A Note on Testing Monotonicity

Oded Goldreich* Shafi Goldwasser† Dana Ron‡

October 1997 (revised April 1998)

Abstract

We show that under a certain conjecture regarding the boolean lattice, there exists an efficient algorithm for testing whether a function is monotone or ϵ-far from monotone.

NOTE: The said combinatorial conjecture has been recently proven in collaboration with Eric Lehman. We plan to write a joint paper presenting these results. [April 12, 1998]

1 Introduction

One of the first problems we considered while working on our paper on Property Testing [3], is testing Monotonicity of (Boolean) functions. A function $f : \{0,1\}^n \mapsto \{0,1\}$ is called monotone if $f(x) \leq f(y)$, for every $x < y$, where the partial order between strings is defined analogously to the set inclusion relation. That is, $x_1 x_2 \cdots x_n < y_1 y_2 \cdots y_n$ if $x_i \leq y_i$ for all i's and $0 = x_j < y_j = 1$ for some j.

A testing algorithm is given a distance parameter ϵ, and oracle access to an unknown function. It is required to accept with high probability any monotone function, and to reject with high probability any function that is ϵ-far from being monotone. A function $f : \{0,1\}^n \mapsto \{0,1\}$ is said to be ϵ-far from monotone if for every monotone function g,

$$|\{x \in \{0,1\}^n : f(x) \neq g(x)\}| \ > \ \epsilon \cdot 2^n$$

One natural idea is to test monotonicity by repeating poly(n/ϵ) many time the following step: Uniformly select $x \in \{0,1\}^n$, and $i \in [n] \stackrel{\rm def}{=} \{1,2,...,n\}$, query the function at x and

*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@theory.lcs.mit.edu.

‡Laboratory for Computer Science, MIT, 545 Technology Sq., Cambridge, MA 02139. E-mail: danar@theory.lcs.mit.edu. Supported by an NSF postdoctoral fellowship.

at $x' \stackrel{\text{def}}{=} x \oplus 0^{i-1}10^{n-i}$ (i.e., x with the ith bit flipped), and reject if monotonicity is violated (e.g., in case $x < x'$ we reject if $f(x) > f(x')$).

We were able to reduce the correctness of this natural algorithm to a conjecture regarding the structure of the Boolean Lattice, but were not able to prove this conjecture. The current note presents what we know.

2 Preliminaries

For any pair of functions $f, g : \{0,1\}^n \rightarrow \{0,1\}$, we define the *distance* between f and g, denoted, $\text{dist}(f,g)$, to be the fraction of instance $x \in \{0,1\}^n$ on which $f(x) \neq g(x)$. In other words, $\text{dist}(f,g)$ is the probability over a uniformly chosen x that f and g differ on x. Let $\mathcal{G}$ be a class of boolean functions over $\{0,1\}^n$. We define $\text{dist}(f,\mathcal{G}) \stackrel{\text{def}}{=} \min_{g\in\mathcal{G}}\{\text{dist}(f,g)\}$. We say that f is *ϵ-far* from $\mathcal{G}$, for $0 \leq \epsilon \leq 1$, if $\text{dist}(f,\mathcal{G}) > \epsilon$. Let $\mathcal{M}_n$ be the class of monotone functions over $\{0,1\}^n$. That is, for every function $g \in \mathcal{M}_n$ the following holds: For every $x, y \in \{0,1\}^n$, if $x < y$ then $g(x) \leq g(y)$.

Definition 1 *A* property-testing *algorithm for the monotonicity property is given an input-size parameter n, a distance parameter ϵ, and oracle access to an unknown function $f : \{0,1\}^n \rightarrow \{0,1\}$. We require that for every n, ϵ and f the following hold:*

- *If f is a monotone function then the algorithm accepts with probability at least $\frac{2}{3}$;*
- *If f is ϵ-far from $\mathcal{M}_n$ then the algorithm rejects with probability at least $\frac{2}{3}$.*

3 The Reduction

Clearly, the algorithm described in the introduction accepts any monotone function. To analyze its behavior on a function that is far from being monotone, we consider an arbitrary function f, and investigate its properties.

Let $\delta = \text{dist}(f,\mathcal{M}_n)$, and let g be a monotone function (over $\{0,1\}^n$) for which $\text{dist}(f,g) = \delta$. Namely, g is a monotone function that is closest to f. For $b \in \{0,1\}$, let

$$\mathrm{D}_b \stackrel{\text{def}}{=} \{x : \ f(x) \neq g(x) \text{ and } g(x) = b\} \quad (1)$$

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 strings in the set then we obtain a monotone function (i.e., g). Since $|\mathrm{D}_0 \cup \mathrm{D}_1| = \delta \cdot 2^n$ and $D_0 \cap \mathrm{D}_1 = \emptyset$, we may assume without loss of generality that $|\mathrm{D}_1| \geq \frac{\delta}{2} \cdot 2^n$. Note that, by definition,

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

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

$$\sigma(S) \stackrel{\text{def}}{=} \{x \notin S : \exists y \in S \text{ s.t. } x < y\}$$

Namely, the shadow of S is the set of all strings not in S that are each smaller than some string in S. For any $S \subseteq D_1$ (where D_1 is as defined above), define

$$\sigma_1(S) \stackrel{\text{def}}{=} \{x \in \sigma(S) : f(x) = g(x) = 1\}$$

As a visualization (See Figure 3), we view g as defining a *boundary* in the boolean lattice 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 $\sigma(D_1)$ contains all strings in the shadow of D_1 that lie above the boundary and f labels 1.

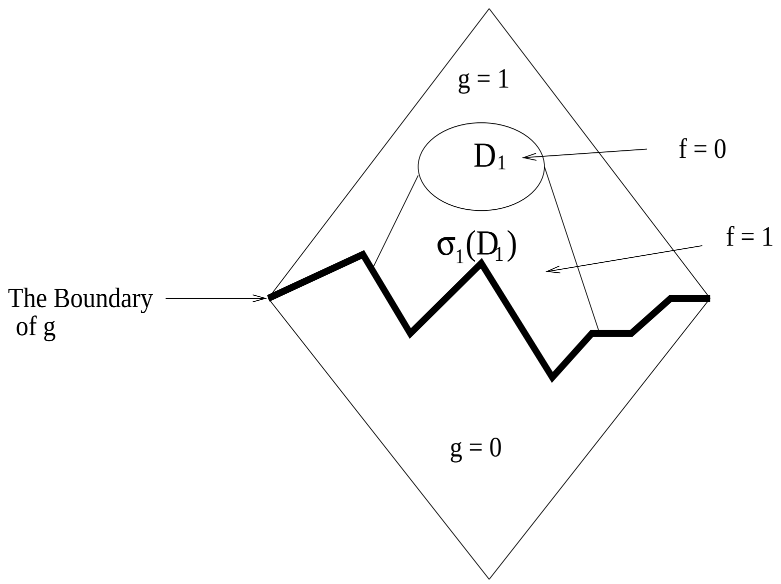

Figure 1: The sets D_1 and $\sigma_1(D_1)$.

Thus, by definition of D_1 and $\sigma_1(D_1)$, we have that for every $x \in \sigma_1(D_1)$, there exists $y \in D_1$ such that $y < x$ but $f(y) < f(x)$ (i.e. $f(y) = 0$ and $f(x) = 1$), thus providing *evidence* to the non-monotonicity of f. We stress that this evidence is not necessarily detectable by the algorithm. The next lemma is the first step towards translating evidence of the above type into phenomena that may be detectable by the algorithm. The lemma asserts (as a special case) that there is a matching of elements of D_1 to elements of $\sigma_1(D_1)$ so that each $y \in D_1$ is matched to a string $x < y$.

Lemma 1 *For every $S \subseteq D_1$, there exists a 1-to-1 mapping ϕ from S into $\sigma_1(S)$, such that for each $x \in S$, $\phi(x) < y$.*

[1] This is not the standard definition of a shadow [1].

Proof: We first show that for every $\mathrm{T} \subseteq \mathrm{D}_1$, $|\sigma_1(\mathrm{T})| \geq |\mathrm{T}|$. Assume towards the contradiction that, for some $\mathrm{T} \subseteq \mathrm{D}_1$, $|\sigma_1(\mathrm{T})| < |\mathrm{T}|$. 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 $x \in \mathrm{T} \cup \sigma(\mathrm{T})$, $g'(x) = 0$. Otherwise, $g'(x) = g(x)$. We need to verify the following two claims.

Claim 1: $\mathrm{dist}(f, g') < \mathrm{dist}(f, g)$.

Claim 2: *g' is a monotone function.*

Proof of Claim 1: For any set $\mathrm{X} \subseteq \{0,1\}^n$ let $\mu(\mathrm{X}) \stackrel{\mathrm{def}}{=} 2^{-n} \cdot |\mathrm{X}|$. By definition of g', the functions g and g' differ on $\Delta \stackrel{\mathrm{def}}{=} (\mathrm{T} \cup \sigma(\mathrm{T})) \cap \{x : g(x)=1\}$. Since $\mathrm{T} \subseteq \mathrm{D}_1 \subseteq \{x : g(x)=1\}$,

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

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

- For every $x \in \mathrm{T}$, we have $f(x) = 0$ and $g(x) = 1$ (since $\mathrm{T} \subseteq \mathrm{D}_1$), and $g'(x) = 0$ (by definition).
- For every $x \in \sigma_1(\mathrm{T})$, we have $f(x) = g(x) = 1$ (by definition of $\sigma_1(\mathrm{T})$), and again $g'(x) = 0$.
- For every $x \in \mathrm{A}$, we have $f(x) = 0$ and $g(x) = 1$ (by definition of A), and again $g'(x) = 0$.

Thus,

$$\begin{aligned}
\mathrm{dist}(f, g') - \mathrm{dist}(f, g) &= \mu(\sigma_1(\mathrm{T})) - \mu(\mathrm{T} \cup \mathrm{A}) \\
&\leq \mu(\sigma_1(\mathrm{T})) - \mu(\mathrm{T}) \\
&< 0
\end{aligned}$$

where the strict inequality is due to the counter-hypothesis regarding T. $\square$

Proof of Claim 2: We need to show that for every x, y such that $x < y$, it holds that $g'(x) \leq g'(y)$. Consider the following cases.

Case 1: $x \in \mathrm{T} \cup \sigma(\mathrm{T})$. In this case $g'(x) = 0$, and so for all y, $g(x) \leq g(y)$.

Case 2: $x \notin \mathrm{T} \cup \sigma(\mathrm{T})$. Note that in this case $g'(x) = g(x)$. We will show that for every y if $y > x$ then $y \notin \mathrm{T} \cup \sigma(\mathrm{T})$ as well, and thus $g'(y) = g(y) \geq g(x) = g'(x)$ as required. Suppose towards contradiction that for some $y \in \mathrm{T} \cup \sigma(\mathrm{T})$ it holds that $y > x$. We consider two cases.

1. If $y \in \mathrm{T}$ and $x < y$ then $x \in \mathrm{T} \cup \sigma(\mathrm{T})$ in contradiction to the case hypothesis.
2. If $y \in \sigma(\mathrm{T})$ and $x < y$ then there exists $z \in \mathrm{T}$ such that $z > y$. Thus $z > x$ and contradiction follows as in Item 1.

The claim follows. $\square$

Consider any set $\mathrm{S} \subseteq \mathrm{D}_1$. We have established that for every $\mathrm{T} \subseteq \mathrm{S}$, $|\sigma_1(\mathrm{T})| \geq |\mathrm{T}|$. The lemma follows from Hall's Theorem [2, Thm. 6.12]: Consider the bipartite graph G whose vertex set is labeled by the strings in $\mathrm{S} \cup \sigma_1(\mathrm{S})$, and whose edge set is $\{(x, y) : \ x \in \sigma_1(\mathrm{S}),\ y \in \mathrm{S},\ x < y\}$. By the above, for each $\mathrm{T} \subseteq \mathrm{S}$, we have $|\Gamma(\mathrm{T})| \geq |\mathrm{T}|$, where $\Gamma(\mathrm{T})$ denotes the neighbor set of T in G. By Hall's Theorem, this implies that there exists a complete matching of S to $\sigma_1(\mathrm{S})$, and the lemma follows. $\blacksquare$

For every string x, let $w(x)$ denote the *weight* of x (i.e., the number of 1's in x). For each i, $0 \leq i \leq n$, let $\mathrm{L}_i \subset \{0,1\}^n$ denote the set of n-bit long strings of weight i (i.e., $L_i = \{x \in \{0,1\}^n : w(x) = i\}$). 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 $x < y$ and $w(x) = w(y) - 1$ (i.e., x and y are in adjacent L_i's). As shown below, the following conjecture implies that the algorithm presented in the introduction constitute a tester of monotonicity.

Conjecture 1 *Let r and s be integers satisfying, $0 \leq r < s \leq n$, and let $\mathrm{R}, \mathrm{S} \subset \{0,1\}^n$, be sets such that $\mathrm{R} \subseteq \mathrm{L}_r$, and $\mathrm{S} \subseteq \mathrm{L}_s$, and $|\mathrm{R}| = |\mathrm{S}| = m$. Suppose that there exists a 1-to-1 mapping ψ from S to R such that for every $y \in \mathrm{S}$, there is a directed path in G_n from y to $\psi(y)$. Then there exist m vertex-disjoint directed paths from S to R in G_n.*

Actually, let us present a slight variant of the algorithm presented in the introduction

Algorithm 1:

Repeat $2n^2/\epsilon$ times

1. Uniformly select $x \in \{0,1\}^n$, and obtain the value $f(x)$;
2. In case $f(x) = 1$, obtain the values of $f(y)$ for all $y > x$ that neighbor x in G_n (i.e., y equals x with one of the zeros in x flipped to 1). If one of these $f(y)$'s is 0 then reject.
3. Analogously, in case $f(x) = 0$, obtain the values of $f(y)$ for all $y < x$ that neighbor x in G_n. If one of these $f(y)$'s is 1 then reject.

If all iterations were completed without rejecting then accept.

Theorem 1 *If Conjecture 1 holds then Algorithm 1 is a property tester for monotonicity. In particular, if f is monotone then Algorithm 1 always accepts, whereas if f is ϵ-far from $\mathcal{M}_n$ then Algorithm 1 rejects with probability at least 2/3.*

The only difference between the algorithm presented in the introduction and Algorithm 1 is that in each iteration, instead of picking a random neighbor of the chosen string x, we consider either all neighbors above x or all neighbors below x. Thus the correctness of Algorithm 1 implies the correctness of the algorithm presented in the introduction.

Proof: Clearly, if f is monotone, then the algorithm always accepts. So we need to consider what happens when f is ϵ-far from $\mathcal{M}_n$. By the above discussion we may assume that D_1 (see Eq. (1)) has size at least $\epsilon \cdot 2^{n-1}$. Let $\mathrm{S}_i \stackrel{\mathrm{def}}{=} \mathrm{D}_1 \cap \mathrm{L}_i$, and let k denote the index of the largest set among the S_i's. It follows that $|\mathrm{S}_k| \geq \frac{\delta}{2n} \cdot 2^n$.

We now invoke Lemma 1 with $\mathrm{S} = \mathrm{S}_k$. Let $\mathrm{R}_k \stackrel{\mathrm{def}}{=} \phi(\mathrm{S}_k)$, where ϕ is as guaranteed by the lemma. Hence, $\mathrm{R}_k \subseteq \sigma_1(\mathrm{S}_k)$, and $|\mathrm{R}_k| = |\mathrm{S}_k|$. Note that while all elements of S_k belong to L_k, the elements of R_k are contained in several L_i's, $i < k$. For each i, $0 \leq i < k$, let $\mathrm{R}_{k,i} \stackrel{\mathrm{def}}{=} \mathrm{R}_k \cap \mathrm{L}_i$. Let $\mathrm{R}_{k,j}$ be the largest such set. Since $|\mathrm{R}_k| = |\mathrm{S}_k| \geq \frac{\delta}{2n} \cdot 2^n$, we have $|\mathrm{R}_{k,j}| \geq \frac{\delta}{2n^2} \cdot 2^n$. Finally let $\mathrm{S}_{k,j} \stackrel{\mathrm{def}}{=} \phi^{-1}(\mathrm{R}_{k,j})$.

We next apply Conjecture 1 with $r = j$, $s = k$, $\mathrm{R} = \mathrm{R}_{k,j}$ and $\mathrm{S} = \mathrm{S}_{k,j}$. We conclude that there exist $m = |\mathrm{S}| \geq \frac{\epsilon}{2n^2} \cdot 2^n$ vertex disjoint paths from S to R (in G_n). Consider any such path $y_0 = y, \ldots, y_t = x$, where $y \in S$, $x \in \mathrm{R}$, and $t = k - j$. Since $y_0 \in \mathrm{S} \subseteq \mathrm{D}_1$, we have $f(y_0) = 0$. On the other hand, since $y_t \in \mathrm{R} \subseteq \sigma_1(\mathrm{D}_1)$, we have $f(y_t) = 1$. Therefore, there must exist some $\ell \in \{0, ..., t-1\}$, such that $f(y_\ell) = 0$ and $f(y_{\ell+1}) = 1$. But $y_\ell > y_{\ell+1}$, and so if the algorithm selects either y_ℓ or $y_{\ell+1}$ at Step (1) then it rejects. Since these m paths are vertex-disjoint, we conclude that the probability that the algorithm rejects in a single iteration is at least

$$\frac{2m}{2^n} \geq \frac{\epsilon}{n^2}$$

Thus, the probability that the algorithm accepts an ϵ-far from monotone function is bounded above by

$$\left(1 - \frac{\epsilon}{n^2}\right)^{2n^2/\epsilon} < \frac{1}{3}$$

and the theorem follows. ■

4 Observations Concerning Conjecture 1

We stress that Conjecture 1 does not require that the vertex-disjoint paths from S to R respect the given 1–1 mapping ψ (i.e., that the new paths also connect each $y \in \mathrm{S}$ to the corresponding $\psi(\mathrm{S})$). In fact, a stronger claim in which these paths are required to respect the given mapping is false. An example is depicted in Figure 4. For the given example $|\mathrm{S}| = |\mathrm{R}| = 8$, and there are no 8 vertex-disjoint paths that respect the given matching (yet there exist 8 vertex-disjoint paths from S to R). This illustrates the non-triviality of the conjecture.

We next show that Conjecture 1 follows from the following seemingly weaker conjecture.

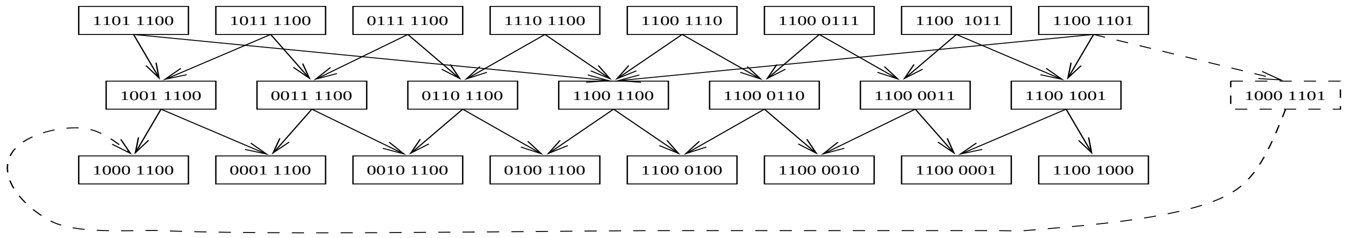

Figure 2: An example in which there *aren't* enough disjoint paths respecting a *particular* 1–1 mapping (but there is the desired number of disjoint paths corresponding to a different mapping). The given 1-1 mapping is from each 8-bit long string at the top level to the 8-bit long string that is aligned 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. Still, if we are willing to use a different matching then 8 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.

Conjecture 2 *Let r, s, R, S and ψ be as in Conjecture 1. Let I be the set of vertices in L_{s-1} that are on a directed path going from some vertex in S to a vertex in R. Then $|\mathrm{I}| \geq |\mathrm{S}|$.*

Clearly Conjecture 2 follows from Conjecture 1. We show that the converse holds too.

Proposition 1 *Conjecture 2 implies Conjecture 1.*

Proof: The proof is by induction on $\ell \stackrel{\rm def}{=} s-r$. The base case of $\ell = 1$ holds vacuously. For the induction step, we assume that the implication holds for $\ell - 1$, and consider arbitrary sets $\mathrm{S} \subseteq \mathrm{L}_s$ and $\mathrm{R} \subseteq \mathrm{L}_r$ with $s - r = \ell$. We will shortly prove that (1) there exists a complete matching from S to I; and (2) there exists a 1–1 mapping ψ' from the matched vertices of I to R so that there is a path from each matched $x \in \mathrm{I}$ to $\psi'(x)$. Given (2) we can apply the induction hypothesis on I and R, and by combining with (1) we get the desired paths from S to R.

We now prove both (1) and (2). 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, I and R, respectively. There is an edge from s to each of the vertices in T, and from each of the vertices in R to t. The edges between S and I are as in G_n and edges between I and R correspond to directed paths in G_n. We show that the minimum $s-t$ vertex-separator in A has size $m \stackrel{\rm def}{=} |\mathrm{S}|$. Claims (1) and (2) follow by Menger's Theorem [2, Thm. 6.4], 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 $m_1 \stackrel{\rm def}{=} |\mathrm{C} \cap \mathrm{S}|$, $m_2 \stackrel{\rm def}{=} |\mathrm{C} \cap \mathrm{I}|$, and $m_3 \stackrel{\rm def}{=} |\mathrm{C} \cap \mathrm{R}|$. Consider the subset of vertices

$S' \subseteq S$ that do not belong to C and are not mapped by ψ to vertices in $R \cap C$. The size of S' is at least $m' = m - (m_1 + m_3) > |C| - (m_1 + m_3) = m_2$. Let $R' \stackrel{\text{def}}{=} \psi(S')$, and I' be the set of vertices in L_{s-1} that are on a directed path going from some vertex in S' to a vertex in R'. Then, by Conjecture 2 (applied to S' and R'), the set I' is of size at least m'. Since $|C \cap I| = m_2 < m'$, there exists at least one vertex v in $I' \setminus C$, but this contradicts the fact that C is an $s - t$ vertex-separator (since we can connect s to t using a path through S', v and R'). ■

Acknowledgements

We would like to thank Dan Kleitmann for a helpful discussion, and in particular for coming up with the counter-example (Figure 4).

References

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

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

[3] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. Extended abstract in *Proc. of the 37th IEEE Symp. on Foundation of Computer Science*, pages 339–348, 1996. Full version available from `http://theory.lcs.mit.edu/~oded/ggr.html`.