting for polynomials and for approximate functions. In *Proceedings of STOC91*, pages 32–42, 1991.

[8] O. Goldreich, S. Goldwasser, E. Lehman, and D. Ron. Testing monotonicity. In *Proceedings of FOCS98*, 1998.

[9] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *JACM*, 45(4):653–750, 1998. An extended abstract appeared in the proceedings of FOCS96.

[10] O. Goldreich and D. Ron. Property testing in bounded degree graphs. In *Proceedings of STOC97*, pages 406–415, 1997.

[11] O. Goldreich and D. Ron. A sublinear bipartite tester for bounded degree graphs. In *Proceedings of STOC98*, pages 289–298, 1998. To appear in *Combinatorica*, 1999.

[12] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of STOC97*, pages 475–484, 1997.

[13] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'