

# An $\tilde{O}(n)$ Queries Adaptive Tester for Unateness\*

Subhash Khot<sup>1</sup> and Igor Shinkar<sup>2</sup>

1 Courant Institute of Mathematical Sciences, New York University, USA  
khot@cims.nyu.edu

2 Courant Institute of Mathematical Sciences, New York University, USA  
ishinkar@cims.nyu.edu

---

## Abstract

We present an adaptive tester for the unateness property of Boolean functions. Given a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  the tester makes  $O(n \log(n)/\varepsilon)$  adaptive queries to the function. The tester always accepts a unate function, and rejects with probability at least 0.9 if a function is  $\varepsilon$ -far from being unate.

**1998 ACM Subject Classification** F.1.1 Models of Computation

**Keywords and phrases** property testing, boolean functions, unateness

**Digital Object Identifier** 10.4230/LIPIcs.APPROX-RANDOM.2016.37

## 1 Introduction

A Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is said to be *unate* if for every  $i \in [n]$  it is either the case that  $f$  is monotone non-increasing in the  $i$ 'th coordinate, or  $f$  is monotone non-decreasing in the  $i$ 'th coordinate. In this work we present an adaptive tester for the unateness property that makes  $O(n \log(n)/\varepsilon)$  adaptive queries to a given function. The tester always accepts a unate function, and rejects with probability at least 0.9 any function that is  $\varepsilon$ -far from being unate.

Testing unateness has been studied first in the paper of Goldreich et al. [10], where the authors present a non-adaptive tester for unateness that makes  $O(n^{1.5}/\varepsilon)$  queries. The tester in [10] is the so-called “edge tester”, that works by querying the function on the endpoints of  $O(n^{1.5}/\varepsilon)$  uniformly random edges of the hypercube, i.e., uniformly random pairs  $(x, y)$  that differ in one coordinate, and checking that there are no violations to the unateness property.

The notion of unateness generalizes the notion of monotonicity. Recall that a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is said to be monotone if  $f(x) \leq f(y)$  for all  $x \prec y$ , where  $\prec$  denotes the natural partial order on Boolean strings, namely,  $x \prec y$  if  $x_i \leq y_i$  for all  $i \in [n]$ . Since the original paper of [10] there has been a lot of research concerning the problem of testing monotonicity of Boolean functions, as well as many closely related problems, such as testing monotonicity on functions with different (non-Boolean) domains [8, 9, 4, 13, 5, 7, 6, 1, 2], culminating in a recent result of [11], which gives a  $\tilde{O}(\sqrt{n}/\varepsilon^2)$ -query non-adaptive tester for monotonicity. In this paper we will use the monotonicity tester of [10], which has a better dependence on  $\varepsilon$ .

► **Theorem 1** (Testing Monotonicity [10]). *For any proximity parameter  $\varepsilon > 0$  there exists a non-adaptive tester for the monotonicity property that given a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  the tester makes  $O(n/\varepsilon)$  queries to the function. The tester always accepts a monotone*

---

\* Research supported by NSF grants CCF 1422159, 1061938, 0832795 and Simons Collaboration on Algorithms and Geometry grant.



function, and if a function is  $\varepsilon$ -far from being monotone, the tester finds a violation to monotonicity with probability at least 0.99.

We remark that the monotonicity testers analyzed in [10, 5, 7, 11] are all pair testers that pick pairs  $x \prec y$  according to some distribution, and check that the given function  $f$  does not violate monotonicity on this pair, i.e., checks that  $f(x) \leq f(y)$ . It is not clear whether a variant of such tester can be applied for testing unateness, since the function can be monotone increasing in some of the coordinates where  $x$  and  $y$  differ, and monotone decreasing in others.

## 1.1 Our result

In this paper we prove the following theorem.

► **Theorem 2.** *For any proximity parameter  $\varepsilon > 0$  there exists an adaptive tester for the unateness property, that given a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  makes  $O(n \log(n)/\varepsilon)$  adaptive queries to  $f$ . The tester always accepts a unate function, and rejects with probability at least 0.9 any function that is  $\varepsilon$ -far from being unate.*

The tester works as follows. Given a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , the tester first finds a subset of coordinates  $T \subseteq [n]$  such that the function is essentially independent of the coordinates outside  $T$ . Specifically, it finds a subset of coordinates  $T \subseteq [n]$  such that  $\mathbb{E}_{z \in \{0, 1\}^T} [\text{Var}_{w \in \{0, 1\}^{[n] \setminus T}} [f(z_T \circ w_{\bar{T}})]]$  is small, i.e., if we pick  $x, y \in \{0, 1\}^n$  that are equal on their coordinates in  $T$  uniformly at random, then with high probability we will have  $f(x) = f(y)$ . Furthermore, for each  $i \in T$  the tester will find an edge  $(x, x + e_i)$  in the hypercube such that  $f(x) \neq f(x + e_i)$  (where  $e_i$  is the unit vector with 1 in the  $i$ 'th coordinate). Querying  $f$  on these two points gives a “direction” for monotonicity for each coordinate in  $T$ .

In the second part of the tester, we define a function that depends only on the coordinates in  $T$  by fixing the variables outside  $T$  uniformly at random. We then apply the monotonicity tester from Theorem 1 on this function with respect to the directions obtained for the coordinates in  $T$  in the previous step, and output the answer of this tester. For the analysis, we use the fact that *on average* the restricted function is close to the original function  $f$ , and hence is far from being unate. In particular, it is far from being a monotone function with respect to the directions for the coordinates in  $T$  obtained in the first step. Hence a monotonicity tester with high probability will find a violation of monotonicity in these directions, which will serve as evidence that the function is not unate.

## 2 Preliminaries

► **Definition 3.** For two Boolean functions  $f, g : \{0, 1\}^n \rightarrow \{0, 1\}$  defined the distance between them as  $\text{distance}(f, g) = \Pr_{x \in \{0, 1\}^n} [f(x) \neq g(x)] = 2^{-n} |\{x \in \{0, 1\}^n : f(x) \neq g(x)\}|$ . We say that  $f$  is  $\varepsilon$ -far from a collection of functions  $\mathcal{P}$  if for any  $g \in \mathcal{P}$  it holds that  $\text{distance}(f, g) \geq \varepsilon$ .

► **Definition 4.** A Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is said to be *monotone non-decreasing* or simply *monotone* if  $f(x) \leq f(y)$  for all  $x \prec y$ , where  $\prec$  denotes the natural partial order on Boolean strings i.e.,  $x \prec y$  if  $x_i \leq y_i$  for all  $i \in [n]$ . In other words,  $f$  is monotone if for every  $i \in [n]$  the function  $f$  is monotone non-decreasing in the  $i$ 'th coordinate.

For directions  $B = (b_i \in \{up, down\} : i \in [n])$  let the partial order  $\prec_B$  be defined as  $x \prec_B y$  if for all  $i \in [n]$  such that  $b_i = up$  it holds that  $x_i \leq y_i$  and for all for all  $i \in [n]$  such that  $b_i = down$  it holds that  $x_i \geq y_i$ . A Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is said to be

monotone with respect to the directions  $B = \{b_i \in \{\text{up}, \text{down}\} : i \in [n]\}$  if  $f(x) \leq f(y)$  for all  $x \prec_B y$ .

A Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is said to be *unate* if it is monotone with respect to some directions, i.e., if for every  $i \in [n]$  it is either the case that  $f$  is monotone non-increasing in the  $i$ 'th coordinate, or  $f$  is monotone non-decreasing in the  $i$ 'th coordinate.

Next we make definitions related to restrictions of Boolean functions by fixing some of the coordinates.

► **Definition 5.** Given a string  $x \in \{0, 1\}^n$  and a subset of coordinates  $T \subseteq [n]$  denote by  $x_T$  the substring of  $x$  whose coordinates are indexed by  $T$ . Given two strings  $x, y \in \{0, 1\}^n$  and two disjoint subsets of coordinates  $S, T \subseteq [n]$  denote by  $x_T \circ y_S$  the string  $z$  whose coordinates are indexed by  $T \cup S$  with  $z_i = x_i$  if  $i \in T$  and  $z_i = y_i$  if  $i \in S$ .

► **Definition 6.** Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. For a subset of coordinates  $T \subseteq [n]$  and  $w \in \{0, 1\}^{[n] \setminus T}$  denote by  $f_{T,w} : \{0, 1\}^n \rightarrow \{0, 1\}$  the Boolean function defined as  $f_{T,w}(z) = f(z_T \circ w_{[n] \setminus T})$ . That is, for each  $w \in \{0, 1\}^{[n] \setminus T}$  the function  $f_{T,w}$  depends only on the coordinates in  $T$ .

► **Definition 7.** Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function, and let  $T \subseteq [n]$  be a subset of coordinates. Define  $\text{Var}_{[n] \setminus T}(f) = \mathbb{E}_{z \in \{0, 1\}^T} [\text{Var}_{w \in \{0, 1\}^{[n] \setminus T}} [f(z_T \circ w_{\overline{T}})]]$ .

This quantity has been used before, e.g., in [12, 3]. It measures how much  $f$  depends on the coordinates outside  $T$ . In particular, if  $f$  depends only on the coordinates in  $T$ , (i.e., is independent of the coordinates in  $[n] \setminus T$ ) then  $\text{Var}_{[n] \setminus T}(f) = 0$ .

The following proposition is straightforward from the definition.

► **Proposition 8.** Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. and let  $T \subseteq [n]$  be a subset of coordinates. Pick  $x, y \in_R \{0, 1\}^n$  such that  $x_i = y_i$  for all  $i \in T$  and  $\{x_i, y_i \in \{0, 1\} : i \in [n] \setminus T\}$  are chosen independently and uniformly at random. Then  $\text{Var}_{[n] \setminus T}(f) = \Pr[f(x) \neq f(y)]$ .

### 3 Proof of Theorem 2

Below we present our tester for the unateness property. The tester uses a subroutine called **Find an influential coordinate** which works as follows. It gets an oracle access to a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , and a subset of the coordinates  $T \subseteq [n]$ , which is given explicitly. The subroutine outputs either  $\perp$  or some  $i^* \in [n] \setminus T$  and  $b \in \{\text{up}, \text{down}\}$  such that there exist  $x, y \in \{0, 1\}^n$  that differ only in the  $i^*$ 'th coordinate, satisfy  $f(x) \neq f(y)$ , and  $b$  is the orientation of  $f$  along the edge  $(x, y)$ .

The subroutine **Find an influential coordinate** has the guarantee that if  $f$  has some non-negligible dependence on the coordinates outside  $T$ , then with some non-negligible probability the subroutine will return some  $i^* \in [n] \setminus T$  and  $b \in \{\text{up}, \text{down}\}$  as above. This is done by picking independently and uniformly at random two inputs  $x, y \in \{0, 1\}^n$  that are equal on their coordinates in  $T$  such that  $f(x) \neq f(y)$ , and then using “binary search” in order to decrease  $\text{distance}(x, y)$  to 1, while preserving the invariant that  $f(x) \neq f(y)$ . Specifically, given  $x, y \in \{0, 1\}^n$  such that  $f(x) \neq f(y)$  we pick an arbitrary  $z \in \{0, 1\}^n$  such that if  $V = \{i \in [n] : x_i \neq y_i\}$  is the set of the coordinates where  $x_i \neq y_i$ , then  $z_i = x_i$  for all  $i \in [n] \setminus V$ , and  $\text{distance}(z, x) = \lfloor |V|/2 \rfloor$  and  $\text{distance}(z, y) = \lceil |V|/2 \rceil$ . Since  $f(x) \neq f(y)$ , it must be the case that  $f(z)$  differs from either  $f(x)$  or  $f(y)$ . We then update either  $x$  or  $y$  to be  $z$  so that  $f(x) \neq f(y)$ . This clearly decreases  $\text{distance}(x, y)$  by roughly a multiplicative

37:4 An  $\tilde{O}(n)$  Queries Adaptive Tester for Unateness

---

```

1: procedure FIND AN INFLUENTIAL COORDINATE( $f : \{0, 1\}^n \rightarrow \{0, 1\}, T$ )
2:   Pick  $x, y \in_R \{0, 1\}^n$  independently and uniformly at random such that  $x_T = y_T$ 
3:   if  $f(x) = f(y)$  then
4:     return  $\perp$ 
5:   else ( $f(x) \neq f(y)$ )
6:     repeat
7:        $U \leftarrow \{i \in [n] : x_i = y_i\}$ 
8:        $V \leftarrow \{j \in [n] : x_j \neq y_j\}$ 
9:       Pick an arbitrary  $z_V \in \{0, 1\}^V$  such that  $|\{i \in V : z_i = y_i\}| = \lfloor |V|/2 \rfloor$ .
10:      Let  $z = x_U \circ z_V \in \{0, 1\}^n$ 
11:      if  $f(x) \neq f(z)$  then
12:         $y \leftarrow z$ 
13:      else ( $f(y) \neq f(z)$ )
14:         $x \leftarrow z$ 
15:      end if
16:    until  $|V| = 1$ 
17:    Let  $i^* \in [n]$  be the unique element in  $V$ 
18:    Let  $b \in \{up, down\}$  be the orientation of  $f$  in the edge  $(x, y)$ 
19:    return  $(i^*, b)$ 
20:  end if
21: end procedure

```

---



---

```

1: procedure UNATENESS TESTER( $f : \{0, 1\}^n \rightarrow \{0, 1\}$ )
2:   Let  $m = O(\frac{n}{\epsilon})$ 
3:   Let  $T = \emptyset$ 
4:   for  $i = 1 \dots m$  do
5:     Find an influential coordinate( $f, T$ )
6:     if returned a coordinate and a direction  $(i^*, b_{i^*})$  then
7:       Add  $i^*$  to  $T$ , and let  $b_{i^*}$  be the corresponding direction.
8:     end if
9:   end for
10:  Pick  $w \in \{0, 1\}^{[n] \setminus T}$ 
11:  Apply the monotonicity tester on  $f_{T,w}$  with respect to the directions  $\{b_i : i \in T\}$ 
12:  Return the output of the monotonicity tester.
13: end procedure

```

---

factor of 2, and so, by repeating this at most  $\log(n)$  times we obtain  $x$  and  $y$  that satisfy  $f(x) \neq f(y)$  and differ in exactly one coordinate.

For the proof of Theorem 2 we need the following two claims.

► **Claim 9.** *Let  $c > 0$  be a small constant and let  $m = \frac{2n}{c\varepsilon}$  be the number of iterations of the for loop in the Unateness tester. Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function, and let  $T \subseteq [n]$  be the set in the Unateness tester after  $m$  iterations of the for loop. Then, with high probability the set  $T$  satisfies*

$$\text{Var}_{[n] \setminus T}(f) < c\varepsilon.$$

**Proof.** Note that if in some iteration we have a subset of coordinates  $T \subseteq [n]$  such that  $\text{Var}_{[n] \setminus T}(f) \geq c\varepsilon$ , then, by Proposition 8 the variables  $x$  and  $y$  chosen in line 2 of Find an influential coordinate( $f, T$ ) will satisfy  $f(x) \neq f(y)$  with probability at least  $c\varepsilon$ . Having such  $x$  and  $y$ , let  $U \subseteq [n]$  be the coordinates where  $x$  and  $y$  are equal, and let  $V \subseteq [n]$  be the coordinates where the two strings differ. Then, in each iteration the procedure chooses  $z$  at random, such that it agrees with  $x$  and  $y$  in the coordinates where they equal, and updates  $x$  or  $y$  according to the value of  $f(z)$ , while preserving the property that  $f(x) \neq f(y)$ . Clearly, if  $z \neq y$  and  $z \neq x$ , then in each step we reduce the distance between  $x$  and  $y$ , until  $|V| = 1$ , i.e.,  $y = x + e_i$  for the unique coordinate  $i \in V$ , which is returned by the procedure together with the orientation of the edge  $(x, y)$ .

Therefore, if  $m = \frac{2n}{c\varepsilon}$ , then by Azuma's inequality with probability  $1 - e^{-\Omega(n)}$  among the  $m$  iterations at least  $\frac{c\varepsilon m}{2} = n$  iterations will have the property that either Find an influential coordinate finds a new coordinate to add to  $T$  or that  $\text{Var}_{[n] \setminus T}(f) \leq c\varepsilon$ .<sup>1</sup> On the other hand, the function  $f$  depends on at most  $n$  coordinates, and hence, after  $m = \frac{2n}{c\varepsilon}$  iterations the set  $T$  will satisfy the property

$$\text{Var}_{[n] \setminus T}(f) \leq c\varepsilon,$$

with probability at least  $1 - e^{-\Omega(n)}$ , as required. ◀

► **Claim 10.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function, and let  $T \subseteq [n]$  be such that*

$$\text{Var}_{[n] \setminus T}(f) \leq c\varepsilon$$

for some  $\varepsilon > 0$  and  $c \in (0, 1/8)$ . Then, for a random  $w \in \{0, 1\}^{[n] \setminus T}$  it holds that

$$\Pr_{w \in \{0, 1\}^{[n] \setminus T}}[\text{distance}(f_{T, w}, f) \geq \varepsilon/2] \leq 8c.$$

**Proof.** Define the function  $\text{Maj}_T : \{0, 1\}^n \rightarrow \{0, 1\}$  as

$$\text{Maj}_T(z) = \begin{cases} 1 & \text{if } \Pr_{w \in \{0, 1\}^{[n] \setminus T}}[f(z_T \circ w_{\overline{T}}) = 1] > 0.5 \\ 0 & \text{otherwise.} \end{cases}$$

That is,  $\text{Maj}_T$  depends only on the coordinates in  $T$ . By the assumption of the claim we have that for a uniformly random  $w \in \{0, 1\}^{[n] \setminus T}$  it holds that

$$\begin{aligned} \mathbb{E}_{w \in \{0, 1\}^{[n] \setminus T}}[\text{distance}(f_{T, w}, \text{Maj}_T)] &= \mathbb{E}_{z \in \{0, 1\}^T} \left[ \Pr_{w \in \{0, 1\}^{[n] \setminus T}}[f(z_T \circ w_{\overline{T}}) \neq \text{Maj}(z_T)] \right] \\ &\leq \mathbb{E}_{z \in \{0, 1\}^T} [2\text{Var}_{w \in \{0, 1\}^{[n] \setminus T}}[f(z_T \circ w_{\overline{T}})]] \\ &\leq 2c\varepsilon. \end{aligned}$$

<sup>1</sup> Formally, let  $(X_i : i \in [m])$  be Bernoulli random variables with  $X_i = 1$  if either  $\text{Var}_{[n] \setminus T}(f) \leq c\varepsilon$  or a new coordinate is added to  $T$  in the  $i$ 'th iteration, and observe that  $\Pr[X_i = 1] \geq c\varepsilon$  for all  $i \in [m]$ .

Hence, by Markov's inequality

$$\Pr_w[\text{distance}(f_{T,w}, \text{Maj}_T) \geq \varepsilon/4] \leq 8c.$$

On the other hand,

$$\text{distance}(f, \text{Maj}_T) = \mathbb{E}_{w \in \{0,1\}^{[n] \setminus T}}[\text{distance}(f_{T,w}, \text{Maj}_T)] \leq 2c\varepsilon \leq \varepsilon/4.$$

Therefore, by triangle inequality we have

$$\Pr_w[\text{distance}(f_{T,w}, f) \geq \varepsilon/2] \leq \Pr_w[\text{distance}(f_{T,w}, \text{Maj}_T) \geq \varepsilon/4] \leq 8c,$$

and the claim follows.  $\blacktriangleleft$

Theorem 2 now follows easily from the above claims.

**Proof of Theorem 2.** For a small constant  $c > 0$  let  $m = O(\frac{n}{c\varepsilon})$  be the number of iterations of the *for* loop in the Unateness tester. Let  $T \subseteq [n]$  be the set in the Unateness tester after  $m$  iterations of the *for* loop. By Claim 9 with probability 0.99 the set  $T$  satisfies

$$\text{Var}_{[n] \setminus T}(f) \leq c\varepsilon.$$

Assuming that  $T$  satisfies the above, by Claim 10 if  $f$  is  $\varepsilon$ -far from being unate, then for a uniformly random  $w \in \{0,1\}^{[n] \setminus T}$  it holds that  $f_{T,w}$  is  $\varepsilon/2$ -far from being unate with probability  $(1 - 8c)$ , and in particular, it is  $\varepsilon/2$  far from being monotone with respect to the directions  $\{b_i : i \in T\}$ . By applying the monotonicity tester on  $f_{T,w}$  with  $w$  such that  $f_{T,w}$  is  $\varepsilon/2$ -far from being unate it follows that with probability at least 0.99 the invocation of the monotonicity tester will find a violation to monotonicity of  $f_{T,w}$  with respect to the directions  $\{b_i : i \in T\}$ . Therefore, for a sufficiently small constant  $c > 0$ , if  $f$  is  $\varepsilon$ -far from unate, then with probability 0.9 the tester will reject.

Finally, we analyze the query complexity of the tester. It is clear that the procedure *Find an influential coordinate* makes at most  $O(\log(n))$  iterations, as in each iteration  $z$  differs from both  $x$  and  $y$  in at most  $\lceil \text{distance}(x, y)/2 \rceil$  coordinates. Therefore, the total number of queries made by the tester in the *for* loop is  $m \cdot O(\log(n))$ . In addition the tester makes at most  $O(n/\varepsilon)$  queries in step 11. Therefore, tester makes at most  $O(n \log(n)/\varepsilon)$  queries.  $\blacktriangleleft$

**Acknowledgements.** We are thankful to the anonymous referees for their helpful comments.

---

## References

- 1 Aleksandrs Belovs and Eric Blais. A polynomial lower bound for testing monotonicity. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2016, pages 1021–1032, New York, NY, USA, 2016. ACM. doi:10.1145/2897518.2897567.
- 2 Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Transitive-closure spanners of the hypercube and the hypergrid. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:46, 2009. URL: <http://eccc.hpi-web.de/report/2009/046>.
- 3 Eric Blais. Testing juntas nearly optimally. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC'09, pages 151–158, New York, NY, USA, 2009. ACM. doi:10.1145/1536414.1536437.
- 4 Jop Briët, Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Monotonicity testing and shortest-path routing on the cube. *Combinatorica*, 32(1):35–53, 2012.

- 5 Deeparnab Chakrabarty and C. Seshadhri. A  $o(n)$  monotonicity tester for boolean functions over the hypercube. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 411–418, 2013. doi:10.1145/2488608.2488660.
- 6 Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. Boolean function monotonicity testing requires (almost)  $n^{1/2}$  non-adaptive queries. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 519–528, 2015. doi:10.1145/2746539.2746570.
- 7 Xi Chen, Rocco A. Servedio, and Li-Yang Tan. New algorithms and lower bounds for monotonicity testing. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 286–295, 2014. doi:10.1109/FOCS.2014.38.
- 8 Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *Randomization, Approximation, and Combinatorial Algorithms and Techniques, Third International Workshop on Randomization and Approximation Techniques in Computer Science, and Second International Workshop on Approximation Algorithms for Combinatorial Optimization Problems RANDOM-APPROX'99, Proceedings. Berkeley, CA, USA, August 8-11, 1999*, pages 97–108, 1999.
- 9 Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings of the Thirty-fourth Annual ACM Symposium on Theory of Computing, STOC'02*, pages 474–483, New York, NY, USA, 2002. ACM.
- 10 Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000. doi:10.1007/s004930070011.
- 11 Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and boolean isoperimetric type theorems. In *Proceedings of the 56th Annual Symposium on Foundations of Computer Science (FOCS 2015)*, 2015.
- 12 Guy Kindler and Shmuel Safra. Noise-resistant boolean-functions are juntas, 2003. Manuscript.
- 13 Eric Lehman and Dana Ron. On disjoint chains of subsets. *J. Comb. Theory, Ser. A*, 94(2):399–404, 2001. doi:10.1006/jcta.2000.3148.