Notes for Lower Bounds Techniques

Oded Goldreich*

February 22, 2016

Summary: We present and illustrate three techniques for proving lower bounds on the query complexity of property testers.

1. By presenting a distribution on instances that have the property and a distribution on instances that are far from the property such that an oracle machine of low query complexity cannot distinguish these two distributions.
2. By reduction from communication complexity. That is, by showing that a communication complexity problem of high complexity can be solved within communication complexity that is related to the query complexity of the property testing task that we are interested in.
3. By reduction from another testing problem. That is, by showing a "local" reduction of a hard testing problem to the testing problem that we are interested in.

We also present simplifications of these techniques for the cases of one-sided error probability testers and non-adaptive testers.

The methodology of reducing from communication complexity was introduced by Blais, Brody, and Matulef [1], and our description of it is based on [5].

1 Preliminary comments

Proving lower bounds is often more challenging than proving upper bounds, since one has to defeat all possible methods (or algorithms) rather than showing that one of them works. Indeed, it seems harder to cope with a universal quantifier than with an existential one, but one should bear in mind that a second quantifier of opposite nature follows the first one. That is, a lower bound has the form "every method fails on some instance" (i.e., $\forall\exists$), whereas an upper bound has form "(there) exists a method that succeeds on all instances" (i.e., $\exists\forall$). Still, the $\forall\exists$ template seems harder to argue about than the $\exists\forall$ template.

Recall that when presenting testers, we presented them in terms of uniform algorithms that get the size parameter n and proximity parameter ϵ as inputs. That is, the same algorithm was used for all values of n and ϵ, making it potentially more useful, especially when it was relatively efficient in terms of computational complexity (i.e., when its running time was closely related to its query complexity). In contrast, when seeking query complexity lower bounds, we drop the

*Department of Computer Science, Weizmann Institute of Science, Rehovot, Israel.

computational complexity requirement, and even allow the potential tester to be non-uniform (i.e., depend arbitrary on n and ϵ).[1] This makes the lower bound results stronger, clarifying that they are due only to "information theoretic" considerations; but the truth is that the techniques presented in this section can not capitalize on uniformity conditions.

2 Indistinguishability of distributions

A popular methodology for proving lower bounds on the complexity of solving a computational problem consists of presenting a distribution of instances on which every algorithm that has lower complexity (i.e., lower than claimed) fails. In the context of randomized algorithms (of error probability at most 1/3), this means presenting a distribution X such that, for every algorithm A having lower complexity, it holds that $A(X)$ is wrong with probability greater than 1/3, where the probability is taken over both X and the internal coin tosses of A. (Typically, X cannot be concentrated on a single instance, since for every instance there exists a "special purpose" algorithm that solves it.)[2]

The foregoing methodology seems to make the job of proving lower bounds harder. Rather than having total freedom in choosing for each "low complexity" algorithm a designated instance (or a distribution of instances) on which this algorithm fails, the prover is required to find a single distribution of instances on which all (low complexity) algorithms fail. Proving lower bounds this way is certainly valid (since if each algorithm fails on the said distribution then for each algorithm there exists an instance on which it fails), but one may wonder about the point of restricting the freedom of the lower bound prover. But such a restriction is manifested in any proof technique; any proof technique represents a restriction of the possible proof strategies to a single one. The point is that a restriction has the benefit of focusing attention, which is often beneficial. In other words, the restriction may turn out to be a simplification, especially when we recall the thesis that the $\exists\forall$ template (underlying the existence of a distribution that foils any algorithm) seems simpler (or more intuitive to handle) than the $\forall\exists$ template (which underlies the task of finding a bad instance for each algorithm).[3]

In the context of non-uniform complexity, as is the case when we only care about the query complexity of oracle machines, we can take a simplification step and observe that it suffices to consider deterministic machines. This is because A is a convex combination of deterministic machines of the same complexity; specifically, for every x, if $A(x) \in \mathbb{R}$, then $A(x) = \mathbb{E}_r[A'(x,r)]$, where $A'(x,r)$ denotes the output of A on input x when the outcome of A's internal coin tosses equals r.[4] In this

[1] In other words, we allow to present a different algorithm for each possible value of n and ϵ, making no requirements regarding the dependence of this algorithm on these values (or about the "uniformity" of this sequence of algorithms).

[2] Indeed, in the context of uniform complexity classes, one may encounter arguments that identify a single instance per each length, but in these cases one refers to an infinite sequence of such instances (whereas the same uniform machine must handle all lengths). Indeed, in a non-uniform complexity setting, picking a single instance per length will not do.

[3] Furthermore, in the context of non-uniform complexity, this methodology is actually "complete" in the sense that any valid lower bound can be proved by presenting a single distribution that foils all "low complexity" algorithms. See further discussion following the statement of Theorem 1.

[4] This description fits well the case that A solves a decision problem (or estimates a numerical parameter of the input). The case of search problems can be handled by introducing a verification algorithm V (which accepts (x,y) if and only if y is a valid solution for x). In this case, we can consider the probability that $V(x,A(x)) = 1$. In this case, $B(X,A(X)) = \mathbb{E}_r[B(X,A'(X,r))]$ (which equals $\mathbb{E}_{x\leftarrow X}[\mathbb{E}_r[B(x,A'(x,r))]]$), and it follows that there exists an r such that the probability that $A'(X,r)$ is a valid solution to X is lower-bounded by the probability that $A(X)$ is a

case, $A(X) = \mathbb{E}_r[A'(X,r)]$, and it follows that there exists an r such that $A'(X,r)$ is correct with probability that is lower-bounded by the probability that $A(X)$ is correct.

Let us detail the argument in the concrete setting of property testing. Recall that in this setting we deal with randomized algorithms, which are allowed error probability at most 1/3, for solving a promise problem (i.e., distinguishing instances that have the property from instances that are far from the property). Hence, the algorithm (i.e., a potential tester) fails only if it outputs a wrong answer, with probability exceeding 1/3, on an instance that satisfies the promise. As stated above, rather than seeking, for each algorithm of low complexity, an instance (that satisfies the promise) on which this algorithm fails, we shall seek a single distribution such that the each algorithm of low complexity fails on this distribution. That is, the algorithm errs with probability exceeding 1/3, where the probability is taken both over the distribution and the internal coin tosses of the algorithm. Furthermore, fixing such a distribution of instances, it will suffice to consider deterministic algorithms.

Theorem 1 (the query complexity of randomized algorithms is lower bounded by the "distributional" query complexity of deterministic algorithms): *Let $\Pi = \cup_{n\in\mathbb{N}}\Pi_n$ such that Π_n contains functions from $[n]$ to R_n, and let $q : \mathbb{N} \times (0,1] \to \mathbb{N}$. Suppose that for some $\epsilon > 0$ and $n \in \mathbb{N}$, there exists a distribution F of functions from $[n]$ to R_n such that for every deterministic oracle machine M that makes at most $q(n,\epsilon)$ queries it holds that*

$$\mathbf{Pr}[F \in \Pi_n \wedge M^F(n,\epsilon) \neq 1] + \mathbf{Pr}[F \in \Gamma_\epsilon(\Pi_n) \wedge M^F(n,\epsilon) \neq 0] \;>\; \frac{1}{3}, \tag{1}$$

where $\Gamma_\epsilon(\Pi_n)$ denotes the set of functions (from $[n]$ to R_n) *that are ϵ-far from Π_n. Then, the query complexity of ϵ-testing Π is greater than $q(\cdot,\epsilon)$.*

The term "distributional complexity" that appears in the title of Theorem 1 refers to the query complexity of deterministic algorithms that are only required to solve the problem "on the average" (or rather on random instances drawn from some fixed distribution). The method underlying Theorem 1 was first employed by Yao [13], and it turns out that it is "complete" in the sense that any valid lower bound can be proved by using it; that is, if Π has query complexity greater than q, then there exists a distribution as in the hypothesis of Theorem 1. (The latter claim is far more difficult to establish; it requires employing von Neumann's Minimax Theorem [11].)[5]

Proof: Suppose towards the contradiction that T is an ϵ-tester of query complexity $q(\cdot,\epsilon)$ for Π. Then, for any $n \in \mathbb{N}$ and every $f \in \Pi_n$ it holds that $\mathbf{Pr}[T^f(n,\epsilon) \neq 1] \leq 1/3$, whereas for every $f : [n] \to R_n$ that is ϵ-far from Π_n it holds that $\mathbf{Pr}[T^f(n,\epsilon) \neq 0] \leq 1/3$, since T has error probability at most 1/3. On the other hand, for every distribution F of functions from $[n]$ to R_n, it holds that

$$\mathbf{Pr}[F \in \Pi_n \wedge T^F(n,\epsilon) \neq 1] \;\leq\; \mathbf{Pr}[F \in \Pi_n] \cdot \max_{f\in\Pi_n}\{\mathbf{Pr}[T^f(n,\epsilon) \neq 1]\} \tag{2}$$

$$\mathbf{Pr}[F \in \Gamma_\epsilon(\Pi_n) \wedge T^F(n,\epsilon) \neq 0] \;\leq\; \mathbf{Pr}[F \in \Gamma_\epsilon(\Pi_n)] \cdot \max_{f\in\Gamma_\epsilon(\Pi_n)}\{\mathbf{Pr}[T^f(n,\epsilon) \neq 0]\}. \tag{3}$$

valid solution to X.

[5] See discussion in [4, Apdx. A.1].

(This represents an averaging argument over the distribution F; that is, $\mathbf{Pr}[\chi(X,R)|X \in S] \leq \max_{x \in S}\{\mathbf{Pr}[\chi(x,R)]\}$, for independent random variables X and R).[6] Recalling that each of the "max-factors" in Eq. (2)&(3) is upper-bounded by 1/3, we get

$$\mathbf{Pr}[F \in \Pi_n \wedge T^F(n,\epsilon) \neq 1] + \mathbf{Pr}[F \in \Gamma_\epsilon(\Pi_n) \wedge T^F(n,\epsilon) \neq 0] \leq \frac{1}{3} \tag{4}$$

since $\mathbf{Pr}[F \in \Pi_n] + \mathbf{Pr}[F \in \Gamma_\epsilon(\Pi_n)] \leq 1$.

> **Teaching note:** Recall that Eq. (2)&(3) were derived by an averaging argument over the distribution F. In contrast, in the following paragraph we shall employ an averaging argument over the internal coin tosses of T.

Denoting by T_r the residual deterministic machine that is obtained by fixing the internal coin tosses of T to r, it follows (by an averaging argument on r)[7] that there exists an r such that

$$\mathbf{Pr}[F \in \Pi_n \wedge T_r^F(n,\epsilon) \neq 1] + \mathbf{Pr}[F \in \Gamma_\epsilon(\Pi_n) \wedge T_r^F(n,\epsilon) \neq 0] \leq \frac{1}{3} \tag{5}$$

which contradicts Eq. (1), since T_r is a deterministic oracle machine that makes at most $q(n,\epsilon)$ queries. ■

A more convenient form – indistinguishability. A more important simplification step is obtained by considering a distribution F (of functions from $[n]$ to R_n) such that $\mathbf{Pr}[F \in \Pi_n] = \mathbf{Pr}[F \in \Gamma_\epsilon(\Pi_n)] = 1/2$. In this case, it suffices to show that no deterministic oracle machine M that makes at most $q(n,\epsilon)$ queries can distinguish the case of $F \in \Pi_n$ from the case of $F \in \Gamma_\epsilon(\Pi_n)$ with a gap of at least 1/3.

Theorem 2 (the method of indistinguishability of distributions): *Let $\Pi = \cup_{n \in \mathbb{N}} \Pi_n$ and $\Gamma_\epsilon(\Pi_n)$ be as in Theorem 1, and let $q : \mathbb{N} \times (0,1] \to \mathbb{N}$. Suppose that for some $\epsilon > 0$ and $n \in \mathbb{N}$, there exists a distribution F_1 of functions in Π_n and a distribution F_0 of functions in $\Gamma_\epsilon(\Pi_n)$ such that for every deterministic oracle machine M that makes at most $q(n,\epsilon)$ queries it holds that*

$$\left|\mathbf{Pr}[M^{F_1}(n,\epsilon) = 1] - \mathbf{Pr}[M^{F_0}(n,\epsilon) = 1]\right| < \frac{1}{3}. \tag{6}$$

Then, the query complexity of ϵ-testing Π is greater than $q(\cdot,\epsilon)$.

(The method captured by Theorem 2 is also complete in the sense that any valid lower bound can be proved by using it; see Exercise 2.)

Proof: Fixing any deterministic oracle machine M of query complexity q, for every $i \in \{0,1\}$, let p_i denote the probability that $M^{F_i}(n,\epsilon)$ equals 1. Then, by Eq. (6), we have $|p_1 - p_0| < 1/3$. Now, let F equal F_1 with probability 1/2, and equal F_0 otherwise. Then, the probability that $M^F(n,\epsilon)$ errs (i.e., outputs 0 when $F = F_1$ or outputs 1 when $F = F_0$) is $0.5 \cdot (1 - p_1) + 0.5 \cdot p_0 \geq 0.5 - |p_1 - p_0|/2 > 1/3$. Hence, F satisfies the hypothesis of Theorem 1, and the current claim follows. ■

[6] Here X represents F and R represents the internal coin tosses of T. In Eq. (2) we used $S = \Pi$ and $\chi(x,R) = 1$ if $T^F \neq 1$ (under coins R), whereas in Eq. (3) we used $S = \Gamma_\epsilon(\Pi)$ and $\chi(x,R) = 1$ if $T^F \neq 0$ (under coins R).

[7] We stress that Eq. (5) is proved by viewing the l.h.s of Eq. (4) as an expected value of the l.h.s of Eq. (5), where the expectation is taken over all possible choices of r. (In contrast, we cannot just pick an r_1 that minimized the first term in the l.h.s of Eq. (4) and an r_2 that minimizes the second term.)

A more flexible form. The reasoning underlying Theorem 2 remains valid also if we allow F_1 and F_0 to reside outside their designated sets with small probability. In such a case, we should reduce the gap accordingly. This yields the following more flexible version, when in typical applications (which are asymptotic) one can make all η_i's arbitrarily small positive constants.

Corollary 3 (a more flexible form of Theorem 2): *Let $\Pi = \cup_{n\in\mathbb{N}}\Pi_n$, $\Gamma_\epsilon(\Pi_n)$, and $q : \mathbb{N}\times(0,1] \to \mathbb{N}$ be as in Theorem 2. Suppose that for some $\epsilon > 0$, $\eta_0, \eta_1, \eta_2 > 0$ and $n \in \mathbb{N}$, there exist distributions F_1 and F_0 such that $\mathbf{Pr}[F_1 \in \Pi_n] \geq 1-\eta_1$ and $\mathbf{Pr}[F_0 \in \Gamma_\epsilon(\Pi_n)] \geq 1-\eta_0$, and for every deterministic oracle machine M that makes at most $q(n,\epsilon)$ queries it holds that*

$$\left|\mathbf{Pr}[M^{F_1}(n,\epsilon) = 1] - \mathbf{Pr}[M^{F_0}(n,\epsilon) = 1]\right| \leq \eta_2. \tag{7}$$

If $\eta_0 + \eta_1 + \eta_2 < 1/3$, then the query complexity of ϵ-testing Π is greater than $q(\cdot,\epsilon)$.

Proof: Let F_1' (resp., F_0') denote the distribution of F_1 (resp., F_0) conditioned on $F_1 \in \Pi_n$ (resp., $F_0 \in \Gamma_\epsilon(\Pi_n)$). Then, for each $i \in \{0,1\}$, the statistical distance between F_i' and F_i is at most η_i. Hence, if (F_1, F_0) satisfies Eq. (7), then (F_1', F_0') satisfies Eq. (6),[8] and the proof is completed by applying Theorem 2. ■

Illustrating the application of the method. We have already used the method of indistinguishability of distributions (twice) in the first lecture. Here we reproduce the proof of the existence of properties that are hard to test, while explicitly using Corollary 3.

Proposition 4 (hardness of testing membership in a linear code): *Let G be a $0.5n$-by-n Boolean matrix in which every $0.05n$ columns are linearly independent. Let $\Pi = \{xG : x \in \{0,1\}^{0.5n}\}$ be the linear code generated by G. Then, 0.1-testing Π requires more than $0.05n$ queries.*

Proof: Let X denote the uniform distribution on Π, and Y denote the uniform distribution on $\{0,1\}^n$. We shall use the following two observations, which were justified in the first lecture.

1. An algorithm that makes at most $0.05n$ queries cannot distinguish X from Y; that is, for any oracle machine M that makes at most $0.05n$ queries, it holds that $\mathbf{Pr}[M^X = 1] = \mathbf{Pr}[M^Y = 1]$.

   (Recall that this follows from the fact that the restriction of each of the two distributions to any $0.05n$ coordinates is uniformly distributed in $\{0,1\}^{0.05n}$.)[9]

[8] This holds since

$$\begin{aligned}\left|\mathbf{Pr}[M^{F_1'}(n,\epsilon) = 1] - \mathbf{Pr}[M^{F_0'}(n,\epsilon) = 1]\right| \;\leq\; &\left|\mathbf{Pr}[M^{F_1'}(n,\epsilon) = 1] - \mathbf{Pr}[M^{F_1}(n,\epsilon) = 1]\right| \\ &+ \left|\mathbf{Pr}[M^{F_1}(n,\epsilon) = 1] - \mathbf{Pr}[M^{F_0}(n,\epsilon) = 1]\right| \\ &+ \left|\mathbf{Pr}[M^{F_0}(n,\epsilon) = 1] - \mathbf{Pr}[M^{F_0'}(n,\epsilon) = 1]\right|\end{aligned}$$

which is at most $\eta_1 + \eta_2 + \eta_0 < 1/3$.

[9] As shown in Exercise 3, the distinguishing gap of an algorithm that makes q adaptive queries (to a Boolean function) is at most 2^q times larger than the distinguishing gap of a corresponding non-adaptive algorithm (which makes q non-adaptive queries).

2. For all sufficiently large n, with probability at least $1-2^{-0.01n}$, it holds that Y is 0.1-far from Π.

   (Recall that this follows from a counting argument that relies on the exponentially vanishing density of Π.)

Invoking Corollary 3, with $q(n) = 0.05n$, $\epsilon = 0.1$, $\eta_1 = \eta_2 = 0$, and $\eta_0 = 0.3$ (and sufficiently large n), the claim follows. (Indeed, $\eta_0 = 2^{-0.01n} < 0.3$ follows by Observation 2 (and $n \geq 200$), $\eta_2 = 0$ follows by Observation 1, and $\eta_1 = 0$ follows by the definition of X.) ■

Digest: On the simplicity of the foregoing proof. The simplicity of the proof of Proposition 4 is due to the fact that the projections of the two distributions on any set of q coordinates are identically distributed, where $q+1$ is the lower bound established by the proof. In more complicated cases, this strong assertion does not hold, and only weaker assertions can be proved. For example, if for some small $\eta > 0$, one can prove that the projections of the two distributions on any fixed set of $i \leq q$ coordinates are within stantistical distance of at most $i \cdot \eta$, then we can only infer that a *non-adaptive* algorithm that makes q queries has a distinguishing gap of at most $q \cdot \eta$. In contrast, an adaptive q-query algorithm may distinguish such distributions with a gap of $\widetilde{\Omega}(\min(2^q \cdot \eta, 1))$, and this is indeed the worst possible (see Exercise 3).

A general comment. The fact that Theorems 1 and 2 (and Corollary 3) allow to restrict the attention to deterministic algorithms (rather than consider all randomized algorithms) is less useful than one may think. In fact, many arguments that use these results can be generalized to relate to the distinguishing gap of randomized algorithms (see, for example, the proof of Proposition 4). The *important aspect of the method is the focus on the distinguishing gap* (between a distribution concentrated on instances that have the property and an distribution concentrated on inputs that are far from the property). Still, in some cases the argument (or its presentation) is simplified by restricting attention to deterministic algorithm. (Note, however, that the proof of Theorem 1 would not have been much simpler if we were to relax it and refer to the behavior of randomized algorithms.)[10]

Further reflection. As stated above, the important aspect of the method is not the apparent gain obtained by restricting attention to deterministic algorithms (rather than randomized ones), but rather the "loss" that arises when confining ourselves to a single distribution of instances (and showing that all "low-complexity" algorithms fail on this distribution). We stress that potentially we gave up on the possibility of tailoring a hard instance (or distribution of instances) to each potential algorithm, although in retrospect it turns out that nothing was lost (since the method is "complete" in the sense that any valid lower bound can be proved by using it). Nevertheless, as is often the case in mathematics and science, proving a stronger statement and/or using more restricted methods is sometimes easier.

Final digest. Note that the path we have taken consisted of four steps, where the first two steps are packed into Theorem 1, and the last two steps are captured by Theorem 2 and Corollary 3, respectively.

[10] Also note that the proofs of Theorem 2 and Corollary 3 would remain intact, since Theorem 2 is proved by a reduction to Theorem 1, whereas Corollary 3 is proved by reduction to Theorem 2.

1. Requiring the lower bound prover to present a single distribution that foils all algorithms of low complexity.

   Recall that potentially this makes the task of the prover harder, since the claim being established is seemingly stronger, but as argued above such a step may turn out beneficial. Furthermore, in the context of non-uniform complexity, this seemingly harder task is actually equivalent to the original task (i.e., the seemingly stronger claim is actually equivalent to the original one).

2. Showing that it suffices to establish the (foiling) claim for deterministic algorithms rather than for randomized ones.

   This step simplifies the presentation of lower bound proofs, but in many cases is less helpful than one may imagine.

3. Requiring the lower bound prover to prove the foiling claim by showing that low complexity algorithms cannot distinguish (a distribution over) instances that should be accepted from (a distribution over) instances that should be accepted.

   As with Step 1, potentially this makes the task of the prover harder, since the claim being established is seemingly stronger, but again such a step may turn out beneficial, and again the claim it seeks to establish is actually not stronger.

4. Showing that it suffices to establish a relaxed version of the indistinguishability claim.

   Like Step 2, the current step simplifies the presentation of lower bound proofs, freeing the prover from the need to deal with the corresponding issues either implicitly or explicitly.

Hence, Steps 1 and 3 make the proving task potentially harder, although they actually help to focus attention on a task that is more intuitive and easier to think about. In contrast, Steps 2 and 4 simplify the proving task either by restricting its scope (see Step 2) or by relaxing the requirements (see Step 4).

3 Reduction from Communication Complexity

A somewhat unexpected methodology for proving lower bounds on the query complexity of property testing consists of reducing communication complexity problems to property testing problems. This is quite surprising because we reduce between two very different models. Specifically, property testing problems have no "topology" that can be naturally 2-partitioned to fit the two-party setting of communication complexity.

> **Teaching note:** Readers who are not familiar with communication complexity may want to skip the following paragraph. On the other hand, readers who are familiar with the communication complexity background may skim through Section 3.1 with the sole purpose of picking the specific notations that we shall use.

The reduction at a glance. In order to derive a lower bound on testing the property Π, one presents a mapping F of pairs of inputs $(x,y) \in \{0,1\}^{\ell+\ell}$ for a two-party communication problem Ψ to $n(\ell)$-bit long inputs for Π such that $(x,y) \in \Psi$ implies $F(x,y) \in \Pi$ and $(x,y) \notin \Psi$ implies that

$F(x,y)$ is ϵ-far from Π. Let $f_i(x,y)$ be the i^{th} bit of $F(x,y)$, and suppose that B is an *upper bound* on the (deterministic) communication complexity of each f_i, and that C is a *lower bound* on the randomized communication complexity of Ψ. Then, ϵ-testing Π requires at least C/B queries.

Tedious comments. For sake of simplicity, we focus on problems that correspond to the binary representation (i.e., to objects that are represented as sequences over a binary alphabet).[11] Also, our main presentation refers to finite problems that correspond to bit strings of fixed lengths, denoted ℓ and $n = n(\ell)$. One should think of these lengths as generic (or varying), and interpret the O-notation (as well as similar notions) as hiding universal constants (which do not depend on any parameter of the problems discussed).

3.1 Communication Complexity

We refer to the standard setting of communication complexity, and specifically to randomized two-party protocols in the model of shared randomness (cf. [9, Sec. 3]). We denote by $\langle A(x), B(y)\rangle(r)$ the (joint) output of the two parties, when the first party uses strategy A and gets input x, the second party uses strategy B and gets input y, and both parties have free access to the shared randomness r. Since many of the known reductions that use the methodology surveyed here actually reduce from promise problems, we present communication problems in this more general setting. The standard case of decision problems is obtained by using a trivial promise (i.e., $P = \{0,1\}^{2\ell}$).[12]

Definition 5 (two-party communication complexity): *Let $\Psi = (P,S)$ such that $P, S \subseteq \{0,1\}^{2\ell}$, and $\eta \geq 0$. A two-party* protocol that solves Ψ with error at most η *is a pair of strategies (A,B) such that the following holds* (w.r.t. some $\rho = \rho(\ell)$):

1. *If $(x,y) \in P \cap S$, then $\mathbf{Pr}_{r\in\{0,1\}^\rho}[\langle A(x), B(y)\rangle(r)=1] \geq 1-\eta$.*
2. *If $(x,y) \in P \setminus S$, then $\mathbf{Pr}_{r\in\{0,1\}^\rho}[\langle A(x), B(y)\rangle(r)=0] \geq 1-\eta$.*

The communication complexity of this protocol *is the maximum number of bits exchanged between the parties when the maximization is over all $x,y \in \{0,1\}^\ell$ and $r \in \{0,1\}^\rho$. The η-*error communication complexity of Ψ*, denoted $\mathtt{CC}_\eta(\Psi)$, is the minimum communication complexity of all protocols that solve Ψ with error at most η.*

For a Boolean function $f : \{0,1\}^{2\ell} \to \{0,1\}$, the two-party communication problem of computing f is the promise problem $\Psi_f \stackrel{\text{def}}{=} (\{0,1\}^{2\ell}, \{(x,y) : f(x,y) = 1\})$. Abusing notation, we let $\mathtt{CC}_\eta(f)$ denote $\mathtt{CC}_\eta(\Psi_f)$.

Note that randomized complexity with zero error (i.e., $\eta = 0$) collapses to deterministic complexity.[13] This is one reason that we kept η as a free parameter rather than setting it to a small

[11] For a treatment of the general case of non-binary alphabets, see [5, Sec. 6]. The bottom-line is that little is lost by considering only the binary representation.

[12] In general, P denotes the promise and S denotes the set of YES-instances. The task is to distinguish between instances in $P \cap S$ and instances in $P \setminus S$.

[13] Note that $\mathtt{CC}_0(\cdot)$ is different from the *standard* notion of zero-error randomized communication complexity, since in the latter one considers the expected number of bits exchanged on the worst-case pair of inputs (whereas we consider the worst-case over both the shared randomness and the pair of inputs). Note that the difference between the expected complexity and the worst-case complexity is not very significant in the case of $\Theta(1)$-error communication complexity, but it is crucial in the case of zero-error.

constant (e.g., $\eta = 1/3$), as is the standard. Another reason for our choice is to allow greater flexibility in our presentation (cf., e.g., Theorem 7, where we use several different values of η). For the same reason, as seen next, we take the rather unusual choice of making the error probability explicit also in the context of property testing (where we also denote it by η).

3.2 The methodology

For sake of clarity, we spell out the version of the definition of property testing that we refer to. In this definition, as in most work on *lower bounds* in property testing, we fix the proximity parameter (denoted ϵ). In contrast to this fixing, as stated above, we treat the error probability as a free parameter (rather than having it fixed to 1/3).

Definition 6 (property testing, redefined): *Let $\Pi \subseteq \{0,1\}^n$, and $\epsilon, \eta > 0$. An ϵ-*tester with error *η* for *Π is a randomized oracle machine T that satisfies the following two conditions.*

1. *If $z \in \Pi$, then $\mathbf{Pr}[T^z(n)=1] \geq 1 - \eta$.*
2. *If $z \in \{0,1\}^n$ is ϵ-far from Π, then $\mathbf{Pr}[T^z(n)=0] \geq 1 - \eta$.*

The query complexity of *T is the maximum number of queries that T makes, when the maximization is over all $z \in \{0,1\}^n$ and all possible outcomes of the coin tosses of T. The η-*error query complexity of *ϵ-*testing *Π, denoted $\mathtt{Q}_\eta(\epsilon, \Pi)$, is the minimum query complexity of all ϵ-testers with error η for Π.*

For any property Π and any constant $\eta > 0$, it holds that $\mathtt{Q}_\eta(\epsilon, \Pi) = O(\mathtt{Q}_{1/3}(\epsilon, \Pi))$, where the O-notation hides a $\log(1/\eta)$ factor. Thus, establishing a lower bound on the ϵ-testing query complexity of Π for any constant error, yields the same asymptotic lower bound for the (standard) error level of 1/3. In light of this fact, we may omit the constant error from our discussion; that is, when we say the query complexity of ϵ-testing Π we mean the 1/3-error query complexity of ϵ-testing Π. Hence, we denote $\mathtt{Q}(\epsilon, \Pi) = \mathtt{Q}_{1/3}(\epsilon, \Pi)$.

With the above preliminaries in place, we are ready to state the main result, which captures the methodology of obtaining lower bounds on the query complexity of property testing based on lower bounds on communication complexity. Using this methodology towards establishing a lower bound on the query complexity of testing the property Π requires finding a suitable communication complexity problem Ψ (for which adequate lower bounds are known) and presenting a reduction that satisfies the hypothesis of Theorem 7.

Theorem 7 (property testing lower bounds via communication complexity): *Let $\Psi = (P, S)$ be a promise problem such that $P, S \subseteq \{0,1\}^{2\ell}$, and let $\Pi \subseteq \{0,1\}^n$ be a property. For $\epsilon, \eta > 0$, suppose that the mapping $F : \{0,1\}^{2\ell} \to \{0,1\}^n$ satisfies the following two conditions:*

1. *For every $(x, y) \in P \cap S$, it holds that $F(x, y) \in \Pi$.*
2. *For every $(x, y) \in P \setminus S$, it holds that $F(x, y)$ is ϵ-far from Π.*

Then, $\mathtt{Q}_\eta(\epsilon, \Pi) \geq \mathtt{CC}_{2\eta}(\Psi)/B$, where $B = \max_{i \in [n]}\{\mathtt{CC}_{\eta/\ell}(f_i)\}$ and $f_i(x, y)$ is the i^{th} bit of $F(x, y)$. Furthermore, if $B = \max_{i \in [n]}\{\mathtt{CC}_0(f_i)\}$, then $\mathtt{Q}_\eta(\epsilon, \Pi) \geq \mathtt{CC}_\eta(\Psi)/B$.

Proof: Given an ϵ-tester with error η for Π and communication protocols for the f_i's, we present a two-party protocol for solving Ψ. The key idea is that, using their shared randomness, the two parties (holding x and y, respectively) can emulate the execution of the ϵ-tester, while providing it with virtual access to $F(x,y)$. Specifically, when the tester queries the i^{th} bit of the oracle, the parties provide it with the value of $f_i(x,y)$ by first executing the corresponding communication protocol.

The protocol for Ψ proceeds as follows: On local input x (resp., y) and shared randomness $r = (r_0, r_1, ..., r_n) \in (\{0,1\}^*)^{n+1}$, the first (resp., second) party invokes the ϵ-tester on randomness r_0, and answers the tester's queries by interacting with the other party. That is, each of the two parties invokes a local copy of the tester's program, but both copies are invoked on the same randomness (i.e., r_0), and are fed with identical answers to their (identical) queries. When the tester issues a query $i \in [n]$, the parties compute the value of $f_i(x,y)$ by using the corresponding communication protocol, and feed $f_i(x,y)$ to (their local copy of) the tester. Specifically, denoting the latter protocol (i.e., pair of strategies) by (A_i, B_i), the parties answer with $\langle A_i(x), B_i(y)\rangle(r_i)$. When the tester halts, each party outputs the very output it has obtained from (its local copy of) the tester.

Turning to the analysis of this protocol, we note that the two local executions of the tester are identical, since they are fed with the same randomness and the same answers (to the same queries).[14] The total number of bits exchanged by the two parties is at most B times the query complexity of ϵ-tester; that is, the communication complexity of this protocol is at most $B \cdot q$, where q denotes the query complexity of the ϵ-tester.

Let us consider first the furthermore clause; that is, assume that $B = \max_{i\in[n]}\{\mathtt{CC}_0(f_i)\}$. In this case, the parties always provide the ϵ-tester, denoted T, with the correct answers to all its queries. Now, if $(x,y) \in P \cap S$, then $F(x,y) \in \Pi$, which implies that $\mathbf{Pr}[T^{F(x,y)}(n)=1] \geq 1-\eta$ (since T has error at most η), which in turn implies that the parties output 1 with probability at least $1-\eta$. On the other hand, if $(x,y) \in P \setminus S$, then $F(x,y)$ is ϵ-far from Π, which implies that $\mathbf{Pr}[T^{F(x,y)}(n)=0] \geq 1-\eta$, which in turn implies that the parties output 0 with probability at least $1-\eta$. Hence, in this case (and assuming that T has query complexity $\mathtt{Q}_\eta(\epsilon,\Pi)$), we get $\mathtt{CC}_\eta(\Psi) \leq B \cdot \mathtt{Q}_\eta(\epsilon,\Pi)$.

Turning to the main claim, we may assume that $q \stackrel{\text{def}}{=} \mathtt{Q}_\eta(\epsilon,\Pi) \leq \ell$, since otherwise we can just use the trivial communication protocol for Ψ (which has complexity ℓ). Recall that if $(x,y) \in P \cap S$, then $\mathbf{Pr}[T^{F(x,y)}(n)=1] \geq 1-\eta$. However, the emulation of T is given access to bits that are each correct only with probability $1-(\eta/\ell)$, and hence the probability that the protocol outputs 1 is at least $1-\eta-q\cdot(\eta/\ell) \geq 1-2\eta$. On the other hand, if $(x,y) \in P \setminus S$, then $\mathbf{Pr}[T^{F(x,y)}(n)=0] \geq 1-\eta$. Taking account of the errors in computing the f_i's, we conclude that the probability that the protocol outputs 0 in this case is at least $1-2\eta$. The claim follows. ■

3.3 Illustrating the application of the methodology

Recall that the set of ℓ-variate linear functions over GF(2) is ϵ-testable within query complexity $O(1/\epsilon)$. In contrast, we shall show that, for every even $k \leq \ell/2$, the set of k-linear functions, defined as linear (ℓ-variate) functions that depend on exactly k of their ℓ variables cannot be 0.499-tested

[14] Each of these answers is correct with a certain probability that depends on the corresponding sub-protocols (A_i, B_i), but by convention both parties always obtain the same answer (from these sub-protocols).

using $o(k)$ queries.[15] This will be shown by reduction from the communication complexity of the $k/2$-disjointness function (in which the two parties are each given a $k/2$-subset of $[\ell]$ and need to determine whether these subsets are disjoint). We start by defining the k-linear property and the communication complexity known as $k/2$-disjointness.

Definition 8 (k-linearity): *A function $f : \mathrm{GF}(2)^\ell \to \mathrm{GF}(2)$ is called k-*linear *if it is linear and depends on exactly k of its variables; that is, $f(z) = \sum_{i \in I} z_i$ for some $I \subseteq [\ell]$ of cardinality k.*

In the following definition, one should think of ℓ-bit long strings as representing subsets of $[\ell]$. Hence, k-subsets are represented by string of Hamming weight k, and set disjointness is represented by strings share no bit position that holds the value 1. (Recall that the Hamming weight of z is denoted $\mathrm{wt}(z)$; that is, $\mathrm{wt}(z) = |\{i \in [|z|] : z_i = 1\}|$.)

Definition 9 ($k/2$-disjointness): *The communication problem called $k/2$-*disjointness *consists of solving $\{\mathrm{DISJ}_\ell^{(k)} = (P_\ell, S_\ell)\}_{\ell \in \mathbb{N}}$, where $P_\ell, S_\ell \subseteq \{0,1\}^{2\ell}$ such that $(x,y) \in P_\ell$ if $\mathrm{wt}(x) = \mathrm{wt}(y) = k(\ell)/2$, and $(x,y) \in S_\ell$ if $I(x,y) \stackrel{\mathrm{def}}{=} \{i \in [\ell] : x_i = y_i = 1\}$ is empty.*

Indeed, recalling that x and y are indicators of sets, the set $I(x,y)$ is the intersection of these sets.

For $k(\ell) \leq \ell/2$, using the celebrated result $\mathtt{CC}_{1/3}(\mathrm{DISJ}_\ell^{(k)}) = \Omega(k(\ell))$, which is implicit in [8] (see also [1, Lem. 2.6]), we shall prove that 0.499-testing k-linearity requires $\Omega(k)$ queries, for every even $k \leq \ell/2$. This will be done by invoking Theorem 7.

Theorem 10 (the complexity of k-linearity): *For every even $k(\ell) \leq \ell/2$, the query complexity of 0.499-testing $k(\ell)$-linearity is $\Omega(k(\ell))$.*

Proof: We present a reduction from the communication complexity problem $\{\mathrm{DISJ}_\ell^{(k)} = (P_\ell, S_\ell)\}_{\ell \in \mathbb{N}}$ to $k(\ell)$-linearity of ℓ-variate functions, where in this case the size of the tested object is $n = 2^\ell$. The reduction $F : \{0,1\}^{2\ell} \to \{0,1\}^n$ maps pairs (x,y) of the communication problem to a function $g_{x,y} : \{0,1\}^\ell \to \{0,1\}$, which may be described by its truth-table $T_{x,y} \in \{0,1\}^n$, such that $g_{x,y}(z) = \sum_{i \in [\ell]} (x_i + y_i) \cdot z_i$. Indeed, $g_{x,y}(z) = \sum_{i : x_i + y_i = 1} z_i$, which means that $g_{x,y}$ is $|\{i \in [\ell] : x_i + y_i = 1\}|$-linear.

Let $k = k(\ell)$. Then, if $(x,y) \in P_\ell \cap S_\ell$ (i.e., x and y are "disjoint"), then $F(x,y) = g_{x,y}$ is k-linear, since $|\{i \in [\ell] : x_i + y_i = 1\}| = \mathrm{wt}(x) + \mathrm{wt}(y) = k$. On the other hand, if $(x,y) \in P_\ell \setminus S_\ell$, then $F(x,y) = g_{x,y}$ is $(k - 2 \cdot |I(x,y)|)$-linear, since

$$\begin{aligned} |\{i \in [\ell] : x_i + y_i = 1\}| &= |\{i \in [\ell] : x_i = 1 \wedge y_i = 0\}| + |\{i \in [\ell] : x_i = 0 \wedge y_i = 1\}| \\ &= \mathrm{wt}(x) + \mathrm{wt}(y) - 2 \cdot |\{i \in [\ell] : x_i = y_i = 1\}|, \end{aligned}$$

which equals $k - 2 \cdot |I(x,y)|$. One key observation is that different linear functions are at distance $1/2$ of one another. Hence, in this case $F(x,y) = g_{x,y}$ is 0.499-far from being k-linear.[16]

Associating $[n]$ with $\{0,1\}^\ell$, this means that the bit associated with z in $F(x,y)$, denoted $F(x,y)_z$ or $f_z(x,y)$, is $\sum_{i \in [\ell]} (x_i + y_i) \cdot z_i$, which in turn equals the sum (mod 2) of $\sum_{i \in [\ell]} x_i \cdot z_i$ and $\sum_{i \in [\ell]} y_i \cdot z_i$. (That is, $F(x,y)_z = f_z(x,y) = \sum_{i : z+_i = 1} x_i + \sum_{i : z+_i = 1} y_i$.) Hence, $F(x,y)_z = f_z(x,y)$ can be computed by the two-party protocol in which the first party (who holds x) sends $\sum_{i \in [\ell]} z_i \cdot x_i$

[15] The cases of odd k and $k > \ell/2$ will be treated in Section 4.

[16] Recall that ϵ-far (from Π) was defined as being at distance (from Π) that is strictly larger than ϵ.

to the second party, who (holds y and) responds with $\sum_{i\in[\ell]} z_i \cdot y_i$. That is, the bit sent by each party is the inner product (mod 2) of the desired location z and its own input.

Invoking the furthermore part of Theorem 7, with $B = 2$, it follows that the query complexity of 0.499-testing $k(\ell)$-linearity is at least $\mathtt{CC}_{1/3}(\mathrm{DISJ}^{(k)}_\ell)/2 = \Omega(k(\ell))$. ■

A generalization which may further clarify the argument. Theorem 10 is a special case of the following result that refers to properties of certain subsets of linear codes. Specifically, for any linear code of constant relative distance, we consider the set of codewords that correspond to the encoding of (ℓ-bit long) strings of a specific Hamming weight (i.e., $k(\ell)$). Theorem 10 refers to the special case in which the code is the Hadamard code (i.e., $n = 2^\ell$).

Theorem 11 (on the hardness of testing some sets of codewords in some codes): *Let* $\{C_\ell : \{0,1\}^\ell \to \{0,1\}^n\}_{\ell\in\mathbb{N}}$ *be a family of* linear *codes* (i.e., $C_\ell(x \oplus y) = C_\ell(x) \oplus C_\ell(y)$) *of constant relative distance. Then, for some constant* $\epsilon > 0$ *and any function* $k : \mathbb{N} \to \mathbb{N}$ *such that* $k(\ell)$ *is even and* $k(\ell) \leq \ell/2$*, the query complexity of* ϵ*-testing the property*

$$\Pi_n \stackrel{\text{def}}{=} \{C_\ell(z) : z\in\{0,1\}^\ell \wedge\ \mathrm{wt}(z)=k(\ell)\} \tag{8}$$

is $\Omega(k(\ell))$*. That is,* $\mathtt{Q}(\epsilon,\Pi_n) = \Omega(k(\ell))$*. Furthermore,* $\epsilon > 0$ *is any constant that is smaller than the relative distance of the code* C_ℓ*.*

Note that Π_n is a code; actually, it is a sub-code of the (linear) code C, but Π_n is not necessarily linear. In the special case that C is the Hadamard code, the property Π_n is $k(\ell)$-linearity; that is, the codewords of the Hadamard code corresponds to linear functions (from $\mathrm{GF}(2)^\ell$ to $\mathrm{GF}(2)$) and the codewords of Π_n are $k(\ell)$-linear functions. The following proof is very similar to the proof of Theorem 10, but it may be more clear because Π_n is now viewed as a property of n-bit strings (rather than as a property of Boolean functions with domain $[n] \equiv \{0,1\}^\ell$).

Proof: Again, we reduce from the communication problem $\{\mathrm{DISJ}^{(k)}_\ell = (P_\ell, S_\ell)\}_{\ell\in\mathbb{N}}$, and invoke Theorem 7. The reduction maps (x, y) to $F(x, y) = C_\ell(x \oplus y)$, and the i^{th} bit of $C_\ell(x \oplus y) = C_\ell(x) \oplus C_\ell(y)$ can be computed by exchanging the i^{th} bits of $C_\ell(x)$ and $C_\ell(y)$.

We again observe that for every $(x, y) \in P_\ell$ it holds that $\mathrm{wt}(x \oplus y) = k(\ell) - 2 \cdot |I(x, y)|$, where $I(x, y) = \{i \in [\ell] : x_i = y_i = 1\}$. Hence, if $(x, y) \in S_\ell$ (i.e., x and y are "disjoint"), then $\mathrm{wt}(x \oplus y) = k(\ell)$ and $F(x, y) = C_\ell(x \oplus y)$ is in Π_n. On the other hand, if $(x, y) \in P_\ell \setminus S_\ell$, then $\mathrm{wt}(x \oplus y) \neq k(\ell)$ and $F(x, y) = C_\ell(x \oplus y)$ is ϵ-far from Π_n, where $\epsilon > 0$ is any constant that is smaller than the relative distance of the code C_ℓ.

Finally, we invoke again the furthermore part of Theorem 7 with $B = 2$, and it follows that the query complexity of ϵ-testing Π_n is at least $\mathtt{CC}_{1/3}(\mathrm{DISJ}^{(k)}_\ell)/2 = \Omega(k(\ell))$. ■

Another implication of Theorem 11. As stated upfront, Theorem 10 follows as a special case of Theorem 11. Another result that follows easily is a generalization of Theorem 10 to the case of k-sparse polynomials of degree d (i.e., polynomials of degree d that have exactly k monomials). We state this result for polynomials over $\mathrm{GF}(2)$, but it can be proved for larger finite fields (while losing a factor that is logarithmic in the field size).[17]

[17] This is done by extending Theorem 11 to codes from $\{0,1\}^\ell$ to Σ^n for any finite Σ, which is viewed as an additive group. In this case, we use the reduction $F(x, y) = C_\ell(x - y)$, but emulating queries to $F(x, y)$ require $\log_2 |\Sigma|$ binary queries.

Corollary 12 (the complexity of k-sparse): *Let $d, m, k \in \mathbb{N}$ and Π_n denote the set m-variate polynomials of degree d over* GF(2) *having excetly k monomials, where $n = 2^m$. Then, if $k \leq \binom{m}{d}/2$ is even, then the query complexity of $0.99 \cdot 2^{-d}$-testing Π_n is $\Omega(k)$.*

Proof: For $\ell = \binom{m}{d}$, consider the Reed-Muller code of order d, which maps the ℓ-bit long description of a m-variate polynomial of degree d over GF(2) to its evaluation at all points of $\mathrm{GF}(2)^m$. This code has relative distance 2^{-d}, and so the claim follows by Theorem 11. ■

4 Reduction between testing problems

A natural method for obtaining lower bounds is via reductions. Indeed, this method is common practice in computability as well as in the theory of NP-completeness and in the study of other computational complexity classes (see, e.g., [3]). In each case, the definition of a reduction should preserve the relevant notion of feasible computation. Hence, when using reductions in the context of property testing, we should use reductions that preserve easy testability. Specifically, when we reduce testing property Π to testing property Π', it should be possible to answer each query to the reduced instance by making few queries to the original instance. In addition, the reduction should preserve the distance to the property, at least to some extent.

> **Teaching note:** Indeed, for the sake of simplicity, we confine ourselves to many-to-one reductions; that is, reductions that map an instance of the original problem to a single instance of the reduced problem such that YES-instances are mapped to YES-instances and NO-instances are mapped to NO-instances. (In the foregoing context of property testing, instances in Π are mapped to instances in Π' and instances that are far from Π are mapped to instances that are far from Π'.) That is, we consider the analogue of Karp-reductions rather than the analogue of Cook-reduction in which the reduction is a machine that given an instance of the original problem may issue queries to various instances of the reduced problem (see [3, Sec. 2.2.1]). We strongly discourage the reader from thinking, at least at this point, about what such a general notion (of a reduction) may mean in the context of property testing.

Definition 13 (local reductions): *Let $\Pi = \cup_{n\in\mathbb{N}}\Pi_n$ and $\Pi' = \cup_{n\in\mathbb{N}}\Pi'_n$ be such that Π_n and Π'_n contains functions from $[n]$ to R_n and R'_n, respectively. A mapping F_n from the set of functions $\{f : [n] \to R_n\}$ to the set of functions $\{f' : [n'] \to R'_{n'}\}$ is called a q-*local (ϵ, ϵ')-reduction of Π_n to $\Pi'_{n'}$ *if for every $f : [n] \to R_n$ the following conditions hold.*

1. Locality (local computation): *The value of $F_n(f)$ at any point $i \in [n']$ is determined by the value of f at q points in $[n]$; that is, there exist functions $Q_n : [n'] \to [n]^q$ and $V_n : [n'] \times R_n^q \to R'_{n'}$ such that $V_n(i, f(i_1), ..., f(i_q)) = (F_n(f))(i)$, where $(i_1, ..., i_q) = Q_n(i)$.*
2. Preservation of the properties: *If $f \in \Pi_n$, then $F_n(f) \in \Pi'_{n'}$.*
3. Partial preservation of distance to the properties: *If f is ϵ-far from Π_n, then $F_n(f)$ is ϵ'-far from $\Pi'_{n'}$.*

*For $q : \mathbb{N} \to \mathbb{N}$, the ensemble $\{F_n\}_{n\in\mathbb{N}}$ is called a q-*local (ϵ, ϵ')-reduction of Π to Π' *if there exists a function $N : \mathbb{N} \to \mathbb{N}$ such that for every $n \in \mathbb{N}$ it holds that F_n is a $q(n)$-local (ϵ, ϵ')-reduction of Π_n to $\Pi'_{N(n)}$. In such a case we say that Π is q-locally (ϵ, ϵ')-reducible to Π' (with length function N).*

Indeed, Definition 13 corresponds to a deterministic reduction, and this suffices in many cases. Nevertheless, we shall present a randomized version of Definition 13 at a later stage. But before doing so, let us examine the effect of such reductions.

Theorem 14 (local reductions preserve testability): *Let* $\Pi = \cup_{n\in\mathbb{N}}\Pi_n$ *and* $\Pi' = \cup_{n'\in\mathbb{N}}\Pi'_{n'}$ *be as in Definition 13. Suppose that* Π *is* q*-locally* (ϵ,ϵ')*-reducible to* Π' *with length function* N*. Then, if* Π' *can be* ϵ'*-tested with* $q'(n',\epsilon')$ *queries, then* Π *can be* ϵ*-tested with* $q'(N(n),\epsilon')\cdot q(n)$ *queries.*

Theorem 14 states the positive effect of a local reduction, but in the context of proving lower bounds one uses its counter-positive which asserts that *if the query complexity of* ϵ*-testing* Π *exceeds* $L(n,\epsilon)$*, then the query complexity of* ϵ'*-testing* $\Pi' = \cup_{n'}\Pi'_{n'}$ *exceeds* $L'(n',\epsilon') = L(n,\epsilon)/q(n)$ *for any* $n \in N^{-1}(n')$. We shall state this counter-positive below, after proving Theorem 14.

Proof: Let us fix any $n \in \mathbb{N}$ and let $n' = N(n)$. Given an ϵ'-tester T' for $\Pi'_{n'}$ as in the hypothesis, we construct an ϵ-tester for Π_n as follows. On input $f : [n] \to R_n$, the new tester invokes T' and answers each of its queries by using the local reconstruction procedure (i.e., Q_n and V_n) that is provided by the local reduction F_n. That is, the query $i \in [n']$ is answered by querying f at $i_1, ..., i_q$, where $(i_1, ..., i_q) = Q_n(i)$, and providing the value $V_n(i, f(i_1), ..., f(i_q))$. Hence, this tester, denoted T, makes $q(n)$ queries per each of the $q'(n',\epsilon')$ queries issued by T'. When T' halts, T just outputs the verdict provided by T'.

Turning to the analysis of T, we first observe that on input f algorithm T answers each query of T' according to $F_n(f)$. Hence, if $f \in \Pi_n$, then $F_n(f)$ is in $\Pi'_{n'}$, and T' will accept (with probability at least 2/3) and so will T. On the other hand, if f is ϵ-far from Π_n, then $F_n(f)$ is ϵ'-far from $\Pi'_{n'}$, and T' will reject (with probability at least 2/3) and so will T. The theorem follows. ■

Corollary 15 (lower bounds via local reductions, a counter-positive of Theorem 14): *Let* $\Pi = \cup_{n\in\mathbb{N}}\Pi_n$ *and* $\Pi' = \cup_{n'\in\mathbb{N}}\Pi'_{n'}$ *be as in Definition 13. Suppose that* Π *is* q*-locally* (ϵ,ϵ')*-reducible to* Π' *with length function* N*. Then, if the query complexity of* ϵ*-testing* Π *exceeds* $L(n,\epsilon)$*, then the query complexity of* ϵ'*-testing* $\Pi' = \cup_{n'}\Pi'_{n'}$ *exceeds* $L'(n',\epsilon') = \max_{n:N(n)=n'}\{L(n,\epsilon)/q(n)\}$.

Typically, $N : \mathbb{N} \to \mathbb{N}$ is non-decreasing and one-to one, and so we get $L'(n',\epsilon') = L(N^{-1}(n'),\epsilon)/q(N^{-1}(n))$.

Illustrating the application of the method. Recall that Theorem 10 provides a lower bound on the query complexity of testing k-linearity (of ℓ-variate Boolean functions) only in the case that $k(\ell) \leq \ell/2$ is even. Using two simple reductions, we establish the following.

Proposition 16 (Theorem 10, extended): *For every* $k : \mathbb{N} \to \mathbb{N}$*, the query complexity of* 0.499*-testing* k*-linearity is* $\Omega(\min(k(\ell), \ell - k(\ell)))$.

Proof Sketch: We first reduce ϵ-testing k-linearity of ℓ-variate Boolean functions to ϵ-testing $(k+1)$-linearity of $(\ell+2)$-variate Boolean functions. (This reduction allows to switch the parity of the linearity parameter.) The reduction just maps $f : \{0,1\}^\ell \to \{0,1\}$ to $f' : \{0,1\}^{\ell+2} \to \{0,1\}$ such that $f'(x_1 \cdots x_\ell x_{\ell+1} x_{\ell+2}) = f(x_1 \cdots x_\ell) + x_{\ell+1}$. Hence, each query to f' can be answered by making a single query to f (i.e., the query $\sigma_1 \cdots \sigma_{\ell+1}\sigma_{\ell+2}$ is answered by querying f at $\sigma_1 \cdots \sigma_\ell$ and returning $f(\sigma_1 \cdots \sigma_\ell) + \sigma_{\ell+1}$). Observe the distance of f from being k-linear equals the distance of f' from being $(k+1)$-linear. In particular, this yields a 1-local $(0.499, 0.499)$-reduction from the

case of even $k \leq \ell/2$ to the case of odd $(k+1) \leq (\ell/2)+1 = (\ell+2)/2$. Hence, applying Corollary 15, the lower bound of Theorem 10 is extended to the case of an odd linearity parameter.

The second reduction is from testing k-linearity of ℓ-variate Boolean functions (when $k \leq \ell/2$) to testing $(\ell-k)$-linearity of ℓ-variate Boolean functions (when $\ell-k \geq \ell/2$). The reduction just maps $f : \{0,1\}^\ell \to \{0,1\}$ to $f' : \{0,1\}^\ell \to \{0,1\}$ such that $f'(x) = f(x) + \sum_{i\in[\ell]} x_i$, where $x = x_1 \cdots x_\ell$. Again, each query to f' can be answered by making a single query to f. In this case the distance of f from being k-linear equals the distance of f' from being $(\ell-k)$-linear. In particular, this yields a 1-local $(0.499, 0.499)$-reduction of k-linearity to $(\ell-k)$-linearity. Hence, applying Corollary 15, the $\Omega(k)$ lower bound for testing k-linearity when $k \leq \ell/2$, yields a lower bound of $\Omega(k)$ for testing $(\ell-k)$-linearity when $\ell - k \geq \ell/2$. ■

Randomized reductions (advanced comment). Definition 13 captures only deterministic reductions. This is reflected by the main deterministic mapping F_n as well as by the auxiliary functions Q_n and V_n (used in the locality condition). Allowing randomized auxiliary algorithms in the locality condition is straightforward (and we should just require that they yield the correct value with probability at least 2/3). More care should be taken when allowing a randomized mapping F_n: In such a case, its randomness should be handed over to the algorithms used in the locality condition, or else different invocations of (the local computation procedure captured by) these algorithms may not yield values that are consistent with a single function $f' : [n'] \to R'_{n'}$ (but may rather yield values that fit different functions $f' : [n'] \to R'_{n'}$).[18] For sake of simplicity, in the following definition, we view the randomized mapping as a distribution of (deterministic) mappings and allow the auxiliary procedures to depend on the specific mapping chosen from that distribution.

Definition 17 (randomized local reductions): *Let $\Pi = \cup_{n\in\mathbb{N}}\Pi_n$ and $\Pi' = \cup_{n\in\mathbb{N}}\Pi'_n$ be be as Definition 13. A distribution of mappings $\mathcal{F}_n$ from the set of functions $\{f : [n] \to R_n\}$ to the set of functions $\{f' : [n'] \to R'_{n'}\}$ is called a* randomized q-local (ϵ, ϵ')-reduction of Π_n to $\Pi'_{n'}$ *if for every $f : [n] \to R_n$ the following conditions hold with probability at least 5/6 when the mapping F_n is selected according to the distribution $\mathcal{F}_n$.*

1. Locality (local computation): *There exist randomized algorithms $Q_n : [n'] \to [n]^q$ and $V_n : [n'] \times R_n^q \to R'_{n'}$, which may depend on F_n, such that for every $i \in [n']$ it holds that*

$$\mathbf{Pr}_{(i_1,...,i_q)\leftarrow Q_n(i)}[V_n(i, f(i_1), ..., f(i_q)) = (F_n(f))(i)] \;\geq\; 2/3. \tag{9}$$

2. Preservation of the properties: *If $f \in \Pi_n$, then $F_n(f) \in \Pi'_{n'}$.*

3. Partial preservation of distance to the properties: *If f is ϵ-far from Π_n, then $F_n(f)$ is ϵ'-far from $\Pi'_{n'}$.*

Randomized local reduction of Π to Π' are defined analogously to Definition 13.

Hence, if $f \in \Pi_n$ (resp., if f is ϵ-far from Π_n), then, with probability at least 5/6, over the choice of F_n, Conditions 1 and 2 both hold (resp., Conditions 1 and 3 both hold). When applying such a reduction, the error probability of the algorithms guaranteed by the locality condition (i.e.,

[18] A similar issue arises in the general definition of local computations, to be discussed in a subsequent lecture.

Condition 1) should be reduced according to the application.[19] (The error probability of the ϵ'-tester for Π' should also be reduced, say, to 0.1.)

Another type of reductions. The result regarding testing a property $\Pi' \cap \Pi''$ such that Π' is self-correctable, which was presented (as an exercise) in the lecture on testing juntas, can be interpreted as a reduction of testing $\Pi' \cap \Pi''$ to testing Π'', especially when Π' is easy to test and it is easy to decide membership in Π'' when given an input (that is promised to be) in Π'. For sake of clarity, we first restate the foregoing result, which refers to the notion of random self-correction defined in the said lecture.

Theorem 18 (restating a result from the lecture on testing juntas):[20] *Let Π' and Π'' be sets of functions ranging over D. Suppose that functions in Π' are randomly self-reducible by q queries, and that Π' and Π'' are ϵ-testable using $q'(\epsilon)$ and $q''(\epsilon)$ queries, respectively. Then, for every $\epsilon_0 < 1/q$, the property $\Pi' \cap \Pi''$ is ϵ-testable using $O(q'(\min(\epsilon, 1/3q))) + q \cdot \widetilde{O}(q''(\epsilon_0))$ queries.*

The positive application of this result is to derive a tester for $\Pi' \cap \Pi''$, when given testers for Π' and Π''. Here, we present a negative application: Given a lower bound on the query complexity of $\Pi' \cap \Pi''$ and assuming that q and $q'(\epsilon)$ are both relatively small, we derive a lower bound on the query complexity of Π''.

Corollary 19 (negative application of Theorem 18): *Let Π' be a set of functions ranging over D such that functions in Π' are randomly self-reducible by q queries, and Π' is ϵ-testable using $q'(\epsilon)$ queries. Suppose that $Q(\epsilon)$ is a lower bound on the query complexity of ϵ-testing $\Pi \cap \Pi''$, where Π'' is also a set of functions ranging over D. Then, for every $\epsilon_0 < 1/q$, the query complexity of ϵ_0-testing Π'' is $\max_{\epsilon \in (0,1/3q]}\{\widetilde{\Omega}((Q(\epsilon) - O(q'(\epsilon)))/q)\}$.*[21]

As an illustration to the application of Corollary 19, we use it to derive a lower bound on testing k-juntas. Towards this application, we consider the set of $k^{\leq}$-linear functions defined as the union of the sets of i-linear function for $i = 0, 1, ..., k$, and note that the lower bound for k-linearity holds also for $k^{\leq}$-linearity (see Exercise 4). The key observation is that the set of $k^{\leq}$-linear functions is the intersection of the set of linear functions and the set of k-juntas.

Corollary 20 (a lower bound on the query complexity of testing k-juntas): *For every $k(\ell) \leq (\ell/2) - 2$, the complexity of 0.24-testing k-juntas is $\widetilde{\Omega}(k)$.*

[19] Specifically, if the ϵ'-tester for Π' makes q' queries, then the error probability of these algorithms should be requced to $1/20q'$.

[20] Recall that the aforementioned exercise is proved based on a theorem that postulates the existence of a decision procedure for the promise problem (Π', Π'') rather than a tester for Π'' as postulated here. But as suggested in the guideline for that exercise, for any $\epsilon_0 < 1/q$, a tester of query complexity $q''(\epsilon)$ for Π'' yields a procedure of query complexity $q''(\epsilon_0)$ for distinguishing inputs in $\Pi' \cap \Pi''$ from inputs in $\Pi' \setminus \Pi''$. This is because every input in $\Pi' \setminus \Pi''$ is at distance at least $1/q$ from Π'' (since self-reducibility by q queries implies that distinct functions in Π' are at distance at least $1/q$ apart). Note that we need to invoke the tester for Π'' with a proximity parameter smaller than $1/q$ so to guarantee that inputs at distance exactly $1/q$ are rejected (w.h.p.).

[21] The poly-logarithmic factor in the $\widetilde{\Omega}$-notation is merely a logarithmic factor. We stress that all constants are universal (i.e., they are independent of Π' and Π''). Note that we lower-bounded $\max_{\epsilon \in (0,1]}\{\widetilde{\Omega}((Q(\epsilon) - O(q'(\min(\epsilon, 1/3q))))/q)\}$ by $\max_{\epsilon \in (0,1/3q]}\{\widetilde{\Omega}((Q(\epsilon) - O(q'(\epsilon)))/q)\}$, losing nothing in the typical cases in which $Q(1/3q) \geq \max_{\epsilon \in [1/3q,1]}\{Q(\epsilon)\}$.

We comment that a linear (in k) lower bound can be obtained by direct reduction from a communication complexity problem; see Exercise 5.

Proof: We let Π' denote the set of linear functions, and Π'' denote the set of $k(\ell)$-juntas. Recall that the set of linear functions is randomly self-reducible by three queries, and that it ϵ-testable by $O(1/\epsilon)$ queries. Observing that $\Pi' \cap \Pi''$ is the set of $k(\ell)^{\leq}$-linear functions, we use the fact that 0.499-testing this set requires $\Omega(k(\ell))$ queries (see Exercise 4). Now, invoking Corollary 19, we infer that 0.24-testing Π'' requires is $\widetilde{\Omega}((k(\ell) - O(1))/3)$ queries. ■

5 Lower bounds for restricted testers

Restricted algorithms may have higher complexity than general ones, and proving lower bounds regarding their complexity may be easier (even when these lower bounds are higher). Two natural restrictions in the context of property testing are the restriction to one-sided error probability and the restriction to non-adaptive queries. We mention that separations between such restricted tester and general testers are known in many (natural) cases (see, e.g., testing graph properties in the bounded-degree model [7, 10]), but there are also cases in which the restriction does not increase the complexity of testing (e.g., testing linear properties [2]).

5.1 One-sided error testers

When analyzing one-sided error testers, the ("indistinguishability") method captured by Corollary 3 takes a simpler form. The point is that in such a case, any function f having the property Π must be accepted by the tester with probability 1 (since the tester is allowed no error when $f \in \Pi$). Hence, it suffices to find a distribution F_0 of functions that are (typically) far from Π such that no low complexity machine that accepts each $f \in \Pi$ with probability 1 can reject F_0 with probability greater than 1/2.[22]

Theorem 21 (the method of indistinguishability, a one-sided error version): *Let $\Pi = \cup_{n\in\mathbb{N}}\Pi_n$, $\Gamma_\epsilon(\Pi_n)$, and $q : \mathbb{N} \times (0,1] \to \mathbb{N}$ be as in Theorem 2. Suppose that for some $\epsilon > 0$, $\eta_0 > 0$ and $n \in \mathbb{N}$, there exist a distribution F_0 such that $\mathbf{Pr}[F_0 \in \Gamma_\epsilon(\Pi_n)] \geq 1 - \eta_0$, and for every deterministic oracle machine M that makes at most $q(n,\epsilon)$ queries and accepts each $f \in \Pi$ with probability 1* (i.e., $|\mathbf{Pr}[M^f(n,\epsilon) = 1] = 1$ for each $f \in \Pi_n$) *it holds that $\mathbf{Pr}[M^{F_0}(n,\epsilon) = 1] > \frac{1}{3} + \eta_0$. Then, the query complexity of ϵ-testing* Π with one-sided error probability *is greater than $q(\cdot,\epsilon)$.*

Considering a machine M as postulated in Theorem 21, note that such a machine cannot reject a function when its partial view of it (i.e., the sequence of query and answer pairs)[23] matches a partial view of a function in f. Hence, the probability that M rejects F_0 (i.e., $\mathbf{Pr}[M^{F_0}(n,\epsilon) = 0]$) may be replaced by the probability that M sees a partial view that does not match any function in Π.[24] Thus, the hypothesis of Theorem 21 may be re-formulated as follows: *There exist a distribution F_0 such that $\mathbf{Pr}[F_0 \in \Gamma_\epsilon(\Pi_n)] \geq 1 - \eta_0$, and for every deterministic oracle machine M that makes at*

[22] Here we assume that "typically" means with probability greater than 5/6.

[23] The partial view that M has of f is the sequence of pairs $((i_1, f(i_1)), ..., (i_q, f(i_q)))$, where i_{j+1} is the $j+1^{\text{st}}$ query made by M after receiving the oracle answers $f(i_1), ..., f(i_j)$.

[24] The one-sided error condition only mandates that M rejects *only if* its partial view does not match any function in Π. (Needless to say, M may well reject *if* its partial view does not match any function in Π.)

most $q(n,\epsilon)$ queries it holds that the probability that M sees a partial view that does not match any function in Π is smaller than $\frac{2}{3}-\eta_0$.

Proof: Suppose that T is a one-sided error probability tester for Π, and let F_0' denote the distribution F_0 conditioned on $F_0 \in \Gamma_\epsilon(\Pi_n)$. Then, $\mathbf{Pr}[T^{F_0'}(n,\epsilon)=1] \leq 1/3$. Let T_r denote a residual deterministic machine (obtained by fixing the coins of T to r) such that $\mathbf{Pr}[T_r^{F_0'}(n,\epsilon)=1] \leq 1/3$. Then, $\mathbf{Pr}[T_r^{F_0}(n,\epsilon)=1] < 1/3+\eta_0$, and it follows that T must have query complexity greater than $q(\cdot,\epsilon)$. ■

The actual methodology. As hinted in the discussion Following the statement of Theorem 21, the methodology that arises here is to find distribution F_0 such that $\mathbf{Pr}[F_0 \in \Gamma_\epsilon(\Pi_n)] \geq 1-\eta_0$, and to show that any oracle machine that makes at most $q(n,\epsilon)$ queries to F_0 sees, with probability greater than $\frac{1}{3}+\eta_0$, a partial view that matches some function in Π.

Reduction from communication complexity. The methodology described in Section 3 can be adapted to provide a reduction between the *one-sided error probability versions* of the two types of problems. For details see [1, 5].

5.2 Non-adaptive testers

Also when analyzing non-adaptive testers, we can obtain a simplification of the ("indistinguishability") method captured by Corollary 3. In such a case, it suffices to consider non-adaptive deterministic machines, which is the same as just considering the projection of the relevant distributions on any size-bounded subset of the function domain.[25]

Theorem 22 (the method of indistinguishability, a non-adaptive version): *Let $\Pi = \cup_{n\in\mathbb{N}}\Pi_n$, $\Gamma_\epsilon(\Pi_n)$, and $q : \mathbb{N}\times(0,1] \to \mathbb{N}$ be as in Theorem 2. Suppose that for some $\epsilon > 0$, $\eta_0,\eta_1,\eta_2 > 0$ and $n \in \mathbb{N}$, there exist distributions F_1 and F_0 such that $\mathbf{Pr}[F_1 \in \Pi_n] \geq 1-\eta_1$ and $\mathbf{Pr}[F_0 \in \Gamma_\epsilon(\Pi_n)] \geq 1-\eta_0$, and for every set $Q \subset [n]$ of size $q(n,\epsilon)$ it holds that the projection of F_1 on Q is η_2-close to the projection of F_0 on Q; that is,*

$$\frac{1}{2}\cdot \sum_{v_1,...,v_q \in R_n} |\mathbf{Pr}[F_1(i_1)\cdots F_1(i_q)=v_1\cdots v_q] - \mathbf{Pr}[F_0(i_1)\cdots F_0(i_q)=v_1\cdots v_q]| \leq \eta_2. \qquad (10)$$

where $q = q(n,\epsilon)$ and $Q = \{i_1,...,i_q\}$. If $\eta_0+\eta_1+\eta_2 < 1/3$, then the non-adaptive *query complexity of ϵ-testing Π is greater than $q(\cdot,\epsilon)$.*

As shown in Exercise 3, the hypothesis of Theorem 22 implies the hypothesis of Corollary 3 with η_2 replaced by $|R_n|^{q(n,\epsilon)}\cdot\eta_2$. Typically, this observation is useful only when $\eta_2 = 0$ (see the proof of Proposition 4).

[25] Recall that the distinguishing gap of any machine when fed with one of two distributions is upper-bounded by the statistical distance between the two distributions, which is captured in Eq. (10).

Reduction from communication complexity. Adapting the methodology described in Section 3 to non-adaptive testers yields a way to lower bound their query complexity based on lower bounds on the complexity of *one-way communication protocols*.[26] Actually, it is even more natural to reduce from an even weaker model of communication protocols, known as the *simultaneous model*. In this model, each of the two parties holding an input, sends a single message to an auxiliary party, called the referee (who only has access to the common random string), and the referee is the sole producer of output. The proof of Theorem 7 is easily adapted to yield the following, where $\mathtt{Q}^{\mathrm{na}}$ and $\mathtt{CC}^{\mathrm{sim}}$ denote the corresponding complexity measures (i.e., the query complexity of non-adaptive testers and the communication complexity of simultaneous protocols).

Theorem 23 (the communication complexity method, a non-adaptive version): *Let $\Psi = (P, S)$ be a promise problem such that $P, S \subseteq \{0,1\}^{2\ell}$, and let $\Pi \subseteq \{0,1\}^n$ be a property. For $\epsilon, \eta > 0$, suppose that the mapping $F : \{0,1\}^{2\ell} \to \{0,1\}^n$ satisfies the following two conditions:*

1. *For every $(x, y) \in P \cap S$, it holds that $F(x, y) \in \Pi$.*
2. *For every $(x, y) \in P \setminus S$, it holds that $F(x, y)$ is ϵ-far from Π.*

Then, $\mathtt{Q}^{\mathrm{na}}_{\eta}(\epsilon, \Pi) \geq \mathtt{CC}^{\mathrm{sim}}_{2\eta}(\Psi)/B$, where $B = \max_{i\in[n]}\{\mathtt{CC}^{\mathrm{sim}}_{\eta/\ell}(f_i)\}$ and $f_i(x, y)$ is the i^{th} bit of $F(x, y)$. Furthermore, if $B = \max_{i\in[n]}\{\mathtt{CC}^{\mathrm{sim}}_{0}(f_i)\}$, then $\mathtt{Q}^{\mathrm{na}}_{\eta}(\epsilon, \Pi) \geq \mathtt{CC}^{\mathrm{sim}}_{\eta}(\Psi)/B$.

Proof Sketch: Given a non-adaptive ϵ-tester with error η for Π and simultaneous communication protocols for the f_i's, we present a simultaneous protocol for solving Ψ. The key idea is that, using their shared randomness, the two parties (holding x and y, respectively) and the referee can emulate the execution of the non-adaptive tester, while providing it with virtual access to $F(x, y)$. Specifically, if the tester queries the i^{th} bit of the oracle, then the two parties provide the referee with messages that allow it to obtain the value of $f_i(x, y)$. The referee feds all answers to the tester, and outputs whatever it has output.

The main difference between this emulation and the one that is carried out in the proof of Theorem 7 is that the tester is non-adaptive, and this fact allows its emulation in the simultaneous communication model. Specifically, the tester generates all its queries as a function of its internal coin tosses (and n) only, which means that both parties obtain these queries based on the shared randomness only (i.e., without interacting). Each party then sends the referee a message that corresponds to the execution of the suitable protocol; that is, if location i in $F(x, y)$ is queried, then each party sends the message that allows for the computation of $f_i(x, y)$ in the simultaneous protocol. The referee gets all these messages, reconstructs the corresponding $f_i(x, y)$'s, feeds them to the tester, obtains its verdict, and outputs it. Hence, the two parties only invoke the query-generation stage of the tester, whereas the referee invokes its decision stage. ■

6 Additional comments and suggested reading

The methodology captured in Theorems 1 and 2 suggests to prove lower bounds on the worst-case complexity of randomized algorithms (e.g., property testers) by proving lower bounds on the "distributional complexity" of corresponding deterministic algorithms (which are only required to solve the problem "on the average"). This methodology is commonly attributed to Yao [13],

[26] In such protocols the first part sends a single message to the second party, who produces the output.

who employed it in the context of several non-uniform models of computation such as Boolean circuits and communication complexity. It was first employed in the context of property testing by Goldreich, Goldwasser, and Ron [6, Sec. 4.1].

Recall that this methodology is "complete" in the sense that any valid lower bound can be proved by using it. The latter assertion, which can be traced to an earlier work of Yao [12] and is proved by employing von Neumann's Minimax Theorem [11], is often confused with the methodology itself. That is, results derived via Theorems 1 and 2 use the methodology of Yao [13], not the "Minimax principle" of von Neumann [11] as employed by Yao [12]. For further discussion of this point, the interested reader is referred to [4, Apdx. A.1].

The methodology of deriving lower bounds on the query complexity of property testing based on lower bounds on communication complexity was introduced by Blais, Brody, and Matulef [1]. As noted in Section 3, we find this connection quite surprising, since property testing problems have no "topology" that can be naturally 2-partitioned to fit the two-party setting of communication complexity. Nevertheless, using this methodology, Blais *et al.* [1] were able to resolve a fair number of open problems (cf., e.g., [1, Thms. 1.1-1.3]). Our presentation of their methodology is based on [5], which generalizes the presentation of Blais *et al.* [1].[27] (We believe that the more general formulation of the methodology is easier to use as well as more intuitive than the original one.)

Exercises

Exercise 1 (generalization of Theorem 1): *Let Π and q be as in Theorem 1. Suppose that for some $p \in (0, 0.5)$ and $\epsilon > 0$ and $n \in \mathbb{N}$, there exists a distribution F of functions from $[n]$ to R_n such that for every deterministic oracle machine M that makes at most $q(n, \epsilon)$ queries it holds that*

$$\mathbf{Pr}[F \in \Pi_n \wedge M^F(n, \epsilon) \neq 1] + \mathbf{Pr}[F \in \Gamma_\epsilon(\Pi_n) \wedge M^F(n, \epsilon) \neq 0] \ > \ p,$$

where $\Gamma_\epsilon(\Pi_n)$ denotes the set of functions (from $[n]$ to R_n) *that are ϵ-far from Π_n. Then, any ϵ-tester of error probability p for Π makes more than $q(\cdot, \epsilon)$ queries.*

Exercise 2 (on the completeness of Theorem 2): *Recall that Theorem 1 is complete in the sense that any valid lower bound can be proved by using it. Prove the essentially same assertion with respect to Theorem 2. That is, show that if Π satisfies the hypothesis of Theorem 1, then it satisfies the hypothesis of Theorem 2 (possiblly with a threshold of $1/2$ rather than $1/3$).*[28]

Guideline: Let F be as the hypothesis of Theorem 1, and let F_1 (resp., F_0) denote the distribution of F conditioned on $F \in \Pi$ (resp., $F \in \Gamma_\epsilon(\Pi)$). Suppose, for simplicity, that $\mathbf{Pr}[F \in \Pi] = \mathbf{Pr}[F \in \Gamma_\epsilon(\Pi)] = 1/2$. Then, by the hypothesis of Theorem 1, $\frac{1}{2} \cdot (1 - \mathbf{Pr}[M^{F_1} = 1]) + \frac{1}{2} \cdot \mathbf{Pr}[M^{F_0} = 1] > 1/3$, which is equivalent to $\mathbf{Pr}[M^{F_1} = 1] - \mathbf{Pr}[M^{F_0} = 1] < 1/3$. By considering the machine that complements the output of M, we also have $\mathbf{Pr}[M^{F_0} = 1] - \mathbf{Pr}[M^{F_1} = 1] < 1/3$, and the hypothesis

[27] Loosely speaking, the formulation of Blais *et al.* [1] refers to the special case (of Theorem 7) in which each $f_i(x, y)$ is a function of the i^{th} bit of x and the i^{th} bit of y (i.e., x_i and y_i). Indeed, in that case, $n = \ell$ and $B = 2$ (by the straightforward protocol in which the two parties exchange the relevant bits (i.e., x_i and y_i)). Typically, using this formulation requires reducing the original communication (complexity) problem into an auxiliary one, and applying the reduction on the latter. Our formulation frees the user from this maneuver, and makes the reduction from the original (communication) problem (to property testing) more transparent. See further discussion in [5].

[28] Note that this good enough for claiming "completeness", since an indistinguishability gap of $1/2$ yields that each algorithm is wrong with probability at least $1/4$ (see the proof of Theorem 2). Still, we would welcome a converse that has no slackness, although we do not know if it is possible.

of Theorem 2 follows. In general, for $q_1 \stackrel{\text{def}}{=} \mathbf{Pr}[F \in \Pi]$ and $q_0 \stackrel{\text{def}}{=} \mathbf{Pr}[F \in \Gamma_\epsilon(\Pi)]$, it holds that $q_1 \cdot (1 - \mathbf{Pr}[M^{F_1} = 1]) + q_0 \cdot \mathbf{Pr}[M^{F_0} = 1] > 1/3$. Observing that $q_0, q_1 \in (1/3, 2/3)$ (since otherwise a trivial algorithm violates the hypothesis), we get $\frac{1}{3} \cdot (1 - \mathbf{Pr}[M^{F_1} = 1]) + \frac{1}{3} \cdot \mathbf{Pr}[M^{F_0} = 1] > 1/6$, which is equivalent to $\mathbf{Pr}[M^{F_1} = 1] - \mathbf{Pr}[M^{F_0} = 1] < 1/2$.

Exercise 3 (on the distinguishing gap of adaptive testers): *Let $\eta \geq 0$ and suppose that X and Y are random variables distributed over $\{0,1\}^n$ such that, for every fixed $I \subseteq [n]$ of size at most q, the statistical difference between X_I and Y_I is at most η.*

1. *Prove that for any* (adaptive) *oracle machine M that makes q queries, it holds that*

$$\left|\mathbf{Pr}[M^X = 1] - \mathbf{Pr}[M^Y = 1]\right| \leq 2^q \cdot \eta.$$

   Note that this holds also for $\eta = 0$, which means that if X_I and Y_I are identically distributed then no q-query oracle machine can distinguish X from Y with any positive gap.

2. *Demonstrate that the upper bound provided in Part 1 is quite tight by considering the following two distributions X and Y that are each almost uniform over $\{0,1\}^n$ such that the first $\log_2 n$ bits indicate a bit-position that is set to 0 in X and to 1 in Y. That is, for $n = q - 1 + 2^{q-1}$, let X* (resp., Y) *be the uniform distribution except the bit that corresponds to location $q + \sum_{j \in [q-1]} 2^{j-1} X_j$ set to 0* (resp., location $q + \sum_{j \in [q-1]} 2^{j-1} Y_j$ set to 1).

We stress that the proof of Part 1 is generic, and better bounds can be obtained in many cases (i.e., for specific pairs (X, Y)).

Guideline: Part 1 is proved by fixing the coins of M and considering all 2^q possible answers to the corresponding sequence of q queries. (Indeed, a possible sequence of answers, uniquely determines the sequence of queries (made by the residual deterministic machine).) By the hypothesis, the difference in the probability that each such sequence of answers occurs in the two distributions is at most η, and the claim follows.[29] Part 2 follows by observing that an adaptive q-query machine

[29] For $\alpha \in \{0,1\}^q$ and $i \in [q]$, let $M'(\alpha_{[i-1]})$ denote the i^{th} query of the residual deteterministic machine when getting the answers $\alpha_1, ..., \alpha_{i-1}$, and $M'(\alpha)$ denote the corresponding final output. Then, $M^z = 1$ if and only if there exist $\alpha \in \{0,1\}^q$ such that $M'(\alpha) = 1$ and $\alpha_i = z_{M'(\alpha_{[i-1]})}$ for every $i \in [q]$. It follows that

$$\begin{aligned}
&\left|\mathbf{Pr}[M^X = 1] - \mathbf{Pr}[M^Y = 1]\right| \\
&= \left| \sum_{\alpha \in \{0,1\}^q : M(\alpha) = 1} \mathbf{Pr}[(\forall i \in [q])\ X_{M'(\alpha_{[i-1]})} = \alpha_i] \ - \sum_{\alpha \in \{0,1\}^q : M(\alpha) = 1} \mathbf{Pr}[(\forall i \in [q])\ Y_{M'(\alpha_{[i-1]})} = \alpha_i] \right| \\
&\leq \sum_{\alpha \in \{0,1\}^q} \left| \mathbf{Pr}[(\forall i \in [q])\ X_{M'(\alpha_{[i-1]})} = \alpha_i] - \mathbf{Pr}[(\forall i \in [q])\ Y_{M'(\alpha_{[i-1]})} = \alpha_i] \right| \\
&= \sum_{\alpha \in \{0,1\}^q} \left| \mathbf{Pr}[X_{M'(\lambda)} \cdots X_{M'(\alpha_{[q-1]})} = \alpha] - \mathbf{Pr}[Y_{M'(\lambda)} \cdots Y_{M'(\alpha_{[q-1]})} = \alpha] \right| \\
&\leq \sum_{\alpha \in \{0,1\}^q} \max_{\beta \in \{0,1\}^q} \left\{ \left| \mathbf{Pr}[X_{M'(\lambda)} \cdots X_{M'(\alpha_{[q-1]})} = \beta] - \mathbf{Pr}[Y_{M'(\lambda)} \cdots Y_{M'(\alpha_{[q-1]})} = \beta] \right| \right\}
\end{aligned}$$

which is at most $2^q \cdot \eta$. (Indeed, we have upper-bounded the max-norm distance between the $X_{M'(\lambda)} \cdots X_{M'(\alpha_{[q-1]})}$ and $Y_{M'(\lambda)} \cdots Y_{M'(\alpha_{[q-1]})}$ by their total variation distance.) Note that the fixing of M's coins does simplify the exposition of the argument, and it can be justified as in the proof of Theorem 1. Alternatively, Theorem 2 asserts that, for the purpose of proving query complexity lower bounds, it suffices to consider deterministic testers.

can perfectly distinguish between X and Y (i.e., has distinguishing gap 1), whereas for every fixed $I \subseteq [n]$ of size at most q the statistical difference between X_I and Y_I is at most $|I|/(n-(q-1)) = \widetilde{\Omega}(2^{-q})$.

Exercise 4 (a lower bound for testing $k^{\leq}$-linearity): *A function $f : \mathrm{GF}(2)^\ell \to \mathrm{GF}(2)$ is called $k^{\leq}$-*linear *if it is linear and depends on at most k of its variables. Show that for every $k(\ell) \leq \ell/2$, the query complexity of 0.499-testing $k(\ell)^{\leq}$-linearity is $\Omega(k(\ell))$.*

Guideline: We reduce from the communication complexity problem that is the complement of $\{\mathrm{DISJ}_\ell^{(k)}\}_{\ell\in\mathbb{N}}$; that is, the YES-instances are pairs (x,y) such that $I(x,y) \stackrel{\text{def}}{=} \{i \in [\ell] : x_i = y_i = 1\} \neq \emptyset$. Note that the communication complexity of problems remains unchanged by complementation. Finally, note that the reduction used in the proof of Theorem 10 maps intersecting pairs to $(k-2)^{\leq}$-linear functions, and non-intersecting pairs to k-linear, which are 0.499-far from being $(k-2)^{\leq}$-linear.[30]

Exercise 5 (a lower bound for testing k-juntas): *Show that for every even $k(\ell) \leq (\ell/2) - 2$, the complexity of 0.499-testing k-juntas is $\Omega(k)$.*

Guideline: Just use the same reduction in Exercise 4, while noting that $(k-2)^{\leq}$-linear functions are $(k-2)$-junta, whereas k-linear functions are 0.499-far from being k-juntas.

References

[1] E. Blais, J. Brody, and K. Matulef. Property Testing Lower Bounds via Communication Complexity. *Computational Complexity*, Vol. 21 (2), pages 311–358, 2012. Extended abstract in *26th CCC*, 2011.

[2] E. Ben-Sasson, P. Harsha, and S. Raskhodnikova. 3CNF Properties are Hard to Test. *SIAM Journal on Computing*, Vol. 35(1), pages 1–21, 2005.

[3] O. Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.

[4] O. Goldreich. On Multiple Input Problems in Property Testing. *18th RANDOM*, pages 704–720, 2014.

[5] O. Goldreich. On the Communication Complexity Methodology for Proving Lower Bounds on the Query Complexity of Property Testing. *ECCC*, TR13-073, 2013.

[6] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, pages 653–750, July 1998. Extended abstract in *37th FOCS*, 1996.

[7] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, pages 302–343, 2002. Extended abstract in *29th STOC*, 1997.

[30] All the above refers to the case of even k; the case of odd k can be handled as in the proof of Proposition 16.

[8] B. Kalyanasundaram and G. Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Disc. Math. and Alg.*, Vol. 5 (4), pages 545–557, 1992.

[9] E. Kushilevitz and N. Nisan. *Communication complexity.* Cambridge University Press, 1997.

[10] S. Raskhodnikova and A. Smith. A note on adaptivity in testing properties of bounded degree graphs. *ECCC*, TR06-089, 2006.

[11] J. von Neumann. Various techniques used in connection with random digits. *Applied Math Series*, Vol. 12, pages 36–38, 1951. Reprinted in *von Neumann's Collected Works*, Vol. 5, pages 768–770, Pergamon, 1963.

[12] A.C.C. Yao. Probabilistic complexity: towards a unified measure of complexity. In *18th IEEE Symposium on Foundations of Computer Science*, pages 222–227, 1977.

[13] A.C.C. Yao. Lower Bounds by Probabilistic Arguments. In *24th IEEE Symposium on Foundations of Computer Science*, pages 420–428, 1983.