Electronic Colloquium on Computational Complexity, Report No. 82 (2013)

Some properties are not even partially testable*

Eldar Fischer[†] Yonatan Goldhirsh[‡] Oded Lachish[§]

June 6, 2013

Abstract

For a property P and a sub-property P', we say that P is *P'-partially testable with q queries* if there exists an algorithm that distinguishes, with high probability, inputs in P' from inputs ϵ-far from P by using q queries. There are natural properties that require many queries to test, but can be partitioned into a small number of subsets for which they are partially testable with very few queries, and in fact the minimal $O(1/\epsilon)$.

We prove that this is not always the case. More than that, we prove the existence of a property P such that the only subsets P' for which P is P'-partially testable are very small. To prove this we introduce new techniques for proving property testing lower bounds. In addition to obtaining some broad-brush criteria for non-testability, this implies a lower bound on the possibility of PCPPs with a sublinear proof size. This also implies lower bounds on MAPs, a notion newly defined by Gur and Rothblum.

The new techniques rely on analyzing a proposed partial tester. We show that the queries performed by a tester must, with high probability, query indexes where a uniformly random member of the sub-property has low entropy. We then show how one can aggregate the "entropy loss" to deduce that a random choice in the sub-property must have low entropy, and therefore the sub-property must be small.

We develop two techniques for aggregating the entropy loss. A simpler technique that applies to non-adaptive testers is based on partitioning the input bits into high query probability parts and parts where there is an entropy loss when conditioned on the high probability parts. Against adaptive testers we develop a more intricate technique based on constructing a decision tree. The root-to-leaf paths in this tree rearrange the input into parts where each part exhibits entropy loss when conditioned on the path prefix. This decision tree is constructed by combining carefully selected decision trees from those used by the adaptive testing algorithm.

[*] The research leading to these results has received funding from the European Union's - Seventh Framework Programme [FP7/2007-2013] under grant agreement n 202405 (PROPERTY TESTING).

[†] Department of Computer Science, Technion, Haifa 32000, Israel. eldar@cs.technion.ac.il

[‡] Department of Computer Science, Technion, Haifa 32000, Israel. jongold@cs.technion.ac.il

[§] Birkbeck, University of London, London, UK. oded@dcs.bbk.ac.uk

ISSN 1433-8092

1 Introduction

Property Testing deals with randomized approximation algorithms that operate under low information situations. Formally, we deal with objects from some universe U parametrized by an integer n, usually Σ^n where Σ is some finite alphabet; with a notion of *distance* between two objects in U, usually the Hamming distance; and with a notion of a *query* to an object in U, usually corresponding to retrieving x_i for an index $i \in \{1, \dots, n\}$.

Definition 1.1 (Testable property)**.** *Let $P \subseteq \{0,1\}^n$. We say that P is* testable with q queries *if there exists an algorithm A that gets as input a parameter $\epsilon > 0$ and query access to an input string $x \in \{0,1\}^n$ and outputs* accept *or* reject *such that:*

- *If $x \in P$, then A accepts with probability at least $2/3$.*
- *If $d(x,P) > \epsilon$, then A rejects with probability at least $2/3$.*

If furthermore all queries performed to the input can be decided before any of them are made, then the algorithm is non-adaptive*, and otherwise it is* adaptive*. If we require that whenever $x \in P$, then the algorithm accepts with probability 1, then the algorithm is* 1-sided*, and otherwise it is* 2-sided*.*

Property Testing was first addressed by Blum, Luby and Rubinfeld [6], and most of its general notions were first formulated by Rubinfeld and Sudan [19]. The first investigated properties were mostly of an algebraic nature, such as the property of a Boolean function being linear. The first investigation of combinatorial properties and the formal definition of testability was by Goldreich, Goldwasser and Ron [11]. Since then Property Testing has attracted significant attention. For surveys see [8, 16, 17].

When proving that testing a property requires many queries, one might ask "how strong is this requirement?", which can be illustrated with an example. Alon et. al. [3] studied the testability of formal languages, and proved that the language $L = \{uu^R vv^R | u, v \in \{0,1\}^*\}$ requires at least $\Omega(\sqrt{n})$ queries to test. Informally, one may say that the "reason" for this language being untestable is the difficulty in guessing the length of uu^R. This can be made formal by considering the languages $L_i = \{uu^R vv^R | u, v \in \{0,1\}^*, |u| = i\}$, which form a partition of L. A simple sampling algorithm can perform $O(\epsilon^{-1})$ queries to an input and distinguish between inputs in L_i and inputs ϵ-far from L. It is also important to note that $|L \cap \{0,1\}^n| = 2^{\Theta(n)}$, but its partition $L_0 \cap \{0,1\}^n, \dots, L_n \cap \{0,1\}^n$ is only to a number of subsets linear in n.

This phenomenon is not unique to the language considered by Alon et. al. Another example is that of graph isomorphism, first considered in the property testing framework by Alon et. al. [2] (and later by Fischer and Matsliah [9]), and shown to require at least $\Omega(n)$ queries to test. In this setting we consider a pair of unknown graphs given by their adjacency matrices, and we are charged with distinguishing the case where they are isomorphic from the case where more than ϵn^2 of their edges must be changed to make them isomorphic. In this case, the size of the property is $2^{\Theta(n^2)}$, and we can partition the property into $n!$ properties $\{P_\pi | \pi \in S_n\}$, each defined by $P_\pi = \{(G_1, G_2) | \pi(G_1) = G_2\}$, such that a sampling algorithm can perform $O(\epsilon^{-1})$ queries to an input and distinguish between inputs in P_π and inputs ϵ-far from the original property.

Thus it is tempting to ask whether this is a general phenomenon. Can any property P be partitioned into $k = |P|^{o(1)}$ properties $P_1, \dots, P_k$ such that the task of distinguishing inputs in P_i from inputs far from P can be performed with a number of queries that depends only on ϵ? The main result of the present paper is to prove that this is not always the case. In fact, there exist properties for which any such partition must be to a number of subsets exponential in n.

To prove this result we consider a simpler setting that implies it.

Definition 1.2 (Partially testable property). *Let $P \subseteq \{0,1\}^n$ and $P' \subseteq P$. We say that P is P'-*partially testable with q queries *if there exists an algorithm A that gets as input a parameter $\epsilon > 0$ and query access to an input string $x \in \{0,1\}^n$ and outputs* accept *or* reject *such that:*

- *If $x \in P'$, then A accepts with probability at least $2/3$.*
- *If $d(x, P) > \epsilon$, then A rejects with probability at least $2/3$.*

If furthermore all queries performed to the input can be decided before any of them are made, then the algorithm is non-adaptive*, and else it is* adaptive*.*

Obviously, if P is testable with q queries, then for any subset $P' \subseteq P$ it is P'-partially testable with the same number of queries. On the other hand, for any property P and any element $x \in P$, we have that P is $\{x\}$-partially testable with $O(\epsilon^{-1})$ queries.

The partitions described above are in fact partitions of P into subsets $P_1, \dots, P_k$ such that P is P_i-partially testable for every $1 \leq i \leq k$. If there exists such a partition into not too many sets, then there must be at least one set that is relatively large. Our main result shows that there exists a property P for which all subsets $P' \subseteq P$ such that P is P'-partially testable are small. In fact, all linear codes with large dual distance define such properties.

Theorem 1.3. *Let $C \subseteq \{0,1\}^n$ be a linear code of size $|C| \leq 2^{\frac{1}{64}n}$ and dual distance Γ. For every $C' \subseteq C$, if C is C'-partially testable with q adaptive queries, then $|C'| \leq |C| 2^{-\Theta\left(\frac{\Gamma}{q}\right)}$.*

We will first prove a weak version of Theorem 1.3 in Section 4 which will apply for q non-adaptive queries and imply the bound $|C'| \leq |C| 2^{-\Theta\left(\frac{\Gamma}{q^3}\right)}$. This proof will use some of the key ideas that will later manifest in the proof of the theorem in its full generality in Section 5.

An important question is the existence of codes with strong parameters. A random linear code C will have $\Gamma = \Theta(n)$ and $|C| = 2^{\Theta(n)}$ with high probability (this is implied by the Gilbert-Varshamov bound [10, 20]; MacWilliams et. al. [15] showed that this can also be obtained by codes which are self-dual and thus also have good distance), and thus by Theorem 1.3 we will have that for any $C' \subseteq C$ such that C is C'-partially testable with q queries, $|C'| \leq |C| 2^{-\Theta\left(\frac{n}{q}\right)}$. For a constant q, this implies that partial testability will only be possible with exponentially small subsets. The best explicit (and low uniform decision complexity) construction known to us is that of [1], which gives $|C| = 2^{\Theta(n)}$ with $\Gamma = \Theta(n/\log n)$, and thus the bound becomes $|C'| \leq n^{O(1)} |C| 2^{-\Theta\left(\frac{n}{q}\right)}$, which is polynomially worse than the non-explicit bound, but is still a strong upper bound on the size of C'.

Theorem 1.3 implies that there exist properties P that require a lot of queries to test, and that every partition of P into subsets $P_1, \dots, P_k$ such that P is P_i-partially testable for every $1 \leq i \leq k$ requires that k will be very big. One might ask if we can prove a converse. That is, if P can be tested with a few queries, can we find such a partition with a small k?

Open Problem 1.4. *Let P be a property testable with r queries. Is it true that we can partition P into subsets $P_1, \dots, P_k$ such that P is P_i-partially testable with $O(1)$ queries for every $1 \leq i \leq k$ and k is bounded by some moderate function of r?*

Theorem 1.3 implies that for some properties, k might be as big as $2^{\Theta\left(\frac{n}{q}\right)}$. It is not clear whether this value of k can always be obtained. The trivial upper bound for every property is by partitioning into 2^{n-q} subsets of size 2^q. Are there properties for which this is required?

Open Problem 1.5. *Does there exist a property P such that for every $P' \subseteq P$ where P is P'-partially testable with q queries we also have $|P'| \leq |P|2^{\Theta(q)-\Theta(n)}$?*

1.1 Related work

Our notion of a partition is similar to existing notions in computational complexity. For a partition $P = P_1 \cup P_2 \cup \ldots \cup P_k$ where for every $1 \leq i \leq k$, P is P_i-partially testable, the designation of P_i can be seen as a "proof" that a certain x is in P. If $x \in P$, then there exists some P_i such that $x \in P_i$ and therefore a P_i-partial tester for P will accept it with high probability. If x is ϵ-far from P, then all P_i-partial testers for P will reject it with high probability.

This is similar to the notion of a *Probabilistically Checkable Proof of Proximity (PCPP)*, first introduced by Ben-Sasson et. al. [4]. PCPPs are to property testing as NP is to P. A q query PCPP for a property $P \subset U$ is an algorithm that gets as input $x \in U$ and a *proof of proximity* $\pi \in \{0,1\}^l$. The algorithm must perform at most q queries to x and π and fulfill the requirement that if $x \in P$ then there exists a proof π that causes the algorithm to accept with high probability, but when x is ϵ-far from P then for any proof π the algorithm rejects with high probability. In our setting, the algorithm is allowed free access to a proof of length $l = \log(k)$, but we expect l to be sublinear in the size of x. In particular, the property we analyze here cannot have a PCPP with a sublinear length proof.

Rothblum et. al. [18] introduced the notion of an *Interactive Proof of Proximity (IPP).* In an IPP for a property P, the tester can also communicate with a *prover* in addition to querying the input x. If $x \in P$ then the prover has a strategy that will cause the tester to accept with high probability. When x is ϵ-far from P, the prover cannot make the tester accept with high probability. Rothblum et. al. show that all languages in NC admit such a protocol with $\sqrt{n}$ query and communication complexity and polylog(n) communication rounds. Protocols of this kind are only interesting for the case where the communication complexity is sublinear, or else the prover may just give the input to the tester.

Independently of the present work, Gur and Rothblum [13] weakened the IPP model to create *Merlin-Arthur Proofs of Proximity (MAP).* Gur and Rothblum define a MAP as a proof-system for a property P where for an input x and a proof π the verifier reads the entire proof π and queries q bits from the input x. If $x \in P$, then there exists a proof π such that the verifier accepts with high probability, and if x is far from P, then for every proof π the verifier rejects with high probability. Since we can trivially set $\pi = x$, the only interesting cases are where the length of π is sublinear.

The notion of a MAP with q queries and proofs of length ℓ for a property P is equivalent to the existence of $k = 2^\ell$ sets $P_1, \ldots, P_k$ such that $P = P_1 \cup P_2 \cup \ldots \cup P_k$ where for every $1 \leq i \leq k$, P is P_i-partially testable with q queries.

Gur and Rothblum give several constructions of properties where a MAP with a sublinear length proof greatly reduces query complexity. Gur and Rothblum also introduce the *Tensor Sum* family of properties, for which they prove that for every constant $\alpha > 0$ there exists an instantiation of Tensor Sum such that any MAP for it that performs q queries must require a proof of length $\Omega\left(\frac{n^{1-\alpha}}{q}\right)$. This is slightly weaker than Theorem 1.3 proved in the present paper. Their lower bound is proved by an extension of the communication complexity technique of Brody et.al. [5] to

Merlin-Arthur communication complexity. This proof technique is fitting for the MAP setting, but does not apply to partial testing in general. Gur and Rothblum also prove that this trade off is almost optimal for the Tensor Sum properties.

Additionally, Gur and Rothblum show separations between the power of MAPs and that of IPPs and PCPPs. They also show that 2-sidedness may only give a MAP a polylog(n) factor improvement in proof length and query complexity over a 1-sided algorithm. Their result implies a connection also between 1-sided and 2-sided partial testability, though not one that would preserve O(1)-query partial testability.

2 Proof plan

For the proofs of our main result we develop new techniques that are in some ways more flexible than the traditional use of Yao's method for proving property testing lower bounds. We believe that these techniques hold promise for other lower bound situations where using Yao's method seems to hit a wall.

2.1 General themes

As with Yao's method, we contrast the behavior of a supposed test when it is run over an input chosen according to some distribution over "yes" instances, with its behavior when it is run over an input chosen according to some distribution over "no" instances. However, while in the traditional method these two distributions are chosen based only on the property (and should work against all possible algorithms of a given class), here the distributions are in fact *tailor made* for the specific analyzed algorithm. Note that special care must be taken in the definition of such an input distribution. It may not depend on the "real-time" behavior of the algorithm (i.e. it may not adapt itself to the identity of the random queries that the algorithm has made), and is instead constructed based only on the *description* of the algorithm.

The second theme is the use of *Shannon entropy.* Our goal here is to prove that if C is C'-partially testable, then C' cannot be too large. For achieving this we assume that a testing algorithm exists, and then contrast a uniformly random choice of a word in C' with another word chosen from a "dangerous" distribution over words far from C. The assumption that the test in fact distinguishes the two distributions allows us to show that a uniformly random choice of a word in C' has low entropy, and hence C' must be small. Using entropy instead of direct counting is crucial to using our main method for obtaining a bound against 2-sided error tests, rather than only 1-sided error ones.

2.2 Proving a bound against non-adaptive algorithms

The bound against non-adaptive algorithms showcases the above general themes. A supposed C'-partial test with q queries is in essence a distribution over query sets of size q, such that with high probability a query set is chosen that highlights a difference between members of C' and inputs far from being in C. Now if the test were additionally 1-sided, this would translate to forbidden values in the meaningful query sets, which would give a cross product bound on the size of C'. For 2-sided tests we use an analogous feature of entropy, namely that of subadditivity.

To construct a "dangerous" distribution over words far from being in C, we first take note of the "heavy" indexes, which are those bits of the input that are with high probability part of the query subset of the investigated testing algorithm. There will be only few of those, and our distribution over far words would be that of starting with a restriction of a uniformly random word in C' to the set of heavy indexes, and augmenting it with independently and uniformly chosen values to all other input bits. When contrasted with the uniform distribution over all members of C', we obtain that there must be many query sets that show a distinction between the two distributions over the non-heavy indexes with respect to the heavy ones. This means that the values of the non-heavy indexes in each such query set do not behave like a uniformly independent choice, and thus have a corresponding entropy (conditioned on the heavy index bits) that is significantly less than the maximal possible entropy. Having many such query sets in essence means that we can find many such sets that are disjoint outside the heavy indexes, which in turn leads to an entropy bound by virtue of subadditivity (when coupled with general properties of linear codes).

2.3 Proving a bound against adaptive algorithms

An adaptive algorithm cannot be described as a distribution over query sets, but rather as a distribution over small decision trees of height q that determine the queries. Therefore low-entropy index sets cannot be readily found (and in fact do not always exist). To deal with this we employ a new technique, that allows us to "rearrange" the input in a way that preserves entropy, but now admits disjoint low-entropy sets.

This new construction is a *reader*, which in essence is an adaptive algorithm that reads the entire input bit by bit (without repetitions). As this adaptive algorithm always eventually reads the entire input, it defines a bijection between the input to be read and the "reading stream", i.e. the sequence of values in the order that the reader has read them.

The construction of this reader is fully based on the description of the q-query adaptive algorithm that C'-partially tests for C (again we assume that such an algorithm exists). In fact we contrast the uniform distribution over members of C' with not one but many possible distributions over inputs far from C. At every stage we obtain that, as long as our reader has not yet read a large portion of the input, the adaptive test can provide a decision tree over the yet-unread bits that shows a difference between a uniformly random member of C' (conditioned on the values of the bits already read) and an independently uniform random choice of values for the unread bits. Our reader will be the result of "concatenating" such decision trees as long as there are enough unread bits. Thus in the "reading stream" we have sets of q consecutive bits, each with low entropy (as it is distinguishable from independently uniform values). When there are not enough unread bits left, we read all remaining bits arbitrarily, and use general properties of codes to bound the entropy on that final chunk.

The method of constructing a reader not only allows us to do away with the exponential penalty usually associated with moving from non-adaptive to adaptive algorithms, but we additionally obtain better bounds for non-adaptive algorithms as well. This is because a reader can do away also with the penalty of moving from the situation of having many low-entropy query sets to having a family of sets disjoint outside the heavy indexes, in essence by constructing the reader for the uniform distribution over C' based on not one but many "dangerous" input distributions.

3 Preliminaries

Below we introduce the reader to some basic definitions and results regarding entropy and the dual distance of codes. We refer the reader who is interested in a more thorough introduction of entropy to [7, Chapter 2].

First, we introduce a standard notion of distance between distributions.

Definition 3.1 (Total variation distance)**.** *Let p and q be two distributions over the domain $\mathcal{D}$. The total variation distance between p and q is defined to be $d_{TV}(p,q) = \frac{1}{2}\sum_{i\in\mathcal{D}} |p(i) - q(i)|$.*

We now introduce the notion of the entropy of a random variable, the entropy of a random variable conditioned on another one, and two well-known lemmas.

Definition 3.2 (Entropy)**.** *Let X be a random variable over the domain $\mathcal{D}$. The* entropy of X *is defined to be $H[X] = -\sum_{i\in\mathcal{D}} \Pr[X = i] \log(\Pr[X = i])$.*

Definition 3.3 (Conditional entropy)**.** *Let X and Y be random variables over the domain $\mathcal{D}$. The* entropy of X conditioned on Y *is defined to be $H[X|Y] = \sum_{y\in\mathcal{D}} \Pr[Y = i]H[X|Y = y]$.*

Lemma 3.4 (The chain rule)**.** *Assume that X and Y are random variables. The entropy of the combined state determined by both random variables is denoted $H[X,Y]$. This quantity obeys the chain rule $H[X,Y] = H[X|Y] + H[Y]$.*

Lemma 3.5 (Subadditivity)**.** *If X and Y are random variables, then $H[X,Y] \leq H[X] + H[Y]$.*

The total variation distance is not a natural fit to the context of entropy. A more fitting notion of distance between distributions is divergence.

Definition 3.6 (Divergence)**.** *Let p and q be two distributions over $\mathcal{D}$. The* divergence of q from p *is defined to be $D(p\|q) = \sum_{i\in\mathcal{D}} p(i) \log\left(\frac{p(i)}{q(i)}\right)$.*

Fortunately, divergence and total variation distance are related via Pinsker's inequality.

Lemma 3.7 (Pinsker's inequality)**.** *Assume that p and q are two distributions over the domain $\mathcal{D}$. The total variation distance between p and q is related to the divergence of q from p by the inequality $\sqrt{\frac{1}{2}D(p||q)} \geq d_{TV}(p,q)$.*

We will actually be using a simpler corollary of it.

Lemma 3.8 (Corollary of Pinsker's inequality)**.** *Assume that X is a random variable distributed according to p over $\mathcal{D}$, and denote the uniform distribution over $\mathcal{D}$ by p_u. The entropy of X is related to its total variation distance from the uniform distribution by $H[X] \leq \log(|\mathcal{D}|) - 2(d_{TV}(p,p_u))^2$.*

Proof.

$$H[X] = -\sum_{i\in\mathcal{D}} \Pr[X = i] \log(\Pr[X = i])$$

$$= -\sum_{i\in\mathcal{D}} \Pr[X = i] \log(\Pr[X = i] \cdot \frac{1}{|\mathcal{D}|} \cdot |\mathcal{D}|)$$

$$= -\sum_{i\in\mathcal{D}} \Pr[X = i] \log(\frac{1}{|\mathcal{D}|}) - \sum_{i\in\mathcal{D}} \Pr[X = i] \log(\Pr[X = i] \cdot |\mathcal{D}|)$$

$$= \log(|\mathcal{D}|) - D(p\|p_u) \leq \log(|\mathcal{D}|) - 2(d_{TV}(p,p_u))^2$$

Where the last step follows from Pinsker's inequality. □

Let $x \in \{0,1\}^n$ and $J \subseteq [n]$. We use $x[J]$ to denote the restriction of x to the indices in J. That is, the vector $< x_j >_{j \in J}$. When $C \subseteq \{0,1\}^n$ we use $C[J] = \{x[J] | x \in C\}$.

Let $C \subseteq \{0,1\}^n$. We denote by $U(C)$ the uniform distribution over C. In accordance with the notation above, when $X \sim U(C)$, $X[J]$ denotes the random variable obtained by drawing uniformly from C and then restricting to the indices in J. As a shorthand we use $U(C)[J]$ for the distribution of $X[J]$. We use $U_J(C)$ to denote the result of first drawing a vector x according to $U(C)$, and then replacing $x\,[[n] \setminus J]$ with a uniformly random vector in $\{0,1\}^{n-|J|}$. In particular, in many cases we will take C to be a singleton, in which case we drop the curly braces and denote this probability distribution by $U_J(x)$.

We will make inherent use of the following result, which can be found e.g. in [14, Chapter 1, Theorem 10].

Lemma 3.9. *If $J \subseteq [n]$ is such that $|J| < \Gamma$ and $X \sim U(C)$, then $X[J]$ is distributed uniformly over $\{0,1\}^{|J|}$.*

We will also need the fact that a mostly random input is far from a code with high probability.

Lemma 3.10. *Let $C \subseteq \{0,1\}^n$ such that $|C| \leq 2^{\frac{1}{64}n}$, $\epsilon < 1/8$, and let $J \subseteq [n]$ be such that $|J| \leq n/2$. $X \sim U_J(C)$ is ϵ-far from C with probability $1 - o(1)$. Furthermore, this is still true when conditioned on any value of $X[J]$.*

Proof. By Chernoff bounds, the probability that a random element $X \sim U_J(C)$ will agree with $c \in C$ in more than $(1-\epsilon)n$ coordinates is at most $\exp\left(-n(1/4-\epsilon)^2\right)$. Taking the union bound over all $c \in C$ gives us $|C| \cdot \exp\left(-n(1/4-\epsilon)^2\right) = o(1)$. Since this calculation assumes that $X[J]$ always agrees with $c[J]$, it holds when conditioned on any value of $X[J]$. □

Finally, we will also need to use Lemma 3.9 to help us calculate the entropy of uniform random variables in codes.

Lemma 3.11. *Let C be a code with dual distance Γ, $J \subseteq [n]$ such that $|J| \leq \Gamma$, $C' \subseteq C$ and $X \sim U(C')$. Then $H[X|X[J]] \leq \log|C| - |J|$. Furthermore, this is true when conditioned on any particular value of $X[J]$.*

Proof. We can partition C according to the values of the bits in J:

$$C = \bigcup_{z \in \{0,1\}^{|J|}} \{c \in C | c[J] = z\}$$

By Lemma 3.9, all sets on the right hand side are of size $2^{-|J|}|C|$. Obviously, for all $z \in \{0,1\}^{|J|}$, we have $\{c' \in C' | c'[J] = z\} \subseteq \{c \in C | c[J] = z\}$, simply because $C' \subseteq C$. Thus for every $x \in C'[J]$, we have that

$$H[X|X[J] = x] \leq \log|\{c' \in C' | c'[J] = z\}| \leq \log|\{c \in C | c[J] = z\}|.$$

This completes the "furthermore" part of the lemma. To obtain the non-conditioned version, note that by the definition of conditional entropy,

$$H[X|X[J]] = \mathrm{E}_{x \sim U(C')[J]} H[X|X[J] = x] \leq \log\left(2^{-|J|}|C|\right) = \log|C| - |J|.$$

□

We note (and use throughout) that trivially $H[X|X[J]] = H[X[\{1, \dots, n\} \setminus J]|X[J]]$.

4 Nonadaptive lower bound

In this section we prove Theorem 1.3 for the case of a non-adaptive tester and with slightly worse quantitative bounds. For the rest of this section, set $C \subset \{0,1\}^n$ to be a code with dual distance Γ and $|C| \leq 2^{\frac{1}{64}n}$. Set $\epsilon < 1/8$ and assume that C is C'-partially testable for $C' \subseteq C$ with q non-adaptive queries.

Next we define a non-adaptive tester for a property. This definition is consistent with the standard one.

Definition 4.1 (Non-adaptive property tester)**.** *A non-adaptive* ϵ-tester *for a code $C \subseteq \{0,1\}^n$ with query complexity $q(\epsilon, n)$ is defined by a collection of query sets $\{Q_i\}_{i \in I}$ of size q together with a predicate π_i for each query set and a distribution μ over I which satisfies:*

- *If $x \in C$, then with probability at least $2/3$ an $i \in I$ is picked such that $\pi_i(x[Q_i]) = 1$.*
- *If $d(x, C) > \epsilon$, then with probability at least $2/3$ an $i \in I$ is picked such that $\pi_i(x[Q_i]) = 0$.*

For a C'-partial tester the first item must hold only for $x \in C'$.

Set a non-adaptive tester for C', and let $\{Q_i\}_{i \in I}$ be its query sets.

We will be interested only in those query sets which are useful for telling a random element in C' from a mostly random element in $\{0,1\}^n$.

Definition 4.2 (J-Discerning query set)**.** *Let $J \subseteq [n]$ be such that $|J| \leq n/2$. A query set Q_i is a* J-discerning *set if $d_{TV}(U(C')[Q_i], U_J(C')[Q_i]) \geq 1/8$.*

Next we prove that a tester must have a lot of such good query sets.

Lemma 4.3. *Set $J \subseteq [n]$ such that $|J| \leq n/2$. With probability at least $1/9$ the query set Q_i picked by the tester is a J-discerning set.*

Proof. Assume the contrary, that is, that with probability greater than $8/9$ the query set Q_i picked by the tester is such that $d_{TV}(U(C')[Q_i], U_J(C')[Q_i]) < 1/8$.

Thus for every such Q_i,

$$|\Pr_{U(C')[Q_i]}[\text{tester accepts}] - \Pr_{U_J(C')[Q_i]}[\text{tester accepts}]| < 1/8.$$

For the case where the query set picked is not discerning, which occurs with probability smaller than $1/9$, we have no bound (better than 1) on the difference in probability.

Overall, over the randomness of the tester,

$$|\Pr_{U(C')}[\text{tester accepts}] - \Pr_{U_J(C')}[\text{tester accepts}]| < 8/9 \cdot 1/8 + 1/9 = 2/9.$$

But by the correctness of the tester and Lemma 3.10, we arrive at $\Pr_{U(C')}[\text{tester accepts}] \geq 2/3$ and $\Pr_{U_J(C')}[\text{tester accepts}] \leq 1/3$, a contradiction. □

We will later want to construct a collection of J-discerning sets disjoint outside of a small fixed portion of the input. Towards this end we prove that J-discerning sets show difference between an element in C' and a mostly random element in $\{0,1\}^n$ even when we only look outside of J.

Lemma 4.4. *Assume that Q_i is a J-discerning set, draw $Z \sim U(C')[J]$ and then draw $X \sim U(C')[Q_i]$ conditioned on $X[J] = Z$. With probability at least $1/15$, the distribution of $X[Q_i \setminus J]$ is $1/16$-far from $U(\{0,1\}^{|Q_i \setminus J|})$.*

Proof. First note that the distance between $U(C')[Q_i]$ and $U_J(C')[Q_i]$ is the expectation over Z of the distance of $X[Q_i \setminus J]$ from $U(\{0,1\}^{|Q_i \setminus J|})$, conditioned on $X[J] = Z$. By definition, that is at least $1/8$. By simple probability bounds, with probability at least $1/15$, Z is such that the distance of $X[Q_i \setminus J]$ from $U(\{0,1\}^{|Q_i \setminus J|})$ conditioned on $X[J] = Z$ is at least $1/16$. □

However, total variation distance is not very handy for counting. We now use Lemma 3.8 to transform our total variation bounds into "entropy loss" bounds.

Lemma 4.5. *If Q_i is a J-discerning set and $X \sim U(C')[Q_i]$, then $H[X[Q_i \setminus J]|X[J]] \leq |Q_i \setminus J| - 0.0005$.*

Proof. Let $L \subseteq \{0,1\}^{|J|}$ be the set of values $z \in \{0,1\}^{|J|}$ such that when drawing $X \sim U(C')[Q_i]$ conditioned on $X[J] = z$, the distribution of $X[Q_i \setminus J]$ is $1/16$-far from $U(\{0,1\}^{|Q_i \setminus J|})$.

Since the entropy is non-negative, we can upper bound

$$H[X[Q_i \setminus J]|X[J]] \leq \sum_{z \in L} \Pr_{Z \sim U(C')[J]}[Z = z] H[[Q_i \setminus J]|X[J] = z] + \sum_{z \in \{0,1\}^J \setminus L} \Pr_{Z \sim U(C')[J]}[Z = z]|Q_i \setminus J|.$$

To treat the first summand on the right hand side, we invoke Lemma 3.8 to obtain

$$H[[Q_i \setminus J]|X[J] = z] \leq |Q_i \setminus J| - 0.007.$$

Overall we get

$$\sum_{z \in L} \Pr_{Z \sim U(C')[J]}[Z = z] H[[Q_i \setminus J]|X[J] = z] + \sum_{z \in \{0,1\}^J \setminus L} \Pr_{Z \sim U(C')[J]}[Z = z]|Q_i \setminus J| \leq |Q_i \setminus J| - 0.0005.$$

□

Next we would try to cover the indices in $[n]$ with as many discerning sets as possible. To later sum up the entropy loss, we need these sets to be disjoint outside a not-too-big set. We determine this set of "bad" indices as the set of bits read (non-adaptively) by the tester with the highest probability.

Definition 4.6. *Define $B = \{k \in [n] | \Pr_{Q \sim \mu}[k \in Q] \geq \frac{2q}{\Gamma}\}$.*

Observation 4.7. *$|B| \leq \Gamma/2 \leq n/2$. Therefore Lemma 3.10 holds with $I = B$.*

Now we can prove that we can find many B-discerning sets which are disjoint outside of B.

Lemma 4.8. *There exists a set I_D such that:*

- *For all $i \in I_D$, Q_i is a B-discerning set*
- *For all $i, j \in I_D$, $Q_i \setminus B$ and $Q_j \setminus B$ are disjoint*

$D = \cup_{i \in I_D}(Q_i \setminus B)$ satisfies $\Gamma/2 \geq |D| \geq \frac{\Gamma}{18q^2}$. Additionally, $|I_D| \geq \frac{\Gamma}{18q^3}$.

Proof. We construct the set I_D greedily. Suppose that we have discerning sets covering k bits that are disjoint outside of B. Choose a set randomly using the tester's distribution conditioned on it being B-discerning. This increases the probability of every query set, and every bit to be in a query set, by at most 9. By the definition of B, if we choose a query set randomly using the tester's distribution, the probability that it intersects our already covered bits outside of B is at most $9\frac{2q^2}{\Gamma}k$. As long as this number is smaller than 1, such a set exists. Therefore, as long as $k < \frac{\Gamma}{18q^2}$ we have a set to add, leading to the bound. To get the upper bound on $|D|$ we can just stop the process before D gets too big. □

Finally, we are ready to calculate the entropy of a uniformly random codeword from C'. We use the chain rule to split this into calculating the entropy of the bits in B, the entropy of the bits in D conditioned on the bits of B, and the entropy of everything else conditioned on the bits in $D \cup B$.

Lemma 4.9. *If $X \sim U(C')$, then $H[X] \leq \log|C| - 0.0005\frac{\Gamma}{18q^3}$*

Proof. First, by the chain rule for entropy and the fact that $D \setminus B = D$,

$$H[X] = H[X|X[D \cup B]] + H[X[D]|X[B]] + H[X[B]]$$

We proceed by bounding each element in the sum. First, trivially:

$$H[X[B]] \leq |B|$$

Next, invoke Lemma 3.11 for $D \cup B$, since $|D \cup B| \leq \Gamma$. This gives us:

$$H[X|X[D \cup B]] \leq \log|C| - |D \cup B|$$

Now, recall that $\cup_{i \in I_D}(Q_i \setminus B) = D$. Since these sets are disjoint outside of B, we employ subadditivity to get:

$$H[X[D \setminus B]|X[B]] \leq \sum_{i \in I_D} H[X[Q_i \setminus B]|X[B]]$$

Now, since these are all B-discerning sets, by Lemma 4.5 we know that for all $i \in I_D$ we have that $H[X[Q_i \setminus B]|X[B]] \leq |Q_i \setminus B| - 0.0005$. By Lemma 4.8 we know that $|I_D| \geq \frac{\Gamma}{18q^3}$. Summing up we get:

$$\sum_{i \in I_D} H[X[Q_i \setminus B]|X[B]] \leq |D| - 0.0005|I_D|$$

$$\leq |D| - 0.0005\frac{\Gamma}{18q^3}$$

That is,

$$H[X[D]|X[B]] \leq |D| - 0.0005\frac{\Gamma}{18q^3}$$

Summing everything up we get the statement of the lemma. □

From this it follows that:

Theorem 4.10 (Weak form of the main theorem)**.** *Let $C' \subseteq C$, if C is C'-partially testable with q non-adaptive queries, then*

$$|C'| = 2^{H[X]} \leq |C|2^{-0.0005\frac{\Gamma}{18q^3}}.$$

5 Adaptive lower bound

In this section we prove Theorem 1.3 in its full generality. We start by introducing the mechanism of a *reader*, which allows us to separate the adaptivity and randomness of the algorithm.

Definition 5.1 (Reader)**.** *A k-*reader *r is a sequence $r_0, r_1, \ldots, r_{k-1}$, where $r_i : \{0,1\}^i \to \{1,\ldots,n\}$ satisfy for all $i < j$ and $y \in \{0,1\}^j$ that $r_i(y[\{1,\ldots,i\}]) \neq r_j(y)$.*

Given an input $x \in \{0,1\}^n$, the reader defines a sequence of its bits. This is the *reading* of x.

Definition 5.2 (Reading)**.** *Given $x \in \{0,1\}^n$ and a k-reader r, the* reading *$R_{r(x)}$ of x according to r is a sequence $y_1, \ldots, y_k$ defined inductively by $y_{i+1} = x_{r_i(y_1,\ldots,y_i)}$. We define $r_i(x)$ to be $r_i(y_1,\ldots,y_i)$. The set of* unread bits *$U_{r(x)}$ is the subset of $\{1,\ldots,n\}$ that did not appear as values of $r_1,\ldots,r_k$ in the reading.*

We can now define an adaptive tester as a distribution over readers and decision predicates.

Definition 5.3 (Adaptive tester)**.** *An* adaptive *ϵ-tester for a code $C \subseteq \{0,1\}^n$ with query complexity $q = q(\epsilon, n)$ is defined by a collection of q-readers $\{r^i\}_{i\in I}$ together with predicates π_i for each reader, and a distribution μ over I which satisfies:*

- *For all $x \in C$, $\Pr_{i\sim\mu}\left[\pi_i(R_{r^i(x)}) = 1\right] \geq 2/3$.*
- *For all $x \in \{0,1\}^n$ such that $d(x, C) > \epsilon$, $\Pr_{i\sim\mu}\left[\pi_i(R_{r^i(x)}) = 0\right] \geq 2/3$.*

Part of the usefulness of readers is that if we can construct a reader that reads the entire input, then reading the property C' through it preserves its size.

Observation 5.4. *If r is an n-reader, then the function mapping every $x \in \{0,1\}^n$ to its reading $R_{r(x)}$ is a bijection.*

Proof. Suppose that $x' \neq x$, and let $i \in \{1,\ldots,n\}$ be the least index such that $x_{r_i(x)} \neq x'_{r_i(x)}$. Such an i must exist since r reads all bits, and $x' \neq x$. Note that $r_i(x) = r_i(x')$, since it is the first bit read to be different (and thus $y_1,\ldots,y_i = y'_1,\ldots,y'_i$). Thus $x_{r_i(x)} \neq x'_{r_i(x')}$ and therefore $R_{r(x)} \neq R_{r(x')}$. □

In light of the above, we will construct an n-reader and bound the size of C' when permuted by its reading. However, while the end product of the construction is an n-reader, the intermediate steps might not be k-readers for any k. Thus we need to introduce a more general notion.

Definition 5.5 (Generalized reader). *A* generalized reader *r is a sequence $r_0, r_1, \ldots, r_{n-1}$ where $r_i : \{0,1\}^i \to \{1,\ldots,n\} \cup \{\star\}$ satisfy for all $i < j$ and $y \in \{0,1\}^j$ one of the following*

- $r_i(y[\{1,\ldots,i\}]) \in \{1,\ldots,n\} \setminus r_j(y)$
- $r_i(y[\{1,\ldots,i\}]) = r_j(y) = \star$

Given a generalized reader r, a terminal sequence *in it is $y \in \{0,1\}^i$ such that $r_i(y_1,\ldots,y_i) = \star$, while $r_{i-1}(y_1,\ldots,y_{i-1}) \neq \star$ or $i = 0$.*

If we fix a certain $x \in \{0,1\}^n$, a generalized reader defines a sequence of non-repeating indices that at some point may degenerate to a constant sequence of $\star$. Note that every k-reader naturally defines a generalized reader by setting all undefined functions to map everything to $\star$.

It is useful to think of a (possibly generalized) reader as a decision tree. With a generalized reader, we will often want to continue the branches of the tree with another reader. This operation is called *grafting*. We start with the notion of a 0-*branch* and a 1-*branch*.

Definition 5.6 (0-branch, 1-branch). *Let r be a (possibly generalized) reader. The* 0-branch *of r is the reader r' defined by $r'_i(y_1,\ldots,y_i) = r_{i+1}(0, y_1,\ldots,y_i)$. Similarly, the 1-branch of r is the reader r'' defined by $r''_i(y_1,\ldots,y_i) = r_{i+1}(1, y_1,\ldots,y_i)$.*

We can now define grafting, and will do so recursively. Informally, grafting a reader t onto r at y means that at every $\star$ in the decision tree of r that can be reached after reading y, we continue the reading according to t.

Definition 5.7 (Grafting). *Let r and t be generalized readers and $x \in \{0,1\}^i$ be a terminal sequence in r. The* grafting of t onto r on the branch y *is a new reader $r^{t,y}$ defined as follows.*

- *If $t_0 \in \{r_0(y_1,\ldots,y_i),\ldots,r_{i-1}(y_1,\ldots,y_i)\}$, graft the y_{t_0}-branch of t onto r at $y_1,\ldots,y_i$.*
- *If $t_0 \notin \{r_0(y_1,\ldots,y_i),\ldots,r_{i-1}(y_1,\ldots,y_i)\}$, set $r_i(y_1,\ldots,y_i) = t_0$, call the new reader r', and graft the 0-branch of t onto r' at $y_0,\ldots,y_i,0$ and the 1-branch of t onto t at $y_0,\ldots,y_i,1$.*

Repeat the above recursively, with the base case being the grafting of an identically $\star$ reader onto r by not changing anything.

Note that the grafting of a generalized reader onto another results in a generalized reader. Note that it is also possible that $r^{t,y} = r$ when all bits that t may read were already read by r according as y.

To introduce the notion of a reader that discerns a random input from an input from C', we will first need to formulate a notion of executing a reader, which is inherently adaptive, on a partly random input.

Definition 5.8 (J-Simulation of a reader). *Let r be a q-reader, $J \subseteq [n]$ and $y \in \{0,1\}^{|J|}$. The J-*simulation of r on y *is the distribution $S(r,y,J)$ over $\{0,1\}^q$ defined to be $R_{r(x)}$ where $x[J] = y[J]$, and all bits of x outside of J are picked independently and uniformly at random from $\{0,1\}$.*

We now introduce the notion of a reader that discerns a random input from an input from C'.

Definition 5.9 (J-Discerning reader)**.** *Let r be a (possibly generalized) reader, $J \subseteq [n]$ and $y \in \{0,1\}^{|J|}$. Let x be a uniform random variable in $\{c \in C' | c[J] = y\}$. We say that r is a J-*discerning reader *for y if $d_{TV}(R_{r(x)}, S(r, y, J)) \geq 1/8$.*

Next, we prove that many readers are indeed discerning.

Lemma 5.10. *Set $J \subseteq [n]$ such that $|J| \leq n/2$ and $y \in \{0,1\}^{|J|}$. With probability at least $1/9$ the q-reader r picked by the tester is J-discerning for y.*

Proof. Let r be a reader that is not J-discerning for y. Let $B \sim U_J(y)$ and $G \sim U(\{c \in C' | c[J] = y\})$. Denote by π_r the predicate associated with r. By our assumption,

$$|\Pr[\pi_r(R_{r(B)}) = 1] - \Pr[\pi_r(R_{r(G)}) = 1]| < 1/8.$$

Now assume that with probability greater than $8/9$, the q-reader picked is not J-discerning for y. Now consider the difference in acceptance probability when drawing a reader according to μ.

$$|\Pr_{r\sim\mu}[\pi_r(R_{r(B)}) = 1] - \Pr_{r\sim\mu}[\pi_r(R_{r(G)}) = 1]| < 8/9 \cdot 1/8 + 1/9 = 2/9.$$

But by Lemma 3.10 and the correctness of the tester, $\Pr_{r\sim\mu}[\pi_r(R_{r(B)}) = 1] \leq 1/3$, and by the correctness of the tester $\Pr_{r\sim\mu}[\pi_r(R_{r(G)}) = 1] \geq 2/3$, a contradiction. □

A common operation will be to graft a discerning reader with additional arbitrary bits. This does not cause a discerning reader to stop being one.

Definition 5.11. *Let r and s be generalized readers. We say that r* contains *s if for every $x \in \{0,1\}^n$, the sequence of non-$\star$ elements in $R_{s(x)}$ is a prefix of $R_{r(x)}$.*

Note that in particular, whenever we graft s onto r along some branch, we obtain a reader which contains r.

Lemma 5.12. *Let r and s be generalized readers such that r contains s. Let $J \subseteq [n]$ and $y \in \{0,1\}^{|J|}$. If s is a J-discerning reader for y, then so is r.*

Proof. Let $B \sim U_J(y)$ and $G \sim U(\{c \in C' | c[J] = y\})$. Consider $R_{r(B)}$. Its outcomes can be partitioned according to their $R_{s(B)}$ prefixes. Thus the probability of every event defined by values of $R_{r(B)}$ can be written as a weighted sum of the probabilities of events defined by values of $R_{s(B)}$. The same is true for $R_{r(G)}$ and $R_{s(G)}$. Therefore $d_{TV}(R_{r(x)}, S(r, y, J)) \geq d_{TV}(R_{s(x)}, S(s, y, J))$. □

To prove that a uniform choice in C' does not have high entropy we graft discerning readers one onto the other. We will want to make sure that all the branches of the decision tree are of the same height throughout the grafting, and thus we define the notion of a *padded grafting.*

Definition 5.13 (q-Padded grafting)**.** *Let r be a generalized reader, t be a q-reader and $y \in \{0,1\}^i$ be a terminal sequence in r. The q-*padded grafting *of t onto r* on the branch *y is defined by the following process. First, let r' be the grafting of t onto r at the branch y. Now perform the following repeatedly: Let $z_1, \ldots, z_j$ with $j < q$ be such that $r'_{i+j-1}(y_1, \ldots, y_i, z_1, \ldots, z_{j-1}) \neq \star$ but $r'_{i+j}(y_1, \ldots, y_i, z_1, \ldots, z_j) = \star$ (or $j = 0$ and $r'_i(y_1, \ldots, y_i) = \star$). Let k be an arbitrary index not in $\{r'_0, \ldots, r'_{i+j-1}(y_1, \ldots, y_i, z_1, \ldots, z_{j-1})\}$, and redefine $r'_{i+j}(y_1, \ldots, y_i, z_1, \ldots, z_j) = k$. Repeat this process as long as such $z_1, \ldots, z_j$ with $j < q$ exist.*

The above is basically grafting additional arbitrary reads, so that the end-result will always read exactly q bits after reading the sequence $y_1, \dots, y_i$. The next observation together with Lemma 5.12 implies that q-padded grafting of a J-discerning reader is equivalent to a grafting of some other J-discerning reader.

Observation 5.14. *Let r be a generalized reader, t a q-reader and $y \in \{0,1\}^i$ a terminal sequence in r. There exists a reader s containing t such that the q-padded grafting of t onto r at y is equivalent to the grafting of s onto r at y.*

Now we can finally prove the main lemma, by performing repeated q-padded grafting of discerning readers one onto another.

Lemma 5.15. *If $X \sim U(C')$, where C is C'-partially testable with q queries, then $H[X] \le \log|C| - \lfloor \frac{1}{32}\Gamma/q \rfloor$*

Proof. Let us construct an n-reader and consider the entropy of C' when permuted by this reader.

Start with a 0-reader r^0. Let s be a $\emptyset$-discerning q-reader for the empty word, which must exist since the adaptive tester must pick one with positive probability. Set r^1 to be the grafting of s onto r^0 on the branch of the empty word.

Assume that we have constructed the jq-reader r^j. If $jq \ge \Gamma$, graft a reader that reads all remaining bits arbitrarily onto r^j on all branches. Else, perform the following for all branches $y \in \{0,1\}^{jq}$ to obtain r^{j+1} (noting that they are all terminal sequences in r^j):

- If there is no member of C' with the reading $R_{r^j(y)}$, perform a q-padded grafting of an arbitrary q-reader onto r^j at the branch y,
- If such a member exists, let s be a $\{r_1^j(y), r_2^j(y), \dots, r_{jq}^j(y)\}$-discerning reader for y. Perform a q-padded grafting of s onto r^j at the branch y.

Now let r be the resulting n-reader, let $r_{R(C')}$ be the image of C' under the reading of r, and let $X \sim U(r_{R(C')})$. By Observation 5.4, the distribution of X is the same as starting with a uniformly random member of C' and then taking its reading according to r. By the chain rule $H[X] = H[X[\{1,\dots,\Gamma\}]] + H[X|X[\{1,\dots,\Gamma\}]]$.

Note that in the case of a word from C', the maximal j in the construction is equal to Γ/q. By the chain rule we may write

$$H[X[\{1,\dots,\Gamma\}]] = \sum_{i=1}^{\Gamma/q} H[X[\{(i-1)q+1,\dots,iq-1\}]|X[\{1,\dots,(i-1)q-1\}]]$$

and since each sequence of q bits is from the grafting of a reader which is discerning with respect to all the previous ones, we may apply Lemma 3.8 to obtain

$$H[X[\{1,\dots,\Gamma\}]] = \sum_{i=1}^{\Gamma/q} H[X[\{(i-1)q+1,\dots,iq-1\}]|X[\{1,\dots,(i-1)q-1\}]]$$

$$\le \sum_{i=1}^{\Gamma/q} \left(q - \frac{1}{32}\right) \le \Gamma - \Gamma/q \cdot \frac{1}{32}$$

.

By Lemma 3.11, $H[X|X[\{1,\ldots,\Gamma\}]] \leq \log|C| - \Gamma$, so by summing it all up we get $H[X] \leq \log|C| - \Gamma/q \cdot \frac{1}{32}$.

□

This gives us Theorem 1.3 in its full generality, as it implies that $|C'| = 2^{H[X]} \leq 2^{-\Gamma/32q} \cdot |C|$.

References

[1] Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, 7(4):567–583, December 1986.

[2] Noga Alon, Eldar Fischer, Michael Krivelevich, and Mario Szegedy. Efficient testing of large graphs. *Combinatorica*, 20(4):451–476, 2000.

[3] Noga Alon, Michael Krivelevich, Ilan Newman, and Mario Szegedy. Regular languages are testable with a constant number of queries. *SIAM J. Comput*, 30(6):1842–1862, 2000.

[4] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM J. Comput*, 36(4):889–974, 2006.

[5] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012.

[6] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993.

[7] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley-Interscience, New York, USA, 2000.

[8] Eldar Fischer. The art of uninformed decisions: A primer to property testing. *Current Trends in Theoretical Computer Science: The Challenge of the New Century*, I:229–264, 2004.

[9] Eldar Fischer and Arie Matsliah. Testing graph isomorphism. *SIAM J. Comput*, 38(1):207–225, 2008.

[10] E. N. Gilbert. A comparison of signalling alphabets. *The Bell System Technical Journal*, 31(3):504–522, May 1952.

[11] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45:653–750, July 1998.

[12] Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. *J. ACM*, 53(4):558–655, 2006.

[13] Tom Gur and Ron Rothblum. Non-interactive proofs of proximity. *Electronic Colloquium on Computational Complexity (ECCC)*, (078), 2013.

[14] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*, volume 16 of *North-Holland Mathematical Library.* North-Holland, 1977.

[15] F. J. MacWilliams, N. J. A. Sloane, and J. G. Thompson. Good self dual codes exist. *Discrete Math.*, 3:153–162, 1972.

[16] Dana Ron. Property testing: A learning theory perspective. *Found. Trends Mach. Learn.*, 1:307–402, March 2008.

[17] Dana Ron. *Algorithmic and Analysis Techniques in Property Testing.* 2010.

[18] Guy N. Rothblum, Salil Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, STOC '13, pages 793–802, New York, NY, USA, 2013. ACM.

[19] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

[20] R. R. Varshamov. Estimate of the number of signals in error correcting codes. In *Dokl. Akad. Nauk SSSR*, volume 117, pages 739–741, 1957.