Testing Monotonicity

Oded Goldreich[*] Shafi Goldwasser[†] Eric Lehman[†] Dana Ron[‡]

September 8, 1998

Abstract

We present a (randomized) test for monotonicity of Boolean functions. Namely, given the ability to query an unknown function $f : \{0,1\}^n \mapsto \{0,1\}$ at arguments of its choice, 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). The complexity of the test is $\mathrm{poly}(n/\epsilon)$.

The analysis of our algorithm relates two natural combinatorial quantities that can be measured with respect to a Boolean function; one being global to the function and the other being local to it.

We also consider the problem of testing monotonicity based only on random examples labeled by the function. We show an $\Omega(\sqrt{2^n/\epsilon})$ lower bound on the number of required examples, and provide a matching upper bound (via an algorithm).

[*]Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, ISRAEL. E-mail: `oded@wisdom.weizmann.ac.il`. Work done while visiting LCS, MIT.

[†]Laboratory for Computer Science, MIT, 545 Technology Sq., Cambridge, MA 02139. E-mail: {`shafi,e_lehman,danar`}`@theory.lcs.mit.edu`.

[‡]Supported by an ONR Science Scholar Fellowship at the Bunting Institute.

1 Introduction

In this work we address the problem of *testing whether a given Boolean function is monotone*. A function $f : \{0,1\}^n \mapsto \{0,1\}$ is said to be monotone if $f(x) \leq f(y)$ for every $x \prec y$, where $\prec$ denotes the natural partial order among strings (i.e., $x_1 \cdots x_n \prec y_1 \cdots y_n$ if $x_i \leq y_i$ for every i and $x_i < y_i$ for some i). The testing algorithm can request the value of the function on arguments of its choice, and is required to distinguish monotone functions from functions that are far from being monotone.

More precisely, the testing algorithm is given a *distance* parameter $\epsilon > 0$, and oracle access to an unknown function f mapping $\{0,1\}^n$ to $\{0,1\}$. If f is a monotone then the algorithm should accept it with probability at least $2/3$, and if f is at distance greater than ϵ from any monotone function then the algorithm should reject it with probability at least $2/3$. Distance between functions is measured in terms of the fraction of the domain on which the functions differ. The complexity measures we focus on are the *query complexity* and the *running time* of the testing algorithm.

We present a randomized algorithm for testing the monotonicity property whose query complexity and running time are polynomial in n and $1/\epsilon$. The algorithm performs a simple local test: It verifies whether monotonicity is maintained for randomly chosen pairs of strings that differ exactly on a single bit. In our analysis we relate this local measure to the global measure we are interested in — the minimum distance of the function to any monotone function.

1.1 Perspective

Property Testing, as explicitly defined by Rubinfeld and Sudan [RS96] and extended in [GGR96], is best known by the special case of *low degree testings* [BLR93, GLR+91, RS96, RS97, AS97] which plays a central role in the construction of probabilistically checkable proofs (PCP) [BFL91, BFLS91, FGL+96, AS98, ALM+98, RS97, AS97]. The recognition that property testing is a general notion has been implicit in the context of PCP: It is understood that low degree tests as used in this context are actually codeword tests (in this case of BCH codes), and that such tests can be defined and performed also for other error-correcting codes such as the Hadamard code [ALM+98, BGLR93, BS94, BCH+95, BGS98, Kiw96, Tre98], and the "Long Code" [BGS98, Hås96, Hås97, Tre98].

Forasmuch as error-correcting codes emerge naturally in the context of PCP, they do not seem to provide a natural representation of familiar objects whose properties we may wish to investigate. That is, one can certainly encode any given object by an error-correcting code — resulting in a (legitimate yet) probably unnatural representation of the object — and then test properties of the encoded object. However, this can hardly be considered as a "natural test" of a "natural phenomena". For example, one may indeed represent a graph by applying an error correcting code to its adjacency matrix (or to its incidence list), but the resulting string is not the "natural representation" of the graph.

The study of Property Testing as applied to natural representation of (non-algebraic) objects was initiated in [GGR96]. In particular, Property Testing as applied to *graphs* has been studied in [GGR96, GR97, GR98] – where the first work considers the *adjacency matrix representation* of graphs (most adequate for dense graphs), and the latter works consider the *incidence list representation* (adequate for sparse graphs).

In this work we consider property testing as applied to the most generic (i.e., least structured) object – an arbitrary Boolean function. In this case the choice of representation is "forced" upon us.

1.2 Monotonicity

In interpreting monotonicity it is useful to view Boolean functions over $\{0,1\}^n$ as subsets of $\{0,1\}^n$, called *concepts*. This view is the one usually taken in the PAC Learning literature. Each position in $\{1,\ldots,n\}$ corresponds to a certain *attribute*, and a string $x = x_1 \ldots x_n \in \{0,1\}^n$ represents an instance where $x_i = 1$ if and only if the instance x has the $i^{\rm th}$ attribute. Thus, a concept (subset of instances) is monotone if the presence of additional attributes maintains membership of instances in the concept (i.e., if instance x is in the concept C then any instance resulting from x by adding some attributes is also in C).

The class of monotone concepts is quite general and rich. On the other hand, monotonicity suggests a certain aspect of simplicity. Namely, each attribute has a uni-directional effect on the value of the function. Thus, knowing that a concept is monotone may be useful in various applications. In fact, this form of simplicity is exploited by Angluin's learning algorithm for monotone concepts [Ang88], which uses membership queries and has complexity that is linear in the number of terms of the target concept's DNF representation.

We note that an efficient tester for monotonicity is useful as a preliminary stage before employing Angluin's algorithm. As is usually the case, Angluin's algorithm relies on the premise that the unknown target concept is in fact monotone. It is possible to simply apply the learning algorithm without knowing whether the premise holds, and hope that either the algorithm will succeed nonetheless in finding a good hypothesis or detect that the target is not monotone. However, due to the dependence of the complexity of Angluin's algorithm on the number of terms of the target concept's DNF representation, it may be much more efficient to first test whether the function is at all monotone (or close to it).

1.3 The natural monotonicity test

The main result of the paper is that a tester for monotonicity is obtained by repeating the following for $\mathrm{poly}(n/\epsilon)$ many times: Uniformly select a pair of strings at Hamming distance 1 and check if monotonicity is satisfied with respect to the value of f on these two strings. That is,

ALGORITHM 1: On input n, ϵ and oracle access to $f : \{0,1\}^n \mapsto \{0,1\}$, repeat the following steps up to n^3/ϵ times

1. Uniformly select $x \in \{0,1\}^n$ and $i \in \{1, ..., n\}$.
2. Obtain the values of $f(x)$ and $f(y)$, where y results from x by flipping the $i^{\rm th}$ bit.
3. If $x, y, f(x), f(y)$ demonstrate that f is not monotone then reject.

   That is, if either $(x \prec y) \wedge (f(x) > f(y))$ or $(y \prec x) \wedge (f(y) > f(x))$ then reject.

If all iterations were completed without rejecting then accept.

Theorem 1 (main result): *Algorithm 1 is a testing algorithm for monotonicity. Furthermore, if the function is monotone then Algorithm 1 always accepts.*

Theorem 1 asserts that a (random) *local check* (i.e., Step 3 above) can establish the existence of a *global property* (i.e., the distance of f to the set of monotone functions). Actually, Theorem 1 is proven by relating two quantities referring to the above: Given $f : \{0,1\}^n \mapsto \{0,1\}$, we denote by $\delta_{\mathrm{M}}(f)$ the fraction of pairs (x,y) in which Step 3 rejects. Observe that $\delta_{\mathrm{M}}(f)$ is actually a combinatorial quantity (i.e., the fraction of pairs of n-bit strings, differing on one bit, which violate the monotonicity condition). We then define $\epsilon_{\mathrm{M}}(f)$ to be the distance of f from the set of monotone functions (i.e., the minimum over all monotone functions g of $|\{x : f(x) \neq g(x)\}|/2^n$). Observing that Algorithm 1 always accepts a monotone function, Theorem 1 follows from Theorem 2, stated below.

Theorem 2 *For any* $f : \{0,1\}^n \mapsto \{0,1\}$,

$$\delta_{\mathrm{M}}(f) \ \geq \ \frac{\epsilon_{\mathrm{M}}(f)}{n^3}.$$

We comment that a slightly more careful analysis yields a better bound than the one stated in the theorem: namely,

$$\delta_{\mathrm{M}}(f) = \Omega\left(\frac{\epsilon_{\mathrm{M}}(f)}{n^2 \log(1/\epsilon_{\mathrm{M}}(f))}\right) . \tag{1}$$

As for the reverse direction; that is, lower bounding $\epsilon_{\mathrm{M}}(f)$ in terms of $\delta_{\mathrm{M}}(f)$, we have

Proposition 3 *For every function* $f : \{0,1\}^n \mapsto \{0,1\}$, $\epsilon_{\mathrm{M}}(f) \geq \delta_{\mathrm{M}}(f)/2$.

Thus, for every function f

$$\frac{\epsilon_{\mathrm{M}}(f)}{\mathrm{poly}(n)} \leq \delta_{\mathrm{M}}(f) \leq O(\epsilon_{\mathrm{M}}(f))$$

A natural question that arises is that of the exact relation between $\delta_{\mathrm{M}}(\cdot)$ and $\epsilon_{\mathrm{M}}(\cdot)$. We observe that this relation is not simple; that is, it does not depend only on the values of δ_{M} and ϵ_{M}.

Proposition 4 *The following holds for every* n *and every* $2^{-c\cdot n} \leq \alpha \leq \frac{1}{2} - O(\frac{1}{\sqrt{n}})$, *where* c *is any constant strictly smaller than 1.*

1. *There exists a function* $f : \{0,1\}^n \mapsto \{0,1\}$ *such that* $\alpha \leq \epsilon_{\mathrm{M}}(f) \leq 2\alpha$ *and* $\delta_{\mathrm{M}}(f) = \Theta\left(\frac{\epsilon_{\mathrm{M}}(f)}{\sqrt{n}}\right)$.
2. *There exists a function* $f : \{0,1\}^n \mapsto \{0,1\}$ *such that* $\alpha \leq \epsilon_{\mathrm{M}}(f) \leq 2\alpha$ *and* $\delta_{\mathrm{M}}(f) = \Theta\left(\epsilon_{\mathrm{M}}(f)\right)$.
3. *For any* $\alpha = O(n^{-\frac{3}{2}})$, *there exists a function* $f : \{0,1\}^n \mapsto \{0,1\}$ *such that* $\alpha \leq \epsilon_{\mathrm{M}}(f) \leq 2\alpha$ *and* $\delta_{\mathrm{M}}(f) = \Theta\left(\frac{\epsilon_{\mathrm{M}}(f)}{n}\right)$.

PERSPECTIVE. Analogous quantities capturing local and global properties of functions were analyzed in the context of *linearity testing*. For a function $f : \{0,1\}^n \mapsto \{0,1\}$ (as above), one may define $\epsilon_{\mathrm{LIN}}(f)$ to be its distance from the set of linear functions and $\delta_{\mathrm{LIN}}(f)$ to be the fraction of pairs, $(x,y) \in \{0,1\}^n \times \{0,1\}^n$ for which $f(x) + f(y) \neq f(x \oplus y)$. A sequence of works [BLR93, BGLR93, BS94, BCH+95] has demonstrated a fairly complex behavior of the relation between δ_{LIN} and ϵ_{LIN}. The interested reader is referred to [BCH+95].

1.4 Monotonicity testing based on random examples

Algorithm 1 makes essential use of queries. We show that this is no coincidence – any monotonicity tester that utilizes only uniformly and *independently* chosen random examples, must have much higher complexity.

Theorem 5 *For any $\epsilon = O(n^{-3/2})$, any tester for monotonicity that only utilizes random examples must use at least $\Omega(\sqrt{2^n/\epsilon})$ such examples.*

Interestingly, this lower bound is tight up to a $\mathrm{poly}(n)$ factor.

Theorem 6 *There exists a tester for monotonicity which only utilizes random examples and uses at most $O(\sqrt{n^3 \cdot 2^n/\epsilon})$ examples. Furthermore, the algorithm runs in time $\mathrm{poly}(n) \cdot \sqrt{2^n/\epsilon}$.*

We note that the above tester is significantly faster than any learning algorithm for the class of all monotone concepts when the allowed error is $O(1/\sqrt{n})$: Learning (under the uniform distribution) requires $\Omega(2^n/\sqrt{n})$ examples (and even that number of queries) [KLV94].[1]

1.5 Extensions and Open Problems

TESTING UNATENESS. A function $f : \{0,1\}^n \mapsto \{0,1\}$ is said to be unate if for every x_i (where $x = x_1 \ldots x_n$ is the input to the function), exactly one of the following holds: whenever the value of x_i is flipped from 0 to 1 then the value of f does not decrease; *or* whenever the value of x_i is flipped from 1 to 0 then the value of f does not decrease. Thus, unateness is a more general notion than monotonicity. We show that our algorithm can be extended to test whether a Boolean function is unate or far from any unate function. The query and time complexities of the (extended) algorithm are bounded by $O(n^{3.5}/\epsilon)$.

OTHER DOMAINS AND RANGES. Let Σ and Ξ be finite sets, and $<_\Sigma$ and $<_\Xi$ (total) orders on Σ and Ξ, respectively. Then we can extend the notion of monotonicity to functions from Σ^n to Ξ, in the obvious manner: Namely, a function $f : \Sigma^n \mapsto \Xi$ is said to be monotone if $f(x) \leq_\Xi f(y)$ for every $x \prec_\Sigma y$, where $x_1 \cdots x_n \prec_\Sigma y_1 \cdots y_n$ if $x_i \leq_\Sigma y_i$ for every i and $x_i <_\Sigma y_i$ for some i. Our algorithm generalizes to testing monotonicity over extended domains and ranges. The complexity of the generalized algorithm scales quadratically with $|\Sigma|$ and linearly with $|\Xi|$. It is an interesting open problem whether these dependencies can be removed (or reduced). In particular, we believe that the dependence on the size of the range Ξ can be removed.

REMOVING THE DEPENDENCE ON n. Our algorithm (even for the base case), has a polynomial dependence on the dimension of the input, n. As shown in Proposition 4, some dependence of the query complexity on n is unavoidable in the case of our algorithm. However, it is an interesting open problem

[1]The claim follows by considering all possible concepts that contain all instances having $\lfloor n/2 \rfloor + 1$ or more 1's, no instances having $\lfloor n/2 \rfloor - 1$ or less 1's, and any subset of the instances having exactly $\lfloor n/2 \rfloor$ 1's. In contrast, "weak learning" [KV94] is possible in polynomial time. Specifically, the class of monotone concepts can be learned in polynomial time with error at most $1/2 - \Omega(1/\sqrt{n})$ (though no polynomial-time learning algorithm can achieve an error of $1/2 - \omega(\log(n)/\sqrt{n})$) [BBL98].

whether other algorithms may have significantly lower query (and time) complexities, and in particular have query complexity independent of n. A candidate alternative algorithm inspects pairs of strings x, y, where x is chosen uniformly in $\{0,1\}^n$, and y is chosen as follows: First select an index (*weight*) $w \in \{0,\ldots,n\}$ with probability $\binom{n}{w} \cdot 2^{-n}$, and then select y uniformly among the strings having w 1's, and being comparable to x (i.e., $y \prec x$ or $y \succ x$).

Related Work

The "spot-checker for sorting" presented in [EKK$^+$98, Sec. 2.1] implies a tester for monotonicity with respect to functions from any fully ordered domain to any fully ordered range, having query and time complexities that are logarithmic in the size of the domain. We note that this problem corresponds to the special case of $n = 1$ of the extension discussed in Subsection 1.5 (to general domains and ranges).

Organization

Theorem 2 is proved in Section 3. Propositions 3 and 4 are proved in Section 4, and Theorems 5 and 6 are proved in Section 5. The extensions are presented in Section 6.

2 Preliminaries

For any pair of functions $f, g : \{0,1\}^n \to \{0,1\}$, we define the *distance* between f and g, denoted, $\mathrm{dist}(f,g)$, to be the fraction of instances $x \in \{0,1\}^n$ on which $f(x) \neq g(x)$. In other words, $\mathrm{dist}(f,g)$ is the probability over a uniformly chosen x that f and g differ on x. Thus, $\epsilon_{\mathrm{M}}(f)$ as defined in the introduction is the minimum, taken over all monotone functions g of $\mathrm{dist}(f,g)$.

A general formulation of Property Testing was suggested in [GGR96], but here we consider a special case formulated previously in [RS96].

Definition 1 (property tester)*: Let* $\mathrm{P} = \cup_{\mathrm{n}\geq 1}\mathrm{P_n}$ *be a subset* (or a property) *of Boolean functions, so that* P_n *is a subset of the functions mapping* $\{0,1\}^n$ *to* $\{0,1\}$. *A* (property) tester for P *is a probabilistic oracle machine*[2]*,* M*, which given* n*, a* distance parameter $\epsilon > 0$ *and oracle access to an arbitrary function* $f : \{0,1\}^n \mapsto \{0,1\}$ *satisfies the following two conditions:*

1. The tester accepts f if it is in P :

   If $f \in \mathrm{P}_n$ *then* $\mathrm{Prob}(M^f(n,\epsilon)=1) \geq \frac{2}{3}$.

2. The tester rejects f if it is far from P :

   If $\mathrm{dist}(f,g) > \epsilon$ *for every* $g \in \mathrm{P}_n$, *then* $\mathrm{Prob}(M^f(n,\epsilon)=1) < \frac{1}{3}$.

TESTING BASED ON RANDOM EXAMPLES. In case the queries made by the tester are uniformly and *independently* distributed in $\{0,1\}^n$, we say that it only uses examples. Indeed, a more appealing way of looking as such a tester is as an ordinary algorithm (rather than an oracle machine) which is given as input a sequence $(x_1, f(x_1)), (x_2, f(x_2)), \ldots$ where the x_i's are uniformly and independently distributed in $\{0,1\}^n$.

[2] Alternatively, one may consider a RAM model of computation, in which trivial manipulation of domain and range elements (e.g., reading/writing an element and comparing elements) is performed at unit cost.

Definition 2 (the Boolean-Lattice graph)*: For every string $x \in \{0,1\}^n$, let $w(x)$ denote the* weight *of x (i.e., the number of 1's in x). For each i, $0 \le i \le n$, let $\mathrm{L}_i \subset \{0,1\}^n$ denote the set of n-bit strings of weight i (i.e., $\mathrm{L}_i = \{x \in \{0,1\}^n : w(x) = i\}$). Let G_n be the leveled* directed *(acyclic) graph over the vertex set $\{0,1\}^n$, where there is a directed edge from y to x if and only if $x \prec y$ and $w(x) = w(y) - 1$ (i.e., x and y are in adjacent L_i's).*

Given the definition of G_n we may view our algorithm as uniformly selecting *edges* in G_n and querying the function f on their end-points. We call an edge directed from y to x in G_n a *violating edge with respect to f* if $f(x) > f(y)$ (whereas $x \prec y$). Thus, $\delta_{\mathrm{M}}(f)$, as defined in the introduction, is the fraction of violating edges in G_n with respect to f.

3 Proof of the Main Technical Result

In order to prove Theorem 2 we prove the following two lemmas. The first lemma shows the existence of a *matching* between two relatively large (with respect to $\epsilon_{\mathrm{M}}(f)$) sets of vertices (strings) belonging to different layers of G_n where each vertex y in the first set is matched to a vertex x such that $x \prec y$ *but* $f(x) > f(y)$. The second lemma shows that for any such matching there exist vertex disjoint (directed) paths in G_n between the two sets (though the paths may correspond to a different matching — see Appendix A for further discussion).

Lemma 7 (existence of large violating matched sets) *For any function $f : \{0,1\}^n \mapsto \{0,1\}$, there exist two sets of vertices $\mathrm{S} \subseteq \mathrm{L}_s$ and $\mathrm{R} \subseteq \mathrm{L}_r$, where $s > r$, for which the following holds:*

1. *$|\mathrm{S}| = |\mathrm{R}| \ \ge \frac{\epsilon_{\mathrm{M}}(f)}{2n^2} \cdot 2^n$;*
2. *For every $y \in \mathrm{S}$, $f(y) = 0$, and for every $x \in \mathrm{R}$, $f(x) = 1$;*
3. *There exists a one-to-one mapping ϕ from S to R such that for every $y \in \mathrm{S}$, $\phi(y) \prec y$.*

Lemma 8 (existence of disjoint paths between matched sets) *Let r and s be integers satisfying, $0 \le r < s \le n$, and let $\mathrm{S} \subseteq \mathrm{L}_s$ and $\mathrm{R} \subseteq \mathrm{L}_r$ be sets each of size m. Suppose that there exists a 1-to-1 mapping ϕ from S to R such that for every $y \in \mathrm{S}$, there is a directed path in G_n from y to $\phi(y)$. Then there exist m vertex-disjoint directed paths from S to R in G_n.*

We prove the two lemmas in the next two subsections. But first we show that Theorem 2 follows by combining the two lemma.

Proof of Theorem 2: Fixing f we first invoke Lemma 7 to obtain the two matched sets S and R of size at least $m = \frac{\epsilon_{\mathrm{M}}(f)}{2n^2} \cdot 2^n$. By Lemma 8 this matching implies the existence of m vertex disjoint paths from S to R. Consider any such path $z_0 = y, \ldots, z_d = x$, where $y \in \mathrm{S}$, $x \in \mathrm{R}$, and $d = s - r$. Since $z_0 \in \mathrm{S}$, we have $f(z_0) = 0$. On the other hand, since $z_d \in \mathrm{R}$, we have $f(z_d) = 1$. Therefore, there must exist some $\ell \in \{0, ..., d-1\}$, such that $f(z_\ell) = 0$ and $f(z_{\ell+1}) = 1$. Thus the edge directed from z_ℓ to $z_{\ell+1}$ is a violating edge with respect to f. Since the paths from S to R are vertex disjoint, they are necessarily edge disjoint, and hence there are at least $m = \frac{\epsilon_{\mathrm{M}}(f)}{2n^2} \cdot 2^n$ such violating edges (at least one per path). Because each vertex in G_n has total degree (indegree plus outdegree) n, the number of edges in G_n is $\frac{1}{2} \cdot 2^n \cdot n$. Therefore, the fraction of violating edges is at least $\frac{\epsilon_{\mathrm{M}}(f)}{n^3}$, and the theorem follows. ■

The strengthening of Theorem 2 stated in Equation (1) is justified by the fact that one may actually show that there exist sets S and R as in Lemma 7 such that $|\mathrm{S}| = |\mathrm{R}| = \Omega\left(\frac{\epsilon_{\mathrm{M}}(f)}{n \log(1/\epsilon_{\mathrm{M}}(f))}\right) \cdot 2^n$. We show how this improvement can be obtained after we prove Lemma 7.

3.1 Proving the existence of large violating matched sets

Fixing f, let g be a monotone function (over $\{0,1\}^n$) for which $\mathrm{dist}(f,g) = \epsilon_{\mathrm{M}}(f)$. Namely, g is a monotone function that is closest to f. For $b \in \{0,1\}$, let

$$\mathrm{D}_b \stackrel{\mathrm{def}}{=} \{x : \; f(x) \neq g(x) \text{ and } g(x) = b\} \tag{2}$$

That is, the set $\mathrm{D}_0 \cup \mathrm{D}_1$ is a set of minimum size such that if we flip the value of f on all elements in the set then we obtain a monotone function (i.e., g). Since $|\mathrm{D}_0 \cup \mathrm{D}_1| = \epsilon_{\mathrm{M}}(f) \cdot 2^n$ and $D_0 \cap \mathrm{D}_1 = \emptyset$, we may assume, without loss of generality, that $|\mathrm{D}_1| \geq \frac{\epsilon_{\mathrm{M}}(f)}{2} \cdot 2^n$. Recall that, by definition,

$$\mathrm{D}_1 = \{x : \; g(x) = 1 \text{ and } f(x) = 0\} \quad \subseteq \quad \{x : \; f(x) = 0\}$$

For any set $\mathrm{Y} \subseteq \{0,1\}^n$, the shadow[3] of Y, denoted $\sigma(\mathrm{Y})$, is defined as follows:

$$\sigma(\mathrm{Y}) \stackrel{\mathrm{def}}{=} \{x \notin \mathrm{Y} : \; \exists y \in \mathrm{Y} \text{ s.t. } x \prec y\} \tag{3}$$

Namely, the shadow of Y is the set of all strings not in Y that are each smaller than some string in Y. For any $\mathrm{Y} \subseteq \mathrm{D}_1$ define

$$\sigma_1(\mathrm{Y}) \stackrel{\mathrm{def}}{=} \{x \in \sigma(\mathrm{Y}) : \; f(x) = g(x) = 1\} \quad \subseteq \quad \{x : \; f(x) = 1\} \tag{4}$$

Namely, $\sigma_1(\mathrm{Y})$ is the subset of the shadow of Y containing all strings on which both f and g have value 1. (Note that for any $\mathrm{Y} \subseteq \mathrm{D}_1$, $\sigma(\mathrm{Y}) \setminus \sigma_1(\mathrm{Y}) \;\subseteq\; \{x : \; g(x) = 0\}$.) As a visualization (see Figure 3.1), we view g as defining a *boundary* in the Boolean Lattice (similarly, in G_n), such that all strings on and above the boundary are labeled 1, and all other strings are labeled 0. The set D_1 contains those strings above the boundary that f labels 0. The set $\sigma_1(\mathrm{D}_1)$ contains all strings in the shadow of D_1 that lie above the boundary. These strings are labeled 1 by f (as otherwise they would be in D_1).

Thus, by definition of D_1 and $\sigma_1(\mathrm{D}_1)$, we have that for every $x \in \sigma_1(\mathrm{D}_1)$, there exists $y \in \mathrm{D}_1$ such that the pair (x,y) satisfies: $x \prec y$ and $f(y) < f(x)$ (i.e., $f(y) = 0$ and $f(x) = 1$). We next show that a stronger statement holds.

Lemma 9 *For every $\mathrm{Y} \subseteq \mathrm{D}_1$, there exists a 1-to-1 mapping ϕ from Y into $\sigma_1(\mathrm{Y})$, such that for each $y \in \mathrm{Y}$, $\phi(y) \prec y$.*

Lemma 9 is the main step in proving Lemma 7 (which also requires that all elements in the set S belong to the same layer in G_n, and that the same hold for all the elements they are mapped to).

Proof: We first show that for every $\mathrm{Y} \subseteq \mathrm{D}_1$, $|\sigma_1(\mathrm{Y})| \geq |\mathrm{Y}|$. Assume towards contradiction that, for some $\mathrm{Y} \subseteq \mathrm{D}_1$, $|\sigma_1(\mathrm{Y})| < |\mathrm{Y}|$. We show, contrary to our hypothesis on g, that there exists another monotone function g' that is (strictly) closer to f.

Define $\boldsymbol{g'}$ as follows: For every $x \in \mathrm{Y} \cup \sigma(\mathrm{Y})$, $g'(x) = 0$. Otherwise, $g'(x) = g(x)$.

We need to verify the following two claims.

Claim 9.1: *g' is a monotone function.*

Claim 9.2: $\mathrm{dist}(f, g') < \mathrm{dist}(f, g)$.

Proof of Claim 9.1: We need to show that for every x, y such that $x \prec y$, it holds that $g'(x) \leq g'(y)$. Consider the following cases.

[3]This is not the standard definition of a shadow, as in [Bol86, Chap. 5].

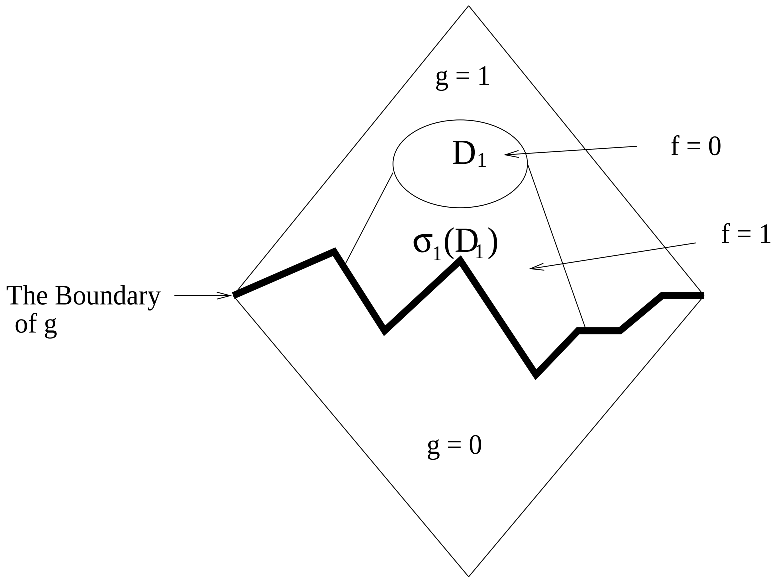

Figure 1: The sets D_1 and $\sigma_1(\mathrm{D}_1)$.

Case 1: $x \in \mathrm{Y} \cup \sigma(\mathrm{Y})$. In this case $g'(x) = 0$, and so $g'(x) \le g'(y)$ for all y,

Case 2: $x \notin \mathrm{Y} \cup \sigma(\mathrm{Y})$. Note that in this case $g'(x) = g(x)$. We will show that for every y if $x \prec y$ then $y \notin \mathrm{Y} \cup \sigma(\mathrm{Y})$ as well, and thus $g'(y) = g(y) \ge g(x) = g'(x)$ as required. Suppose towards contradiction that for some $y \in \mathrm{Y} \cup \sigma(\mathrm{Y})$ it holds that $x \prec y$. We consider two subcases.

1. If $y \in \mathrm{Y}$ then since $x \prec y$ we have that $x \in \mathrm{Y} \cup \sigma(\mathrm{Y})$ in contradiction to the case hypothesis.
2. If $y \in \sigma(\mathrm{Y})$ then there exists $z \in \mathrm{Y}$ such that $y \prec z$. Using $x \prec y$ it follows that $x \prec z$ and so again $x \in \mathrm{Y} \cup \sigma(\mathrm{Y})$ in contradiction to the case hypothesis.

Claim 9.1 follows. $\square$

Proof of Claim 9.2: By definition of g', the functions g and g' differ on the set of strings $\Delta \stackrel{\rm def}{=} (\mathrm{Y} \cup \sigma(\mathrm{Y})) \cap \{x : g(x) = 1\}$. Since $\mathrm{Y} \subseteq \mathrm{D}_1 \subseteq \{x : g(x) = 1\}$, we have

$$\begin{aligned}
\Delta &= \mathrm{Y} \bigcup (\sigma(\mathrm{Y}) \cap \{x : g(x) = 1\}) \\
&= \mathrm{Y} \bigcup (\sigma(\mathrm{Y}) \cap \{x : g(x) = 1 \text{ and } f(x) = 1\}) \bigcup (\sigma(\mathrm{Y}) \cap \{x : g(x) = 1 \text{ and } f(x) = 0\}) \\
&= \mathrm{Y} \bigcup \sigma_1(\mathrm{Y}) \bigcup \mathrm{A}
\end{aligned}$$

where $\mathrm{A} \stackrel{\rm def}{=} \sigma(\mathrm{Y}) \cap \{x : g(x) = 1 \text{ and } f(x) = 0\}$. Consider the three (disjoint) subsets of Δ: Y, $\sigma_1(\mathrm{Y})$, and A.

- For every $x \in \mathrm{Y}$, we have $f(x) = 0$ and $g(x) = 1$ (since $\mathrm{Y} \subseteq \mathrm{D}_1$), and $g'(x) = 0$ (by definition). Such x contributes to $\mathrm{dist}(f, g)$ but not to $\mathrm{dist}(f, g')$.
- For every $x \in \sigma_1(\mathrm{Y})$, we have $f(x) = g(x) = 1$ (by definition of $\sigma_1(\mathrm{Y})$), and again $g'(x) = 0$. Such x do not contribute to $\mathrm{dist}(f, g)$ but do contribute to $\mathrm{dist}(f, g')$.

- For every $x \in \mathrm{A}$, we have $f(x) = 0$ and $g(x) = 1$ (by definition of A), and again $g'(x) = 0$.
  Such x contribute to $\mathrm{dist}(f, g)$ but not to $\mathrm{dist}(f, g')$.

Thus,

$$2^n \cdot (\mathrm{dist}(f, g') - \mathrm{dist}(f, g)) \; = \; |\sigma_1(\mathrm{Y})| - |\mathrm{Y} \cup \mathrm{A}| \; \leq \; |\sigma_1(\mathrm{Y})| - |\mathrm{Y}| \; < \; 0$$

where the strict inequality is due to the assumption that $|\sigma_1(\mathrm{Y})| < |\mathrm{Y}|$. Claim 9.2 follows. $\square$

Consider any set $\mathrm{Y} \subseteq \mathrm{D}_1$. We have established that for every $\mathrm{Y}' \subseteq \mathrm{Y}$, $|\sigma_1(\mathrm{Y}')| \geq |\mathrm{Y}'|$. Lemma 9 follows from Hall's Theorem (*cf.* [Eve79, Thm. 6.12]): Consider the auxiliary bipartite graph B whose vertex set is labeled by the strings in $\mathrm{Y} \cup \sigma_1(\mathrm{Y})$, and whose edge set is $\{(x, y) : \; x \in \sigma_1(\mathrm{Y}), \; y \in \mathrm{Y}, \; x \prec y\}$. By the above, for each $\mathrm{Y}' \subseteq \mathrm{Y}$, we have $|\Gamma(\mathrm{Y}')| \geq |\mathrm{Y}'|$, where $\Gamma(\mathrm{Y}')$ denotes the neighbor set of Y' in B. By Hall's Theorem, this implies that there exists a perfect matching between Y and a subset of $\sigma_1(\mathrm{Y})$. Lemma 9 follows. $\blacksquare$

Proof of Lemma 7: As noted previously, we may assume that D_1 (see Eq. (2)) has size at least $\epsilon_{\mathrm{M}}(f) \cdot 2^{n-1}$ (the case $|\mathrm{D}_0| \geq \epsilon_{\mathrm{M}}(f) \cdot 2^{n-1}$ is analogous). Let $\mathrm{Y}_i \overset{\mathrm{def}}{=} \mathrm{D}_1 \cap \mathrm{L}_i$, and let s denote the index of the largest set among the Y_i's. It follows that $|\mathrm{Y}_s| \geq \frac{\epsilon_{\mathrm{M}}(f)}{2n} \cdot 2^n$.

We now invoke Lemma 9 with $\mathrm{Y} = \mathrm{Y}_s$. Let $\mathrm{X}_s \overset{\mathrm{def}}{=} \phi(\mathrm{Y}_s)$, where ϕ is as guaranteed by the lemma. Hence, $\mathrm{X}_s \subseteq \sigma_1(\mathrm{Y}_s)$, and $|\mathrm{X}_s| = |\mathrm{Y}_s|$. Note that while all elements of Y_s belong to L_s, the elements of X_s are contained in several L_j's, $j < s$. For each j, $0 \leq j < s$, let $\mathrm{X}_{s,j} \overset{\mathrm{def}}{=} \mathrm{X}_s \cap \mathrm{L}_j$. Let $\mathrm{X}_{s,r}$ be the largest such set. Since $|\mathrm{X}_s| = |\mathrm{Y}_s| \geq \frac{\epsilon_{\mathrm{M}}(f)}{2n} \cdot 2^n$, we have $|\mathrm{X}_{s,r}| \geq \frac{\epsilon_{\mathrm{M}}(f)}{2n^2} \cdot 2^n$. Finally, let $\mathrm{Y}_{s,r} \overset{\mathrm{def}}{=} \phi^{-1}(\mathrm{X}_{s,r})$. Then Lemma 7 holds with $\mathrm{S} = \mathrm{Y}_{s,r} \subseteq \mathrm{L}_s$ and $\mathrm{R} = \mathrm{X}_{s,r} \subseteq \mathrm{L}_r$. $\blacksquare$

Comment: To obtain the stronger bound on the sizes of S and R we do the following. Let

$$\mathrm{dev} \overset{\mathrm{def}}{=} \sqrt{\frac{1}{2} n \cdot \ln(8/\epsilon_{\mathrm{M}}(f))} \quad .$$

Then we have that the total number of strings in layers L_i where $i > \frac{n}{2} + \mathrm{dev}$ is at most $\frac{\epsilon_{\mathrm{M}}(f)}{8} \cdot 2^n$. Similarly, the total number of strings in layers L_i where $i < \frac{n}{2} - \mathrm{dev}$ is at most $\frac{\epsilon_{\mathrm{M}}(f)}{8} \cdot 2^n$. Assuming (without loss of generality) that $|\mathrm{D}_1| \geq \frac{\epsilon_{\mathrm{M}}(f)}{2} \cdot 2^n$, we have that the number of strings in D_1 that belong to layers L_i where $i \leq \frac{n}{2} + \mathrm{dev}$ is at least $\frac{3\epsilon_{\mathrm{M}}(f)}{8} \cdot 2^n$. By invoking Lemma 9 on the set

$$\mathrm{Y} = \mathrm{D}_1 \cap \left(\bigcup_{i \leq \frac{n}{2} + \mathrm{dev}} \mathrm{L}_i \right)$$

we have a one-to-one mapping ϕ from Y to $\mathrm{X} = \phi(\mathrm{Y}) \subseteq \sigma_1(\mathrm{Y})$. Note that by definition of Y,

$$\mathrm{X} \subseteq \bigcup_{i < \frac{n}{2} + \mathrm{dev}} \mathrm{L}_i \quad .$$

Since $|\mathrm{X}| = |\mathrm{Y}| \geq \frac{3\epsilon_{\mathrm{M}}(f)}{8} \cdot 2^n$, and the total number of strings in layers L_i where $i < \frac{n}{2} - \mathrm{dev}$ is at most $\frac{\epsilon_{\mathrm{M}}(f)}{8} \cdot 2^n$, we have that

$$\left| \mathrm{X} \cap \left(\bigcup_{i = \frac{n}{2} - \mathrm{dev}}^{\frac{n}{2} + \mathrm{dev}} \mathrm{L}_i \right) \right| \geq \frac{\epsilon_{\mathrm{M}}(f)}{4} \cdot 2^n \quad .$$

For each i, $\frac{n}{2} - \mathrm{dev} \leq i < \frac{n}{2} + \mathrm{dev}$, let $\mathrm{X}_i \stackrel{\mathrm{def}}{=} \mathrm{X} \cap \mathrm{L}_i$ and let X_r be the largest such set. Then $|\mathrm{X}_r| \geq \frac{\epsilon_{\mathrm{M}}(f)}{4 \cdot 2\mathrm{dev}} \cdot 2^n$. Let $\mathrm{Y}_r \stackrel{\mathrm{def}}{=} \phi^{-1}(\mathrm{X}_r)$, and for each i, $\frac{n}{2} - \mathrm{dev} < i \leq \frac{n}{2} + \mathrm{dev}$, define $\mathrm{Y}_{i,r} \stackrel{\mathrm{def}}{=} \mathrm{Y}_r \cap \mathrm{L}_i$. Then there exists a set $\mathrm{Y}_{s,r} \subseteq \mathrm{L}_s$ such that

$$|\mathrm{Y}_{s,r}| \geq \frac{\epsilon_{\mathrm{M}}(f)}{16 \cdot \mathrm{dev}^2} \cdot 2^n = \Omega(\frac{\epsilon_{\mathrm{M}}(f)}{n \cdot \log(1/\epsilon_{\mathrm{M}}(f))}) \cdot 2^n \quad .$$

We then let $\mathrm{S} = \mathrm{Y}_{s,r}$ and $\mathrm{R} = \mathrm{X}_{s,r}$.

3.2 Existence of disjoint paths between matched sets

Let $\mathrm{S} \subseteq \mathrm{L}_s$ and $\mathrm{R} \subseteq \mathrm{L}_r$ be as stated in Lemma 8, and let $d = s - r$. Recall that for each $0 \leq i \leq n$, L_i is the set of all vertices in G_n corresponding to strings with exactly i 1's. We shall prove Lemma 8 by induction on m and d. The base cases, i.e., the case where $m = 1$ and $d \geq 1$, and the case where $d = 1$ and $m \geq 1$, clearly hold. Consider general $m > 1$ and $d > 1$, and assume by induction that the claim holds for every pair m' and d' such that either $m' < m$ and $d' \leq d$ or $m' \leq m$ and $d' < d$. Let Q be the set of vertices in L_{s-1} that are on a directed path going from some vertex in S to some vertex in R, and let P be the set of vertices in L_{r+1} that are on such directed paths from S to R (see Figure 3.2). We shall prove the induction claim in two steps. In the first step we use the induction hypothesis (for $m' < m$ and $d' = d$) to show that either $|\mathrm{Q}| \geq m$ or $|\mathrm{P}| \geq m$ (or both). In the second step we use this fact together with the induction hypothesis (for $m' < m$ and $d' = d$ *and* for $m' = m$ and $d' < d$) to prove the induction claim.

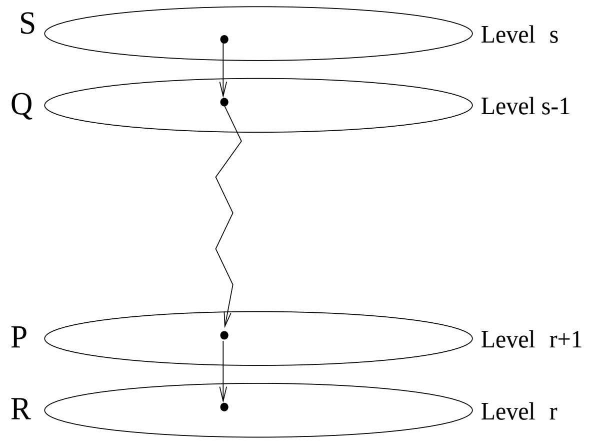

Figure 2: The sets S, R, Q, and P.

Step 1: Either $|\mathrm{Q}| \geq m$ or $|\mathrm{P}| \geq m$.

Proof: Consider the subgraph G'_n of G_n containing S, R and all vertices and edges that belong to paths between S and R.

Claim 8.1: *Let v be a vertex in S and let u be a vertex in some level L_i, where $r + 1 \leq i \leq s - 1$, such that there is a directed path from v to u in G'_n. Then the outdegree of v in G'_n is at least as large as the outdegree of u in G'_n. Similarly, if $w \in \mathrm{R}$ and $z \in \mathrm{L}_i$ where $r + 1 \leq i \leq s - 1$, such that there is a* *directed path from z to w in G'_n, then the indegree of w in G'_n is at least as large as the indegree of z in G'_n.*

Proof: We prove the claim concerning outdegrees. The claim about indegrees is proved analogously. Let $a \geq 1$ be the outdegree of u and consider the vertices $u^1, \ldots, u^a$ in L_{i-1} such that there is an edge in G'_n from u to each u^i. Recall that by definition of G'_n there are paths in G'_n from the u^i's to vertices in R. Therefore, any vertex that is on a path from v to one of the u^i's is in G'_n as well.

For each u^i, let $b^i \in [n]$ be the index of the bit on which the strings corresponding to u and u^i, differ; i.e., $u_{b^i} = 1$ while $u^i_{b^i} = 0$. By definition, $b^1, \ldots, b^a$ are distinct indices, and since $u_{b^i} = 1$ for every i, it also holds that $v_{b^i} = 1$ for every i. For each b^i, let v^i be the vertex in L_{s-1} that differs from v on the b^i'th bit; i.e., $v^i_{b^i} = 0$, and for every $j \neq b^i$, $v^i_j = v^i_j$. Then each of the a v^i's is on a path from v to u^i, and the claim follows. □

We note that the above claim can be strengthened to show that the outdegree of v (respectively, indegree of w) is greater than the outdegree of u (respectively, indegree of z), by at least $s - i$ (respectively, $i - r$). This is done by taking into account the bits on which v and u (respectively, w and z) differ.

Let k be the maximum outdegree of vertices in S, and let t be the maximum indegree of vertices in R. We partition S, R, Q, and P into subsets according to their degrees in G'_n as follows. For every $i \leq k$ we let S_i be the subset of vertices in S that have outdegree exactly i, and for every $j \leq t$, we let R^i be the subset of vertices in R that have indegree exactly j. Similarly we let Q^j_i (respectively, P^j_i) be the subset of vertices in Q (respectively, P) with outdegree exactly i and indegree exactly j. First note that by Claim 8.1, the maximum outdegree of vertices in Q and P is at most k, and the maximum indegree is at most t. Therefore, for every j and $i > k$, $|\mathrm{Q}^j_i|, |\mathrm{P}^j_i| = 0$, and for every i and $j > t$, $|\mathrm{Q}^j_i|, |\mathrm{P}^j_i| = 0$.

Furthermore, by Claim 8.1, for every i, and each vertex $v \in \mathrm{S}_i$, the vertices u in P such that there exists a directed path from v to u must belong to $\cup_{i' \leq i} \cup_j \mathrm{P}_{i',j}$. For any $q \leq k$, let $\mathrm{S}_{\leq q} = \cup_{i=1}^{q} \mathrm{S}_i$. By definition of k (as the maximum degree of vertices in S), the set S_k is nonempty and hence for every $q < k$, $|\mathrm{S}_{\leq q}| < m$. Therefore, we can apply the induction hypothesis and obtain that there exist vertex disjoint paths between $\mathrm{S}_{\leq q}$ and $\phi(\mathrm{S}_{\leq q})$ (where ϕ is the matching guaranteed by the hypothesis of Lemma 8). For any $q < k$ let $\Pi(\mathrm{S}_{\leq q}) \subseteq Q$ denote the set of neighbors of vertices in $\mathrm{S}_{\leq q}$ that lie on these paths to $\phi(\mathrm{S}_{\leq q})$. Since these paths are disjoint, $|\Pi(\mathrm{S}_{\leq q})| = |\mathrm{S}_{\leq q}|$. Using the above and the fact that the S_i's are disjoint and the P^j_i's are disjoint, the following inequality holds for every $q < k$:

$$\sum_{i=1}^{q} |\mathrm{S}_i| \;=\; |\mathrm{S}_{\leq q}| \;=\; |\Pi(\mathrm{S}_{\leq q})| \;\leq\; \left|\cup_{i=1}^{q} \cup_{j=1}^{\infty} \mathrm{P}^j_i\right| \;=\; \sum_{i=1}^{q}\sum_{j=1}^{\infty} |\mathrm{P}^j_i| \;=\; \sum_{i=1}^{q}\sum_{j=1}^{k} |\mathrm{P}^j_i| \,. \tag{5}$$

Similarly, we can obtain that for every $p < s$,

$$\sum_{j=1}^{p} |\mathrm{R}^j| \;\leq\; \sum_{i=1}^{k}\sum_{j=1}^{p} |\mathrm{Q}^j_i| \,. \tag{6}$$

Recall that we would like to show that either $|\mathrm{Q}| \geq m$ or $|\mathrm{P}| \geq m$. Thus, assume in contradiction that both $|\mathrm{Q}| < m$ and $|\mathrm{P}| < m$. Therefore, by Equation (5), for every $q < k$,

$$\sum_{i=q+1}^{k} |\mathrm{S}_i| \;=\; |\mathrm{S}| - \sum_{i=1}^{q} |\mathrm{S}_i| \;=\; m - \sum_{i=1}^{q} |\mathrm{S}_i| \;\geq\; m - \sum_{i=1}^{q}\sum_{j=1}^{t} |\mathrm{P}^j_i| \;>\; |\mathrm{P}| - \sum_{i=1}^{q}\sum_{j=1}^{t} |\mathrm{P}^j_i| \tag{7}$$

and so

$$\sum_{i=q+1}^{k} |\mathrm{S}_i| \;\; > \;\; \sum_{i=q+1}^{k} \sum_{j=1}^{t} |\mathrm{P}_i^j| \, . \tag{8}$$

Similarly, for every $p < t$,

$$\sum_{j=p+1}^{k} |\mathrm{R}^j| \;\; > \;\; \sum_{i=1}^{k} \sum_{j=p+1}^{t} |\mathrm{Q}_i^j| \, . \tag{9}$$

By summing both sides of Equation (8) over all $q < k$ we get

$$\sum_{q=0}^{k-1} \sum_{i=q+1}^{k} |\mathrm{S}_i| > \sum_{q=0}^{k-1} \sum_{i=q+1}^{k} \sum_{j=1}^{t} |\mathrm{P}_i^j| \tag{10}$$

or equivalently,

$$\sum_{i=1}^{k} i \cdot |\mathrm{S}_i| > \sum_{i=1}^{k} \sum_{j=1}^{t} i \cdot |\mathrm{P}_i^j| \, . \tag{11}$$

Similarly, from Equation (9) we get

$$\sum_{j=1}^{t} j \cdot |\mathrm{R}^j| > \sum_{i=1}^{k} \sum_{j=1}^{t} j \cdot |\mathrm{Q}_i^j| \, . \tag{12}$$

Summing Equations (11) and (12), we get

$$\sum_{i=1}^{k} i \cdot |\mathrm{S}_i| + \sum_{j=1}^{t} j \cdot |\mathrm{R}^j| > \sum_{i=1}^{k} \sum_{j=1}^{t} i \cdot |\mathrm{P}_i^j| + \sum_{i=1}^{k} \sum_{j=1}^{t} j \cdot |\mathrm{Q}_i^j| \, . \tag{13}$$

However, since the number of edges going out of vertices in S equals the number of edges entering vertices in Q we have that:

$$\sum_{i=1}^{k} i \cdot |\mathrm{S}_i| \; = \; \sum_{i=1}^{k} \sum_{j=1}^{t} j \cdot |\mathrm{Q}_i^j| \tag{14}$$

and similarly for R and P we have

$$\sum_{j=1}^{t} j \cdot |\mathrm{R}^j| \; = \; \sum_{i=1}^{k} \sum_{j=1}^{t} i \cdot |\mathrm{P}_i^j| \, . \tag{15}$$

Summing Equations (14) and (15) we get

$$\sum_{i=1}^{k} i \cdot |\mathrm{S}_i| + \sum_{j=1}^{t} j \cdot |\mathrm{R}^j| \; = \; \sum_{i=1}^{k} \sum_{j=1}^{t} j \cdot |\mathrm{Q}_i^j| + \sum_{i=1}^{k} \sum_{j=1}^{t} i \cdot |\mathrm{P}_i^j| \tag{16}$$

contradicting Equation (13). ■ (Step 1.)

Step 2: There exist vertex disjoint paths from S to R.

Proof: From Step 1 we have that either $|\mathrm{Q}| \geq m$ or $|\mathrm{P}| \geq m$. Assume the former is true — we shall see that this can be done without loss of generality. We next show that (1) there exists a perfect matching between S and (a subset of) Q; and (2) there exists a 1-to-1 mapping ϕ' from the matched vertices of Q to R so that there is a path from each matched $u \in \mathrm{Q}$ to $\phi'(u)$. Given (2) we can apply the induction hypothesis for $d' = d-1$ (and $m' = m$) on Q and R, and by combining with (1) we get the desired paths from S to R.

We actually prove both (1) and (2) together. Consider the following auxiliary network, A. It has a single source vertex s, a single target vertex t, and the rest of the vertices are partitioned into three layers corresponding to S, Q and R, respectively. There is an edge from s to each of the vertices in S, and from each of the vertices in R to t. The edges between S and Q are as in G'_n and edges between Q and R correspond to directed paths in G'_n. We show that the minimum $s-t$ vertex-separator in A has size m. Items (1) and (2) follow by one of the variations of Menger's Theorem (see [Eve79, Thm. 6.4 and discussion on pp. 130]), which guarantees the existence of m vertex-disjoint paths from s to t.

Assume in contradiction that there exists a vertex-separator C of size smaller than m in A. Let $m_1 \stackrel{\mathrm{def}}{=} |\mathrm{C} \cap \mathrm{S}|$, $m_2 \stackrel{\mathrm{def}}{=} |\mathrm{C} \cap \mathrm{Q}|$, and $m_3 \stackrel{\mathrm{def}}{=} |\mathrm{C} \cap \mathrm{R}|$. Consider the subset of vertices $\mathrm{S}' \subseteq \mathrm{S}$ that do not belong to C and are not mapped by ϕ to vertices in $\mathrm{R} \cap \mathrm{C}$. The size of S' is at least $m' = m - (m_1 + m_3) > |\mathrm{C}| - (m_1 + m_3) = m_2$. Let $\mathrm{R}' \stackrel{\mathrm{def}}{=} \phi(\mathrm{S}')$, and Q' be the subset of vertices in Q that are on a directed path in G'_n going from some vertex in S' to a vertex in R'.

We consider two cases. If $\mathrm{S}' = \mathrm{S}$ (i.e., $\mathrm{C} \subseteq \mathrm{Q}$) then $\mathrm{Q}' = \mathrm{Q}$, and since $|\mathrm{C}| < m \leq |\mathrm{Q}|$, there exists at least one vertex in $\mathrm{Q} \setminus \mathrm{C}$ on a path from a vertex in S to a vertex in R, contradicting the assumption that C is a vertex separator. If $\mathrm{S}' \subset \mathrm{S}$, then by the induction hypothesis (for $m' = |\mathrm{S}'| < m$ and $d' = d$), there exist vertex disjoint paths in G'_n from S' to $\phi(\mathrm{S}')$ and hence necessarily $|\mathrm{Q}'| \geq |\mathrm{S}'| > m_2$. Since $|\mathrm{C} \cap \mathrm{Q}| = m_2$, we again reach contradiction to the assumption that C is a vertex separator. ■

4 Proofs of Propositions 3 and 4

Below we restate and prove the propositions concerning the relations between $\epsilon_{\mathrm{M}}(f)$ and $\delta_{\mathrm{M}}(f)$ that were stated in the introduction.

Proposition 3 *For every function* $f : \{0,1\}^n \mapsto \{0,1\}$, $\epsilon_{\mathrm{M}}(f) \geq \delta_{\mathrm{M}}(f)/2$.

Proof: Let us fix f and consider the set E of its violating edges. In order to make f monotone, we must modify the value of f on at least one end-point of each of its violating edges. Since each vertex (string) is incident to at most n violating edges, the number of strings whose value must be modified is at least

$$\frac{|\mathrm{E}|}{n} = \frac{\delta_{\mathrm{M}}(f) \cdot \left(\frac{1}{2} \cdot 2^n n\right)}{n} = \frac{\delta_{\mathrm{M}}(f)}{2} \cdot 2^n$$

and the proposition follows. ■

Comment: taking into account the fact that the number of violating edges incident to a vertex is at most the maximum between its indegree and outdegree and that for most vertices this maximum values is roughly $n/2$, the above bound can be improved to yield $\epsilon_{\mathrm{M}} \geq (1 - o(1)) \cdot \delta_{\mathrm{M}}(f)$, provided $\delta_{\mathrm{M}}(f) \geq 2^{-cn}$ for any constant $c < 1$.

Proposition 4 *The following holds for every* n *and every* $2^{-c \cdot n} \leq \alpha \leq \frac{1}{2} - O(\frac{1}{\sqrt{n}})$, *where* c *is any constant strictly smaller than 1.*

1. *There exists a function $f : \{0,1\}^n \mapsto \{0,1\}$ such that $\alpha \leq \epsilon_{\mathrm{M}}(f) \leq 2\alpha$ and*
$$\delta_{\mathrm{M}}(f) = \Theta\left(\frac{\epsilon_{\mathrm{M}}(f)}{\sqrt{n}}\right) \ .$$

2. *There exists a function $f : \{0,1\}^n \mapsto \{0,1\}$ such that $\alpha \leq \epsilon_{\mathrm{M}}(f) \leq 2\alpha$ and*
$$\delta_{\mathrm{M}}(f) = \Theta\left(\epsilon_{\mathrm{M}}(f)\right) \ .$$

3. *For any $\alpha = O(n^{-\frac{3}{2}})$, there exists a function $f : \{0,1\}^n \mapsto \{0,1\}$ such that $\alpha \leq \epsilon_{\mathrm{M}}(f) \leq 2\alpha$ and*
$$\delta_{\mathrm{M}}(f) = \Theta\left(\frac{\epsilon_{\mathrm{M}}(f)}{n}\right) \ .$$

Proof:

Items 1 and 2. We start by proving the first two items for the case where $\alpha = \frac{1}{2} - O(\frac{1}{\sqrt{n}})$.

1. Let f be the (symmetric) function that has value 0 on all vertices belonging to layers L_j where $j \geq \frac{n}{2}$ and is 1 on all vertices belonging to layers L_i where $i < \frac{n}{2}$. Then on one hand, all edges between the layers, $\mathrm{L}_{\lceil\frac{n}{2}\rceil}$ and $\mathrm{L}_{\lceil\frac{n}{2}\rceil-1}$ are violating edges, and so $\delta_{\mathrm{M}}(f) = \Theta(\frac{1}{\sqrt{n}})$. On the other hand, we next show that $\epsilon_{\mathrm{M}}(f) = \frac{1}{2} - O(\frac{1}{\sqrt{n}})$. Clearly, $\epsilon_{\mathrm{M}} \leq \frac{1}{2}$ as the all 0 function is monotone and at distance at most $\frac{1}{2}$ from f. It remains to show that we cannot do better.

   To this end we show the existence of a one-to-one mapping ψ between the vertices in the layers L_i where $i > \frac{n}{2}$ and the vertices in the layers L_i where $i < \frac{n}{2}$ so that for every x, $\psi(x) \prec x$. In particular for each i, $0 \leq i < \frac{n}{2}$, there exists such a one-to-one mapping between L_{n-i} and L_i: Consider the auxiliary bipartite graph over vertex sets L_{n-i} and L_i, where there is an edge between $y \in \mathrm{L}_{n-i}$ and $x \in \mathrm{L}_i$ if an only if $x \prec y$. Since this auxiliary graph is a regular bipartite graph (with degree $\binom{n-i}{i}$), where both sides are of the same size, there exists a perfect matching between the two sides. We let ψ be defined by such $\lceil\frac{n}{2}\rceil - 1$ perfect matchings, where for odd n all strings in $\{0,1\}^n$ are matched, and for even n only the strings in the middle layer, $\mathrm{L}_{\frac{n}{2}}$, are left unmatched. To make f monotone, we must modify the value of at least one vertex in each matched pair, and since these pairs are disjoint (and their number is at least $(1 - O(\frac{1}{\sqrt{n}}) \cdot 2^n)$, the claim follows.

2. Let f be the (symmetric) function that has value 0 on all vertices belonging to layers L_i where i is even, and has value 1 on all vertices belonging to layers L_i where i is odd. Since all edges going from even layers to odd layers are violating edges, $\delta_{\mathrm{M}}(f) = 1/2$. We next show that $\epsilon_{\mathrm{M}}(f) \geq \frac{1}{2} - O(\frac{1}{\sqrt{n}})$ (where once again, $\epsilon_{\mathrm{M}}(f) \leq \frac{1}{2}$ since it is at distance at most $1/2$ from the all-0 function or the all-1 function). Consider any pair of adjacent layers such that the top layer is labeled 0 (so that all edges between the two layers are violating edges). It can be shown (*cf.* [Bol86, Chap. 2, Cor. 4]) using Hall's Theorem, that for any such pair of adjacent layers, there exists a perfect matching between the smallest among the two layers and a subset of the larger layer. Since we must modify the value of at least one end-point of each violating edge, the claim follows.

To generalize the above two constructions for smaller α we do the following. Let $n' = n - \lfloor \log(1/(2\alpha)) \rfloor$, and consider the set S of all strings whose last $n - n'$ bits are set to 0 (thus forming a *sub-cube* of the n-dimensional cube). The size of the set S is at least $2\alpha \cdot 2^n$ and at most $4\alpha \cdot 2^n$.

1. Let f' be defined on S analogously to the way f is defined on $\{0,1\}^n$ in Item 1 above (i.e., it has value 0 on all strings in S having weight at least $\frac{n'}{2}$ and is 1 on all strings having weight less than $\frac{n'}{2}$). On all strings not in S, the function f' has value 1. By this definition, there are no violating edges (w.r.t. f') between vertices not in S and vertices in S, and the only violating edges are between the middle two layers of the subgraph of G_n induced by S. The number of these edges is $\Theta(\frac{|\mathrm{S}|}{\sqrt{n'}} \cdot \frac{n'}{2})$, which by our assumption on α (and the definition of n') is $\Theta(\frac{\alpha}{\sqrt{n'}} \cdot n \cdot 2^n)$. On the other hand, as argued in the first item above, we can show that $\epsilon_{\mathrm{M}}(f')$ is approximately $\frac{1}{2} \cdot \frac{|\mathrm{S}|}{2^n}$, which ranges between α and 2α as required.

2. Here too f' has value 1 on all strings not in S, and is defined on S analogously to the way f is defined on $\{0,1\}^n$ in Item 2 above, alternating between 0 and 1 on the layers of the subgraph of G_n induced by S. The rest of the argument follows as in Item 2 when restricting the attention to this subgraph.

Item 3. We start by proving the case $\alpha = \Omega(n^{-3/2})$. We consider the vertices in L_k and L_{k-1}, where $k = \lceil \frac{n}{2} \rceil$. We know that $|\mathrm{L}_k|, |\mathrm{L}_{k-1}| = \Omega(n^{-1/2} \cdot 2^n)$. As noted in the proof of Item 2, between any two adjacent layers there exists a matching whose size equals the size of the smaller among the two layers. Let such a matching, between L_k and L_{k-1}, be denoted $\mathrm{M} = \{((v_i, u_i)\}_{i=1}^{t}$, where $t = |\mathrm{L}_{k-1}|$. Using a greedy approach, we find a large matching $\mathrm{M}' = \{(v_{i_j}, u_{i_j})\} \subset \mathrm{M}$ such that there are no edges (in G_n) between pairs v_{i_j} and u_{i_ℓ} such that $i_j \neq i_\ell$. Since each edge $(v_{i_j}, u_{i_j}) \in \mathrm{M}'$ "rules out" at most $(k-1) + (n-(k-1)-1) < n$ other edges in M (i.e., an edge (v_{i_ℓ}, u_{i_ℓ}) is ruled out if either (v_{i_j}, u_{i_ℓ}) or (v_{i_ℓ}, u_{i_j}) is an edge in G_n), we can obtain $|\mathrm{M}'| \geq \frac{t}{n} = \Omega(n^{-3/2} \cdot 2^n)$. Since we can always drop edges from M', we can have $|\mathrm{M}'| = \Theta(n^{-3/2} \cdot 2^n)$.

Using M' we define f as follows. For each matched pair (v_{i_j}, u_{i_j}) in M', the function f has value 0 on v_{i_j}, and value 1 on u_{i_j}. All other vertices in layers k and higher have value 1, and those in layers $k-1$ and lower have value 0. Hence, the violating edges with respect to f are only those that belong to M', and so $\delta_{\mathrm{M}}(f) = \frac{|\mathrm{M}'|}{2^n \cdot n/2} = \Theta(n^{-1/2})$. On the other hand, $\epsilon_{\mathrm{M}}(f) = \frac{|\mathrm{M}'|}{2^n} = \Theta(n^{-3/2})$ (as in order to make f monotone we must modify the value of at least one end-point of each edge in M'). For smaller values of α we simply define f based on a subset of M' of size $\lceil \alpha \cdot 2^n \rceil$. ■

5 Testing based on Random Examples

In this section we prove Theorems 5 and 6: establishing a lower bound on the sample complexity of such testers and a matching algorithm, respectively. For convenience, we first restate the theorems.

Theorem 5 *For any $\epsilon = O(n^{-3/2})$, any tester for monotonicity which only utilizes random examples must use at least $\Omega(\sqrt{2^n/\epsilon})$ such examples.*

Theorem 6 *There exists a tester for monotonicity which only utilizes random examples and uses at most $O(\sqrt{n^3 \cdot 2^n/\epsilon})$ examples. Furthermore, the algorithm runs in time $\mathrm{poly}(n) \cdot \sqrt{2^n/\epsilon}$.*

5.1 A Lower bound on sample complexity

Let M' be as defined in the proof of Item 3 in Proposition 4. By possibly dropping edges from M' we can obtain a matching M'' so that $|\mathrm{M}''|$ is even and of size $2\epsilon \cdot 2^n$ (recall that $\epsilon = O(n^{-3/2})$). Using M'' we define two families of functions. A function in each of the two families is determined by a partition of M'' into two sets, A and B, of equal size.

1. A function f in the first family is defined as follows
   - For every $(v, u) \in A$, define $f(v) = 1$ and $f(u) = 0$.
   - For every $(v, u) \in B$, define $f(v) = 0$ and $f(u) = 1$.
   - For x with $w(x) \geq k$, for which f has not been defined, define $f(x) = 1$.
   - For x with $w(x) \leq k - 1$, for which f has not been defined, define $f(x) = 0$.
2. A function f in the second family is defined as follows
   - For every $(v, u) \in A$, define $f(v) = 1$ and $f(u) = 1$.
   - For every $(v, u) \in B$, define $f(v) = 0$ and $f(u) = 0$.
   - For x's on which f has not been defined, define $f(x)$ as in the first family.

It is easy to see that every function in the second family is monotone, whereas for every function f in the first family $\epsilon_{\mathrm{M}}(f) = |B|/2^n = \epsilon$. Theorem 5 is established by showing that an algorithm which obtains $o(\sqrt{|B|})$ random examples cannot distinguish a function uniformly selected in the first family (which needs to be rejected with probability at least $2/3$) from a function uniformly selected in the second family (which needs to be accepted with probability at least $2/3$). That is, we show that the statistical distance between two such samples is too small.

Claim 10 *The statistical difference between the distributions induced by the following two random processes is bounded above by* $\binom{m}{2} \cdot \frac{|\mathrm{M}''|}{2^{2n}}$. *The* first process (resp., second process) *is define as follows*

- *Uniformly select a function f in the* first (resp., second) *family.*
- *Uniformly and independently select m strings, $x_1, ..., x_m$, in $\{0,1\}^n$.*
- *Output $(x_1, f(x_1)), ..., (x_m, f(x_m))$.*

Proof: The randomness in both processes amounts to the choice of B (uniform among all $(|\mathrm{M}''|/2)$-subsets of M'') and the uniform choice of the sequence of x_i's. The processes differ only in the labelings of the x_i's which are matched by M'', yet for u (resp., v) so that $(u, v) \in \mathrm{M}''$ the label of u (resp., v) is uniformly distributed in both processes. The statistical difference is due merely to the case in which for some i, j the pair (x_i, x_j) resides in M''. The probability of this event is bounded by $\binom{m}{2}$ times the probability that a specific pair (x_i, x_j) resides in M''. The latter probability equals $\frac{|\mathrm{M}''|}{2^n} \cdot 2^{-n}$. $\square$

Conclusion. By the above claim, $m < 2^n/\sqrt{3|\mathrm{M}''|}$ implies that the statistical difference between these processes is less than $\frac{m^2}{2} \cdot \frac{|\mathrm{M}''|}{2^{2n}} < 1/6$ and thus an algorithm utilizing m queries will fail to work for the parameter $\epsilon = |B|/2^n$. Theorem 5 follows. ■

5.2 A matching algorithm

The algorithm consists of merely emulating Algorithm 1. That is, the algorithm is given $m \stackrel{\text{def}}{=} O(\sqrt{n^3 \cdot 2^n/\epsilon})$ uniformly selected examples and tries to find a violating pair as in Step 3 of Algorithm 1.

ALGORITHM 2: Input n, ϵ and $(x_1, f(x_1)), \ldots, (x_m, f(x_m))$.

1. Place all $(x_j, f(x_j))$'s on a heap arranged according to any ordering on $\{0,1\}^n$.
2. For $j = 1, \ldots, m$ and $i = 1, \ldots, n$, try to retrieve from the heap the value $y \stackrel{\text{def}}{=} x_j \oplus 0^{i-1}10^{n-i}$. If successful then consider the values $x_j, y, f(x_j), f(y)$ and in case they demonstrate that f is not monotone then reject.

If all iterations were completed without rejecting then accept.

ANALYSIS. Clearly, Algorithm 2 always accepts a monotone function, and can be implemented in time $\text{poly}(n) \cdot m$. Using a Birthday Paradox argument, we show that for a suitable choice of m, Algorithm 2 indeed rejects ϵ-far from monotone functions with high probability. We merely need to show the following.

Lemma 11 *There exists a constant c so that the following holds. If $m \geq \sqrt{cn^3 2^n/\epsilon_{\mathrm{M}}(f)}$ and if the x_i's are uniformly and independently selected in $\{0,1\}^n$ then Algorithm 2 rejects the function f with probability at least $2/3$.*

Proof: We use the fact that the proof of Theorem 2 provides two disjoint sets, V and U, with the following properties

1. Each set has size at least $\frac{\epsilon_{\mathrm{M}}(f)}{2n^3} \cdot 2^n$.
2. There is 1-1 mapping, ψ, of V to U.
3. For every $v \in \mathrm{V}$ it holds that $f(v) = 0$, $f(\psi(v)) = 1$, and $\psi(v)$ is obtained from v by setting a single bit to 0.

We will show that with probability at least $2/3$, there exist i and j so that $x_i \in \mathrm{V}$ and $x_j = \psi(x_i) \in \mathrm{U}$, and the lemma will follow.

We split the sample into two equal parts. Using a Multiplicative Chernoff Bound,[4] with probability at least 0.9 the number of x_i's in the first part which hit V is at least $\frac{1}{2} \cdot \frac{m}{2} \cdot \frac{|\mathrm{V}|}{2^n}$. Denote the set of examples hitting V by V', and consider the set $\mathrm{U}' \subseteq \mathrm{U}$ of vertices which are matched by ψ to V'. Then, the probability that none of the $m/2$ examples in the second part hits U' is at most

$$\left(1 - \frac{|\mathrm{U}'|}{2^n}\right)^{m/2} = \left(1 - \frac{|\mathrm{V}'|}{2^n}\right)^{m/2} < \exp\left(-\frac{|\mathrm{V}'|}{2^n} \cdot \frac{m}{2}\right) \leq \exp\left(-\frac{|\mathrm{V}|}{4 \cdot 2^{2n}} \cdot m^2\right) \tag{17}$$

The lemma follows by substituting $|\mathrm{V}|$ with $\frac{\epsilon_{\mathrm{M}}(f)}{2n^3} \cdot 2^n$ and m^2 with $cn^3 2^n/\epsilon_{\mathrm{M}}(f)$. $\Box$

[4]We assume for simplicity that $\epsilon_{\mathrm{M}}(f) \gg n^3/2^n$, which implies $m \gg n^3/\epsilon_{\mathrm{M}}(f)$. Otherwise, $\epsilon_{\mathrm{M}}(f) = O(n^3/2^n)$, in which case $m = \Omega(\sqrt{n^3 2^n/\epsilon_{\mathrm{M}}(f)}) = \Omega(2^n)$, which in turn suffices to hit even a single edge with constant probability.

6 Extensions

6.1 Testing whether a function is unate

By our definition of monotonicity (used throughout the paper), a function is said to be monotone if, for any string, flipping any bit of the string from 0 to 1, does not decrease the value of the function. A more general notion is that of *unate* functions. A function f is *unate* if there exists a string $\zeta = \zeta_1 \ldots \zeta_n \in \{0,1\}^n$ for which the following holds: For any string $x = x_1 \ldots x_n$, and for any i such that $x_i = \zeta_i$, if we let $y = x_1, \ldots, x_{i-1}, \neg x_i, x_{i+1}, \ldots, x_n$ (i.e., y is the same as x except for the i^{th} bit, which is flipped from ζ_i to $\neg\zeta_i$), then $f(y) \geq f(x)$. We say in such a case the f is *monotone with respect to* ζ. In particular, if a function is monotone with respect to the all-0 string, then we simply say that it is a monotone function, and if a function is monotone with respect to *some* ζ, then it is unate. Thus, the generalization of monotonicity to unateness allows that for each position there be a (possibly different) *direction* (i.e., not necessarily the $0 \rightarrow 1$ direction), such that the value of the function cannot decrease when the bit is flipped in that direction.

Similarly to Algorithm 1 (for testing monotonicity), which searches for evidence to non-monotonicity, the testing algorithm for unateness tries to find evidence to non-unateness. However, here it does not suffice to find a pair of strings x, y that differ on the i^{th} bit such that $x < y$ while $f(x) > f(y)$, since f could be monotone with respect to ζ such that $\zeta_i = 1$. Instead we search for *two pairs* of strings, $x^1 < y^1$ and $x^2 < y^2$, where each pair differs on the (same) i^{th} bit, such that $f(x^1) > f(y^1)$ and $f(x^2) < f(y^2)$ (or vica versa). This implies that there is no ζ such that f is monotone with respect to ζ (since, in particular, ζ_i can be neither 0 nor 1).

ALGORITHM 3 (TESTING UNATENESS): On input n, ϵ and oracle access to $f : \{0,1\}^n \mapsto \{0,1\}$, do the following:

1. Uniformly select $m = O(n^{3.5}/\epsilon)$ strings in $\{0,1\}^n$, denoted $x^1, \ldots, x^m$, and m indices in $\{1, \ldots, n\}$. denoted $i^1, \ldots, i^m$.
2. For each selected x^j, obtain the values of $f(x^j)$ and $f(y^j)$, where y^j results from x^j by flipping the i^j'th bit.
3. If unateness is found to be violated then reject.

   Violation occurs, if among the string-pairs $\{x^j, y^j\}$, there exist two pairs and an index i, such that in both pairs the strings differ on the i^{th} bit, but in one pair the value of the function increases when the bit is flipped for 0 to 1, and in the other pair the value of the function increases when the bit is flipped from 1 to 0.

If no contradiction to unateness was found then accept.

Theorem 12 *Algorithm 3 is a testing algorithm for unateness. Furthermore, if the function is unate, then Algorithm 3 always accepts.*

We shall need the following notation. For $\zeta \in \{0,1\}^n$, let $\prec_\zeta$ denote the partial order on strings with respect to ζ. Namely, $x \prec_\zeta y$ if and only if $x \oplus \zeta \prec y \oplus \zeta$. Let $\epsilon_{\mathrm{M},\zeta}(f)$ denote the minimum distance between f and any function g that is monotone with respect to ζ, and let $\delta_{\mathrm{M},\zeta}(f)$ denote the fraction of pairs x, y that differ on a single bit such that $x \prec_\zeta y$ but $f(x) > f(y)$. It follows from the above definitions that for any f and ζ, $\epsilon_{\mathrm{M},\zeta}(f) = \epsilon_{\mathrm{M}}(f_\zeta)$ and $\delta_{\mathrm{M},\zeta}(f) = \delta_{\mathrm{M}}(f_\zeta)$, where f_ζ is defined by $f_\zeta(x) = f(x \oplus \zeta)$. Hence, as a corollary to Theorem 2, we have

Corollary 13 *For any $f : \{0,1\}^n \mapsto \{0,1\}$, and for any $\zeta \in \{0,1\}^n$,* $\delta_{\mathrm{M},\zeta}(f) \;\geq\; \frac{\epsilon_{\mathrm{M},\zeta}(f)}{n^3}$.

Proof of Theorem 12: For each $i \in \{1,\ldots,n\}$, let $\gamma_{i,0}(f)$ denote the fraction, among all pairs of strings that differ on a single bit, of the pairs x, y such that x and y differ only on the i^{th} bit, $x_i = 0$, $y_i = 1$, and $f(x) > f(y)$. Similarly, let $\gamma_{i,1}(f)$ denote the fraction of pairs of strings x, y such that x and y differ only on the i^{th} bit, $x_i = 1$, $y_i = 0$, and $f(x) > f(y)$. In other words, $\gamma_{i,0}(f)$ is the fraction of pairs that can serve as evidence to f not being monotone with respect to any ζ such that $\zeta_i = 0$, while $\gamma_{i,1}(f)$ is the fraction of pairs that can serve as evidence to f not being monotone with respect to any ζ such that $\zeta_i = 1$. Note that in case f is monotone with respect to some ζ, then for every i, $\gamma_{i,\zeta_i}(f) = 0$. More generally, $\delta_{\mathrm{M},\zeta}(f) = \sum_{i=1}^{n} \gamma_{i,\zeta_i}(f)$ holds for every $\zeta \in \{0,1\}^n$ (since each edge contributing to $\delta_{\mathrm{M},\zeta}(f)$ contributes to exactly one γ_{i,ζ_i}).

Let us define $\epsilon_{\mathrm{U}}(f)$ to be $\min_\zeta(\epsilon_{\mathrm{M},\zeta}(f))$ so that it equals the minimum distance of f to any unate function (i.e., any function that is monotone with respect to some ζ).

Claim 12.1. $\sum_{i=1}^{n} \min(\gamma_{i,0}(f), \gamma_{i,1}(f)) \geq \frac{\epsilon_{\mathrm{U}}(f)}{n^3}$.

Proof: Let $\zeta = \zeta_1 \ldots \zeta_n$ be defined as follows: For each i, if $\gamma_{i,0}(f) \leq \gamma_{i,1}(f)$ then $\zeta_i = 0$, and otherwise, $\zeta_i = 1$. In other words, $\zeta_i = \mathrm{argmin}_{b\in\{0,1\}}(\gamma_{i,b})$. The key observation is

$$\delta_{\mathrm{M},\zeta}(f) \;=\; \sum_{i=1}^{n} \gamma_{i,\zeta_i} \;=\; \sum_{i=1}^{n} \min(\gamma_{i,0}(f), \gamma_{i,1}(f))$$

where the first equality holds for any ζ, and the second follows from the definition of this specific ζ. Invoking Corollary 13, we have $\delta_{\mathrm{M},\zeta}(f) \geq \frac{\epsilon_{\mathrm{M},\zeta}(f)}{n^3} \geq \frac{\epsilon_{\mathrm{U}}(f)}{n^3}$. $\square$

For each i, let $\Gamma_{i,0}(f)$ be the set of all pairs of strings x, y that differ only on the i^{th} bit, where $x_i = 0$ and $y_i = 1$, and such that $f(x) > f(y)$. Similarly, let $\Gamma_{i,1}(f)$ be the set of all pairs x, y that differ only on the i^{th} bit, where $x_i = 1$ and $y_i = 0$, and such that $f(x) > f(y)$. Claim 12.1 gives us a lower bound on the sum $\sum_i \min(|\Gamma_{i,0}|, |\Gamma_{i,1}|)$. To prove Theorem 12, it suffices to show that if we uniformly select $\Omega(n^{3.5}/\epsilon_{\mathrm{U}}(f))$ pairs of strings that differ on a single bit, then with probability at least $2/3$, for some i we shall obtain both a pair belonging to $\Gamma_{i,0}(f)$ and a pair belonging to $\Gamma_{i,1}(f)$. The above is derived from the following technical claim, which can be viewed as a generalization of the *Birthday Paradox*.

Claim 12.2. *Let $\mathrm{S}_1, \ldots, \mathrm{S}_n, \mathrm{T}_1, \ldots, \mathrm{T}_n$ be disjoint sets of elements belonging to domain X. For each i, let the probability of selecting an element x in S_i (when x is chosen uniformly in X), be p_i, and the probability of selecting an element in T_i, be q_i. Suppose that for all i, $q_i \geq p_i$, and that $\sum_i p_i \geq \rho$ for some $\rho > 0$. Then, for some constant c, if we uniformly select $c \cdot \sqrt{n}/\rho$ elements in X, then with probability at least $2/3$, for some i we shall obtain one element in S_i and one in T_i.*

Proof: As a mental experiment, we partition the sample of elements into two parts of equal size, $c \cdot \sqrt{n}/(2\rho)$. Let I be a random variable denoting the (set of) indices of sets S_i hit by the first part of the sample. We show below that with probability at least $5/6$ over the choice of the first part of the sample,

$$\sum_{i\in \mathrm{I}} p_i \;\; \geq \;\; \frac{\rho}{\sqrt{n}} \tag{18}$$

The claim then follows since conditioned on Equation (18) holding, and by Claim 12.2's hypothesis that $q_i \geq p_i$ for all i, the probability that the second part of the sample does not include any elements from $\bigcup_{i\in \mathrm{I}} \mathrm{T}_i$, is at most

$$\left(1-\sum_{i\in \mathrm{I}} q_i\right)^{c\cdot\sqrt{n}/(2\rho)} \;\; \leq \;\; \left(1-\frac{\rho}{\sqrt{n}}\right)^{c\cdot\sqrt{n}/(2\rho)} \;\; < \;\; \exp(-c/2)$$

which is less than $1/6$ for an appropriate choice of c.

To prove that Equation (18) holds with probability at least $5/6$, we assume without loss of generality that the sets S_i are ordered according to size. Let $\mathrm{S}_1, \ldots, \mathrm{S}_k$ be all sets with probability weight at least $\rho/(2n)$ each (i.e., $p_1 \geq \ldots \geq p_k \geq \rho/(2n)$). Then, the total probability weight of all other sets $\mathrm{S}_{k+1}, \ldots, \mathrm{S}_n$ is less than $\rho/2$, and $\sum_{i=1}^{k} p_i \geq \rho/2$ follows. We first observe that by a (multiplicative) Chernoff bound (for an appropriate choice of c), with probability at least $11/12$, the first part of the sample contains at least $4\cdot\sqrt{n}$ elements in $\bar{\mathrm{S}} \stackrel{\mathrm{def}}{=} \bigcup_{i=1}^{k} \mathrm{S}_i$.

Let $\mathrm{I}' \stackrel{\mathrm{def}}{=} \mathrm{I} \cap \{1, \ldots, k\}$. That is, I' is a random variable denoting the indices of sets S_i, $i \in \{1, \ldots, k\}$ that are hit by the first part of the sample. Conditioned on there being at least $4\cdot\sqrt{n}$ elements from $\bar{\mathrm{S}}$ in the first part of the sample, we next show that with probability at least $11/12$, $\sum_{i\in \mathrm{I}'} p_i \geq \frac{\rho}{\sqrt{n}}$ (from which Equation (18) follows). Since conditioned on an element belonging to $\bar{\mathrm{S}}$ it is uniformly distributed in that set, we may bound the probability of the above event, when selecting $4\sqrt{n}$ elements uniformly in $\bar{\mathrm{S}}$. Consider the choice of the j^{th} element from $\bar{\mathrm{S}}$, and let I'_{j-1} denote the indices of sets S_i, $i \in \{1, \ldots, k\}$, among the first $j-1$ selected elements of $\bar{\mathrm{S}}$. If

$$\sum_{i\in \mathrm{I}'_{j-1}} p_i \;\; \geq \;\; \frac{2\cdot\sum_{i=1}^{k} p_i}{\sqrt{n}}$$

then, since $\sum_{i=1}^{k} p_i \geq \frac{\rho}{2}$, we are done. Otherwise ($\sum_{i\in \mathrm{I}'_{j-1}} p_i < (2\sum_{i=1}^{k} p_i)/\sqrt{n}$), the probability that the j^{th} element belongs to $\mathrm{I}' \setminus \mathrm{I}'_{j-1}$ (i.e., it hits a set in $\{\mathrm{S}_1, \ldots, \mathrm{S}_k\}$ that was not yet hit), is at least $1 - 2/\sqrt{n}$, which is at least $3/4$ for $n \geq 36$. Since we are assuming that the first part of the sample includes at least $4\cdot\sqrt{n}$ elements from $\bar{S}$, with probability at least $11/12$, we succeed in obtaining a new element in at least $2\cdot\sqrt{n}$ of these trials. Since the sets $\mathrm{S}_1, \ldots, \mathrm{S}_k$ all have probability weight at least $\rho/(2n)$, the claim follows. $\square$

6.2 Other Domains and Ranges

As defined in the introduction, for finite sets Σ and Ξ and orders $<_\Sigma$ and $<_\Xi$ on Σ and Ξ, respectively, we say that a function $f : \Sigma^n \mapsto \Xi$ is monotone if $f(x) \leq_\Xi f(y)$ for every $x \prec_\Sigma y$, where $x_1\cdots x_n \prec_\Sigma y_1\cdots y_n$ if $x_i \leq_\Sigma y_i$ for every i and $x_i <_\Sigma y_i$ for some i.

Without loss of generality we may think of Σ as being the set $\{0, \ldots, |\Sigma|-1\}$ (so that $<_\Sigma$ is simply the order $<$ over integers). Similarly to the $\Sigma = \{0, 1\}$ case, the partial order $\prec_\Sigma$ induces a layered directed graph, denoted $\mathrm{G}_{n,\Sigma}$, where the i^{th} layer L_i contains all strings x such that $\sum_j x_j = i$.

Hence, this graph has $n \cdot (|\Sigma| - 1)$ layers. For each vertex x and every j such that $x_j > 0$, there is an edge directed from x to $x' = x_1, \ldots, x_{j-1}, x_j - 1, x_{j+1}, \ldots, x_n$.

The algorithm we analyze is very similar to Algorithm 1. It uniformly selects $\Theta(n^3 \cdot |\Sigma|^2 \cdot |\Xi|/\epsilon)$ strings and for each string x chosen it performs the following local test: It uniformly selects an index $j \in 1, \ldots, n$, and queries the function f on x and on either $x' = x_1, \ldots, x_{j-1}, x_j - 1, x_{j+1}, \ldots, x_n$ or on $x' = x_1, \ldots, x_{i-1}, x_j + 1, x_{i+1}, \ldots, x_n$ (where this decision is done randomly unless $x_j = 0$ or $x_j = |\Sigma| - 1$). The algorithm rejects if for some x, $f(x) >_\Xi f(x')$ while $x \prec_\Sigma x'$ (or $f(x') >_\Xi f(x)$ while $x' \prec_\Sigma x$).

6.2.1 General Domains

Consider first the case in which Σ may be any finite ordered set, but $\Xi = \{0, 1\}$. As in the case $\Sigma = \{0, 1\}$, we want to bound $\delta_{\mathrm{M}}(f)$ in terms of $\epsilon_{\mathrm{M}}(f)$, where $\epsilon_{\mathrm{M}}(f)$ and $\delta_{\mathrm{M}}(f)$ are generalized in the straightforward manner. Here we have that

Theorem 14 *For any finite ordered set Σ, and for every $f : \Sigma^n \mapsto \{0, 1\}$, $\delta_{\mathrm{M}}(f) \geq \frac{\epsilon_{\mathrm{M}}(f)}{n^3 \cdot (|\Sigma|-1)^2}$.*

(Where similarly to the $\Sigma = \{0, 1\}$ case a slightly stronger bound actually holds.)

The proof of Theorem 14 is analogous to the proof of Theorem 2. In particular, the theorem follows by combining slightly modified versions of Lemmas 7 and 8, as done in the proof of Theorem 2. In the modified version of Lemma 7, the only change is in Item 1, where the sets S and R are of size at least $\frac{\epsilon_{\mathrm{M}}(f)}{2(n \cdot (|\Sigma|-1))^2} \cdot |\Sigma|^n$ (recall that $|\Sigma|^n$ is the size of the domain). The cause for this modification is that the number of layers in the graph $\mathrm{G}_{n,\Sigma}$ is $n \cdot (|\Sigma| - 1)$. More precisely, when invoking Lemma 9 (which can be easily verified to hold as is) in order to prove Lemma 7, we “break” the set D_1 (as defined in Equation (2)) into subsets according to the layers of $\mathrm{G}_{n,\Sigma}$. We then take the largest such subset Y, whose size we can bound by $\frac{\epsilon_{\mathrm{M}}(f)}{2(n(|\Sigma|-1))} \cdot |\Sigma|^n$. When breaking $\phi(\mathrm{Y})$ into layers, we lose another factor of $n \cdot (|\Sigma| - 1)$.

Lemma 8 essentially holds as stated. The only part of the proof that directly depends on the underlying graph is Claim 8.1, and it is easily verified that Claim 8.1 (in the proof of Lemma 8) is in fact still true in this case. The rest of the proof remains unaltered.

6.2.2 General Ranges

Let Ξ be any ordered set, and for ease of the exposition, assume $\Sigma = \{0, 1\}$ (the generalization to other domains is done as described above in Subsection 6.2.1). In this case we can show that

Theorem 15 *For any finite ordered set Ξ, and for every $f : \{0, 1\}^n \mapsto \Xi$, $\delta_{\mathrm{M}}(f) \geq \frac{\epsilon_{\mathrm{M}}(f)}{n^3 \cdot |\Xi|}$, where $\delta_{\mathrm{M}}(f)$ and $\epsilon_{\mathrm{M}}(f)$ are generalized in the natural manner.*

In case Ξ is not finite, we can replace $|\Xi|$ in the above expression with the size of the “effective” domain of f (that is, the number of different values assigned by f.)

The proof of Theorem 15 also follows similar lines to those in the proof of Theorem 2. The statement of Lemma 8 and its proof remain unaltered, since the underlying graph, G_n is the same. The statement of Lemma 7 is modified as follows:

Lemma 16 *For any ordered set Ξ, and for any function $f : \{0,1\}^n \mapsto \Xi$, there exist two sets of vertices $\mathrm{S} \subseteq \mathrm{L}_s$ and $\mathrm{R} \subseteq \mathrm{L}_r$, where $s > r$, for which the following holds:*

1. $|\mathrm{S}| = |\mathrm{R}| \ \geq \frac{\epsilon_{\mathrm{M}}(f)}{2n^2} \cdot 2^n$;
2. *There exists a one-to-one mapping ϕ from S to R such that for every $y \in \mathrm{S}$, $\phi(y) \prec y$, while $f(\phi(y)) >_{\Xi} f(y)$.*

We prove Lemma 16 momentarily, but first show how it can be applied together with Lemma 8 to obtain Theorem 15. Fixing f we invoke Lemma 7 to obtain the two matched sets S and R of size at least $m = \frac{\epsilon_{\mathrm{M}}(f)}{2n^2} \cdot 2^n$. Unfortunately, we cannot continue by simply applying Lemma 8 to the sets S and R as done in the proof of Theorem 2. The reason is that Lemma 8 only tells us that there exist *some* vertex-disjoint paths between S and R, but these paths do not necessarily respect the matching ϕ. In the case of a Boolean range, this was sufficient. However, when the range is larger, the disjoint paths might be from $y \in \mathrm{S}$ to $x \in \mathrm{R}$ such that $f(y) >_{\Xi} f(x)$, and the argument breaks down. Thus, instead of invoking Lemma 8 directly on S and R, we do the following. For each $\xi \in \Xi$, let $\mathrm{S}_\xi \stackrel{\text{def}}{=} \{y \in \mathrm{S} : \ f(y) = \xi\}$. Let S' be the largest among these subsets of S, so that $|\mathrm{S}'| \geq m/|\Xi|$. Since the value of f is constant on S', we have that for every $y \in \mathrm{S}'$ and *every* $x \in \phi(\mathrm{S}')$, $f(y) <_{\Xi} f(x)$. We then invoke Lemma 8 on S' and $\mathrm{R}' \stackrel{\text{def}}{=} \phi(\mathrm{S}')$, and the proof of Theorem 15 follows by the same argument used in the proof of Theorem 2.

One possible way to avoid the introduction of the factor of $|\Xi|$, is by proving the following conjecture which is a variation of Lemma 8: While we relax the requirement that the paths between the matched sets be vertex disjoint to being edge disjoint (which suffices for our purposes), we ask that these paths respect the matching.

Conjecture 1 *Let r and s be integers satisfying, $0 \leq r < s \leq n$, and let $\mathrm{S} \subseteq \mathrm{L}_s$ and $\mathrm{R} \subseteq \mathrm{L}_r$ be sets each of size m. Suppose that there exists a 1-to-1 mapping ϕ from S to R such that for every $y \in \mathrm{S}$, there is a directed path in G_n from y to $\phi(y)$. Then there exist m edge-disjoint directed paths in G_n connecting each $y \in \mathrm{S}$ with $\phi(y) \in \mathrm{R}$.*

In fact, it would be interesting to show even the existence of $m/\mathrm{poly}(n)$ edge-disjoint paths that respect the matching ϕ (instead of exactly m).

Proof of Lemma 16: Fixing f, we let g be a monotone function closest to f, so that $\mathrm{dist}(f,g) = \epsilon_{\mathrm{M}}(f)$. The proof of Lemma 16 is analogous to the proof of Lemma 7. We start by extending the definition of D_0 and D_1 (as given in Equation (2)) to a non-Boolean range. We define:

$$\mathrm{D}_> \stackrel{\text{def}}{=} \{x : \ g(x) > f(x)\} \quad \text{and} \quad \mathrm{D}_< \stackrel{\text{def}}{=} \{x : \ g(x) < f(x)\} \qquad (19)$$

so that $|\mathrm{D}_>| + |\mathrm{D}_<| = \epsilon_{\mathrm{M}}(f) \cdot 2^n$. Without loss of generality we assume $|\mathrm{D}_>| \geq \epsilon_{\mathrm{M}}(f) \cdot 2^{n-1}$. We next extend the operator σ_1 (defined in Equation (4)). For any $\mathrm{Y} \subseteq \mathrm{D}_>$ let

$$\sigma_>(\mathrm{Y}) \stackrel{\text{def}}{=} \{x : \ \exists y \in \mathrm{Y} \text{ s.t. } y \succ x \text{ and } f(x) > f(y)\}$$

where recall that $\sigma(\mathrm{Y})$ denotes the *shadow* of Y (and is defined in Equation (3)). Thus, $\sigma_>(\mathrm{Y})$ can be viewed as the *cause* (or *witness set*) to the need to change (raise) the value of f on the points in Y.

We next slightly depart from the course taken in the proof of Lemma 7. Namely, instead of showing analogously to Lemma 9 that for *every* subset Y of $\mathrm{D}_>$, there exists a 1-to-1 mapping that maps each element $y \in \mathrm{Y}$ to an $x \in \sigma_>(\mathrm{Y})$ such that $x \prec y$, we prove this claim only for sets Y whose elements all belong to a single layer in G_n. While this suffices for our purposes (as it actually did for the proof of Lemma 7), it is still interesting to note that it is not clear whether the stronger claim (referring to all subsets of $\mathrm{D}_>$) holds for a general range or not. In particular, the proof technique we use does not seem to be extendible (as we note in the proof below).

Lemma 17 *For every s, $0 < s \leq n$, and for every $\mathrm{Y} \subseteq (\mathrm{D}_> \cap \mathrm{L}_s)$, there exists a 1-to-1 mapping ϕ from Y into $\sigma_>(\mathrm{Y})$, such that for each $y \in \mathrm{Y}$, $\phi(y) \prec y$.*

Proof: We follow the same proof strategy of Lemma 9. Fixing s, we first show that for every $\mathrm{Y} \subseteq (\mathrm{D}_> \cap \mathrm{L}_s)$, $|\sigma_>(\mathrm{Y})| \geq |\mathrm{Y}|$. Assume towards contradiction that for some $\mathrm{Y} \subseteq (\mathrm{D}_> \cap \mathrm{L}_s)$, $|\sigma_>(\mathrm{Y})| < |\mathrm{Y}|$. We show, contrary to our hypothesis on g, that there exists another monotone function g' that is (strictly) closer to f.

Define g' as follows:

- For every $y \in \mathrm{Y}$, $g'(y) = f(y)$;
- For every $x \in \sigma(\mathrm{Y})$, $g'(x) = \min\left(g(x), \min_{y \in \mathrm{Y}, y \succ x}\{f(y)\}\right)$;[5]
- For $z \notin \mathrm{Y} \cup \sigma(\mathrm{Y})$, $g'(z) = g(z)$;

Thus, while g raises the value f has on points in Y so as to obtain monotonicity, g' maintains the value of f on points in Y but reduced the value of points below Y.[6]

We need to verify the following two claims.

Claim 17.1: *g' is a monotone function.*

Claim 17.2: $\mathrm{dist}(f, g') < \mathrm{dist}(f, g)$.

Proof of Claim 17.1: We need to show that for every x, y such that $x \prec y$, it holds that $g'(x) \leq g'(y)$. Consider the following four cases.

Case 1: $x, y \notin \mathrm{Y} \cup \sigma(\mathrm{Y})$. In this case $g'(x) = g(x) \leq g(y) = g'(y)$, where $g(x) \leq g(y)$ follows from the monotonicity of g, and the two equalities from the third item in the definition of g'.

Case 2: $x \in \mathrm{Y} \cup \sigma(\mathrm{Y})$ and $y \notin \mathrm{Y} \cup \sigma(\mathrm{Y})$. If $x \in \mathrm{Y}$ then $g'(x) = f(x) < g(x) \leq g(y) = g'(y)$, where the inequality $f(x) < g(x)$ follows from $\mathrm{Y} \subseteq \mathrm{D}_>$ and the equalities from the first and third item, respectively, in the definition of g'. If $x \in \sigma(\mathrm{Y})$ then $g'(x) \leq g(x) \leq g(y) = g'(y)$, where $g'(x) \leq g(x)$ follows from the second item in the definition of g'.

Case 3: $x \notin \mathrm{Y} \cup \sigma(\mathrm{Y})$ and $y \in \mathrm{Y} \cup \sigma(\mathrm{Y})$. By definition of $\sigma(\cdot)$, this case does not occur for $x \prec y$.

[5]Note that in the Boolean case, this minimum is always 0.

[6]Here we encounter the main difficulty in trying to prove the lemma for arbitrary $\mathrm{Y} \subseteq \mathrm{D}_>$. In particular, if, as done above, we set g' to equal f on all points in Y, then it might not be monotone.

Case 4: $x, y \in \mathrm{Y} \cup \sigma(\mathrm{Y})$. Since $x \prec y$ and $\mathrm{Y} \subseteq \mathrm{L}_s$, it cannot be the case that both x and y belong to Y. Thus we have two sub-cases.

1. If $y \in \mathrm{Y}$ and $x \in \sigma(\mathrm{Y})$ then $g'(x) \leq \min_{z \in \mathrm{Y}, z \succ x}\{f(z)\} \leq f(y) = g'(y)$, where the first inequality is due to the second item in the definition of g', and the last equality is due to the first item in the definition.
2. If $x, y \in \sigma(\mathrm{Y})$, then since $g(x) \leq g(y)$ (as g is monotone), and $\min_{z \in \mathrm{Y}, z \succ x}\{f(z)\} \leq \min_{z \in \mathrm{Y}, z \succ y}\{f(z)\}$ (as the first minimum is taken over a larger set containing all $z \succ y \succ x$), by definition of g' we have (by the second item in the definition of g'),

$$g'(x) = \min\left(g(x), \min_{z \in \mathrm{Y}, z \succ x}\{f(z)\}\right) \leq \min\left(g(y), \min_{z \in \mathrm{Y}, z \succ y}\{f(z)\}\right) = g'(y) .$$

Claim 17.1 follows. $\square$

Proof of Claim 17.2: By definition of g', the functions g and g' differ on the set of strings $\Delta \stackrel{\text{def}}{=} \mathrm{Y} \cup \mathrm{A}$, where $\mathrm{A} \stackrel{\text{def}}{=} \sigma(\mathrm{Y}) \cap \{x : g(x) > \min_{y \in \mathrm{Y}, y \succ x}\{f(y)\}\}$. For each $x \in \mathrm{Y}$, we have $g'(x) = f(x)$ and $g(x) \neq f(x)$, so that such x contributes to $\mathrm{dist}(f, g)$ but not to $\mathrm{dist}(f, g')$. Next consider any $x \in \mathrm{A}$. Since $\mathrm{A} \subseteq \sigma(\mathrm{Y})$, by the second item in the definition of g', $g'(x) = \min(g(x), \min_{z \in \mathrm{Y}, z \succ x}\{f(z)\})$, and since by definition of A, $g(x) > \min_{y \in \mathrm{Y}, y \succ x}\{f(y)\}$, we have $g'(x) = \min_{y \in \mathrm{Y}, y \succ x}\{f(y)\} < g(x)$. There are hence three sub-cases.

1. If $f(x) = g'(x)$ ($< g(x)$), then x does not contribute to $\mathrm{dist}(f, g')$ but does contributed to $\mathrm{dist}(f, g)$.
2. If $f(x) < g'(x)$ ($< g(x)$), then x contributes both to $\mathrm{dist}(f, g')$ and to $\mathrm{dist}(f, g)$.
3. If $f(x) > g'(x)$ then x contributes to $\mathrm{dist}(f, g')$, and may or may not contribute to $\mathrm{dist}(f, g)$.

Thus,

$$2^n \cdot (\mathrm{dist}(f, g') - \mathrm{dist}(f, g)) \ \leq \ |\sigma_>(\mathrm{Y})| - |\mathrm{Y}| \ < \ 0$$

where the strict inequality is due to the assumption that $|\sigma_>(\mathrm{Y})| < |\mathrm{Y}|$. Claim 17.2 follows. $\square$

Consider any set $\mathrm{Y} \subseteq (\mathrm{D}_> \cap \mathrm{L}_s)$. We have established that for every $\mathrm{Y}' \subseteq \mathrm{Y}$, $|\sigma_>(\mathrm{Y}')| \geq |\mathrm{Y}'|$. Similarly to the proof of Lemma 9, Lemma 17 follows from Hall's Theorem. $\blacksquare$

The proof of Lemma 16 follows from Lemma 17 similarly to the way Lemma 7 was shown to follows from Lemma 9, and is hence omitted. $\blacksquare$

Acknowledgments

We would like to thank Dan Kleitmann for a helpful discussion, and in particular for coming up with the counter-example (Figure 3).

References

[ALM$^+$98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. *JACM*, 45(3):501–555, 1998.

[Ang88] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, April 1988.

[AS97] S. Arora and S. Sudan. Improved low degree testing and its applications. In *Proceedings of STOC97*, pages 485–495, 1997.

[AS98] S. Arora and S. Safra. Probabilistic checkable proofs: A new characterization of NP. *JACM*, 45(1):70–122, 1998.

[BBL98] A. Blum, C. Burch, and J. Langford. On learning monotone Boolean functions. In *Proceedings of FOCS98*, 1998.

[BCH$^+$95] M. Bellare, D. Coppersmith, J. Håstad, M. Kiwi, and M. Sudan. Linearity testing in characteristic two. In *Proceedings of FOCS95*, pages 432–441, 1995.

[BFL91] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, 1991.

[BFLS91] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of STOC91*, pages 21–31, 1991.

[BGLR93] 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.

[BGS98] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and non-approximability – towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998.

[BLR93] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *JACM*, 47:549–595, 1993.

[Bol86] B. Bollobás. *Combinatorics*. Cambridge University Press, 1986.

[BS94] M. Bellare and M. Sudan. Improved non-approximability results. In *Proceedings of STOC94*, pages 184–193, 1994.

[DL98] Y. Doddis and E. Lehman. Private communications. 1998.

[EKK$^+$98] F. Ergun, S. Kannan, S. R. Kumar, R. Rubinfeld, and M. Viswanathan. Spot-checkers. In *Proceedings of STOC98*, pages 259–268, 1998.

[Eve79] S. Even. *Graph Algorithms*. Computer Science Press, 1979.

[FGL$^+$96] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. *JACM*, 43(2):268–292, 1996.

[GGR96] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. In *Proceedings of FOCS96*, pages 339–348, 1996.

[GLR$^+$91] 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.

[GR97] O. Goldreich and D. Ron. Property testing in bounded degree graphs. In *Proceedings of STOC97*, pages 406–415, 1997.

[GR98] O. Goldreich and D. Ron. A sublinear bipartite tester for bounded degree graphs. In *Proceedings of STOC98*, 1998.

[Hås96] J. Håstad. Testing of the long code and hardness for clique. In *Proceedings of STOC96*, pages 11–19, 1996.

[Hås97] J. Håstad. Getting optimal in-approximability results. In *Proceedings of STOC97*, pages 1–10, 1997.

[Kiw96] M. Kiwi. *Probabilistically Checkable Proofs and the Testing of Hadamard-like Codes*. PhD thesis, MIT, 1996.

[KLV94] M. Kearns, M. Li, and L. Valiant. Learning boolean formulae. *JACM*, 41(6):1298–1328, 1994.

[KV94] M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *JACM*, 41(1):67–95, 1994.

[RS96] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[RS97] 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.

[Tre98] L. Trevisan. Recycling queries in pcps and in linearity tests. In *Proceedings of STOC98*, pages 299–308, 1998.

A Paths that Respect the Mapping

It is interesting to note that Lemma 8 *does not* hold if one requires that the vertex-disjoint paths from S to R respect the given 1–1 mapping ϕ (i.e., that the paths connect each $y \in \mathrm{S}$ to the corresponding $\phi(y)$). An example is depicted in Figure A. For the given example $|\mathrm{S}| = |\mathrm{R}| = 8$, and there are no 8 vertex-disjoint paths that respect the given matching. More generally, it can be shown [DL98] that if the paths are required to correspond to a particular matching then the number of disjoint paths can be as small as $O(m/n)$ where m is the number of matched vertices.

However, if we only require that the paths be *edge*-disjoint (which actually suffices for our purposes), then we have no counter-example to the conjecture that such paths always exist (i.e., Conjecture 1).

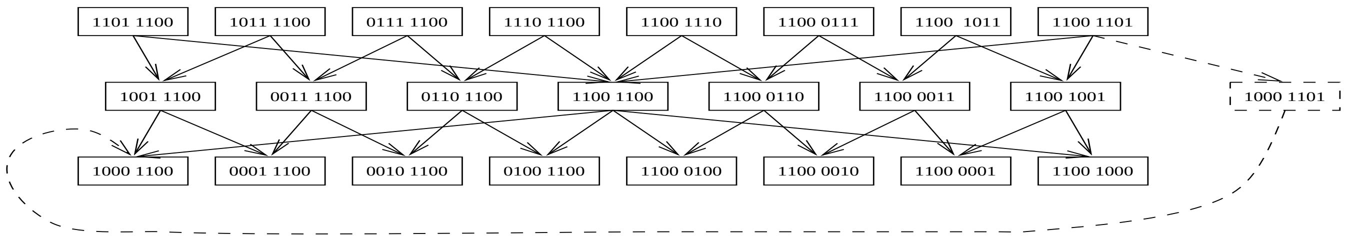

Figure 3: An example in which there *aren't* enough disjoint paths respecting a *particular* 1–1 mapping (and so the disjoint paths guaranteed by Lemma 7 correspond to a different mapping). The given 1-1 mapping is from each 8-bit long string at the top level to the 8-bit long string that is aligned with it in the bottom level. For each such "matched" pair there are (two) paths from the top vertex to the corresponding bottom one. All possible paths connecting these matched pairs appear in the picture in solid arrows. (There are only two paths between each pair of strings that are at Hamming distance 2.) Since the paths that respect the matching use only 7 intermediate vertices, there exist no 8 vertex-disjoint paths respecting this mapping. However, there are other 1–1 mappings for which vertex-disjoint paths from the top vertices to the bottom one do exist. For example, consider the "circular shift-to-right mapping" and use the auxiliary vertex on the right.