An Exponential Separation Between MA and AM Proofs of Proximity*

Tom Gur[1]
UC Berkeley, Berkeley, USA
tom.gur@berkeley.edu

Yang P. Liu
MIT, Cambridge, MA
yangpatil@gmail.com

Ron D. Rothblum[2]
MIT and Northeastern University, Cambridge, MA
ronr@mit.edu

Abstract

Interactive proofs of proximity allow a sublinear-time verifier to check that a given input is *close* to the language, using a small amount of communication with a powerful (but untrusted) prover. In this work we consider two natural *minimally interactive* variants of such proofs systems, in which the prover only sends a single message, referred to as the proof.

The first variant, known as MA-proofs of Proximity (MAP), is fully *non-interactive*, meaning that the proof is a function of the input *only*. The second variant, known as AM-proofs of Proximity (AMP), allows the proof to additionally depend on the verifier's (entire) random string. The complexity of both MAPs and AMPs is the total number of bits that the verifier observes – namely, the sum of the proof length and query complexity.

Our main result is an exponential separation between the power of MAPs and AMPs. Specifically, we exhibit an explicit and natural property Π that admits an AMP with complexity $O(\log n)$, whereas any MAP for Π has complexity $\tilde{\Omega}(n^{1/4})$, where n denotes the length of the input in bits. Our MAP lower bound also yields an alternate proof, which is more general and arguably much simpler, for a recent result of Fischer *et al.* (ITCS, 2014).

Lastly, we also consider the notion of *oblivious* proofs of proximity, in which the verifier's queries are oblivious to the proof. In this setting we show that AMPs can only be quadratically stronger than MAPs. As an application of this result, we show an exponential separation between the power of public and private coin for oblivious interactive proofs of proximity.

2012 ACM Subject Classification Theory of computation → Interactive computation

Keywords and phrases Property testing, Probabilistic proof systems, Proofs of proximity

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.73

Related Version A full version of the paper is available at [31], `https://eccc.weizmann.ac.il/report/2018/083/`.

Acknowledgements We thank Oded Goldreich and Justin Thaler for very helpful discussions.

[*] Extended abstract. See full version [31].

[1] Research supported in part by the UC Berkeley Center for Long-Term Cybersecurity.

[2] Research supported in part by NSF Grants CNS-1350619 and CNS-1414119, Alfred P. Sloan Research Fellowship, Microsoft Faculty Fellowship and in part by the Defense Advanced Research Projects Agency (DARPA), the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236 and by the Cybersecurity and Privacy Institute at Northeastern University.

© Tom Gur, Yang P. Liu, and Ron D. Rothblum;
licensed under Creative Commons License CC-BY

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 73; pp. 73:1–73:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The field of property testing [47, 23] deals with sublinear algorithms for deciding whether a given object has a predetermined property or is far from any object having this property. Such algorithms, called testers, obtain local views of the object by performing queries; that is, the object is seen as a function and the tester receives oracle access to this function. The goal of the tester is to ascertain a global property of the function based only on its local view.

In the last couple of decades, the area of property testing has attracted much attention (see surveys [44, 45, 11] and recent textbook [22]). However, while much success was found in designing testers for a myriad of natural properties, which only make a small number of queries, many other natural properties were shown to require a very large number of queries to test (often linear in the length of the input).

Proofs of proximity, first considered by Ergün, Kumar and Rubinfeld [18], are both intrinsically interesting as a natural notion of proof systems for sublinear algorithms, as well as provide means to significantly reduce the number of queries that the tester needs to make in order to verify, rather than decide. These probabilistic proof systems can be viewed as augmenting testers with a help from a powerful, yet untrusted prover. In a recent line of works [46, 33, 20, 27, 36, 25, 43, 34, 9, 14] various types of interactive [46] and non-interactive proofs of proximity [35] were studied, including arguments of proximity [36], zero-knowledge proofs of proximity [9], and proofs of proximity for distribution testing [14].

In this work we study the relation between two types of proofs of proximity that are *minimally interactive*; namely, MA and AM proofs of proximity, which can be viewed as the property testing analogue of the class MA (i.e., "randomized NP") and AM, respectively, and are described in more detail next.

Informally speaking, an MA proof of proximity (MAP) protocol consists of a tester (or rather a verifier) that receives oracle access to an input function f but also receives explicit access to a short purported proof w. Based on the proof string and a few oracle queries to f, the verifier should decide whether f has some property Π (i.e., whether $f \in \Pi$). More specifically, after reading the proof w, the verifier tosses random coins, makes queries to the oracle f, and decides whether to accept or reject. We require the following completeness and soundness conditions: if $f \in \Pi$, then there exists a proof w that the verifier accepts with high probability, and if f is "far" (in Hamming distance) from any function in Π, then the verifier rejects with high probability. Following the literature, the complexity of an MAP is the total number of bits that the verifier observes - namely, the sum of its proof length and query complexity.[3]

The reason that the foregoing model is referred to as a "Merlin-Arthur" protocol is that we think of the prover as being Merlin (the all powerful magician) and the verifier as Arthur (a mere mortal). Then in the MAP model Merlin "speaks" first (i.e., sends the proof) and Arthur "speaks" second (i.e., tosses his random coins).

It is natural to ask what happens if we switch the order - letting Arthur toss his coins first and Merlin send his proof after seeing Arthur's coin tosses. This type of protocol is typically referred to as an "Arthur-Merlin" protocol. More precisely, an AM proof of proximity (AMP) is defined similarly to an MAP, except that now the proof oracle is a function of the verifier's entire random string. We view AMPs as *minimally interactive*, since given a common random

[3] Alternatively, one could view the running time of the verifier (which serves as an upper bound on the query and communication complexities) as the main resource to be minimized. However, for simplicity (and following the property testing literature), we focus on combinatorial resources. This only makes our lower bounds stronger, whereas for all of our upper bounds the verifier is also computationally efficient.

string, the protocol consists of a single message (i.e., the proof). Analogously to MAPs, the complexity of an AMP is the sum of its proof length and query complexity. We emphasize that the prover's message can depend on all of the verifier's coin tosses. Namely, the verifier cannot toss an additional coins after receiving its message from the prover.[4]

While the difference between these two proof systems may appear minor, MA-type and AM-type proofs naturally admit very different types of strategies. In particular, note that AM proofs provide the additional power of allowing the prover and verifier to jointly restrict their attention to a random subset of the input function's domain. On the other hand, the AM model also significantly hampers the power of the verifier to detect malicious prover strategies, since the prover knows the entire randomness of the verifier, and in particular the prover knows which queries the verifier will make.

At first glance, it may seem that AMPs are extremely limited, since the prover can predict exactly what the verifier will check (knowing the verifier's *entire* random string allows the prover to deduce which queries the verifier will make). However, it turns out that a straightforward adaptation of the classical MA $\subseteq$ AM inclusion [6] implies that any MAP can be emulated by an AMP at a quadratic cost. (More precisely, an MAP with proof complexity p and query complexity q can be emulated by an AMP with proof length p and query complexity $O(p \cdot q)$.[5])

It is natural to ask the following converse question:

Can any AMP *protocol be emulated by an* MAP*, or is there a gap between the power of these two models?*

Note that any AMP can be easily emulated by an MAP with at most an exponential overhead.[6] Thus, the question that we would really like to answer is whether such an exponential overhead is inherent.

1.1 Our results

Our main result shows that it is indeed the case that AMPs can be *exponentially* stronger than MAPs and so the foregoing emulation strategy is optimal, up to polynomial factors:

▶ **Theorem 1.** *There exists a property* $\Pi \subseteq \{f : [n] \to [n]\}$ *such that:*

- Π *has an* AMP *of complexity* $O(\log(n)/\varepsilon)$*, with respect to proximity parameter* $\varepsilon > 0$*; and*
- *Every* MAP *for* Π*, with respect to proximity parameter* $\varepsilon \leq \frac{1}{10}$*, must have complexity* $\Omega(n^{\frac{1}{4}})$*.*

The property Π that we use to prove Theorem 1 is actually very simple and natural. Specifically, Π is the set of all permutations over $[n]$; the goal of the verifier is to check

[4] In contrast, the complexity class AM is sometimes defined as any constant-round public-coin interactive proof-system. Indeed, if one does not care about polynomial factors, then by a result of Babai and Moran [6], any public-coin constant-round interactive proof can be reduced to just 2 messages.

[5] The idea is to first reduce the soundness error of the MAP to $2^{-O(p)}$ (by repetition). Now suppose that the verifier reveals its randomness to the prover before receiving the proof-string. For soundness, observe that when f is far from the having the property, for any fixed proof-string the probability that the verifier would accept is at most $2^{-\Omega(p)}$ and so by a union bound, with high probability there simply does not exist a proof-string that will make the verifier accept.

[6] Any AMP with proof complexity p and query complexity q can be emulated by a *tester* (i.e., an MAP which does not use a proof at all) with query complexity $q \cdot 2^p$ by simply trying all possible candidate proof strings.

whether a given function $f : [n] \to [n]$ is close to being a permutation by querying the function in a few locations and with a short interaction with the prover.

The AMP protocol for deciding whether a given function $f : [n] \to [n]$ is a permutation is extremely simple. The idea is that the random string specifies some random element $y \in [n]$ and the prover should specify an inverse x of y (under f). If f is a permutation such an element must exist whereas if f is ε-far from being a permutation, then with probability ε it holds that y simply does not have an inverse. We can repeat the base protocol $O(1/\varepsilon)$ times to get constant soundness error. This protocol can actually be traced back to a result of Bellare and Yung [7] who used it resolve a gap in the [19] construction of non-interactive zero-knowledge proofs for NP based on trapdoor permutations.

Our MAP lower bound is the technically more challenging part of this work, and is actually a special case of a more general MAP lower bound that we prove. We show that any property that satisfies a relaxed notion of k-wise independence requires MAPs with complexity roughly $\sqrt{k}$. This result generalizes a recent result of Fischer, Goldhirsh and Lachish [20] which can be interpreted as an MAP lower bound of $\sqrt{k}$ for properties that are exactly k-wise independent.[7] Our proof is also (arguably) significantly simpler than that of [20] and in particular uses only elementary arguments, see further discussion in Section 1.2.

1.1.1 Oblivious Proofs of Proximity

Having established Theorem 1, we revisit the MA versus AM problem within the context of proofs of proximity *with prover-oblivious queries* (a notion first considered in [46] and further explored in [35]), or in short *oblivious proofs of proximity.* These are proofs of proximity that have the special feature that the queries that the verifier makes are independent of the proof. Viewed from a temporal perspective, in these proof systems the verifier *first* makes its queries to the input function, and only after making all of its queries does it receive the proof. One reason that makes this feature appealing is because it allows the verifier to probe the object and obtain a certificate, which can then be used later when interacting with a prover, even if the object is no longer accessible. Another reason is that many of the interactive proof systems from the literature (e.g., the sumcheck protocol of [39]) are *oblivious.*

Surprisingly, it turns out that the gap between the power of *oblivious* AMPs and MAPs is dramatically smaller than the one exhibited in Theorem 1. Loosely speaking, we show that oblivious AMPs can only be *quadratically* stronger than oblivious MAPs, and in fact, standard testers (that do not use a proof).

▶ **Theorem 2.** *For any property* Π*, if there exists an* oblivious AMP *for* Π *with proof complexity* p *and query complexity* q*, then there also exists a tester (i.e.,* MAP *with proof complexity* 0*) for* Π *with query complexity* $O(p \cdot q)$*.*

As an application, we use Theorem 2 to derive lower bounds on public-coin oblivious interactive proofs of proximity, and show an exponential separation between public-coin and private-coin protocols in this setting. See further discussion in the full version [31].

1.2 Related works

The notion of proofs of proximity was originally proposed by Ergün, Kumar and Rubinfeld. Ben Sasson *et al.* [8] and Dinur and Reingold [17] considered such proofs in the context of

[7] More precisely, [20] show that any linear code with large dual distance requires MAPs of complexity that is roughly square root of the code's blocklength.

PCPs. Rothblum, Vadhan and Wigderson [46], considered *interactive* proofs of proximity and showed that every language computable by a low-depth circuit has an interactive proof of proximity (IPP) with a sublinear time verifier. Reingold, Rothblum, and Rothblum [43] showed *constant-round* IPPs for any language computable in polynomial-time and bounded polynomial-space. Goldreich and Gur [24, 25] showed general-purpose IPP with only 3 rounds, albeit for a much smaller class.

Proofs of proximity were further studied by [27] who showed more efficient constructions for certain restricted complexity classes, such as functions accepted by small read-once branching programs and context-free languages. Gur and Rothblum proved a round hierarchy theorem for IPPs [34], showing that the power of IPPs gradually increases with the number of rounds of interaction. Several works focused on studying non-interactive (MA) proofs of proximity [35, 20, 26] (see also [30]). In addition, recent works studied (computationally sound) interactive *arguments* of proximity [36], zero-knowledge proofs of proximity [9], and proofs of proximity for distribution testing [14]. Proofs of proximity have also found applications to property testing and related models [33, 21, 34]. We remark that a concurrent work of Berman *et al.* [9], utilizes our results (specifically Theorem 1) to derive a separation between the power of MAPs and zero-knowledge IPPs.

The notion of MA and AM proofs plays a central role in the study of proofs system in various computational models, other than in the setting of polynomial-time Turing machines in which they were originally conceived [6]. For example, in quantum computation, the class QMA (quantum MA proofs) captures the most fundamental type of quantum proof systems (since quantum algorithms are inherently randomized) and it has been extensively studied in the last couple of decades (see survey [4]). Of particular relevance, Aaronson [1] considered the problem of deciding whether a function is close to a permutation to derive a quantum query complexity separation between the class QMA and the class of statistical zero knowledge SZK, showing that every QMA query complexity algorithm with a w-qubit witness and query complexity q must satisfy $q + w = \Omega(n^{1/6})$.

In addition, MA and AM proof systems received much attention in the setting of communication complexity [5, 37, 41, 38, 29, 48] and streaming algorithms [15, 16, 12, 32, 13, 49]. The former also has an interesting connection to the algebrization barrier [2] and recently found important applications to distributed PCPs and hardness of approximation [3].[8] The latter can be viewed as the property testing analogue of online annotated data streams (there, instead of oracle access to the input, the algorithm has one-pass sequential access to the input, and the goal is to minimize *space* complexity rather than *query* complexity). Indeed, part of our results concerning oblivious proofs of proximity are inspired by the techniques for online annotated data streams in [13].

Perhaps most relevant to us, the notion of MA and AM proofs for decision tree complexity (or the "query complexity model"), which can be thought of as property testing for *exact* (rather than approximate) decision problems, is closely related to proofs of proximity, though the query complexity model is much simpler to analyze than property testing. We remark that the high-level approach of our main lower bound for MAPs is inspired by the work of Raz *et al.* [42].

[8] We remark that there are several similarities between MA and AM proof systems in the setting of property testing and communication complexity. In particular, simulating MA communication complexity protocols by their AM counterparts can also be done while only incurring a quadratic blow-up in complexity, and on the other hand AM protocols can also be exponentially more powerful than MA protocols [38]. In addition, oblivious MA proofs of proximity can be viewed as analogous to *online* MA communication complexity protocols [13].

Comparison with the techniques in [20]. As we discussed above, our MAP lower bound generalizes the main result of Fischer, Goldhirsh, and Lachish [20]. The latter result can be interpreted as an MAP lower bound for any k-wise independent property. Our lower bound extends to a natural generalization of this family. We stress that this extension is *crucial* for our main result, as the permutation property (with respect to which we prove Theorem 1) is *not* k-wise independent, but does satisfy our more general notion.[9]

The proof in [20] is technically quite involved and includes several subtle and non-trivial arguments. For example, while typically property testing lower bounds are shown by exhibiting two distributions that are chosen only as a function of the property, the argument in [20] crucially relies on distributions that are functions of both the property and the description of the specific analyzed algorithm. This entails the usage of several complex mechanisms. For example, they rely on an involved treatment of adaptivity, which consists of procedures for "grafting" decision trees, and use a special type of algorithms (called "readers") that expose low-entropy portions. Perhaps the most significant complication is that their argument uses a delicate information theoretic analysis to handle MAPs that have a two-sided error.

In contrast, our proof is much shorter and consists purely of a combinatorial argument, which does not require any special treatment of adaptivity and two-sided error, and does not use information theory.

1.3 Organization

In Section 2, we introduce the notations and definitions that we use throughout this work. In Section 3, we prove our main technical contribution, which is an MAP lower bound for relaxed k-wise independent properties. Finally, in Section 4, we conclude with a discussion and raise open problems. See full version [31] for the proof of our main result: an exponential separation between MAPs and AMPs, as well as for our results regarding oblivious proofs of proximity.

2 Preliminaries

In this section we establish the definitions and notions that we will need throughout this work.

2.1 Properties and Distance

We focus on testing properties of *functions* and identify a "property" with the set of functions having that property. More accurately, for each $n \in \mathbb{N}$, let D_n and R_n be sets. Let $\mathcal{F}_n$ be the set of functions from D_n to R_n. We define a property as an ensemble $\Pi = \bigcup_n \Pi_n$, where $\Pi_n \subseteq \mathcal{F}_n$ for all n.

For an alphabet Σ, we denote the Hamming distance between two strings $x, y \in \Sigma^n$ by $\Delta(x, y) := |\{x_i \neq y_i : i \in [n]\}|$. If $\Delta(x, y) \leq \varepsilon \cdot n$, we say that x is ε-close to y, otherwise we say that x is ε-far from y. For a non-empty set $S \subseteq \Sigma^n$, we similarly define $\Delta(x, S) := \min_{y \in S} \Delta(x, y)$. Again, if $\Delta(x, S) \leq \varepsilon \cdot n$, we also say that x is ε-close to S and otherwise x is ε-far from S. We extend these definitions to functions by identifying functions with their truth tables (viewed as strings).

[9] Jumping ahead, we remark that our relaxed notion of k-wise independence refers to distributions for which the probability that any subset of k indices is equal to any given sequence of k values is upper bounded by the same probability given the uniform distribution *up to a multiplicative constant* (whereas the standard (i.e., non-relaxed) notion requires exact equality). See further details in Section 3.

Integrality. Throughout this work, for simplicity of notation, we use the convention that all (relevant) integer parameters that are stated as real numbers are implicitly rounded to the closest integer.

2.2 Proofs of Proximity

We recall the definitions of MA and AM proofs of proximity (i.e., MAPs and AMPs), following [35]. Throughout, for an algorithm V we denote by $V^f(n, \varepsilon, w)$ the output of V given oracle access to a function f and explicit access to inputs n, ε, and w; if V is a probabilistic algorithm, we write $\Pr[V^f(n, \varepsilon, w) = z]$ to represent the probability *over the internal randomness of* V that this outcome is z.

▶ **Definition 3** (MAP). A Merlin-Arthur proof of proximity (MAP) for a property $\Pi = \bigcup_n \Pi_n$ consists of a probabilistic algorithm V, called the verifier, that is given as explicit inputs an integer $n \in \mathbb{N}$, a proximity parameter $\varepsilon > 0$, and a proof string $w \in \{0,1\}^*$; in addition, it is given oracle access to a function $f \in \mathcal{F}_n$. The verifier satisfies the following conditions.

1. **Completeness:** For every $n \in \mathbb{N}$ and $f \in \Pi_n$, there exists a string w (the proof) such that for every $\varepsilon > 0$ the verifier accepts with high probability; that is,
$$\Pr\left[V^f(n, \varepsilon, w) = 1\right] \geq \frac{2}{3}.$$
2. **Soundness:** For every $n \in \mathbb{N}$, function $f \in \mathcal{F}_n$, string w, and proximity parameter $\varepsilon > 0$, if f is ε-far from Π_n, then the verifier rejects with high probability; that is,
$$\Pr\left[V^f(n, \varepsilon, w) = 0\right] \geq \frac{2}{3}.$$

A MAP is said to have query complexity $q : \mathbb{N} \times \mathbb{R}^+ \to \mathbb{N}$ if for every $n \in \mathbb{N}, \varepsilon > 0$, $f \in \mathcal{F}_n$, and string $w \in \{0,1\}^*$, the verifier reads at most $q(n, \varepsilon)$ bits in its queries to f. We say that a MAP has proof complexity $p : \mathbb{N} \to \mathbb{N}$ if for every $n \in \mathbb{N}$, there always exists a $w \in \{0,1\}^{p(n)}$ satisfying the conditions of Definition 3. We define the complexity of the MAP to be $t(n, \varepsilon) = q(n, \varepsilon) + p(n)$.

Next, we define AM proofs of proximity (AMPs) similarly to MAPs, except that here the proof is also a function of the inner randomness of the verifier (alternatively, the verifier first sends the prover its entire random string).

▶ **Definition 4** (AMP). An Arthur-Merlin proof of proximity (AMP) for a property $\Pi = \bigcup_n \Pi_n$ consists of a probabilistic algorithm V, called the verifier, that is given as explicit inputs an integer $n \in \mathbb{N}$, a proximity parameter $\varepsilon > 0$, and a proof string w that *depends on the verifier's random string* r, as well as oracle access to a function $f \in \mathcal{F}_n$. The verifier must also be deterministic given the random string r. The protocol satisfies the following conditions.

1. **Completeness:** For every $n \in \mathbb{N}$ and $f \in \Pi_n$,
$$\Pr_r\left[\exists w = w(r) \text{ such that } V^f(n, \varepsilon, w; r) = 1\right] \geq \frac{2}{3}.$$
2. **Soundness:** For every $n \in \mathbb{N}$, function $f \in \mathcal{F}_n$, and proximity parameter $\varepsilon > 0$, if f is ε-far from Π_n, then:
$$\Pr_r\left[\exists w \text{ such that } V^f(n, \varepsilon, w; r) = 1\right] \leq \frac{1}{3}.$$

Analogously to MAPs, an AMP is said to have *query complexity* $q : \mathbb{N} \times \mathbb{R}^+ \to \mathbb{N}$ if for every $n \in \mathbb{N}, \varepsilon > 0$, $f \in \mathcal{F}_n$, and string $w \in \{0,1\}^*$, the verifier reads at most $q(n,\varepsilon)$ bits in its queries to f; and *proof complexity* $p : \mathbb{N} \times \mathbb{R}^+ \to \mathbb{N}$ if for every $n \in \mathbb{N}$ and $f \in \mathcal{F}_n$, with probability at least $\frac{2}{3}$ over coin tosses in the first round, there exists a $w \in \{0,1\}^{p(n,\varepsilon)}$ satisfying the completeness condition of Theorem 4. We define the complexity of the AMP to be $t(n,\varepsilon) = q(n,\varepsilon) + p(n,\varepsilon)$.

We note that we do not include the randomness complexity of the verifier in the complexity of the protocol (although the randomness complexity in all the protocols described in this work is not large). This is a similar choice to what is done in similar contexts such as AM query and communication complexities. Moreover, we show in the full version [31] that if a property Π of functions $f : D_n \to R_n$, such that $|R_n|^{|D_n|} = O(\exp(\text{poly}(n)))$, admits an AMP verifier with query complexity q and proof complexity p, then it also admits an AMP verifier with query complexity $O(q)$, proof complexity $O(p)$, and randomness complexity $O(\log n)$.[10] This transformation is similar to known results of Newman[40] in the context of communication complexity, Goldreich and Sheffet [28] in the context of property testing, and Gur and Rothblum [35] for MAPs. Its main disadvantage however is that it does not preserve the *computational* complexity of the verifier.

3 MAP Lower Bound for (Relaxed) k-wise Independence

In this section we show a general MAP lower bound for a large class of properties. More specifically, we show that any MAP for a (non-degenerate) property that is k-wise independent, must have complexity $\Omega(\sqrt{k})$. By a k-wise independent property we mean that if we sample a random element having the property, than its restriction to any k coordinates looks uniform. As mentioned in the introduction, this generalizes a result due to Fischer *et al.* [20].

We would like to apply this lower bound to the permutation property. However, the permutation property is not k-wise independent and so we cannot apply it directly.[11]. Rather, we give a relaxed notion of k-wise independence that does capture the permutation property and for which we can similarly derive an MAP lower bound.

We proceed to define our relaxed notion of k-wise independence. Recall that we use $\mathcal{F}_n$ to denote the set of all functions from D_n to R_n (see Section 2).

▶ **Definition 5** (Relaxed k-wise Independence). Let $\Pi = \bigcup_{n\geq 1} \Pi_n$ be a property, where $\Pi_n \subset \mathcal{F}_n$ for every n. We say that Π is *relaxed k-wise independent*, for $k = k(n)$, if there exists a constant $C \geq 1$ such that for all positive integers n, all pairwise distinct k-tuples $(i_1, i_2, \ldots, i_k) \in (D_n)^k$ and arbitrary $(t_1, t_2, \ldots, t_k) \in (R_n)^k$, we have that

$$\Pr_{f \in \Pi_n}\Big[f(i_j) = t_j \text{ for all } j \in [k]\Big] \leq \frac{C}{|R_n|^k}. \tag{1}$$

Note that standard definition of a k-wise independence corresponds to the special case of Definition 5 when $C = 1$ (in which case the inequality in Eq. (1) can be replaced with an equality).

At first glance it may seem that the relaxation that we allow in Definition 5 is relatively minor and any lower bound that holds for the full-fledged definition should easily be extendable to our relaxed variant. We argue that it is not the case. For example, in a seminal

[10] For most properties, we have that both the domain and range have size that is polynomial in n. Indeed, the case that $|R_n|^{|D_n|} = \omega(\exp(\text{poly}(n)))$ seems quite pathological.

[11] Indeed, it is not even pairwise independent: the chance of seeing the same element twice is *zero*.

work, Braverman [10] showed that any k-wise independent distribution (for k that is poly-logarithmic) fools AC_0 circuits. Now consider the permutation property (defined formally in the full version [31]) which as noted above is not even pairwise independent but does satisfy our relaxed variant (with $k = \sqrt{n}$). It is not too hard to see that there is a very simple AC_0 circuit for checking whether a function is a permutation: simply by checking whether there exist a pair of entries in the truth table that are identical - thus, our seemingly minor relaxation completely sidesteps Braverman's result. As a matter of fact, a similar situation occurs in the context of AMPs: Rothblum *et al.* [46] showed an AMP lower bound for exact k-wise independent distribution, whereas we show a protocol for the permutation property with logarithmic complexity.

Having defined our notion of relaxed k-wise independence, we proceed to describe a second important condition that we require: namely, that the property is *sparse*, in the sense that a random function is far from the property. Sparsity is essential for our result since there are trivial properties that are k-wise independent but are testable with very few queries (e.g., the property that consists of all functions).

▶ **Definition 6** (Sparse Property). Fix the proximity parameter $\varepsilon = \frac{1}{10}$. We say that a property $\Pi_n = \bigcup_{n\in\mathbb{N}} \Pi_n$ is $t(n)$-*sparse* if:

$$\Pr_{f\in\mathcal{F}_n}[f \text{ is } \varepsilon\text{-far from } \Pi_n] \geq 1 - |R_n|^{-t(n)}.$$

We can now state our main theorem for this section.

▶ **Theorem 7.** *Let Π be a relaxed k-wise independent and k-sparse property. Then, any* MAP *for Π, with respect to proximity parameter $\varepsilon = 1/10$, with proof complexity p and query complexity q must satisfy $p \cdot q = \Omega(k)$.*

The intuition and high level approach for the proof are as follows. First, we use the duality of an MAP as a collection of partial testers [20]. More specifically, the existence of an MAP for a property Π implies that there is some large "sub-property" $\Pi' \subseteq \Pi$ and a *tester* T that distinguishes between inputs in Π' from those that are far from Π.

This simple observation reduces lower bounding MAPs for Π to lower bounding a partial tester for an arbitrary, but large, sub-property. To show such a lower bound, consider the uniform distribution on Π' vs. the uniform distribution over functions that are far from Π. We would like to argue that these two distributions look the same to T, which therefore cannot distinguish between them.

As a matter of fact, we will argue that both these distributions are "close" to being k-wise independent, which suffices as long as k is larger than the tester's query complexity. First, by the sparsity condition we have that the uniform distribution over functions that are far from Π is close to the uniform distribution over *all* functions. Clearly the latter is k-wise independent.

As for the uniform distribution over Π', we would like to argue that since Π' covers a substantial part of Π, which is relaxed k-wise independent, then also Π' is relaxed k-wise independent. The problem with this argument is that Π' only consists of a 2^{-p} fraction of Π, and so it could be quite far from being even relaxed k-wise independent (e.g., it could be that the value of functions in Π' on some fixed elements of R_n is constant over all functions in Π').

This seems like a significant difficulty and was overcome using highly elaborate techniques in [20]. In contrast, we suggest a much simpler argument. The idea is that we first reduce the soundness error of the MAP to $2^{-O(p)}$ by repetition. This increases the query complexity of the tester to $O(p \cdot q)$ but now that the soundness error is so small, that the fact that Π' covers a 2^{-p} fraction of Π is sufficient to make the argument go through.

We proceed to the actual proof.

3.1 Proof of Theorem 7

Let C be a constant such that Π_n satisfies the constraints of Definition 5.

Let V be an MAP verifier, with respect to proximity parameter ε, for Π_n, and denote its proof complexity by p and query complexity by q. Note that any MAP with standard 2/3 completeness and soundness probability (as in Definition 3) can be amplified, via $O(p)$ repetitions, to have completeness and soundness errors $\frac{1}{10C} \cdot 2^{-p}$ at the cost of increasing the query (but not the proof) complexity by a multiplicative factor of $O(p)$, to $O(p \cdot q)$. For concreteness, let us fix a constant C' such that a $(C' \cdot p)$-fold repetition of V has completeness and soundness errors $\frac{1}{10C} \cdot 2^{-p}$ (while having proof complexity p and query complexity $C' \cdot p \cdot q$). Assume towards a contradiction that $p \cdot q \leq \frac{k}{10C'}$.

Recall that for $\Pi' \subseteq \Pi$, a (Π, Π')-partial tester (a notion due to [20]) is a tester that is required to accept functions in the subset Π' and reject functions that are ε-far from the superset Π. As pointed out by Fischer *et al.* [20] an MAP as we assumed above, implies a covering of the property by partial testers as follows. For every possible proof string $w \in \{0,1\}^p$, let

$$S_w = \left\{ f \in \Pi_n \ : \ \Pr\left[V^f(n, \varepsilon, w) = 1\right] \geq 1 - \frac{1}{10C} \cdot 2^{-p} \right\}.$$

By the completeness requirement of an MAP, these sets cover the property Π_n. That is, $\bigcup_w S_w = \Pi_n$.

Since the number of sets S_w is at most 2^p, there exists a proof w that corresponds to a large S_w. Namely, such that $|S_w| \geq |\Pi_n| \cdot 2^{-p}$. We fix such a proof w and argue that the corresponding (Π_n, S_w)-partial tester must make $\Omega(k)$ queries, which would contradict our assumption, thereby proving Theorem 7. Hence, we have reduced proving an MAP lower bound for Π_n to proving a partial testing lower bound for (Π_n, S_w).

Let $V_w^f(n, \varepsilon) := V^f(n, \varepsilon, w)$ be the (S_w, Π_n)-partial tester that is induced by V when we fix the proof string w (and with respect to parameters n and ε). We use the notation $V_w^f(n, \varepsilon; r)$ to denote the *deterministic* output V_w^f when its random string is set to r.

Let $B_\varepsilon = \{f \in \mathcal{F}_n : f \text{ is } \varepsilon\text{-far from } \Pi_n\}$ (i.e., the no-instances). As standard in the property testing literature, we prove a lower bound on the query complexity q' of a tester by presenting a distribution over YES-instances ($f \in S_w$) and a distribution over NO-instances ($f \in B_\varepsilon$) and bounding away from 1 the distinguishing probability for every *deterministic* algorithm making q' queries. Specifically, we give distributions over S_w and B_ε such that any deterministic algorithm making q' queries to f has at most a $1 - \frac{1}{4C} \cdot 2^{-p}$ probability of distinguishing between them, which is sufficient for our purposes. In our case, we simply consider the uniform distributions over S_w and B_ε.

More formally, we first observe that

$$\begin{aligned} &\mathbb{E}_{f \in S_w}\left[\Pr_r\left[V_w^f(n, \varepsilon; r) = 1\right]\right] - \mathbb{E}_{f \in B_\varepsilon}\left[\Pr_r\left[V_w^f(n, \varepsilon; r) = 1\right]\right] \\ &= \mathbb{E}_r\left[\Pr_{f \in S_w}\left[V_w^f(n, \varepsilon; r) = 1\right] - \Pr_{f \in B_\varepsilon}\left[V_w^f(n, \varepsilon; r) = 1\right]\right], \end{aligned} \tag{2}$$

By Eq. (2), it suffices to bound the distinguishing probably for any deterministic verifier. We do this via the following lemma.

▶ **Lemma 8.** *For any deterministic verifier W with query complexity at most $\frac{k}{10}$, we have that*

$$\Pr_{f\in S_w}\left[W^f(n,\varepsilon)=1\right]-\Pr_{f\in B_\varepsilon}\left[W^f(n,\varepsilon)=1\right]\le 1-\frac{1}{4C}\cdot 2^{-p}.$$

Proof. We first show that

$$\Pr_{f\in B_\varepsilon}\left[W^f(n,\varepsilon)=1\right]\ge \frac{1}{2C}\cdot 2^{-p}\cdot \Pr_{f\in S_w}\left[W^f(n,\varepsilon)=1\right]. \tag{3}$$

We can view the verifier W as a decision tree of depth $q'=k/10$. Each leaf of the decision tree is associated with indices $i_1,i_2,\dots,i_{q'}\in D_n$ and values $t_1,t_2,\dots,t_{q'}\in R_n$ such that a function $f\in\mathcal{F}_n$ is accepted at that leaf if and only if $f(i_j)=t_j$ for all $j\in[q']$. We may assume without loss of generality that the sets of indices $i_1,\dots,i_{q'}$ for all paths in the decision tree are pairwise distinct. Fix such a sequence of indices $i_1,\dots,i_{q'}\in D_n$ and values $t_1,\dots,t_{q'}\in R_n$. Then,

$$\begin{aligned}\Pr_{f\in B_\varepsilon}\left[f(i_j)=t_j \text{ for all } j\in[q']\right]&\ge \frac{|\{f\in\mathcal{F}_n: f(i_j)=t_j \text{ for all } j\in[q']\}|-|\mathcal{F}_n\backslash B_\varepsilon|}{|B_\varepsilon|}\\ &\ge \frac{1}{|R_n|^{q'}}-\frac{1}{|R_n|^k-1}\\ &\ge \frac{1}{2|R_n|^{q'}}.\end{aligned} \tag{4}$$

Here we have used k-sparsity to note that $\frac{|\mathcal{F}_n\backslash B_\varepsilon|}{|B_\varepsilon|}\le\frac{1}{|R_n|^k-1}$, and we used that $q'=\frac{k}{10}$. On the other hand, we also have that:

$$\Pr_{f\in S_w}\left[f(i_j)=t_j \text{ for all } j\in[q']\right]\le\frac{\Pr_{f\in\Pi_n}\left[f(i_j)=t_j \text{ for all } j\in[q']\right]}{\Pr_{f\in\Pi_n}[f\in S_w]}\le\frac{C\cdot 2^p}{|R_n|^{q'}} \tag{5}$$

by relaxed q'-wise independence and the lower bound on the size of S_w.

Dividing Eq. (4) by Eq. (5), we obtain that

$$\Pr_{f\in B_\varepsilon}\left[f(i_j)=t_j \text{ for all } j\in[q']\right]\ge\frac{1}{2C}\cdot 2^{-p}\cdot\Pr_{f\in S_w}\left[f(i_j)=t_j \text{ for all } j\in[q']\right].$$

Now, summing the above equation over all leaves of the decision tree corresponding to W (since these correspond to disjoint events) gives us Eq. (3).

Given Eq. (3), we now consider two cases. First, if

$$\Pr_{f\in S_w}\left[W^f(n,\varepsilon)=1\right]\le 1-\frac{1}{2C}\cdot 2^{-p}$$

we are obviously done. Otherwise, we can assume that $\Pr_{f\in S_w}\left[W^f(n,\varepsilon)=1\right]>1-\frac{1}{2C}\cdot 2^{-p}$ and so:

$$\begin{aligned}\Pr_{f\in S_w}\left[W^f(n,\varepsilon)=1\right]-\Pr_{f\in B_\varepsilon}\left[W^f(n,\varepsilon)=1\right]&\le 1-\frac{1}{2C}\cdot 2^{-p}\cdot\Pr_{f\in S_w}\left[W^f(n,\varepsilon)=1\right]\\ &\le 1-\frac{1}{2C}\cdot 2^{-p}\cdot\left(1-\frac{1}{2C}\cdot 2^{-p}\right)\\ &\le 1-\frac{1}{4C}\cdot 2^{-p},\end{aligned}$$

where the first inequality is by Eq. (3). The lemma follows. ◀

Now we are ready to use Lemma 8 to complete our proof of Theorem 7. Because V_w^f has completeness and soundness errors $\frac{1}{10C} \cdot 2^{-p}$, we have that

$$\begin{aligned}\mathbb{E}_{f\in S_w}\left[\Pr_r\left[V_w^f(n,\varepsilon;r)=1\right]\right]-\mathbb{E}_{f\in B_\varepsilon}\left[\Pr_r\left[V_w^f(n,\varepsilon;r)=1\right]\right] &\geq 1-\frac{1}{10C}\cdot 2^{-p}-\frac{1}{10C}\cdot 2^{-p}\\ &= 1-\frac{1}{5C}\cdot 2^{-p}.\end{aligned}$$

On the other hand, by Eq. (2) and Lemma 8, it holds that

$$\mathbb{E}_{f\in S_w}\left[\Pr_r\left[V_w^f(n,\varepsilon;r)=1\right]\right]-\mathbb{E}_{f\in B_\varepsilon}\left[\Pr_r\left[V_w^f(n,\varepsilon;r)=1\right]\right]\leq 1-\frac{1}{4C}\cdot 2^{-p},$$

which is a contradiction. Therefore, we can conclude that $p \cdot q \geq \frac{k}{10C'}$, as desired.

4 Discussion and Open Problems

The complexity of the permutation property for testers, which do not use a proof, is $\tilde{\Theta}(\sqrt{n})$. In this work we showed a lower bound of $\tilde{\Omega}(n^{\frac{1}{4}})$ for MAPs for Perm. Thus, the MAP complexity of Perm is somewhere between $\tilde{\Omega}(n^{\frac{1}{4}})$ and $\tilde{O}(\sqrt{n})$ - resolving the exact complexity is an interesting open problem:

▶ **Problem 9.** *Does every* MAP *for* PERMUTATION *have complexity* $\tilde{\Omega}(\sqrt{n})$*?*

Second, our work shows that AMPs can be exponentially more efficient than MAPs. It is natural to ask whether the converse also holds - can MAPs be much more efficient than AMPs? A partial answer to this question is known. As mentioned in Footnote 5, every MAP with complexity c can be emulated by an AMP with complexity (roughly) c^2.

Thus, MAPs can be at most *quadratically* more efficient than AMPs. However, we do not know a property for which this gap is tight. In particular, the following problem is open:

▶ **Problem 10.** *Does this exist a property* Π *that has an* MAP *with complexity* $O(\sqrt{n})$ *but every* AMP *for* Π *must have complexity* $\Omega(n)$*?*

References

1 Scott Aaronson. Impossibility of succinct quantum proofs for collision-freeness. *Quantum Information & Computation*, 12(1-2):21–28, 2012. URL: http://www.rintonpress.com/xxqic12/qic-12-12/0021-0028.pdf.

2 Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *TOCT*, 1(1):2:1–2:54, 2009. doi:10.1145/1490270.1490272.

3 Amir Abboud, Aviad Rubinstein, and R. Ryan Williams. Distributed PCP theorems for hardness of approximation in P. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 25–36, 2017. doi:10.1109/FOCS.2017.12.

4 Dorit Aharonov and Tomer Naveh. Quantum np-a survey. *arXiv preprint quant-ph/0210077*, 2002.

5 László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 337–347, 1986. doi:10.1109/SFCS.1986.15.

6 László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988. doi:10.1016/0022-0000(88)90028-1.

7 Mihir Bellare and Moti Yung. Certifying permutations: Noninteractive zero-knowledge based on any trapdoor permutation. *J. Cryptology*, 9(3):149–166, 1996.

8 Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust pcps of proximity, shorter pcps, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006. doi:10.1137/S0097539705446810.

9 Itay Berman, Ron D. Rothblum, and Vinod Vaikuntanathan. Zero-knowledge proofs of proximity. *IACR Cryptology ePrint Archive*, 2017:114, 2017. URL: http://eprint.iacr.org/2017/114.

10 Mark Braverman. Poly-logarithmic independence fools bounded-depth boolean circuits. *Commun. ACM*, 54(4):108–115, 2011. doi:10.1145/1924421.1924446.

11 Clément L. Canonne. A survey on distribution testing: Your data is big. but is it blue? *Electronic Colloquium on Computational Complexity (ECCC)*, 22:63, 2015. URL: http://eccc.hpi-web.de/report/2015/063.

12 Amit Chakrabarti, Graham Cormode, Andrew McGregor, and Justin Thaler. Annotations in data streams. *ACM Trans. Algorithms*, 11(1):7:1–7:30, 2014. doi:10.1145/2636924.

13 Amit Chakrabarti, Graham Cormode, Andrew McGregor, Justin Thaler, and Suresh Venkatasubramanian. Verifiable stream computation and arthur-merlin communication. In *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, pages 217–243, 2015. doi:10.4230/LIPIcs.CCC.2015.217.

14 Alessandro Chiesa and Tom Gur. Proofs of proximity for distribution testing. *ECCC*, 24:155, 2017. URL: https://eccc.weizmann.ac.il/report/2017/155.

15 Graham Cormode, Michael Mitzenmacher, and Justin Thaler. Streaming graph computations with a helpful advisor. In *Algorithms - ESA 2010, 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part I*, pages 231–242, 2010. doi:10.1007/978-3-642-15775-2_20.

16 Graham Cormode, Justin Thaler, and Ke Yi. Verifying computations with streaming interactive proofs. *PVLDB*, 5(1):25–36, 2011. URL: http://www.vldb.org/pvldb/vol5/p025_grahamcormode_vldb2012.pdf.

17 Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM J. Comput.*, 36(4):975–1024, 2006. doi:10.1137/S0097539705446962.

18 Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate probabilistically checkable proofs. *Inf. Comput.*, 189(2):135–159, 2004. doi:10.1016/j.ic.2003.09.005.

19 Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs under general assumptions. *IAM J. Comput.*, 29(1), 1999. Preliminary version in *FOCS'90*. doi:10.1137/S0097539792230010.

20 Eldar Fischer, Yonatan Goldhirsh, and Oded Lachish. Partial tests, universal tests and decomposability. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 483–500, 2014. doi:10.1145/2554797.2554841.

21 Eldar Fischer, Oded Lachish, and Yadu Vasudev. Trading query complexity for sample-based testing and multi-testing scalability. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1163–1182, 2015. doi:10.1109/FOCS.2015.75.

22 Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.

23 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. doi:10.1145/285055.285060.

24 Oded Goldreich and Tom Gur. Universal locally testable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:42, 2016. URL: http://eccc.hpi-web.de/report/2016/042.

25 Oded Goldreich and Tom Gur. Universal locally verifiable codes and 3-round interactive proofs of proximity for CSP. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:192, 2016. URL: `http://eccc.hpi-web.de/report/2016/192`.

26 Oded Goldreich, Tom Gur, and Ilan Komargodski. Strong locally testable codes with relaxed local decoders. In *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, pages 1–41, 2015. `doi:10.4230/LIPIcs.CCC.2015.1`.

27 Oded Goldreich, Tom Gur, and Ron D. Rothblum. Proofs of proximity for context-free languages and read-once branching programs - (extended abstract). In *International Colloquium on Automata, Languages and Programming ICALP*, 2015. `doi:10.1007/978-3-662-47672-7_54`.

28 Oded Goldreich and Or Sheffet. On the randomness complexity of property testing. *Computational Complexity*, 19(1):99–133, 2010. `doi:10.1007/s00037-009-0282-4`.

29 Mika Göös, Toniann Pitassi, and Thomas Watson. Zero-information protocols and unambiguity in arthur-merlin communication. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 113–122, 2015. `doi:10.1145/2688073.2688074`.

30 Tom Gur. *On Locally Verifiable Proofs of Proximity.* PhD thesis, Weizmann Institute, 2017.

31 Tom Gur, Yang P. Liu, and Ron D. Rothblum. An exponential separation between ma and am proofs of proximity, 2018. URL: `https://eccc.weizmann.ac.il/report/2018/083/`.

32 Tom Gur and Ran Raz. Arthur-merlin streaming complexity. *Inf. Comput.*, 243:145–165, 2015. `doi:10.1016/j.ic.2014.12.011`.

33 Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. *Computational Complexity*, June 2016.

34 Tom Gur and Ron D. Rothblum. A hierarchy theorem for interactive proofs of proximity. In *Innovations in Theoretical Computer Science ITCS*, 2017.

35 Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. *Computational Complexity*, 27(1):99–207, 2018.

36 Yael Tauman Kalai and Ron D. Rothblum. Arguments of proximity - [extended abstract]. In *CRYPTO*, 2015. `doi:10.1007/978-3-662-48000-7_21`.

37 Hartmut Klauck. Rectangle size bounds and threshold covers in communication complexity. In *18th Annual IEEE Conference on Computational Complexity (Complexity 2003), 7-10 July 2003, Aarhus, Denmark*, pages 118–134, 2003. `doi:10.1109/CCC.2003.1214415`.

38 Hartmut Klauck. On arthur merlin games in communication complexity. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, June 8-10, 2011*, pages 189–199, 2011. `doi:10.1109/CCC.2011.33`.

39 Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992. `doi:10.1145/146585.146605`.

40 Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, 1991. `doi:10.1016/0020-0190(91)90157-D`.

41 Ran Raz and Amir Shpilka. On the power of quantum proofs. In *19th Annual IEEE Conference on Computational Complexity (CCC 2004), 21-24 June 2004, Amherst, MA, USA*, pages 260–274, 2004. `doi:10.1109/CCC.2004.1313849`.

42 Ran Raz, Gábor Tardos, Oleg Verbitsky, and Nikolai K. Vereshchagin. Arthur-merlin games in boolean decision trees. In *Proceedings of the 13th Annual IEEE Conference on Computational Complexity, Buffalo, New York, USA, June 15-18, 1998*, pages 58–67, 1998. `doi:10.1109/CCC.1998.694591`.

43 Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In *Proceedings of the 48th Annual ACM SIGACT Sym-*

posium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016, pages 49–62, 2016. doi:10.1145/2897518.2897652.

44 Dana Ron. Property testing: A learning theory perspective. *Foundations and Trends in Machine Learning*, 1(3):307–402, 2008. doi:10.1561/2200000004.

45 Dana Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends in Theoretical Computer Science*, 5(2):73–205, 2009. doi:10.1561/0400000029.

46 Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In *Symposium on Theory of Computing, STOC*, 2013. doi:10.1145/2488608.2488709.

47 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996. doi:10.1137/S0097539793255151.

48 Alexander A. Sherstov. The multiparty communication complexity of set disjointness. *SIAM J. Comput.*, 45(4):1450–1489, 2016. doi:10.1137/120891587.

49 Justin Thaler. Semi-streaming algorithms for annotated graph streams. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 59:1–59:14, 2016. doi:10.4230/LIPIcs.ICALP.2016.59.