Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 115 (2014)

Deconstructions of Reductions from Communication Complexity to Property Testing using Generalized Parity Decision Trees*

Roei Tell
Department of Computer Science
Weizmann Institute of Science
roei.tell@weizmann.ac.il

September 25, 2014

Abstract

A few years ago, Blais, Brody, and Matulef (2012) presented a methodology for proving lower bounds for property testing problems by reducing them from problems in communication complexity. Recently, Bhrushundi, Chakraborty, and Kulkarni (2014) showed that some reductions of this type can be deconstructed to two separate reductions, from communication complexity to randomized parity decision trees and from the latter to property testing.

This work follows up on these ideas. We introduce a model called *linear-access algorithms*, which is a generalization of randomized parity decision trees, and show several methods to reduce communication complexity problems to problems for linear-access algorithms and problems for linear-access algorithms to property testing problems. This approach yields a new interpretation for several well-known reductions, since we present these reductions as a composition of two steps with fundamentally different functionalities.

Furthermore, we demonstrate the potential of proving lower bounds on property testing problems by reducing them directly from problems for linear-access algorithms. In particular, we provide an alternative and simple proof for a known lower bound of $\Omega(k)$ queries on testing "k-linearity"; that is, the property of k-sparse linear functions over $\mathbb{F}_2$. This alternative proof relies on a theorem by Linial and Samorodnitsky (2002). We then extend this result to a new lower bound of $\Omega(s)$ queries for testing s-sparse degree-d polynomials over $\mathbb{F}_2$, for any $d \in \mathbb{N}$. In addition we provide a simple proof for the hardness of testing some families of linear subcodes.

We present an unrelated result in an appendix. In property testing, testers that always accept inputs that are in the property (i.e., testers with one-sided error) are natural and common. We show that the dual notion, testers that always reject inputs that are far from the property, seems to be a notion of limited scope.

Keywords: Property Testing, Communication Complexity, Parity Decision Trees, Linear-Access Algorithms, Affine Subspaces, Linear Codes, One-Sided Error.

*An alternative title: Property Testing Lower Bounds via a Generalization of Randomized Parity Decision Trees.

ISSN 1433-8092

Contents

1 Introduction **3**
1.1 Background . . . 3
1.2 The current study . . . 3

2 Preliminaries **5**
2.1 Standard notations . . . 5
2.2 The Hadamard code and the Reed-Muller code . . . 5
2.3 Three computational models . . . 5

3 Reductions Between the Computational Models **7**
3.1 Reducing linear-access problems to property testing problems . . . 7
3.2 Reducing communication problems to linear-access problems . . . 11
3.3 Detour: A communication model limited to linear functions . . . 12

4 Deconstructions of Reductions from Communication Complexity to Property Testing **14**
4.1 The deconstruction and a generic observation . . . 14
4.2 Deconstructions of known results . . . 15

5 Proving Lower Bounds in Property Testing by Reductions from Linear-access Algorithms **18**
5.1 Reductions of the form of the Hadamard code . . . 18
5.2 A technique for proving lower bounds on linear-access algorithms . . . 19
5.3 A lower bound on testing a family of linear subcodes . . . 21
5.4 A lower bound on testing sparse linear functions and polynomials . . . 23

6 Digest and Open Questions **25**
6.1 Proving lower bounds in property testing via linear-access algorithms . . . 25
6.2 Linear-access algorithms and parity decision trees . . . 26
6.3 Investigating the connection between communication complexity and property testing . . . 27

Acknowledgements **27**

References **27**

Appendix: Testers that Always Reject Inputs that are Far from the Property **29**

1 Introduction

1.1 Background

Property testing (see [21, 13]) is the study of probabilistic algorithms that inspect a given object in few selected locations, and try to decide whether the object has some predetermined property or is significantly different from any object having that property. This is a widely-studied model in theoretical computer science, which is closely related to probabilistically checkable proofs, coding theory, computational learning theory and more.

A few years ago, Blais, Brody, and Matulef [3] discovered a connection between property testing and communication complexity — another widely-studied model in theoretical computer science. In communication complexity, two parties communicate with each other to jointly compute some function of their inputs. Blais, Brody and Matulef presented a methodology, generalized later by Goldreich [11], to reduce any communication complexity decision problem (where the parties compute a Boolean function) to some corresponding property testing problem.

Loosely speaking, the methodology consists of having the two parties in a communication setting use a suitable *combining function* that combines their inputs into an object for property testing. Both parties separately run identical copies of a tester for the combined object, and provide it with virtual access to that object: Whenever the tester wants to inspect the object at some location, the two parties communicate with each other to compute the relevant part of the combined object, and answer the query accordingly. At the end, the parties decide according to the decision of the tester.

This methodology has been useful in proving lower bounds for property testing problems, relying on known lower bounds in communication complexity. That is, a known "hard" problem in communication complexity is reduced, using a suitable *combining function*, to a target property testing problem, thereby proving that the latter problem is also "hard". Further details can be found in [3, 11].

The main observation leading to this work is that *in many known uses of this methodology, the combining function is computationally very simple*, scarcely using the unlimited computational power of the communicating parties. In particular, in several well-known reductions the two parties only need to compute *linear functions of their inputs* during the communication protocol. In these cases, the same property testing problem could be reduced from a weaker model, in which both parties are only allowed to compute linear functions of their inputs (and not any arbitrary function, as in standard communication complexity). The main question motivating this work is therefore:

> *Can lower bounds in property testing be proved by reducing from a model in which the algorithms involved only compute linear functions of their input?*

Such reductions have the potential of proving lower bounds that are tighter than currently known bounds, and of providing simpler proofs for known lower bounds.

1.2 The current study

In this study we examine one candidate for such a weaker model from which to reduce to property testing. Towards introducing the model, consider a communication setting in which both parties can only compute and send each other linear functions of their respective inputs. Since the parties only compute linear functions of the input pair (x, y), this model is equivalent, up to a constant factor in the number of queries, to a model outside the realm of communication complexity, in which a single party computes linear functions of its input, $w = x \circ y$, and needs to decide whether w belongs to some predetermined subset of strings $\mathcal{W}$. This latter model is known as *randomized parity decision trees*.

In fact, Bhrushundi, Chakraborty, and Kulkarni [2] recently showed two reductions from randomized parity decision trees to property testing, and suggested that tighter lower bounds on property testing problems might be achieved by reducing to them directly from the intermediary model. They further showed that some reductions from communication complexity to property testing can be deconstructed to two separate reductions, from communication complexity to randomized parity decision trees and from the latter to property testing.

This work follows up on these ideas. We consider a generalized intermediary model in which, for a finite field $\mathbb{F}$, a probabilistic algorithm tries to decide whether an input $w \in \mathbb{F}^n$ belongs to a predetermined subset of strings $\mathcal{W} \subseteq \mathbb{F}^n$ or does not belong to it. The algorithm may only issue *linear queries* to its input, that is, queries of the form $q = (q_1, ..., q_n) \in \mathbb{F}^n$ to be answered by $q(w) = \sum_{i=1}^{n} q_i w_i$ (over $\mathbb{F}$) and it tries to minimize the number of queries it makes. We call these probabilistic algorithms *linear-access algorithms*, and the problem of deciding a subset $\mathcal{W} \subseteq \{0,1\}^n$ with a linear-access algorithm is called a *linear-access problem*.

Indeed, when $\mathbb{F} = \mathbb{F}_2$ the linear-access model essentially coincides with the model of *randomized parity decision trees*[1]. The latter is a probabilistic version of the known model of *deterministic* parity decision trees, that has received much recent attention (see, e.g., [9, 19, 25, 23]). We discuss the differences in the underlying techniques in Section 6.2, after presenting our results.

Organization and main contributions. After presenting basic definitions in Section 2, including the definition of linear-access algorithms, in Section 3 we present several methods to reduce communication complexity problems to linear-access problems and linear-access problems to property testing problems. In particular, in Section 3.1 we show that *all properties of low-degree rational functions over finite fields* are reducible from linear-access algorithms; and ditto with respect to *all subcodes of linear codes with constant relative distance*. The main contributions of this paper are presented in Sections 4 and 5:

1. In Section 4 we use the linear-access model to offer a *new interpretation of several existing results*. Specifically, we deconstruct several well-known reductions from communication complexity to property testing, presenting each of them as the *composition of two reductions with fundamentally different functionalities*: The first from a communication problem to a linear-access problem, and the second from the linear-access problem to property testing.

2. In Section 5 we demonstrate the potential of proving lower bounds for property testing problems by *reducing them directly from linear-access problems*. We start by presenting, in Section 5.2, a simple technique for proving lower bounds on linear-access problems, which relies on analysis of affine subspaces in finite fields. This technique enables proving lower bounds on properties reducible from linear-access algorithms (e.g., all properties of low-degree polynomials) by tackling a potentially simpler challenge of analyzing affine subspaces.

   In Sections 5.3 and 5.4 we use the aforementioned technique, followed by reductions to property testing, to prove lower bounds in property testing. Specifically, we provide an *alternative and simple proof* for a known lower bound of $\Omega(\min\{k, n-k\})$ queries for testing "k-linearity"; that is, the property of k-sparse n-variate linear functions over $\mathbb{F}_2$.[2] We then extend this result to a *new lower bound* of $\Omega(\{s, \binom{n}{d} - s\})$ queries for testing s-sparse n-variate polynomials of total degree

[1] The equivalence of linear-access algorithms over $\mathbb{F} = \mathbb{F}_2$ and randomized parity decision trees depends on the definition of the latter. Specifically, randomized parity decision trees are sometimes defined as arbitrary distributions over parity decision trees (cf., e.g., [2]), whereas we define linear-access algorithms as randomized oracle machines (see Definition 2.4). A gap between the models exists when considering distributions over parity decision trees that cannot be computed by randomized oracle machines.

[2] This proof also appears in a self-contained form (see our technical report [24]).

d over $\mathbb{F}_2$, for any $d \in \mathbb{N}$. In addition, we show an $\Omega(n)$ lower bound on testing the property $\{C(x \circ y) : x, y \in \{0,1\}^n \wedge \langle x, y \rangle = 1\}$, where C is an arbitrary linear code with constant relative distance and $\langle \cdot, \cdot \rangle$ denotes inner product mod 2.

3. In Section 5.1 we highlight a limitation of certain reductions from linear-access algorithms to property testing. Following [2], we show that reductions of the form corresponding to the Hadamard code are unlikely to be helpful in proving lower bounds on target properties (where the target properties in this case are properties of linear functions).

In Section 6 we present several open questions and suggest research directions related to linear-access algorithms.

2 Preliminaries

2.1 Standard notations

Some standard notations that we will use include $\mathbb{F}^n$ for an n-dimensional vector space over a finite field $\mathbb{F}$; and v_i for the i^{th} coordinate of $v \in \mathbb{F}^n$. When $n = |\mathbb{F}|^m$ (for some integer m), we sometimes identify $\mathbb{F}^m$ with $[n]$, and for $x \in \mathbb{F}^m$ we denote by v_x the x^{th} coordinate of $v \in \mathbb{F}^n$. When referring to a specific field with q elements we denote it by $\mathbb{F}_q$.

We denote the addition mod 2 operator by $\oplus$. For $u, v \in \{0,1\}^n$, we define $\langle u, v \rangle$ to be their inner product mod 2, that is $\oplus_{i=1}^n u_i v_i$. We also define the Hamming weight of $w \in \{0,1\}^n$ to be $\|w\|_1 = \sum_{i=1}^n w_i$.

2.2 The Hadamard code and the Reed-Muller code

We define two standard linear error-correcting codes that will be used throughout the paper — the Hadamard code and the Reed-Muller code.

Definition 2.1 (Hadamard code): *Let $n \in \mathbb{N}$ and $\mathbb{F}$ be a finite field. The* Hadamard code *is a function $H : \mathbb{F}^n \to \mathbb{F}^{|\mathbb{F}|^n}$ such that for $w \in \mathbb{F}^n$, the coordinates of $H(w)$ are the evaluations of $\langle w, x \rangle$ at any $x \in \mathbb{F}^n$. In other words, for any $w, x \in \mathbb{F}^n$, $H(w)_x = \langle w, x \rangle$.*

Definition 2.2 (Reed-Muller code): *For $m, d \in \mathbb{N}$, let $n = \binom{m+d}{d}$ and $\mathbb{F}$ be a finite field. The* $[|\mathbb{F}|, d, m]$-Reed-Muller code *is a function $RM_{d,m} : \mathbb{F}^n \to \mathbb{F}^{|\mathbb{F}|^m}$ such that the coordinates of $RM_{d,m}(w)$ are the evaluations, at any $x \in \mathbb{F}^m$, of the m-variate polynomial of total degree at most d whose coefficients are represented in the coordinates of w. That is, fixing a bijection between the coefficients of m-variate polynomials of degree at most d and the set $[n]$, denote by p_w the polynomial whose coefficients are represented in w, and let $RM_{d,m}(w)_x = p_w(x)$ for any $w \in \mathbb{F}^n$ and $x \in \mathbb{F}^m$.*

2.3 Three computational models

We define each of the three computational models referred to in this paper — property testing, linear-access algorithms, and communication complexity protocols. Since some of the reductions we will discuss involve promise problems, we present all three models in this more general setting. We also define respective complexity measures for each of the models.

Definition 2.3 (property testing): *For $l \in \mathbb{N}$ and a finite set Σ, let $\mathcal{U}, \mathcal{P} \subseteq \Sigma^l$ and $\Pi = (\mathcal{U}, \mathcal{P})$, and let $\epsilon > 0$. An* ϵ-tester *for Π is a randomized oracle machine T that satisfies the following two conditions:*

1. *If $z \in \mathcal{U} \cap \mathcal{P}$ then $Pr[T^z(1^l) = 1] \geq \frac{2}{3}$*
2. *If $z \in \mathcal{U}$ is ϵ-far from $\mathcal{P}$, then $Pr[T^z(1^l) = 0] \geq \frac{2}{3}$, where the distance between z and $\mathcal{P}$ is $\min_{z' \in \mathcal{P}} \left\{ \frac{|i \in [l]: z_i \neq z'_i|}{l} \right\}$.*

The query complexity *of T is the maximum (over all $z \in \Sigma^l$ and the internal coin tosses of T) number of oracle queries that T makes. The* query complexity of ϵ-testing Π, *denoted $\mathtt{PT}(\epsilon, \Pi)$, is the minimum query complexity of all ϵ-testers for Π.*

Definition 2.4 (linear-access algorithms): *For $n \in \mathbb{N}$ and a finite field $\mathbb{F}$, let $\mathcal{Q}, \mathcal{W} \subseteq \mathbb{F}^n$ and $\Phi = (\mathcal{Q}, \mathcal{W})$. A* linear-access algorithm *solving Φ is a randomized oracle machine M that satisfies the following two conditions:*

1. *If $w \in \mathcal{Q} \cap \mathcal{W}$ then $Pr[M^{H(w)}(1^n) = 1] \geq \frac{2}{3}$*
2. *If $w \in \mathcal{Q} \setminus \mathcal{W}$ then $Pr[M^{H(w)}(1^n) = 0] \geq \frac{2}{3}$*

where H is the Hadamard code (as in Definition 2.1).

The query complexity *of M is the maximum (over all $w \in \mathbb{F}^n$ and internal coin tosses of M) number of oracle queries that M makes. The* query complexity of Φ, *denoted $\mathtt{LA}(\Phi)$, is the minimum query complexity of all linear-access algorithms solving Φ.*

In defining the communication setting we refer to the standard setting of communication complexity and specifically to randomized two-party protocols in the model of shared randomness (see [17] for details). We denote by $\mathbb{P}((x, y), r)$ the output of the interaction between the two parties when the first party gets input x, the second party gets input y, and both parties follow protocol $\mathbb{P}$ and have free access to shared randomness r. Without loss of generality, we assume that the output of the interaction is specified in $\mathbb{P}$ (as a function of the exchanged communication and the shared randomness) and need not be explicitly communicated between the two parties.

Definition 2.5 (two-party public-coin communication complexity): *For $n \in \mathbb{N}$, let $\mathcal{R}, \mathcal{S} \subseteq \{0,1\}^{2n}$ and $\Psi = (\mathcal{R}, \mathcal{S})$. A* two-party protocol $\mathbb{P}$ *solves Ψ if it satisfies the following two conditions:*

1. *If $(x, y) \in \mathcal{R} \cap \mathcal{S}$, then $Pr[\mathbb{P}((x, y), r) = 1] \geq \frac{2}{3}$*
2. *If $(x, y) \in \mathcal{R} \setminus \mathcal{S}$, then $Pr[\mathbb{P}((x, y), r) = 0] \geq \frac{2}{3}$*

The communication complexity *of $\mathbb{P}$ is the maximum (over all $(x, y) \in \{0,1\}^{2n}$ and $r \in \{0,1\}^*$) number of bits exchanged between the parties. The* communication complexity of Ψ, *denoted $\mathtt{CC}(\Psi)$, is the minimum communication complexity of all protocols solving Ψ.*

In Section 3 we shall also use the definition of *deterministic communication complexity* of a function $f : \{0,1\}^{2n} \to \Sigma$ (for some finite set Σ). A deterministic protocol $\mathbb{P}$ computes f if for any $(x, y) \in \{0,1\}^{2n}$ it holds that $\mathbb{P}(x, y) = f(x, y)$, where $\mathbb{P}(x, y)$ is the output of the (deterministic) interaction between the two parties when the first party gets input x and the second party gets input y.

In all three models, although we considered the general notion of promise problems, we will frequently consider the special case of decision problems where the promise equals the entire space of possible inputs. When this is the case we will frequently abuse notation, and simply denote the problem by the set of "yes" instances. For example, abusing notations for linear-access problems from Definition 2.4, we may consider a problem which consists of the trivial promise $\mathcal{Q} = \mathbb{F}^n$ and a set of "yes" instances $\mathcal{W} \subseteq \mathbb{F}^n$. In this case, instead of denoting the problem by $\Phi = (\mathcal{Q}, \mathcal{W})$ we will simply denote it by $\mathcal{W}$, and its query complexity by $\mathtt{LA}(\mathcal{W})$.

Additionally, in all three models we defined the two conditions on probabilistic algorithms (or protocols) solving the described problems by fixing the required probabilities to be $\frac{2}{3}$. We also consider the notion of a probabilistic algorithm solving a problem with (a constant) error μ. For example, a linear-access algorithm satisfying the two conditions mentioned in Definition 2.4 with probability $1-\mu$ (instead of $\frac{2}{3}$) solves the linear-access problem Φ with error μ. Note that by standard error reduction, the μ-error query complexity of Φ (i.e., the minimum query complexity of all linear-access algorithms solving Φ with error μ), denoted $\mathtt{LA}_\mu(\Phi)$, satisfies $\mathtt{LA}_\mu(\Phi) = \Theta(\mathtt{LA}(\Phi))$.

3 Reductions Between the Computational Models

In this section we set the stage for the rest of the paper by introducing methods to reduce communication problems to linear-access problems and from them to property testing problems. We also examine several forms of such possible reductions. In particular, we highlight error-correcting codes as appealing candidates for reductions from linear-access algorithms to property testing, and linear functions as appealing candidates for reductions from communication complexity to linear-access algorithms.

This section generalizes ideas of [2, 19, 25], who showed a specific reduction from communication complexity to parity decision trees, and ideas of [2, 6], who considered two reductions from randomized parity decision trees to properties of linear functions and of quadratic functions.

3.1 Reducing linear-access problems to property testing problems

Towards the first definition, consider a linear-access algorithm that transforms its input w into a (possibly longer) input $F(w)$ for a tester and emulates oracle access to $F(w)$ for the tester. The linear-access algorithm has no direct access to its input w, but has the ability to perform linear queries on w (i.e., it can query $H(w)$). Therefore, a necessary condition for the algorithm to be able to provide the tester with oracle access to $F(w)$ is that for any coordinate i of $F(w)$, the value $F(w)_i$ can be computed based on a bounded number of linear queries on w. This gives rise to the following definition.

Definition 3.1 (reductions from linear-access algorithms to property testing). *For $n \in \mathbb{N}$ and a finite field $\mathbb{F}$, let $\mathcal{Q}, \mathcal{W} \subseteq \mathbb{F}^n$ and $\Phi = (\mathcal{Q}, \mathcal{W})$. For $l \in \mathbb{N}$ and a finite set Σ, let $\mathcal{U}, \mathcal{P} \subseteq \Sigma^l$ and $\Pi = (\mathcal{U}, \mathcal{P})$. For $\epsilon > 0$ and $k \in \mathbb{N}$ we call $F : \mathbb{F}^n \to \Sigma^l$ an (ϵ, k)*-reduction of the linear-access problem Φ to the property testing problem Π *if the following two conditions hold:*

1. (F's projections are computable with k linear queries)*: For every $i \in [l]$ there exists a function, $\phi_i : \mathbb{F}^k \to \Sigma$, and k linear functions, $q_1^{(i)}, ..., q_k^{(i)} : \mathbb{F}^n \to \mathbb{F}$, such that for every $w \in \mathbb{F}^n$ it holds that $F(w)_i = \phi_i(q_1^{(i)}(w), ..., q_k^{(i)}(w))$.*
2. (F is an ϵ-reduction of Φ to Π)*: If $w \in \mathcal{Q} \cap \mathcal{W}$, then $F(w) \in \mathcal{U} \cap \mathcal{P}$; whereas if $w \in \mathcal{Q} \setminus \mathcal{W}$, then $F(w) \in \mathcal{U}$ and $F(w)$ is ϵ-far from $\mathcal{P}$ (where distance is measured as in Definition 2.3).*

If the above holds, we say that Π *is* (ϵ, k)*-reducible from* Φ*. A property is reducible from* Φ *if it is* (ϵ, k)*-reducible from it for some* $\epsilon > 0$ *and* $k \in \mathbb{N}$*.*

We now show that if a property is reducible from a linear-access problem Φ, then its query complexity is asymptotically lower bounded by the query complexity of Φ. Similar to [3, 11], we show this by proving that, given oracle access to $H(w)$ for some $w \in \mathbb{F}^n$, a linear-access algorithm M can emulate an execution of a tester for $F(w)$ by making a constant number of queries to its own (i.e., $M's$) oracle per each query of the tester to $F(w)$.

Theorem 3.2 (property testing lower bounds via reductions from linear-access algorithms). *Let* $n, l, \Sigma, \mathbb{F}, \Pi$ *and* Φ *be as in Definition 3.1. If there exist* $\epsilon > 0$ *and* $k \in \mathbb{N}$ *such that* Π *is* (ϵ, k)*-reducible from* Φ *then* $\mathtt{LA}(\Phi) \leq k \cdot \mathtt{PT}(\epsilon, \Pi)$*.*

Proof. Given an (ϵ, k)-reduction F of Φ to Π and an ϵ-tester T for Π that makes at most $\mathtt{PT}(\epsilon, \Pi)$ queries, we show a linear-access algorithm M with query complexity $k \cdot \mathtt{PT}(\epsilon, \Pi)$ that solves Φ.

We construct M in the straightforward manner: Given oracle access to $H(w)$, for $w \in \mathbb{F}^n$, machine M invokes T, feeding T its own (i.e., $M's$) randomness, and emulating oracle access to $F(w)$ for it. Whenever T queries $F(w)_i$, for some $i \in [l]$, machine M queries its own oracle for $q_1^{(i)}(w), ..., q_k^{(i)}(w)$, computes $\phi_i(q_1^{(i)}(w), ..., q_k^{(i)}(w))$, and answers T accordingly. The output of M is simply the output that T returns. Note that the query complexity of M is k times the query complexity of T.

By Condition (1) of Definition 3.1, M can indeed emulate oracle access to $F(w)$ as described. By Condition (2) of Definition 3.1 and the hypothesis that T is an ϵ-tester,

1. If $w \in \mathcal{Q} \cap \mathcal{W}$ then $F(w) \in \mathcal{U} \cap \mathcal{P}$. Hence, $Pr[M^{H(w)}(1^n) = 1] = Pr[T^{F(w)}(1^l) = 1] \geq \frac{2}{3}$.
2. If $w \in \mathcal{Q} \setminus \mathcal{W}$ then $F(w) \in \mathcal{U}$ and $F(w)$ is ϵ-far from $\mathcal{P}$. Hence, $Pr[M^{H(w)}(1^n) = 0] = Pr[T^{F(w)}(1^l) = 0] \geq \frac{2}{3}$.

□

We turn to explore classes of properties that are reducible from linear-access problems, and classes of possible reductions. In particular, we prove that all properties of low-degree rational functions over finite fields and all subcodes of linear codes with constant relative distance are reducible from corresponding linear-access problems

Condition (2) of Definition 3.1 implies that error-correcting codes, and especially linear codes, are appealing candidates for reductions from linear-access algorithms to property testing. Indeed, most of the reductions from linear-access algorithms to property testing that we show in this paper are variations on error-correcting codes. Nevertheless, in Proposition 3.6 we will also demonstrate a useful reduction that is not of this form; that is, a reduction that does not generate distance between pairs of inputs.

Proposition 3.3 (subcodes of linear codes). *For* $n, l \in \mathbb{N}$ *and a finite field* $\mathbb{F}$*, let* $C : \mathbb{F}^n \to \mathbb{F}^l$ *be a linear code with constant relative distance* $\epsilon > 0$*. Then for any* $\mathcal{W} \subseteq \mathbb{F}^n$*, the property* $\mathcal{P} = \{C(w) \in \mathbb{F}^l : w \in \mathcal{W}\}$ *is* $(\epsilon, 1)$*-reducible from* $\mathcal{W}$*.*

Proof. We show that $F(w) = C(w)$ is an $(\epsilon, 1)$-reduction of $\mathcal{W}$ to $\mathcal{P}$.

1. *F's projections are computable with a single linear query.* Let $G_{l\times n}$ be a generator matrix for C. Since $C(w) = Gw$, it means that $C(w)_i = (Gw)_i$, which is computable with a single linear query on w. Following the notations of Definition 3.1, we define $q_1^{(i)}(w) \stackrel{\text{def}}{=\!=} (Gw)_i$ and ϕ_i to be the identity function.

2. *F is an ϵ-reduction of $\mathcal{W}$ to $\mathcal{P}$.* If $w \in \mathcal{W}$ then by definition $F(w) \in \mathcal{P}$. If $w \notin \mathcal{W}$, then by the fact that C has relative distance ϵ it holds that $F(w) = C(w)$ is ϵ-far from $\mathcal{P}$.

□

As a simple corollary, we deduce that all properties of low-degree polynomials are reducible from corresponding linear-access problems.

Corollary 3.4 (properties of low-degree polynomials). *For $m, d \in \mathbb{N}$ and a finite field $\mathbb{F}$, let $n = \binom{m+d}{d}$. Then for any $\mathcal{W} \subseteq \mathbb{F}^n$, the property $\mathcal{P} = \left\{RM_{d,m}(w) \in \mathbb{F}^{|\mathbb{F}|^m} : w \in \mathcal{W}\right\}$ is $(\delta, 1)$-reducible from $\mathcal{W}$, where $\delta = 2^{-d}$ if $\mathbb{F} = \mathbb{F}_2$, and $\delta = 1 - \frac{d}{|\mathbb{F}|}$ otherwise. (Recall that according to Definition 2.2, $RM_{d,m}(w)$ is the evaluation of the degree-d m-variate polynomial associated with w, at all points in $\mathbb{F}^m$.)*

Corollary 3.4 follows directly from Proposition 3.3, since $RM_{d,m}$ is a linear code with relative distance δ. Note that, since δ depends on d, there is a trade-off between the degree bound of the polynomials in the property and the proximity parameter for which we can reduce the property from $\mathcal{W}$. In the special case of $d = 1$, the reduction is the Hadamard code and the property is one of linear functions.

If we are willing to make slightly stricter requirements with respect to the degree bound, then we can generalize Corollary 3.4 and reduce all properties of *rational functions over* $\mathbb{F}$ from linear-access problems. To prove this we use a reduction that is a variant of the Reed-Muller code, but is not a linear code by itself; therefore, this time we cannot simply rely on Proposition 3.3.

Proposition 3.5 (properties of low-degree rational functions). *For $m \in \mathbb{N}$ and a finite field $\mathbb{F}$, let $\mathcal{P} \subseteq (\mathbb{F} \cup \{\infty\})^{|\mathbb{F}|^m}$, where ∞ is a special symbol indicating division by zero. If any $z \in \mathcal{P}$ is the evaluations, at all points $x \in \mathbb{F}^m$, of a rational function $\frac{f}{g}$, where f and g are polynomials of degree at most $d < \frac{|\mathbb{F}|}{4}$, then $\mathcal{P}$ is $(\epsilon, 2)$-reducible from a corresponding linear-access problem $\mathcal{W} \subseteq \mathbb{F}^{2n}$, where $\epsilon = 1 - \frac{4\cdot d}{|\mathbb{F}|}$ and $n = \binom{m+d}{d}$.*

Proof. The linear-access problem that we reduce to the property $\mathcal{P}$ is $\mathcal{W} = \left\{w_1 w_2 \in \mathbb{F}^{2n} : F(w_1 w_2) \in \mathcal{P}\right\}$, where $F : \mathbb{F}^{2n} \to (\mathbb{F} \cup \{\infty\})^{|\mathbb{F}|^m}$ is defined as follows: For any $w_1 w_2 \in \mathbb{F}^{2n}$ and $x \in \mathbb{F}^m$, let $F(w_1 w_2)_x \stackrel{\text{def}}{=\!=} \frac{RM_{d,m}(w_1)_x}{RM_{d,m}(w_2)_x}$. We show that F is an $(\epsilon, 2)$-reduction of $\mathcal{W}$ to $\mathcal{P}$:

1. *F's projections are computable with two linear queries.* Given $x \in \mathbb{F}^m$, for $i = 1, 2$ let $q_x^{(i)}(w_1 w_2) = RM_{d,m}(w_i)$. Let ϕ_x be the division function in $\mathbb{F}$ (which returns ∞ if the denominator is zero). Then, $F(w)_x = \phi_x(q_x^{(1)}(w), q_x^{(2)}(w)) = \frac{RM_{d,m}(w_1)_x}{RM_{d,m}(w_2)_x}$.

2. *F is an ϵ-reduction of $\mathcal{W}$ to $\mathcal{P}$.* By definition, if $w \in \mathcal{W}$ then $F(w) \in \mathcal{P}$. We show that any two distinct rational functions $\frac{f}{g}, \frac{f'}{g'} : \mathbb{F}^m \to \mathbb{F}_{,}$ where all polynomials are of degree at most d, are

ϵ-far from each other. From this it follows that for any $w \notin \mathcal{W}$ it holds that $F(w)$ is ϵ-far from $\mathcal{P}$. To prove the distance claim, we rely on the Schwartz-Zippel lemma to get

$$Pr_{x\in\mathbb{F}^m}\left[\frac{f(x)}{g(x)} = \frac{f'(x)}{g'(x)} \;\middle|\; \text{both } g(x), g'(x) \neq 0\right] = Pr\left[f(x)g'(x) = f'(x)g(x)\right] \leq \frac{2\cdot d}{|\mathbb{F}|}$$

whereas on the other hand $Pr[g(x) = 0 \vee g'(x) = 0] = Pr[g(x)g'(x) = 0] \leq \frac{2\cdot d}{|\mathbb{F}|}$. By union-bound, $\frac{f}{g}$ and $\frac{f'}{g'}$ are at least $(1 - \frac{4\cdot d}{|\mathbb{F}|})$-far from each other.

□

Corollary 3.4 and Proposition 3.5 present two "canonical" reductions of properties of low-degree polynomials or rational functions from corresponding linear-access problems, using variants of the Reed-Muller code. We finish this section by showing that useful reductions (to natural properties) need not be error-correcting codes at all. Furthermore, they need not even generate distance between all pairs of inputs.

To show this we adapt a reduction from [3, Thm 1.8]; we reduce a linear-access promise problem in which "yes" instances and "no" instances are guaranteed to be sufficiently far, to a corresponding property testing problem, by a reduction that does not generate distance between pairs of inputs. Loosely speaking, the property in this case consists of all Boolean functions that are computable by decision trees of size that is not too large.

Proposition 3.6 (reductions that do not generate distance). *For a sufficiently large integer n, let $\mathcal{F}$ be the set of functions $f : \{0,1\}^n \to \{0,1\}$ that are computable by decision trees of size at most $\frac{15}{16}\cdot 2^n$, and $\mathcal{P} \subseteq \{0,1\}^{2^n}$ be the set of truth tables of $\mathcal{F}$. Let $\mathcal{W} \subseteq \{0,1\}^{2^{n-1}}$ be the set of vectors with Hamming weight exactly $\frac{2^{n-1}}{8}$, and $\mathcal{Q} = \mathcal{W} \cup \{0^{2^{n-1}}\}$. Then $\mathcal{P}$ is $(\frac{1}{32}, 1)$-reducible from $\Phi = (\mathcal{Q}, \mathcal{W})$.*

Note that in this case each $w \in \mathcal{W}$ is $\frac{2^{n-1}}{8}$-far from the single (zero) vector in $\mathcal{Q}\setminus\mathcal{W}$. The reduction we will present does not generate additional distance, but merely preserves the existing absolute distance.

Proof. We define a reduction $F : \{0,1\}^{2^{n-1}} \to \{0,1\}^{2^n}$ as follows. For $x \in \{0,1\}^n$, let $Par(x) = \oplus_{i=1}^n x_i$, and we fix some bijection φ between $\{x \in \{0,1\}^n : Par(x) = 0\}$ and the set $[2^{n-1}]$. Then for every $w \in \{0,1\}^{2^{n-1}}$ and every $x \in \{0,1\}^n$, we let

$$F(w)_x = \begin{cases} 1 & Par(x) = 1 \\ w_{\varphi(x)} & Par(x) = 0 \end{cases}$$

One may think of F as mapping vectors of the form $(w_1, ..., w_{2^{n-1}})^T$ to vectors of the form $(1, ..., 1, w_1, ..., w_{2^{n-1}})^T$, where the first half of the coordinates in the 2^n-bit long vector correspond to the positions $x \in \{0,1\}^n$ with $Par(x) = 1$. We show that F reduces Φ to $\mathcal{P}$:

1. *F's projections are computable with a single linear query.* If $Par(x) = 1$, then $\phi_x \equiv 1$. Otherwise, ϕ_x is the identity function, and the single query q_1^x satisfies $q_1^x(w) = \langle e_{\varphi(x)}, w\rangle$ (where $e_{\varphi(x)}$ is the standard unit vector corresponding to coordinate $\varphi(x)$).

2. *F $\frac{1}{32}$-reduces Φ to $\mathcal{P}$.* If $w \in \mathcal{W}$, then $F(w)$ is computable by a decision tree of size at most $\frac{15}{16}\cdot 2^n$; a proof of this fact, adapted from [3], appears below. If $w \in \mathcal{Q}\setminus\mathcal{W}$ then w is the zero vector; in this case, $F(w)$ is the parity function, which is $\frac{1}{32}$-far from any function computable by a decision tree of size $\frac{15}{16}\cdot 2^n$ (the proof of the latter fact appears in [3, Lemma 5.3]).

Lemma. *If $w \in \mathcal{W}$, then $F(w)$, viewed as a function $f_w : \{0,1\}^n \to \{0,1\}$, is computable by a decision tree of size $\frac{15}{16} \cdot 2^n$.*

Proof. Consider the complete binary decision tree of depth n, in which all nodes at level $i \in [n]$ are labeled with the element i. This tree has 2^n leaves, each corresponding to some $x \in \{0,1\}^n$, and we label each leaf by $f_w(x)$.

The key observation is that if $x, x' \in \{0,1\}^n$ correspond to a pair of sibling leaves (where x corresponds to the left one), then $Par(x) = 0$ and $Par(x') = 1$, and in particular $f_w(x') = 1$. Therefore every leaf corresponding to some $x \in \{x : Par(x) = 0\}$ has a sibling x' such that $f_w(x') = 1$.

Since $w \in \mathcal{W}$, there are exactly $\frac{2^{n-1}}{8}$ leaves corresponding to $x \in \{x : Par(x) = 0\}$ that are labeled with 1; in all these cases we can merge the leaf with its sibling, reducing the size of the tree by one leaf. The resulting tree computes f_w and is of size exactly $\frac{15}{16} \cdot 2^n$.

Hence, F is a $(\frac{1}{32}, 1)$-reduction of Φ to $\mathcal{P}$. □

The query complexity of solving Φ with *one-sided error* (i.e., by a linear-access algorithm that accepts each "yes" instance with probability 1) is $\Omega(n)$. The proof, which we do not present fully here, is a straight forward adaptation of a reduction that appeared in [3, Thm 1.8]; it consists of reducing Φ from the communication problem of Gap-Equality. It follows that the query complexity of solving $\mathcal{P}$ with one-sided error is also $\Omega(n)$.

3.2 Reducing communication problems to linear-access problems

In this section we show a method to reduce communication complexity problems to linear-access problems, again in the spirit of [3, 11]. Combined with Section 3.1, this will allow us to transitively reduce communication problems to property testing problems, via linear-access problems.

Definition 3.7 (reductions from communication complexity to linear-access algorithms). *For $m \in \mathbb{N}$, let $\mathcal{R}, \mathcal{S} \subseteq \{0,1\}^{2m}$ and $\Psi = (\mathcal{R}, \mathcal{S})$. For $n \in \mathbb{N}$ and a finite field $\mathbb{F}$, let $\mathcal{Q}, \mathcal{W} \subseteq \mathbb{F}^n$ and $\Phi = (\mathcal{Q}, \mathcal{W})$. For $B \in \mathbb{N}$, we say that a function $G : \{0,1\}^{2m} \to \mathbb{F}^n$ is a B-*reduction of the communication complexity problem Ψ to the linear-access problem Φ *if the following two conditions hold:*

1. (linear queries on G are computable using B bits of communication)*: For any linear function $q : \mathbb{F}^n \to \mathbb{F}$, the deterministic communication complexity of the function $q \circ G : \{0,1\}^{2m} \to \mathbb{F}$ is at most B.*
2. (G is a reduction of Ψ to Φ)*: If $(x, y) \in \mathcal{R} \cap \mathcal{S}$ then $G(w) \in \mathcal{Q} \cap \mathcal{W}$; whereas if $w \in \mathcal{R} \setminus \mathcal{S}$ then $G(w) \in \mathcal{Q} \setminus \mathcal{W}$.*

If the above holds, we say that Φ is B-reducible from Ψ. A problem Φ is reducible from a problem Ψ if it is B-reducible from it for some $B \in \mathbb{N}$.

We now show (analogously to Theorem 3.2) that if a linear-access problem Φ is reducible from a communication problem Ψ, then its query complexity is asymptotically lower bounded by the communication complexity of Ψ.

Theorem 3.8 (lower bounds on linear-access algorithms via reductions from communication complexity). *Let $m, n, \mathbb{F}, \Psi$ and Φ be as in Definition 3.7. If there exists $B \in \mathbb{N}$ such that Φ is B-reducible from Ψ, then $\mathtt{CC}(\Psi) \leq B \cdot \mathtt{LA}(\Phi)$.*

Proof. Let M be a linear-access algorithm solving Φ with query complexity m. We construct a two-party protocol $\mathbb{P}$ solving Ψ with communication complexity $B \cdot m$.

When receiving an input pair (x, y), both parties locally invoke identical copies of M, feeding it the shared randomness and emulating oracle access to $H(G(x, y))$ for it. That is, whenever M makes a linear query q (to $G(x, y)$), both parties compute the result by communicating B bits (via the protocol guaranteed for the function $q \circ G$) and then answer the query accordingly. The protocol's output is simply the output of M.

For any $(x, y) \in \mathcal{R}$ it holds that $\mathbb{P}$ outputs the correct answer if and only if M outputs the correct answer; and the communication complexity of $\mathbb{P}$ is at most $B \cdot m$. □

Just as error-correcting codes are appealing candidates for reductions from the linear-access model to the property testing model, linear functions are appealing candidates for reductions from communication complexity to linear-access algorithms. Indeed, if each coordinate of $G(x, y)$ is a linear combination of x and of y (viewed as vectors over $\mathbb{F}$), then any linear query $q : \mathbb{F}^n \to \mathbb{F}$ on $G(x, y)$ is computable by communicating $2 \cdot \lceil \log_2 |\mathbb{F}| \rceil$ bits.

All the reductions from communication complexity to linear-access algorithms that we will present in this paper are linear functions. Among them are the addition function over $\mathbb{F}^n$, that is $G(x, y) = x+y$, and the concatenation function, $G(x, y) = x \circ y$. We mention, however, that our formulation in Definition 3.7 is not restricted to linear reductions.

3.3 Detour: A communication model limited to linear functions

Recall (from the beginning of Section 1.2) that linear-access algorithms over $\mathbb{F}_2$ are equivalent, up to a factor of 2 in the number of queries, to communication protocols in which both parties only compute linear functions of their respective inputs. In this subsection we shortly discuss the latter model, which we call the *linear communication model*. In particular, we prove the statement that this model is equivalent to linear-access algorithms up to a factor of 2 in the number of queries, and show a strong separation of this model from the standard communication model. After concluding this subsection, we will not refer to the linear communication model again in this paper.

We define the *linear communication model* as a special case of the standard communication model (i.e., of Definition 2.5) that considers only *communication protocols that are limited to linear functions*: These are communication protocols in which in every round of communication, the communicating party chooses a linear function f according to the shared randomness and the communication history, computes the value of f on its input, and communicates the result to the other party. The communication complexity of a problem Ψ in the linear communication model, denoted $\mathtt{CC}^{\mathtt{lin}}(\Psi)$, is the minimum communication complexity of all probabilistic communication protocols that are limited to linear functions and solve Ψ. We start by formally proving the statement about equivalence of this model to the linear-access model in the case of $\mathbb{F}_2$.

Proposition 3.9 (equivalence of linear-access algorithms and the linear communication model): *For $n \in \mathbb{N}$, any sets $\mathcal{R}, \mathcal{S} \subseteq \{0,1\}^{2n}$ and a problem $\Psi = (\mathcal{R}, \mathcal{S})$, the communication complexity of Ψ in the linear communication model is identical, up to a factor of 2, to its query complexity as a linear-access problem. Specifically, it holds that $\mathtt{LA}(\Psi) \leq \mathtt{CC}^{\mathtt{lin}}(\Psi) \leq 2 \cdot \mathtt{LA}(\Psi)$.*

Proof. Note that a linear-access algorithm M that gets input $w = (x, y) \in \{0,1\}^{2n}$ can emulate the execution of a communication protocol that is limited to linear functions on (x, y) using the same number of queries. Specifically, whenever the protocol requires that one of the parties compute a

linear function of its input (i.e., of either x or y), machine M can compute the same function by making a single query to its oracle.

On the other hand, a communication protocol that is limited to linear functions and gets an input pair (x,y) can emulate the execution of any linear-access algorithm M on $w=(x,y)$, using twice the number of queries that M makes. This is since for every linear query $q=(q_1,...,q_{2n})$ that M makes, the communication protocol can compute the value $q(x,y)$ by communicating two bits: The first party computes $\oplus_{i=1}^{n} q_i x_i$ and communicates the result to the second party, the second party computes $\oplus_{i=n+1}^{2n} q_i y_i$ and sends it to the first party, and both parties compute $\oplus_{i=1}^{2n} q_i w_i = q(w)$. □

We now present a property of the linear communication model that does not hold in the standard communication model. Recall that in the standard model, the complexity of a problem $\mathcal{S} \subseteq \{0,1\}^{2n}$ might be significantly different than the complexity of the problem of deciding $\mathcal{S}$ when the bits of the input are distributed to the two parties in a non-standard way; that is, when the first party gets some predetermined subset of n bits of $w \in \{0,1\}^{2n}$ (rather than the n-bit prefix of w) and the second party gets the remaining n bits (rather than the n-bit suffix of w). For example, consider the inner-product communication problem; that is, $\mathcal{IP} = \{(x,y) \in \{0,1\}^{2n} : \langle x,y \rangle = 1\}$. The communication complexity of $\mathcal{IP}$ is $\Omega(n)$ (see, e.g., [8]); however, for $n=2k$, if the first party gets $x_1, y_1, ..., x_k, y_k$ and the second party gets $x_{k+1}, y_{k+1}, ..., x_n, y_n$ then the parties can decide whether $(x,y) \in \mathcal{IP}$ by communicating two bits (since each party can compute the parity of $x_i y_i$'s that are part of its input).

In contrast, in the linear communication model it does not matter which of the two parties gets which bits of the input. Note that partitioning every input $w \in \{0,1\}^{2n}$ to two n-bit subsets, using a predetermined partition, and giving the corresponding bits as inputs to each of the parties, can be achieved by permuting the bits of w, using a corresponding permutation, and giving the n-bit prefix of the permuted string to the first party and the n-bit suffix of the permuted string to the second party. We prove that this action (i.e., permuting the bits of every input) can only change the complexity of a given problem in the linear communication model by a constant factor. Denote the symmetric group over $[2n]$ by S_{2n}, and, for $\sigma \in S_{2n}$ and $w = w_1 w_2 ... w_{2n} \in \{0,1\}^{2n}$, let $\sigma(w) = w_{\sigma(1)} w_{\sigma(2)} ... w_{\sigma(2n)}$.

Proposition 3.10 (permutations on the input in the linear communication model): *For $n \in \mathbb{N}$, let $\mathcal{S} \subseteq \{0,1\}^{2n}$. Then, up to a constant factor of 2, for every permutation $\sigma \in S_{2n}$, the communication complexity of $\mathcal{S}$ in the linear communication model is identical to the communication complexity of $\mathcal{S}_\sigma = \left\{\sigma(x,y) \in \{0,1\}^{2n} : (x,y) \in \mathcal{S}\right\}$ in the linear communication model.*

Proof. Let $\sigma \in S_n$. Note that the query complexity of $\mathcal{S}$ and $\mathcal{S}_\sigma$ as linear-access problems is identical, because permuting the input bits does not affect the ability of a linear-access algorithm to perform any linear query on these input bits. Relying on Proposition 3.9, the communication complexity of $\mathcal{S}$ and $\mathcal{S}_\sigma$ in the linear communication model can only differ by a constant factor of 2. □

We now prove that there exist communication problems that have $O(1)$ communication complexity in the standard model, yet $\Omega(n)$ communication complexity in the linear communication model.

Proposition 3.11 (separation of the linear communication model from the standard communication model): *For $n \in \mathbb{N}$ and any communication problem $\mathcal{S} \subseteq \{0,1\}^{2n}$ with communication complexity $\mathtt{CC}(\mathcal{S})$ in the standard model, there exists a corresponding communication problem $\mathcal{S}'$ such that the communication complexity of $\mathcal{S}'$ in the standard communication model is $O(1)$ but the communication complexity of $\mathcal{S}'$ in the linear communication model is lower bounded by $\mathtt{CC}(\mathcal{S})/2$.*

Proof. Let $\mathcal{S}'' = \left\{((x,x'),(y,y')) \in \{0,1\}^{4n} : (x,y) \in \mathcal{S}\right\}$. Note that the communication complexity of $\mathcal{S}''$ in the standard model is $\mathtt{CC}(\mathcal{S})$, since given inputs $((x,x'),(y,y')) \in \{0,1\}^{4n}$ the two parties can

consider the truncated inputs (x, y) and execute any protocol for solving $\mathcal{S}$; and on the other hand, given inputs $(x, y) \in \{0,1\}^{2n}$ the two parties can pad them to $((x, 0^n), (y, 0^n)) \in \{0,1\}^{4n}$ and execute any protocol for solving $\mathcal{S}''$. It follows that the complexity of $\mathcal{S}''$ in the linear communication model is lower bounded by $\mathtt{CC}(\mathcal{S})$, since communication protocols that only compute linear functions are a special case of standard communication protocols.

Let $\mathcal{S}' = \{((x, y), (x', y')) \in \{0,1\}^{4n} : (x, y) \in \mathcal{S}\}$. Clearly, the communication complexity of $\mathcal{S}'$ in the standard model is $O(1)$, since the first party can compute the result by itself. However, according to Proposition 3.10, the communication complexity of $\mathcal{S}'$ in the linear communication model is identical to the communication complexity of $\mathcal{S}''$ in the linear communication model, up to a factor of 2, and the latter is lower bounded by $\mathtt{CC}(\mathcal{S})$. □

We finish this section by clarifying the implications of Proposition 3.11 on reductions from communication complexity to linear-access algorithms. Proposition 3.11 contrasts the complexity of a set $\mathcal{S}'$ in the standard communication model and the complexity of the same set $\mathcal{S}'$ in the linear communication model, which according to Proposition 3.9 is equivalent to the linear-access model over $\mathbb{F}_2$. This perspective implicitly considers a reduction from communication complexity to linear-access algorithms that is the concatenation function, $G(x, y) = x \circ y$. Therefore, from a perspective of linear-access algorithms, Proposition 3.11 can be phrased as follows: There exist sets $\mathcal{S}' \subseteq \{0,1\}^n \times \{0,1\}^n$ that have communication complexity $O(1)$ and that can be reduced to corresponding linear-access problems with query complexity $\Omega(n)$ via concatenation.

We stress, however, that this statement *does not* rule out the possibility that there exists another communication problem with higher complexity that can be reduced to the linear-access problem $\mathcal{S}'$ via a reduction that is not the concatenation reduction. In particular, in the specific construction presented in the proof of Proposition 3.11, the communication problem $\mathcal{S}$ can be reduced to the linear-access problem $\mathcal{S}'$ (i.e., to the set $\mathcal{S}'$ when treated as a linear-access problem) via the linear reduction $G(x, y) = (x, y, 0^n, 0^n)$. Indeed it is an interesting question whether there exists a set $\mathcal{S}'$ such that its query complexity as a linear-access problem is higher than the communication complexity of any communication problem reducible to it, and we pose it as an open question in Section 6.

4 Deconstructions of Reductions from Communication Complexity to Property Testing

In this section we offer a new interpretation for known results, by deconstructing known reductions from communication complexity to property testing, using linear-access algorithms as an intermediary model. Bhrushundi, Chakraborty, and Kulkarni [2] demonstrated one such deconstruction, and we re-examine it (and offer a new interpretation of it) in Example 4.3.

4.1 The deconstruction and a generic observation

We first note that the reductions underlying Theorems 3.8 and 3.2 can be composed to yield:

Theorem 4.1 *Let Ψ be a communication problem (as in Definition 2.5) and Π be a property (as in Definition 2.3). Suppose that for some $B, k \in \mathbb{N}$ and $\epsilon > 0$ there exist two reductions as follows:*

1. *A function G (as in Theorem 3.8) B-reducing Ψ to a linear-access problem Φ.*
2. *A function F (as in Theorem 3.2) (ϵ, k)-reducing Φ to Π.*

Then, the query complexity of ϵ-testing Π is asymptotically lower bounded by $\frac{1}{k \cdot B}$ times the communication complexity of Ψ. That is, $\mathtt{CC}(\Psi) \leq B \cdot k \cdot \mathtt{PT}(\epsilon, \Pi)$. In this case we say that $F \circ G$ is a reduction from communication complexity to property testing.

Interestingly, deconstructing some well-known reductions in this manner reveals G and F that have fundamentally different functionalities.

Observation 4.2 *(informal). Some known reductions from communication complexity to property testing can be presented as the composition of two reductions, as in Theorem 4.1, each having a distinctly characterized functionality:*

$$\Psi \xrightarrow{\quad G \quad} \Phi \xrightarrow{\quad F \quad} \Pi$$

1. "Combining step": *The first reduction $\Psi \xrightarrow{G} \Phi$ combines two inputs (x, y), given to two parties in the communication setting, into a single input w for a linear-access algorithm. This combination changes the nature of the computational problem: While (by definition) problems in two-party communication complexity have a structure corresponding with two separate parties, linear-access problems do not have such an apparent structure.*

2. "Distance creation step": *The second reduction $\Phi \xrightarrow{F} \Pi$ takes a problem that consists of "yes" instances and "no" instances, and* creates distance *between these instance sets, by mapping them to a (possibly larger) metric space. Following this perspective, it is not surprising that many appealing examples of such reductions involve error-correcting codes.*

We do not claim that *all* reductions from communication complexity to property testing may be deconstructed in this manner. Furthermore, even when such a deconstruction is possible, we do not claim that the two steps can necessarily be characterized as a "combining step" and a "distance creation step" (e.g., Proposition 3.6 demonstrates a useful reduction, adapted from [3], in which the second step does not generate distance). Yet in several well-known cases such a deconstruction is possible, and the two steps correspond to Observation 4.2.

4.2 Deconstructions of known results

The first example we present considers the property of *k-linear functions*, which consists of all n-variate linear Boolean functions over $\mathbb{F}_2$ that are k-sparse, meaning that exactly k of their coefficients are non-zero. The first proof for an $\Omega(k)$ lower bound[3] on the query complexity of this property was provided by Blais, Brody, and Matulef [3], who proved it using a reduction from the communication complexity of unique-$\frac{k}{2}$-set-disjointness. Later on, Blais and Kane [4] proved the lower bound by analyzing the problem directly in property testing. In Section 5.4 we present an alternative proof for this lower bound, relying on techniques presented in Section 5 (Proposition 5.1 and Technique 5.3). We now deconstruct the first proof (i.e., the reduction from communication complexity).

Example 4.3 (k-linearity). *Reducing the problem of testing k-sparse linear functions from the communication problem of unique-$\frac{k}{2}$-set-disjointness, via the linear-access problem of identifying vectors with Hamming weight k.*

[3] The exact complexity of the problem is $\Omega(\min\{k, n-k\})$; see Section 5.4 for further details

The communication problem called *unique-$\frac{k}{2}$-set-disjointness* is defined on $\{0,1\}^{2n}$ by setting the set of "yes" instances to be

$$\mathcal{S} = \left\{ (x,y) \in \{0,1\}^{2n} : \|x\|_1 = \|y\|_1 = \frac{k}{2} \quad \text{and} \quad \forall i \in [n], x_i \wedge y_i = 0 \right\}$$

That is, $\mathcal{S}$ consists of pairs that when treated as indicator strings represent disjoint sets of cardinality $\frac{k}{2}$. The promise $\mathcal{R}$ consists of all pairs of strings $(x,y) \in \{0,1\}^{2n}$ such that $\|x\|_1 = \|y\|_1 = \frac{k}{2}$ having at most one coordinate $i \in [n]$ such that $x_i \wedge y_i = 1$. We denote this well-known promise problem by $\frac{k}{2}$-$\mathcal{UDISJ} = (\mathcal{R}, \mathcal{S})$, and note that its communication complexity is $\Omega(k)$ (see, e.g., [20, 3, 15]).

1. "Combining step": We reduce $\frac{k}{2}$-$\mathcal{UDISJ}$ to a natural linear-access problem that requires identifying strings with Hamming weight exactly k; that is, k-$\mathcal{WT} = \{w \in \{0,1\}^n : \|w\|_1 = k\}$ (with the trivial promise $\{0,1\}^n$). The reduction is simply $G(x,y) = x \oplus y$. Indeed, if (x,y) represent disjoint $\frac{k}{2}$-sets then $\|x \oplus y\|_1 = k$, and if they represent $\frac{k}{2}$-sets that intersect on a single coordinate then $\|x \oplus y\|_1 = k-2$. Furthermore, any linear query on $G(x,y)$ is computable by 2 communication bits (since G is linear over $\{0,1\}$).

2. "Distance creation step": We reduce the problem of k-$\mathcal{WT}$ to the property of k-linear functions simply by using the Hadamard code. Indeed, treating vectors as coefficients of linear functions, if the vector has Hamming weight k then the function is k-linear, and otherwise it is a $(k-2)$-linear function, which is $\frac{1}{2}$-far from being k-linear.

Hence, in this case we have —

$$(x,y) \xrightarrow{x \oplus y} \|w\|_1 \in \{k, k-2\} \xrightarrow{H(w)} H(w) \text{ is } k\text{-linear or } (k-2)\text{-linear}$$

In this example, the first reduction takes a problem with a clear two-party structure — deciding whether sets held by two parties are disjoint — and reduces it to a problem that has no such apparent structure: Deciding by linear queries if the Hamming weight of a vector is k. The second reduction, on the other hand, is merely an error-correcting code creating distance between "yes" instances (vectors of weight k) and "no" instances (vectors of weight $k-2$). These two functionalities reflect two separate functional components that exist in the "composed" reduction $F \circ G$.

Interestingly, both "direct" proofs of a lower bound on testing k-sparse linear functions, which do not involve a reduction from communication complexity (the one provided by [4] and the one that appears in Section 5.4), can easily be adapted to serve as a direct proof for a lower bound on the linear-access problem k-$\mathcal{WT}$. A more general explanation for this phenomena is provided after Proposition 5.1. □

The next example is a deconstruction of a family of reductions that is similar to two families of reductions presented in [11, Thms 4.1 and 4.2]. Loosely speaking, for a linear error-correcting code C with constant relative distance $\epsilon > 0$ and a "hard" communication problem $\Psi = (\mathcal{R}, \mathcal{S})$, we prove that ϵ-testing the property $\{C(x \circ y) : (x,y) \in \mathcal{R} \cap \mathcal{S}\}$ is also "hard". Since error-correcting codes are appealing candidates for reductions for the "distance creation step", this property is an appealing one to think of in the current context.

Example 4.4 *For $n \in \mathbb{N}$, let $\Psi = (\mathcal{R}, \mathcal{S})$ be a communication problem over $\{0,1\}^{2n}$ with linear communication complexity; that is, $\mathtt{CC}(\Psi) = \Omega(n)$. For $l \in \mathbb{N}$, let $C : \{0,1\}^{2n} \to \{0,1\}^l$ be a linear code with constant relative distance $\epsilon > 0$. Then the query complexity of ϵ-testing the property $\mathcal{P} = \left\{C(x \circ y) \in \{0,1\}^l : (x,y) \in \mathcal{S} \cap \mathcal{R}\right\}$ is linear, that is $\mathtt{PT}(\epsilon, \mathcal{P}) = \Omega(n)$.*

1. "Combining step": We reduce Ψ to a corresponding linear-access problem $\mathcal{W}$, defined by $\mathcal{W} = \left\{w = x \circ y \in \{0,1\}^{2n} : (x,y) \in \mathcal{S} \cap \mathcal{R}\right\}$. The reduction G is the concatenation function $G(x,y) = x \circ y$, and since it is linear in x and y, any linear query on $G(x,y)$ (over $\mathbb{F}_2$) is computable by communicating 2 bits.

2. "Distance creation step": We reduce $\mathcal{W}$ to $\mathcal{P}$ with the code C. According to Proposition 3.3 this is an $(\epsilon, 1)$-reduction.

This deconstruction demonstrates that the "combining step" does not have to explicitly add x and y as vectors over some field in order to eliminate the original two-party structure of the problem. Indeed, the problem of deciding $\mathcal{W}$ using arbitrary linear queries does not have an apparent structure corresponding with two separate parties. We will consider Example 4.4 again in Section 5.3, where we prove a lower bound on a subfamily of properties from this family by reducing them directly from the intermediary linear-access model.

The last example we present relies on the fact that we allowed linear-access algorithms to operate over an arbitrary finite field $\mathbb{F}$ (rather than only over $\mathbb{F}_2$, as in the case of randomized parity decision trees). Specifically, we show a reduction from a communication problem in $\{0,1\}^n$ to a linear-access problem in $\mathbb{F}_3^n$, and then to a property of functions from $\mathbb{F}_3^n$ to $\mathbb{F}_3$.

We define the property of *linear functions with* $\{0,1\}$*-coefficients over* $\mathbb{F}_3$ as the set of functions from $\mathbb{F}_3^n$ to $\mathbb{F}_3$ that are linear and whose coefficients are either 0 or 1. Goldreich proved [10] a lower bound of $\Omega(\sqrt{n})$ on testing this property working directly in the property testing model, and Brais, Brody, and Matulef [3] proved an $\Omega(n)$ lower bound by a reduction from the communication problem of set-disjointness. We now deconstruct the latter reduction.

Example 4.5 (linear functions over $\mathbb{F}_3$ with coefficients in $\{0,1\}$). *Reducing the testing problem of linear functions over* $\mathbb{F}_3$ *with coefficients in* $\{0,1\}$ *from the communication problem of set-disjointness, via the linear-access problem of identifying vectors in* $\mathbb{F}_3^n$ *with coordinates in* $\{0,1\}$.

The communication problem of *set-disjointness* (a general version of unique-$\frac{k}{2}$-set-disjointness, presented earlier) is defined on $\{0,1\}^{2n}$ by considering the trivial promise and setting the set of "yes" instances to be $\mathcal{DISJ} = \{(x,y) : \forall i \in [n], x_i \wedge y_i = 0\}$. That is, $\mathcal{DISJ}$ consists of pairs of n-bit strings that when treated as indicators of subsets of $[n]$ represent disjoint sets. The communication complexity of this well-known problem is $\Omega(n)$ (see, e.g., [20]).

1. "Combining step": We reduce $\mathcal{DISJ}$ to a linear-access problem that requires identifying vectors in $\mathbb{F}_3^n$ whose coefficients are 0 or 1. That is, $\{0,1\}$-$\mathcal{W} = \{w \in \mathbb{F}_3^n : \forall i \in [n], w_i \in \{0,1\}\}$ and the promise is the trivial one. The reduction $G : \{0,1\}^{2n} \to \mathbb{F}_3^n$ is $G(x,y) = x + y$ over $\mathbb{F}_3$; that is, x and y are treated as vectors in $\mathbb{F}_3^n$ and G is the component-wise addition of these vectors. Indeed, any linear query on $G(x,y)$ over $\mathbb{F}_3$ can be computed by communicating four (i.e., $2 \cdot \lceil \log_2(3) \rceil$) bits, and it is easy to see that G reduces $\mathcal{DISJ}$ to $\{0,1\}$-$\mathcal{W}$.

2. "Distance creation step": We reduce the problem of $\{0,1\}$-$\mathcal{W}$ to the property of linear functions with coefficients in $\{0,1\}$ simply by using the Hadamard code (over $\mathbb{F}_3$).

In this example too, the first reduction takes a problem with a clear two-party structure (the set-disjointness communication problem) and reduces it to a problem that has no such apparent structure (the $\{0,1\}$-$\mathcal{W}$ linear-access problem). The second reduction is a linear error-correcting code. □

5 Proving Lower Bounds in Property Testing by Reductions from Linear-access Algorithms

In this section we study the potential of proving lower bounds on property testing problems by reducing them directly from linear-access problems. We start by showing a limitation of this approach: Specifically, in Section 5.1, we show that reductions that correspond to the Hadamard code are unlikely to be helpful in proving lower bounds in property testing, since in this case any linear-access algorithm is essentially a tester for the target property testing problem (see Proposition 5.1).

In contrast, we show that in other cases reducing property testing problems from linear-access algorithms is beneficial. In Section 5.2 we show a simple technique for proving lower bounds on linear-access algorithms, relying on the analysis of affine subspaces in $\mathbb{F}^n$ (see Proposition 5.2 and Technique 5.3). We demonstrate this technique in the following two subsections by proving lower bounds on natural linear-access problems, and deriving corresponding lower bounds in property testing.

In particular, in Section 5.3 we show a lower bound of $\Omega(n)$ queries for testing the property $\{C(x \circ y) : x, y \in \{0,1\}^n \wedge \langle x, y \rangle = 1\}$, where C is an arbitrary linear code with constant relative distance. Furthermore, in Section 5.4 we provide an alternative proof for the known lower bound of $\Omega(\min\{k, n-k\})$ queries for testing k-sparse linear Boolean functions over $\{0,1\}^n$, which was presented in Example 4.3. We then extend this result to a lower bound of $\Omega(\min\{s, \binom{n}{d} - s\})$ queries for testing s-sparse polynomials of degree d over $\{0,1\}^n$, for any $d \in \mathbb{N}$.

5.1 Reductions of the form of the Hadamard code

Bhrushundi, Chakraborty, and Kulkarni noted [2] that a randomized parity decision tree of size m solving $\mathcal{W} \subseteq \{0,1\}^n$ exists if and only if a $\frac{1}{2}$-tester with query complexity m exists for the property $\mathcal{P} = \{H(w) : w \in \mathcal{W}\}$, under the promise that the input for the tester is the evaluations of a linear function. Buhrman *et al.* [6] also pointed out this phenomena for the specific property of k-sparse linear functions. We present a proof of the analogous result in the general setting of linear-access algorithms and comment on its implications towards proving lower bounds for property testing.

Intuitively, this phenomena happens since performing a linear query on the coefficients of a linear function (which is what a linear-access algorithm does) is equivalent to querying the linear function at a corresponding point (which is what a tester does). In both cases the answers to these queries are encoded in the Hadamard code of the coefficients of the linear function.

Proposition 5.1 *For $n \in \mathbb{N}$ and a finite field $\mathbb{F}$, let $\mathcal{W} \subseteq \mathbb{F}^n$. Then there exists a linear-access algorithm M that solves $\mathcal{W}$ if and only if there exists an ϵ-tester T with the same query complexity for the promise problem $\Pi = (\mathcal{U}, \mathcal{P})$, where $\mathcal{U} = \{H(w) \in \mathbb{F}^{|\mathbb{F}|^n} : w \in \mathbb{F}^n\}$ and $\mathcal{P} = \{H(w) \in \mathbb{F}^{|\mathbb{F}|^n} : w \in \mathcal{W}\}$ and $\epsilon \leq \frac{|\mathbb{F}|-1}{|\mathbb{F}|}$. Furthermore, given access to the same oracle, of the form $H(w)$ for some $w \in \mathbb{F}^n$, machines M and T issue the same queries and output the same decision*

Proof. Note that an ϵ-tester for Π and a linear-access algorithm for $\mathcal{W}$ are both oracle machines that get access to an oracle of the form $H(w)$, for some $w \in \mathbb{F}^n$, and need to decide with probability at least $\frac{2}{3}$ whether $w \in \mathcal{W}$ or $w \notin \mathcal{W}$. For a linear-access algorithm this is true by definition, whereas for an ϵ-tester this is true since any $z \in \mathcal{U}$ is of the form $z = H(w)$, for some $w \in \mathbb{F}^n$, whereas $H(w) \in \mathcal{P}$ if and only if $w \in \mathcal{W}$, and the Hadamard code guarantees that any two codewords are ϵ-far from each other.

The only difference between an ϵ-tester for Π and a linear-access algorithm for $\mathcal{W}$ is that for $w \in \mathbb{F}^n$, the ϵ-tester gets $1^{|\mathbb{F}|^n}$ as input whereas the linear-access algorithm gets 1^n as input. It follows that any

oracle machine that solves one problem can be modified to an oracle machine that solves the other problem, by changing its dependence on its explicit input (1^n or $1^{|\mathbb{F}|^n}$), and this modification satisfies the "furthermore" clause in the proposition. □

Proposition 5.1 suggests that reductions of the form of the Hadamard code are unlikely to be helpful in proving lower bounds on the query complexity of the target property Π: Any analysis of linear-access algorithms solving $\mathcal{W}$ can serve as an analysis for ϵ-testers solving Π, and vice versa. Furthermore, since hardness of the promise problem Π implies hardness of the property $\mathcal{P}$ (without the promise), the reduction from linear-access algorithms seems redundant also towards proving lower bounds on the query complexity of ϵ-testing $\mathcal{P}$.[4]

A good demonstration of this phenomena is provided by two existing proofs for a lower bound on testing k-sparse linear functions, which analyze the problem without reducing it from a communication problem. Both the existing proof by Blais and Kane [4] and the proof we provide in this paper (Theorems 5.6 and 5.8) can be easily adapted to serve as lower bounds both for testers for k-sparse linear functions and for linear-access algorithms solving the corresponding linear-access problem k-$\mathcal{WT}$, which can be reduced to the property via the Hadamard code (see Example 4.3 for definitions of these problems).

As a last comment on this subject, we note that Proposition 5.1 can be slightly generalized to account for artificial reductions that are similar to the Hadamard code. For example, a similar proposition is true for reductions in which redundant information is added to the code (e.g., $F(w) = H(w) \circ H(w)$) or in which a permutation on $\mathbb{F}$ is applied coordinate-wise (e.g., $F(w)_x = H(w)_x + 1$): In these cases an ϵ-tester for the property $\{F(w) : w \in \mathcal{W}\}$ exists if and only if a linear-access algorithm for $\mathcal{W}$ with the same query complexity exists (the two machines, however, do not issue the exact same queries to their respective oracles). Since these are rather artificial reductions, we chose not to highlight them in the proposition itself.

5.2 A technique for proving lower bounds on linear-access algorithms

We now present a technique for proving lower bounds for linear-access problems. We start by showing that the problem of proving lower bounds in the linear-access model can be reduced to the *analysis of affine subspaces of* $\mathbb{F}^n$.

Proposition 5.2 (deterministic linear-access algorithms partition $\mathbb{F}^n$ to affine subspaces). *For $n \in \mathbb{N}$ and a finite field $\mathbb{F}$, let M be a* deterministic oracle machine *that, when given input 1^n and oracle access to an oracle of the form $H(w)$ for $w \in \mathbb{F}^n$, makes m queries and outputs either 0 or 1. Then M induces a partition of $\mathbb{F}^n$ to $t \leq |\mathbb{F}|^m$ affine subspaces $(\mathcal{V}_1, ..., \mathcal{V}_t)$ such that for any $i \in [t]$ and $w, w' \in \mathcal{V}_i$ it holds that $M^{H(w)}(1^n) = M^{H(w')}(1^n)$ and during both executions the same queries were issued and the same responses were given.*

Note that in the case of $\mathbb{F} = \mathbb{F}_2$, a deterministic linear-access algorithm is a parity decision tree, and the affine subspaces in the partition correspond to the leaves of the tree.

Proof. For $w \in \mathbb{F}^n$, denote the m queries issued by $M^{H(w)}(1^n)$ during its execution by $Q_{m \times n}$ (i.e., the queries are depicted in $Rows(Q)$) and the responses received by $r \in \mathbb{F}^m$. Let

$$\mathcal{V}_{Q,r} = \{w' \in \mathbb{F}^n : Qw' = r\}$$

[4] Regarding upper bounds for $\mathcal{P}$, to obtain a tester for $\mathcal{P}$ from an existing tester for Π one can add a linearity test [5] and use self-correction (see, e.g., [2, Appendix A] for details). However, self-correction may increase the tester's query complexity by a logarithmic multiplicative factor.

be an affine subspace.

Clearly $w \in \mathcal{V}_{Q,r}$. Let $w' \in \mathcal{V}_{Q,r}$. Since M is deterministic, the first query issued by $M^{H(w')}(1^n)$ is identical to the first query issued by $M^{H(w)}(1^n)$, and since $w' \in \mathcal{V}_{Q,r}$, the first response is also identical in both cases. By induction, all m queries and responses will be identical in both cases, and in particular the final output will also be identical. We stress that this is true for both adaptive and non-adaptive machines.

To see that these subspaces are a partition of $\mathbb{F}^n$, consider two subspaces $\mathcal{V}^{(1)}_{Q^{(1)},r^{(1)}}$ and $\mathcal{V}^{(2)}_{Q^{(2)},r^{(2)}}$ such that for $i = 1, 2$, for every input $w \in \mathcal{V}^{(i)}_{Q^{(i)},r^{(i)}}$ it holds that $M^{H(w)}(1^n)$ executes queries $Q^{(i)}$ and receives responses $r^{(i)}$. If there exists $w \in \mathcal{V}^{(1)}_{Q^{(1)},r^{(1)}} \cap \mathcal{V}^{(2)}_{Q^{(2)},r^{(2)}}$ then it follows that $Q^{(1)} = Q^{(2)}$ and $r^{(1)} = r^{(2)}$, which implies that $\mathcal{V}^{(1)}_{Q^{(1)},r^{(1)}} = \mathcal{V}^{(2)}_{Q^{(2)},r^{(2)}}$. Also, any $w \in \mathbb{F}^n$ belongs to some affine subspace of this form (induced by the queries and responses during the execution of $M^{H(w)}(1^n)$).

Further note that there are at most $|\mathbb{F}|^m$ subspaces in the partition. If we assume that all queries made by M are linearly independent, then the response to M's first query induces a partition of $\mathbb{F}^n$ to $|\mathbb{F}|$ distinct subspaces; and on each of these subspaces, the response to M's second query will induce a partition of the subspace to $|\mathbb{F}|$ smaller subspaces. By induction, the response to the m^{th} query induces a partition of $\mathbb{F}^n$ to $|\mathbb{F}|^m$ subspaces. In the general case, if some queries are dependent on previous ones, then the number of subspaces in the partition can only be smaller. □

Using Proposition 5.2, we suggest a simple technique for proving lower bounds in the linear-access model.

Technique 5.3 (a technique for showing lower bounds on linear-access algorithms). *For $\mathcal{Q}, \mathcal{W} \subseteq \mathbb{F}^n$ and $\Phi = (\mathcal{Q}, \mathcal{W})$, we show a lower bound of $\mathtt{LA}(\Phi) = \Omega(m)$ as follows:*

1. *(Standard reduction to deterministic algorithms): In order to lower bound the error probability of any linear-access algorithm with query complexity m, it suffices to lower bound the error probability of all deterministic oracle machines of query complexity m over some (arbitrary) distribution on the inputs. We therefore focus on lower bounding the latter.*

2. *(Partition to affine subspaces): Without loss of generality we assume that each deterministic oracle machine we examine makes exactly m linearly independent queries when given input 1^n and oracle access to $H(w)$, for any $w \in \mathbb{F}^n$. According to Proposition 5.2, any such machine induces a partition of $\mathbb{F}^n$ to $|\mathbb{F}|^m$ affine subspaces of dimension $n-m$ such that its output is fixed on each of them.*

3. *(Key step): Show a distribution over the inputs that assigns an $\Omega(1)$ probabilistic mass to $\mathcal{Q} \cap \mathcal{W}$, and an $\Omega(1)$-fraction of the probabilistic mass of every affine subspace of dimension $n - m$ to $\mathcal{Q} \setminus \mathcal{W}$ (alternatively, the roles of $\mathcal{Q} \cap \mathcal{W}$ and $\mathcal{Q} \setminus \mathcal{W}$ in this requirement can be switched).*

These steps yield a lower bound of m queries on linear-access algorithms solving Φ with some constant error μ, that is $\mathtt{LA}_\mu(\Phi) > m$. It follows that $\mathtt{LA}(\Phi) = \Omega(m)$.

To see that indeed these steps yield an $\Omega(m)$ lower bound, assume that we prove Step (3) by presenting a distribution $\mathcal{D}$ that satisfies both requirements (of the first alternative). Consider an arbitrary deterministic linear-access algorithm M with query complexity m. Since the probabilistic mass of "yes" instances is lower bounded by some $p \in (0, 1)$, if M accepts subspaces of probabilistic mass at most $\frac{p}{2}$, then it incorrectly rejects a probabilistic mass of $\frac{p}{2}$ "yes" instances. On the other hand,

whenever M accepts a subspace $\mathcal{V}$ in its partition of $\mathbb{F}^n$, it suffers an error of magnitude $\mu' \cdot \mathcal{D}(\mathcal{V})$, for a fixed constant $\mu' > 0$. Therefore, if M accepts subspaces of probabilistic mass larger than $\frac{p}{2}$, it suffers an error of at least $\mu' \cdot \frac{p}{2}$, due to incorrectly accepted "no" instances. Either way, the algorithm M suffers a constant error. Proving Step (3) with the alternative formulation yields a symmetric argument.

The key step in the technique is Step (3) — analyzing the intersection of large affine subspaces in $\mathbb{F}^n$ with $\mathcal{Q} \cap \mathcal{W}$ or with $\mathcal{Q} \setminus \mathcal{W}$. While analyzing affine subspaces is a straight forward approach when trying to prove lower bounds for properties of linear functions (see, e.g., [4]), it follows from our results that lower bounds on broader classes of properties (e.g., all properties of low-degree polynomials) can also be provable with this technique. Indeed, we use Technique 5.3 in Section 5.3 to prove a lower bound on testing subcodes of linear codes, and in Section 5.4 to prove lower bounds on a property of polynomials over $\mathbb{F}_2$.

Technique 5.3 is reminiscent of a known lower bound technique in communication complexity (see, e.g., [16, Method 1]). We stress, however, that in communication complexity one needs to analyze products of arbitrary sets (of size that is not too small), whereas here we just need to analyze affine subspaces of a fixed large size, which is a potentially simpler challenge.

5.3 A lower bound on testing a family of linear subcodes

In this section we apply Technique 5.3 to prove a lower bound on the *inner-product linear-access problem*; that is, the problem of recognizing strings of the form $x \circ y$ such that $\langle x, y \rangle = 1$. The proof is relatively easy, using only Technique 5.3 and elementary linear algebra. Following this proof, for any linear code C with constant relative distance, we show how to reduce the inner-product linear-access problem to the property consisting of codewords of the form $C(x \circ y)$, where $\langle x, y \rangle = 1$. Thus, we derive a lower bound on testing this family of properties, which is a family of subcodes of linear codes.

A lower bound on similar families of properties was originally proved by Goldreich [11, Thms 4.1 and 4.2] by reducing the properties from the *inner-product communication complexity problem*, that requires identifying input pairs (x, y) such that $\langle x, y \rangle = 1$. The original proof for an $\Omega(n)$ lower bound on the communication complexity of the inner-product problem was provided by Chor and Goldreich [8] and relied on Lindsey's lemma. In Example 4.4 we presented a deconstruction of reductions from communication complexity to property testing of a corresponding form. Here, we reduce the property directly from the intermediary model.

Proposition 5.4 (inner-product linear-access problem). *For an even integer $n \in \mathbb{N}$ let $\mathcal{W} = \{w = (x, y) \in \{0,1\}^n : \langle x, y \rangle = 1\}$. Then the query complexity of $\mathcal{W}$ as a linear-access problem is $\Omega(n)$.*

Proof. We prove that every affine subspace of dimension $\frac{4}{5} \cdot n$ contains a balanced proportion of vectors from $\mathcal{W}$ and from $\{0,1\}^n \setminus \mathcal{W}$. This is a well-known result in the area of randomness extraction, which follows from the fact that the inner-product function (sometimes referred to as the Hadamard function) is an *affine extractor*. Two proofs for this general fact were recently presented in writing by Cohen and Shinkar [9], and we provide a third proof (with weaker parameters) using elementary linear algebra. To finish the proof we consider the uniform distribution over $\{0,1\}^n$, and note that it translates the balanced proportions of "yes" instances and "no" instances inside every affine subspace of dimension $\frac{4}{5} \cdot n$ to an identically balanced probabilistic mass assigned to both sets.

We therefore focus on showing that every affine subspace of dimension $\frac{4}{5} \cdot n$ contains a balanced proportion of vectors from $\mathcal{W}$ and from $\{0,1\}^n \setminus \mathcal{W}$. For a sufficiently large even integer $n = 2k$ and

$m = \lfloor \frac{n}{5} \rfloor$, let $\mathcal{V}$ be an arbitrary affine subspace of dimension $n - m$. We partition $\mathcal{V}$ into 2^m product subspaces and prove the claim for each of these subspaces. The intuitive reason for this partition is that $\mathcal{W}$ has a structure corresponding to two separate parts (x and y for input (x, y)), and hence it will be easier to analyze its intersection with product subspaces.

To define these product subspaces, we present the affine subspace as $\mathcal{V} = \{w \in \{0,1\}^n : Qw = r\}$, where Q is an $m \times n$ matrix and $r \in \{0,1\}^m$. We also denote $Q \stackrel{\text{def}}{=\!=} (Q'|Q'')$, where Q' and Q'' are of dimensions $m \times k$. Now, for every $s \in \{0,1\}^m$, let

$$\mathcal{V}^{(s)} \stackrel{\text{def}}{=\!=} \left\{ w = (x, y) \in \{0,1\}^{2k} : \begin{array}{c} Q'x = s \\ Q''y = r \oplus s \end{array} \right\}$$

Note that $w \in \mathcal{V}$ if and only if $w \in \mathcal{V}^{(s)}$ for some $s \in \{0,1\}^m$, and that for $s \neq s'$ it holds that $\mathcal{V}^{(s)} \cap \mathcal{V}^{(s')} = \emptyset$, hence this is indeed a partition of $\mathcal{V}$. Furthermore, note that for any $s \in \{0,1\}^m$ it holds that $\mathcal{V}^{(s)}$ is the Cartesian product (i.e., external sum) of the following two subspaces:

$$\mathcal{X}^{(s)} = \{x \in \{0,1\}^k : Q'x = s\}$$

$$\mathcal{Y}^{(s)} = \{y \in \{0,1\}^k : Q''y = r \oplus s\}$$

That is, $\mathcal{V}^{(s)} = \mathcal{X}^{(s)} \times \mathcal{Y}^{(s)}$.

Lemma. *For every $s \in \{0,1\}^m$ it holds that $|\mathcal{V}^{(s)} \cap \mathcal{W}| \leq \frac{3}{4} \cdot |\mathcal{V}^{(s)}|$.*

Proof. If $\mathcal{V}^{(s)} = \emptyset$ then the claim clearly holds. Otherwise, let $m' \stackrel{\text{def}}{=\!=} \mathit{Rank}(Q')$ and $m'' \stackrel{\text{def}}{=\!=} \mathit{Rank}(Q'')$, where both m' and m'' are upper bounded by m. Then $|\mathcal{X}^{(s)}| = 2^{k-m'}$ and $|\mathcal{Y}^{(s)}| = 2^{k-m''}$ and $|\mathcal{V}^{(s)}| = 2^{2k-m'-m''}$. We upper bound $|\mathcal{V}^{(s)} \cap \mathcal{W}|$ in this case by upper bounding the size of the following two sets:

1. $\texttt{Dep} = \left\{(x, y) \in \mathcal{V}^{(s)} : y \in \mathit{Span}(\mathit{Rows}(Q'))\right\}$. Since $|\mathit{Span}(\mathit{Rows}(Q'))| = 2^{m'}$, it holds that $|\texttt{Dep}| \leq |\mathcal{X}^{(s)}| \cdot 2^{m'} = 2^k$.
2. $\texttt{Ind} = \left\{(x, y) \in \mathcal{V}^{(s)} : y \notin \mathit{Span}(\mathit{Rows}(Q')) \wedge \langle x, y \rangle = 1\right\}$. Note that for any fixed $y \in \mathcal{Y}^{(s)} \setminus \mathit{Span}(\mathit{Rows}(Q'))$ there are exactly $\frac{1}{2} \cdot |\mathcal{X}^{(s)}|$ vectors x such that $(x, y) \in \texttt{Ind}$. This is the case since for such y we can add the independent row y to Q' and the coordinate 1 to s, enforcing the additional constraint $\langle y, x \rangle = 1$ on $\mathcal{X}^{(s)}$. Therefore $|\texttt{Ind}| \leq \frac{1}{2} \cdot |\mathcal{X}^{(s)}| \cdot |\mathcal{Y}^{(s)}| = \frac{1}{2} \cdot |\mathcal{V}^{(s)}|$.

Since $\mathcal{V}^{(s)} \cap \mathcal{W} \subseteq \texttt{Dep} \cup \texttt{Ind}$, it follows that

$$|\mathcal{V}^{(s)} \cap \mathcal{W}| \leq 2^k + \frac{1}{2} \cdot |\mathcal{V}^{(s)}| = (2^{m'+m''-k} + \frac{1}{2}) \cdot |\mathcal{V}^{(s)}| \leq \frac{3}{4} \cdot |\mathcal{V}^{(s)}|$$

where the last inequality is since for $n \geq 12$ it holds that $m \leq \frac{k}{2} - 1$, implying that $m' + m'' \leq 2m \leq k - 2$. $\square$

Using a nearly identical argument we can deduce that $|\mathcal{V}^{(s)} \setminus \mathcal{W}| \leq \frac{3}{4} \cdot |\mathcal{V}^{(s)}|$. Since this is true for all $\mathcal{V}^{(s)}$ in the partition of $\mathcal{V}$, it is also true for $\mathcal{V}$ itself, and therefore

$$\frac{1}{4} \leq \frac{|\mathcal{V} \cap \mathcal{W}|}{|\mathcal{V}|} \leq \frac{3}{4}$$

To finish the proof, let $\mathcal{D}$ be the uniform distribution over $\{0,1\}^n$. Then it holds that $\frac{1}{4} \leq \frac{\mathcal{D}(\mathcal{V} \cap \mathcal{W})}{\mathcal{D}(\mathcal{V})} \leq \frac{3}{4}$, and hence also overall $\frac{1}{4} \leq \mathcal{D}(\mathcal{W}) \leq \frac{3}{4}$. Therefore $\mathcal{D}$ satisfies both requirements in Step (3) of Technique 5.3, and the proposition follows. □

Digest. The lower bound on the linear-access inner-product problem follows from the fact that the inner-product function $IP(x,y) = \langle x,y \rangle$ is an *affine extractor* (i.e., is balanced on affine subspaces of sufficient dimension). This is analogous to a result by Chor and Goldreich [8], who proved a lower bound on the communication complexity of the inner-product function by showing that it is a *two-source extractor* (for a definition and further details see, e.g., [8, 22]). Continuing the analogy, since communication complexity is a stronger computational model than linear-access algorithms, the result used to prove a lower bound in communication complexity (i.e., that IP is a two-source extractor) is stronger than the result used to prove a lower bound on linear-access algorithms (i.e., that IP is an affine extractor).

Combining Proposition 5.4, Proposition 3.3, and Theorem 3.2 we get

Corollary 5.5 *For an even integer n and $l \in \mathbb{N}$, let $C : \{0,1\}^n \to \{0,1\}^l$ be a linear code of constant relative distance $\epsilon > 0$. Let $\mathcal{P} = \{C(x \circ y) : (x,y) \in \{0,1\}^n \wedge \langle x,y \rangle = 1\}$. Then $\mathtt{PT}(\epsilon, \mathcal{P}) = \Omega(n)$.*

5.4 A lower bound on testing sparse linear functions and polynomials

We start this section by applying Technique 5.3 to lower bound the query complexity of the linear-access problem k-$\mathcal{WT}$, presented in Example 4.3 (recall that for $k \in [n]$ we define k-$\mathcal{WT}$ $= \{w \in \{0,1\}^n : \|w\|_1 = k\}$). Specifically, we show that the query complexity of k-$\mathcal{WT}$ is $\Omega(\min\{k, n-k\})$. We then rely on Proposition 5.1 to show that this lower bound is essentially equivalent to a property testing lower bound of $\Omega(\min\{k, n-k\})$ queries for testing "k-linearity"; that is, for testing the property of k-sparse linear Boolean functions over $\{0,1\}^n$. We thus provide an alternative proof for this known result.

We finish the section by proving a new lower bound on a property that is generalization of "k-linearity"; specifically, we show that $\Omega(\min\{s, \binom{n}{d} - s\})$ queries are needed to test the property of s-sparse polynomials of degree d over $\{0,1\}^n$, for any $d \in \mathbb{N}$. This result too is proved via a reduction from the k-$\mathcal{WT}$ linear-access problem, with $k = s$.

Theorem 5.6 (the k-$\mathcal{WT}$ linear-access problem): *For $n, k \in \mathbb{N}$, let k-$\mathcal{WT}$ be the linear-access problem defined by the set of "yes" instances $W_k = \{w \in \{0,1\}^n : \|w\|_1 = k\}$ and the trivial promise. Then the query complexity of k-$\mathcal{WT}$ is $\Omega(\min\{k, n-k\})$.*

We start by proving the result with parameter $k = \frac{n}{2}$; that is, for $\frac{n}{2}$-$\mathcal{WT}$. Then, we extend this to all values of $k \in [0, \frac{n}{2}]$ by reducing to the $k = \frac{n}{2}$ case. For any $k \in [n]$, the problems of k-$\mathcal{WT}$ and of $(n-k)$-$\mathcal{WT}$ are computationally equivalent, and therefore it suffices to focus on $k \in [0, \frac{n}{2}]$: The equivalence follows since $w \in k$-$\mathcal{WT}$ if and only if $w \oplus 1^n \in (n-k)$-$\mathcal{WT}$, and computing a linear query on either of the vectors, w or $w \oplus 1^n$, is possible by performing only a single linear query on the other vector (see [4, Apdx. B] for a full proof of a similar fact).

Recall that in the proof of Proposition 5.4 we considered the uniform distribution and proved that every large affine subspace contains a balanced proportion of "yes" entries and of "no" entries. In the case of $\frac{n}{2}$-$\mathcal{WT}$ this approach will not work, since the overall fraction of "yes" instances (i.e., of vectors with Hamming weight $\frac{n}{2}$) in $\{0,1\}^n$ is $O(\frac{1}{\sqrt{n}})$. We therefore rely on a general result by Linial and Samorodnitsky that states:

Linial and Samorodnitsky [18, Thm 4.4]: The fraction of vectors with the same Hamming weight in every affine subspace of dimension $\lambda \cdot n$ (for $\lambda > \frac{1}{2}$) is upper-bounded by $O_\lambda(\frac{1}{\sqrt{n}})$.

Proposition 5.7 (the $\frac{n}{2}$-$\mathcal{WT}$ linear-access problem): *For $n \in \mathbb{N}$, the query complexity of the linear-access problem $\frac{n}{2}$-$\mathcal{WT}$, that is the problem defined by the set of "yes" instances $W_{n/2} = \{w \in \{0,1\}^n : \|w\|_1 = \frac{n}{2}\}$ and the trivial promise, is $\Omega(n)$.*

Proof. Let $\mathcal{V}$ be an arbitrary affine subspace of $\{0,1\}^n$ of dimension at least $\frac{2}{3} \cdot n$. For $p \in (0,1)$, let $\mathcal{D}_p$ be a distribution that with probability p is uniform over $W_{n/2}$ and is otherwise uniform over $\{0,1\}^n \setminus W_{n/2}$. Note that for an arbitrary $u \in W_{n/2}$ and $v \in \{0,1\}^n \setminus W_{n/2}$ it holds that $\mathcal{D}_p(u) = \frac{p}{|W_{n/2}|}$ and $\mathcal{D}_p(v) = \frac{1-p}{|\{0,1\}^n \setminus W_{n/2}|}$. Therefore

$$\frac{\mathcal{D}_p(u)}{\mathcal{D}_p(v)} = \frac{p}{1-p} \cdot \frac{|\{0,1\}^n \setminus W_{n/2}|}{|W_{n/2}|} = \frac{p}{1-p} \cdot O(\sqrt{n})$$

From this it follows that

$$\begin{aligned}
\frac{\mathcal{D}_p(\mathcal{V} \cap W_{n/2})}{\mathcal{D}_p(\mathcal{V})} &= \frac{\mathcal{D}_p(u) \cdot |\mathcal{V} \cap W_{n/2}|}{\mathcal{D}_p(v) \cdot |\mathcal{V} \setminus W_{n/2}| + \mathcal{D}_p(u) \cdot |\mathcal{V} \cap W_{n/2}|} \\
&\leq \frac{\mathcal{D}_p(u)}{\mathcal{D}_p(v)} \cdot \frac{|\mathcal{V} \cap W_{n/2}|}{|\mathcal{V}|} \\
&= \frac{p}{1-p} \cdot O(\sqrt{n}) \cdot \frac{|\mathcal{V} \cap W_{n/2}|}{|\mathcal{V}|} \qquad (1)
\end{aligned}$$

According to Linial and Samorodnitsky's result it holds that (1) is upper bounded by $\frac{p}{1-p} \cdot c$ for some $c > 0$. By setting $p = \frac{1}{1+2 \cdot c} \in (0,1)$ we get that $\frac{\mathcal{D}_p(\mathcal{V} \cap W_{n/2})}{\mathcal{D}_p(\mathcal{V})} \leq \frac{1}{2}$. The proposition follows. □

Recall that to complete the proof of Theorem 5.6 (i.e., extend the lower bound for every $k \in [n]$) it suffices to show a lower bound of $\Omega(k)$ for any $k \in [0, \frac{n}{2})$. We prove this lower bound by a simple black-box reduction to the case of Proposition 5.7. This black-box reduction is implicit in a padding argument presented in [4] for similar purposes.

Proof of Theorem 5.6. For $k \in [0, \frac{n}{2})$, let $m = 2 \cdot k < n$. Assuming that there exists a linear-access algorithm M' for k-$\mathcal{WT}$ over $\{0,1\}^n$, we construct a corresponding algorithm for $\frac{m}{2}$-$\mathcal{WT}$ over $\{0,1\}^m$ with the same error probability and query complexity. Since the query complexity of $\frac{m}{2}$-$\mathcal{WT}$ is $\Omega(m)$, it follows that the query complexity of k-$\mathcal{WT}$ over $\{0,1\}^n$ is $\Omega(m) = \Omega(k)$. The construction itself is straight forward: The algorithm M is given access to $H(w)$, for some $w \in \{0,1\}^m$, and simulates the execution of M' when M' is given access to $H(w')$, where $w' = w \circ 0^{n-m}$. Note that M can answer any oracle query that M' makes by making a single query to its own oracle, and also that $\|w\|_1 = \|w'\|_1$ and therefore $\|w\|_1 = \frac{m}{2}$ if and only if $\|w'\|_1 = \frac{m}{2} = k$. □

Recall that according to Proposition 5.1, if a property Π is reducible from a linear-access problem Φ via the Hadamard code, then both problems are essentially equivalent. In Example 4.3 we showed that the Hadamard code reduces k-$\mathcal{WT}$ to the property of k-sparse linear Boolean functions. Therefore, Theorem 5.6 is essentially equivalent to the following proposition:

Theorem 5.8 (k-linearity, alternative formulation of Theorem 5.6): *For $n, k \in \mathbb{N}$, the query complexity of testing the property of k-sparse linear Boolean functions over $\{0,1\}^n$ is $\Omega(\min\{k, n-k\})$.*

A self-contained proof of Theorem 5.8, using the argument presented in the proof of Theorem 5.6 instead of relying on Proposition 5.1, appears in our technical report [24].

Testing sparse polynomials. We now extend Theorem 5.8 to a lower bound on a broader family of properties. For integers n, s, and d, we define the property of *s-sparse degree-d polynomials* as all n-variate polynomials over $\mathbb{F}_2$ that are of total degree d such that exactly s of their coefficients are non-zero. This problem is a straightforward generalization of the problem of testing k-sparse linear functions (i.e., of Theorem 5.8), which is the special case of $d = 1$.

A lower bound on a related property was proved by Blais, Brody, and Matulef [3]: They considered the property that consists of all n-variate s-sparse polynomials, of any degree, and showed that its query complexity is $\Omega(\min\{s, n-s\})$. Our formulation is a parametrization of their problem, since we consider a property that only consists of polynomials of a predetermined degree $d \in \mathbb{N}$. Furthermore, we show a lower bound of $\Omega(\min\{s, \binom{n}{d} - s\})$, which is stronger when $s = \omega(n)$.[5]

Theorem 5.9 *Let $n, s, d \in \mathbb{N}$ where $n > d$, and let s-$\mathcal{SP} \subseteq \{0,1\}^{2^n}$ be the property of s-sparse degree-d polynomials. Then, the query complexity of (2^{-d})-testing the property is $\Omega(\min\{s, \binom{n}{d} - s\})$.*

We prove Theorem 5.9 by reducing from the linear-access problem of s-$\mathcal{WT}$. While the proof of Theorem 5.8 uses the Hadamard code as a reduction, in the following proof we use a variant of the Reed-Muller code.

Proof of Theorem 5.9. Let $m = \binom{n}{d}$, and s-$\mathcal{WT} = \{w \in \{0,1\}^m : \|w\|_1 = s\}$. By Theorem 5.6, the query complexity of s-$\mathcal{WT}$ is $\Omega(\min\{s, m-s\})$. Let $F : \{0,1\}^m \to \{0,1\}^{2^n}$ be defined as follows: Every $w \in \{0,1\}^m$ represents the coefficients of a polynomial that has total degree d and whose coefficients of all monomials of total degree less than d are zero. Correspondingly, $F(w)$ is the evaluations of this polynomial on all points in $\mathbb{F}_2^n$. Note that for every $w \in s$-$\mathcal{WT}$ it holds that $F(w)$ it an s-sparse polynomial of total degree d, since it has exactly s non-zero coefficients and all of them correspond to monomials with d variables. By the properties of the Reed-Muller code, for every $w \notin s$-$\mathcal{WT}$ it holds that $F(w)$ is (2^{-d})-far from being s-sparse, and in particular is (2^{-d})-far from being an s-sparse degree-d polynomial. Furthermore, the projections of F are computable with a single linear query. Hence F is a $(2^{-d}, 1)$-reduction of s-$\mathcal{WT}$ to the property of s-sparse degree-d polynomials. □

6 Digest and Open Questions

6.1 Proving lower bounds in property testing via linear-access algorithms

In this work we discussed two classes of properties that can be reduced from linear-access algorithms: Properties of low-degree rational functions over finite fields (and in particular, properties of low-degree polynomials), and subcodes of linear codes with constant relative distance. Correspondingly, we proved lower bounds on testing the sparsity of polynomials over $\mathbb{F}_2$ (Theorem 5.9) and on testing certain families of linear subcodes (Corollary 5.5). These results lead to the following questions:

[5] Note that one might expect a lower bound of $\Omega(\min\{s, \binom{n+d}{d} - s\})$ for this property, since n-variate degree-d polynomials have $\binom{n+d}{d}$ coefficients. However, since for a fixed $d \in \mathbb{N}$ it holds that both $\binom{n+d}{d}$ and $\binom{n}{d}$ are $\Theta(n^d)$, the difference between such a lower bound and the one presented in Theorem 5.9 is not significant.

Open question 1: Can additional classes of natural properties be reduced from linear-access algorithms?

Open question 2: Can additional new (or tighter) lower bounds on natural properties be proved via reductions from linear-access algorithms?

We mention, however, that many natural properties of low-degree polynomials are known to be testable in $O(1)$ (and even testable with Proximity-Oblivious testers, see [1]). Yet, as demonstrated by the lower bound on testing the sparsity of polynomials over $\mathbb{F}_2$, other properties of low-degree polynomials may admit significant lower bounds.

Interestingly, all property testing lower bounds we showed in this work by reductions from linear-access algorithms can also be proved by reductions from communication complexity: Theorem 5.8 was indeed proved in [3, Thm 1.1] using a reduction from the set-disjointness communication problem (see Example 4.3); Theorem 5.9 can be proved via a similar reduction from the set-disjointness communication problem (substituting the Hadamard code for the Reed-Muller code); and Corollary 5.5 can be proved similar to [11, Thm 4.2], using a reduction from the inner-product communication problem. A natural question is therefore whether this represents a more general phenomena.

Open question 3: Is there a linear-access problem with higher query complexity than the communication complexity of *every communication problem* that is reducible to it?

In this work we were able to show (Proposition 3.11) that there exist sets $\mathcal{S}' \subseteq \{0,1\}^n \times \{0,1\}^n$ that have communication complexity $O(1)$ and that can be reduced to corresponding linear-access problems with query complexity $\Omega(n)$ via concatenation.

6.2 Linear-access algorithms and parity decision trees

Lower bounds on deterministic parity decision trees follow from the fact that some functions are *affine dispersers*, that is, are not constant on affine subspaces of sufficiently large dimension. This is since a parity decision tree needs to partition the input space into affine subspaces of sufficiently small dimension such that the function to be computed is constant on every subspace in the partition.

However, when considering linear-access algorithms, which are a generalization of *randomized* parity decision trees, affine dispersers *do not* yield lower bounds in the same way. Technique 5.3 and Proposition 5.4 demonstrate that lower bounds on linear-access algorithms follow from the fact that some functions are *affine extractors*, that is, are *far from being constant* on affine subspaces of sufficiently large dimension. As Proposition 5.4 demonstrates, in this case we can reduce the problem to an analysis of deterministic testers and consider a uniform distribution on the inputs. To prove a lower bound in this manner it suffices that the corresponding function be a weak affine extractor: In particular, any fixed distance from being constant on affine subspaces of any linear co-dimension ($\lambda \cdot n$ for any constant $\lambda > 0$) suffices.

We stress that there are also *other ways to show lower bounds on linear-access algorithms* (i.e., besides considering affine extractors and the uniform distribution). For example, the proof of Theorem 5.6 relies on Technique 5.3 but considers a distribution that is very different from the uniform distribution.

6.3 Investigating the connection between communication complexity and property testing

In addition to proving lower bounds, the current work further investigates the connection between communication complexity and property testing, continuing the line of work started by Blais, Brody, and Matulef [3], and followed by Goldreich [11] and by Bhrushundi, Chakraborty, and Kulkarni [2].

The decomposition we suggested (of reductions from communication complexity to property testing) will not necessarily work for all reductions between the models; specifically, as mentioned in the discussion following Theorem 3.8, our form of decomposition is to a large extent appealing for reductions that only compute linear functions of the inputs. Yet, since the decomposition sheds light on the functionality of the two parts of these reductions, the question remains whether reductions of a more general form can be deconstructed in a similar (or other) fashion.

Acknowledgements

The author thanks Tom Gur for suggesting the initial observation motivating the study and for several helpful discussions during the research process. The author is grateful to Avishay Tal for pointing him to the work of Linial and Samorodnitsky. The author also thanks his advisor, Oded Goldreich, for his guidance and support in the research and writing process. This research was partially supported by the Israel Science Foundation (grant No. 671/13).

References

[1] Arnab Bhattacharyya, Eldar Fischer, Hamed Hatami, Pooya Hatami, and Shachar Lovett. Every locally characterized affine-invariant property is testable. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 429–436, New York, NY, USA, 2013. ACM.

[2] Abhishek Bhrushundi, Sourav Chakraborty, and Raghav Kulkarni. Property testing bounds for linear and quadratic functions via parity decision trees. In *CSR*, pages 97–110, 2014.

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

[4] Eric Blais and Daniel M. Kane. Tight bounds for testing k-linearity. In *APPROX-RANDOM*, pages 435–446, 2012.

[5] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 73–83, New York, NY, USA, 1990. ACM.

[6] Harry Buhrman, David García-Soriano, Arie Matsliah, and Ronald de Wolf. The non-adaptive query complexity of testing k-parities. *Chicago J. Theor. Comput. Sci.*, 2013, 2013.

[7] Deeparnab Chakrabarty and C. Seshadhri. A o(n) monotonicity tester for boolean functions over the hypercube. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 411–418, New York, NY, USA, 2013. ACM.

[8] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity (extended abstract). In *FOCS*, pages 429–442, 1985.

[9] Gil Cohen and Igor Shinkar. The complexity of dnf of parities. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:99, 2014.

[10] Oded Goldreich. On testing computability by small width obdds. In *APPROX-RANDOM*, pages 574–587, 2010.

[11] Oded Goldreich. On the communication complexity methodology for proving lower bounds on the query complexity of property testing. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:73, 2013.

[12] Oded Goldreich, Shafi Goldwasser, Eric Lehman, and Dana Ron. Testing monotonicity. In *FOCS*, pages 426–435, 1998.

[13] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, July 1998.

[14] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. In *STOC*, pages 406–415, 1997.

[15] Johan Håstad and Avi Wigderson. The randomized communication complexity of set disjointness. *Theory of Computing*, 3(11):211–219, 2007.

[16] Hartmut Klauck. Rectangle size bounds and threshold covers in communication complexity. In *IEEE Conference on Computational Complexity*, pages 118–134, 2003.

[17] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

[18] Nathan Linial and Alex Samorodnitsky. Linear codes and character sums. *Combinatorica*, 22(4):497–522, 2002.

[19] Ashley Montanaro and Tobias Osborne. On the communication complexity of xor functions. *CoRR*, abs/0909.3392, 2009.

[20] Alexander A. Razborov. On the distributional complexity of disjointness. In *ICALP*, pages 249–253, 1990.

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

[22] Ronen Shaltiel. An introduction to randomness extractors. In *ICALP (2)*, pages 21–41, 2011.

[23] Amir Shpilka, Avishay Tal, and Ben lee Volk. On the structure of boolean functions with small spectral norm. In *ITCS*, pages 37–48, 2014.

[24] Roei Tell. An alternative proof of an $\Omega(k)$ lower bound for testing k-linear boolean functions. *Electronic Colloquium on Computational Complexity (ECCC)*, 21, 2014.

[25] Zhiqiang Zhang and Yaoyun Shi. On the parity complexity measures of boolean functions. *Theor. Comput. Sci.*, 411(26-28):2612–2618, 2010.

Appendix: Testers that Always Reject Inputs that are Far from the Property

Referring to Definition 2.3, recall that the standard definition of an ϵ-tester T with *one-sided error* for a property testing problem $\Pi = (\mathcal{U}, \mathcal{P})$ is an ϵ-tester that always accepts inputs that are "yes" instances; that is, T also satisfies the following condition: If $z \in \mathcal{U} \cap \mathcal{P}$ then $Pr[T^z(1^l) = 1] = 1$. Testers with one-sided error are common, and many natural properties admit such testers with $o(n)$ query complexity (e.g., linearity testing [5], monotonicity of Boolean functions [12, 7], testing various graph properties in various models [13, 14] etc.).

We consider the dual notion of testing with one-sided error, that is, testers that always reject inputs that are "no" instances. We call such machines *ϵ-testers with perfect soundness*. Formally, a randomized oracle machine is an ϵ-tester with perfect soundness for $\Pi = (\mathcal{U}, \mathcal{P})$, where $\mathcal{U}, \mathcal{P} \subseteq \{0,1\}^n$, if it satisfies the following two conditions:

1. If $z \in \mathcal{U} \cap \mathcal{P}$ then $Pr[T^z(1^n) = 1] > 0$.
2. If $z \in \mathcal{U}$ is ϵ-far from $\mathcal{P}$ then $Pr[T^z(1^n) = 0] = 1$.

Indeed, we consider a very relaxed notion with respect to the acceptance probability of "yes" instances: That is, we require acceptance with any positive probability, instead of a constant value such as $\frac{1}{2}$. Yet, as shown next, even this relaxed notion seems quite limited in scope: Specifically, we show that a property (considered with the trivial promise) admits an ϵ-tester with perfect soundness and query complexity q only if every input is $(\epsilon + \frac{q}{n})$-close to the property. This follows as a special case of the following, more general, result:

Theorem 1 *For $n \in \mathbb{N}$, let $\mathcal{U}, \mathcal{P} \subseteq \{0,1\}^n$ such that $\mathcal{U} \cap \mathcal{P} \neq \emptyset$ and let $\Pi = (\mathcal{U}, \mathcal{P})$. If there exists an ϵ-tester with perfect soundness and query complexity q for Π, then every $z \in \{0,1\}^n$ is either $\frac{q}{n}$-close to $\{0,1\}^n \setminus \mathcal{U}$ or $(\frac{q}{n} + \epsilon)$-close to $\mathcal{U} \cap \mathcal{P}$.*

Proof. Let $x \in \mathcal{U} \cap \mathcal{P}$. Then, there exists a random string r such that the residual deterministic tester $T^x(1^n, r)$ accepts after making q queries (we assume for simplicity and without loss of generality that T always makes exactly q queries). Denote the coordinates of these q queries by $(i_1, i_2, ..., i_q)$.

Note that every $z' \in \{0,1\}^n$ such that $(z'_{i_1}, z'_{i_2}, ..., z'_{i_q}) = (x_{i_1}, x_{i_2}, ..., x_{i_q})$ is accepted by the residual deterministic tester with random string r. Since T has perfect soundness, this implies that every such z' either violates the promise (i.e., $z' \notin \mathcal{U}$) or is ϵ-close to $\mathcal{P}$.

Hence, for any $z \in \{0,1\}^n$, by changing the q coordinates $(z_{i_1}, z_{i_2}, ..., z_{i_q})$ to equal $(x_{i_1}, x_{i_2}, ..., x_{i_q})$ we obtain a string z' that is either ϵ-close to $\mathcal{U} \cap \mathcal{P}$ or in $\{0,1\}^n \setminus \mathcal{U}$. □

Corollary 2 *For $n \in \mathbb{N}$, let $\mathcal{P} \subseteq \{0,1\}^n$ be a non-empty set. If there exists an ϵ-tester T for the property $\mathcal{P}$ (i.e., for the promise problem $\Pi = (\{0,1\}^n, \mathcal{P})$) that has perfect soundness and query complexity q, then every $z \in \{0,1\}^n$ is $(\epsilon + \frac{q}{n})$-close to $\mathcal{P}$.*

Corollary 2 implies that most natural properties do not admit testers with *perfect soundness* and query complexity $o(n)$, because in natural cases not all inputs are close to the property.

Nevertheless, testers with perfect soundness and query complexity $o(n)$ do exist for classes of promise problems. In particular, when any two inputs in a promise $\mathcal{U}$ are ϵ-far from each other, then any ϵ-tester with one-sided error for a property $\Pi = (\mathcal{U}, \mathcal{P})$ yields an ϵ-tester with perfect soundness for the property $\Pi^C = (\mathcal{U}, \mathcal{U} \setminus \mathcal{P})$ by complementing the output. This is because in this case an ϵ-tester

for Π is simply an oracle machine deciding, for every $z \in \mathcal{U}$, whether $z \in \mathcal{P}$ or not. Also note that in this case, every input $z \in \{0,1\}^n$ is $\frac{1}{n}$-close to violating the promise.

Appealing examples for such promise problems are problems of testing subcodes of error-correcting codes, under the promise that the input is a codeword. Specifically, this applies to properties of low-degree polynomials and of linear functions with the corresponding promise.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de