Simple Doubly-Efficient Interactive Proof Systems for Locally-Characterizable Sets*

Oded Goldreich[1] **and Guy N. Rothblum**[†2]

1 **Weizmann Institute of Science, Rehovot, Israel**
oded.goldreich@weizmann.ac.il
2 **Weizmann Institute of Science, Rehovot, Israel**
rothblum@alum.mit.edu

Abstract

A proof system is called doubly-efficient if the prescribed prover strategy can be implemented in polynomial-time and the verifier's strategy can be implemented in almost-linear-time.

We present direct constructions of doubly-efficient interactive proof systems for problems in $\mathcal{P}$ that are believed to have relatively high complexity. Specifically, such constructions are presented for t-CLIQUE and t-SUM. In addition, we present a generic construction of such proof systems for a natural class that contains both problems and is in $\mathcal{NC}$ (and also in $\mathcal{SC}$). The proof systems presented by us are significantly simpler than the proof systems presented by Goldwasser, Kalai and Rothblum (*JACM*, 2015), let alone those presented by Reingold, Rothblum, and Rothblum (*STOC*, 2016), and can be implemented using a smaller number of rounds.

1998 ACM Subject Classification F.0 Theory of Computation: General

Keywords and phrases Interactive Proofs, Fine-Grained Complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.18

1 Introduction

The notion of interactive proof systems, put forward by Goldwasser, Micali, and Rackoff [16], and the demonstration of their power by Lund, Fortnow, Karloff, and Nisan [18] and Shamir [23] are among the most celebrated achievements of complexity theory. Recall that an interactive proof system for a set S is associated with an interactive verification procedure, V, that can be made to accept any input in S but no input outside of S. That is, there exists an interactive strategy for the prover that makes V accepts any input in S, but no strategy can make V accept an input outside of S, except with negligible probability. (See [12, Chap. 9] for a formal definition as well as a wider perspective.)

The original definition does not restrict the complexity of the strategy of the prescribed prover and the constructions of [18, 23] use prover strategies of high complexity. This fact limits the applicability of these proof systems in practice. (Nevertheless, such proof systems may be actually applied when the prover knows something that the verifier does not know, such as an NP-witness to an NP-claim, and when the proof system offers an advantage such as zero-knowledge [16, 13].)

Seeking to make interactive proof systems available for a wider range of applications, Goldwasser, Kalai and Rothblum put forward a notion of *doubly-efficient* interactive proof systems (also called *interactive proofs for muggles* [15] and *interactive proofs for delegating*

* A full version of the paper is available at https://eccc.weizmann.ac.il/report/2017/018/
† This research was supported by the ISRAEL SCIENCE FOUNDATION (grant No. 529/17).

computation [22]). In these proof systems the prescribed prover strategy can be implemented in polynomial-time and the verifier's strategy can be implemented in almost-linear-time. (We stress that unlike in *argument systems*, the soundness condition holds for all possible cheating strategies, and not only for feasible ones.) Restricting the prescribed prover to run in polynomial-time implies that such systems may exist only for sets in $\mathcal{BPP}$, and thus a polynomial-time verifier can check membership in such sets by itself. However, restricting the verifier to run in almost-linear-time implies that something can be gained by interacting with a more powerful prover, even though the latter is restricted to polynomial-time.

The potential applicability of doubly-efficient interactive proof systems was demonstrated by Goldwasser, Kalai and Rothblum [15], who constructed such proof systems for any set that has log-space uniform circuits of bounded depth (e.g., log-space uniform $\mathcal{NC}$). A recent work of Reingold, Rothblum, and Rothblum [22] provided such (constant-round) proof systems for any set that can be decided in polynomial-time and a bounded amount of space (e.g., for all sets in $\mathcal{SC}$).

1.1 The current work

In this work, we aim to develop a more "algorithmic" understanding of doubly-efficient interactive proofs. Studying $\mathcal{BPP}$ problems on a case-by-case basis, our goal is to identify structures and patterns that facilitate the design of efficient proof systems. Towards this goal, our main contributions are *identifying a rich and natural class of polynomial-time computations*, and *constructing far simpler doubly-efficient proof systems for this class*. The aforementioned class consists of all sets that can be locally-characterized by the conjunction of polynomially many local conditions, each of which can be expressed by Boolean formulae of polylogarithmic size (see definition in Section 4.1). The class of locally-characterizable sets is believed not to be in $\text{Dtime}(p)$ for any fixed polynomial p, and contains natural problems of interest such as determining whether a given graph *does not* contain a clique of constant size t. In particular, several problems in this class have played a central role in the recent theory of "hardness within $\mathcal{P}$" [27]. We note that the class of locally-characterizable sets is a sub-class of $\mathcal{NC} \cap \mathcal{SC}$, yet the interactive proofs we present are significantly simpler than those in [15, 22].

▶ **Theorem 1** (main result, loosely stated)**.** *Every locally-characterizable set has a simple doubly-efficient interactive proof system. Specifically, on input of length n, the verifier runs in $\widetilde{O}(n)$-time and the strategy of the prescribed prover can be implemented in $\widetilde{O}(n^{c+1})$-time, where n^c denotes the number of local conditions in the characterization. Furthermore, the interactive proof system uses public coins, has logarithmic round-complexity, and uses polylogarithmic communication.*

Studying proof systems for locally-characterizable sets has also shed light on the complexity of problems in this class. In our subsequent work [14], building on the work of [5], we leverage techniques developed in the current work to show worst-case to average-case reductions between problems in a closely related class.[1] These reductions run in nearly-linear time.

[1] The class we study in [14] contains problems that *count* the number of local conditions that are violated by the input, rather than simply checking their conjunction. The interactive proofs we construct in this work can readily be modified to verify these "local-counting" problems, see Remark 4. In addition, the size of formulae considered in [14] is slightly larger, but the interactive proofs presented here apply to this too (except that the verification time becomes slightly larger (i.e., $2^{\widetilde{O}(\log \log n)} \cdot n$ rather than $\widetilde{O}(n)$)).

Thus, if the class contains problems that are hard to solve in the worst case, then it also contains problems that are hard to solve on-average.

High-level overview of our proof systems.

Analogously to [18, 23], our first step is recasting membership in a locally-characterizable set as an algebraic problem. Specifically, the algebraic problem consists of computing the sum of polynomially many evaluations of a low-degree polynomial, where the particular polynomial is derived from the description of the locally-characterizable set. The interactive proof uses the sum-check protocol [18] to verify the correctness of the sum. We stress that we check a sum with polynomially many terms, and so the sum-check protocol uses logarithmically many rounds, whereas the celebrated results of [18, 23] deal with a sum having exponentially (or more) many terms, and so the sum-check protocol requires a linear (or more) number of rounds.

1.1.1 Extensions

Round/communication trade-offs

Using a different setting of parameters, the interactive proofs constructed in Theorem 1 can also give new trade-offs between the number of rounds and communication/verification complexity (i.e., we obtain $O(r)$-round doubly-efficient interactive proofs with $\widetilde{O}(n^{1+(1/r)})$ verification time). In particular, the number of rounds can be significantly smaller than the protocols of [15, 22] (see Theorem 5 and the comparison in Section 1.2).

A richer class

Theorem 1 can be extended to a richer class of sets. This extended class also checks a polynomial number of local conditions, but rather than checking their conjunction, we allow a constant number of alternations between conjunctions and disjunctions. For example, this extended class contains the problem of determining whether a given graph contains no dominating set of constant size t (i.e., whether for every set D of t vertices, there exists a vertex v that is not adjacent to at least one member of D). This extension corresponds to the levels of a known hierarchy of *parameterized complexity classes* [11]. See the discussion following Remark 8.

This non-trivial extension is our most technically involved contribution. The proof system we construct leverages ideas from the proof of Toda's Theorem [26], where we need to take care and use a small bias space with additional algebraic structure. See the details and digest in Section 4.4.

1.1.2 Two cases of interest

We also present direct constructions of proof systems for verifying two particularly appealing t-parameterized problems which are locally-characterizable. Both of these problems are widely believed not to be in, say, Dtime($n^{t/5}$):[2]

[2] Currently, t-CLIQUE is conjectured to require time $n^{c' \cdot t}$, where c' is any constant smaller than one third of the Boolean matrix multiplication exponent (see, e.g., [1]). Recall that t-CLIQUE is $\mathcal{W}[1]$-complete [10] and that solving it in time $n^{o(t)}$ refutes the ETH [9]. The 3-SUM conjecture was popularized in [20], and lower bounds for t-SUM were shown to follow from lower bounds for t-CLIQUE (see [2]).

t-no-CLIQUE: The set of n-vertex graphs that do not contain a clique of size t.

t-no-SUM: The set of n-long sequences of integers that contain no t elements that sum-up to zero.[3]

(Indeed, the aforementioned sets are the complements of the NP-sets called t-CLIQUE and t-SUM.)

We present direct constructions of proof systems for these two sets. The corresponding prover strategies can be implemented in time $\widetilde{O}(n^t)$ and $\widetilde{O}(n^{t+1})$, resp. Although the generic construction for the class of "locally-characterizable" sets, which is presented in Section 4.2, almost meets the parameters of constructions tailored for t-no-CLIQUE and t-no-SUM, we believe that direct constructions for natural problems are of interest. In particular, the construction tailored for t-no-CLIQUE, which is presented in Section 3, is simpler and more efficient than the general construction. It can also be viewed as a warm-up for the more general result. The construction tailored for t-no-SUM makes use of technical ideas that may be of independent interest, it is presented in Section 5 (see the digest at the end of that section).

1.2 Relation to prior work

We first compare our results to the interactive proofs obtained by [15, 22], as well as an interactive proof system of Thaler [25] for counting the number of t-cliques in a graph. We then discuss the relationship to the non-interactive proofs presented in [8, 28, 7].

Complexity comparison to [15, 22]

On top of yielding simpler proof systems, the round complexity of our system is smaller than those of [15, 22]. This is most striking in the case of [15], which uses a protocol of $O(\log^2 n)$ rounds (whereas Theorem 1 uses $O(\log n)$ rounds). As for [22], its constant round-complexity is exponential in the degree of the polynomial bounding the complexity of the set, whereas the constant round-complexity in Theorem 5 is linear in the latter constant.

Comparison to [25]

The recent work [25] gives an interactive proof system for counting the number of t-cliques in a graph.[4] His system uses a constant number of rounds, with $\widetilde{O}(n)$ communication, $O(|E|+n)$ verification time, and $O(|E| \cdot n^{t-2})$ proving time. The system that we present for the t-no-CLIQUE problem can also be used to verify the number of t-cliques (see Remark 4). In that system the verification time is $\widetilde{O}(|E|+n)$, the number of rounds is logarithmic, the communication is polylogarithmic, and the prover work is $\widetilde{O}(n^t)$. As remarked above, we can also trade off the communication complexity and the number of rounds (see Section 4.3, where this is done for the generic protocol for "locally-characterizable" sets). That would result in parameters similar to those obtained in [25], except that the prover complexity is $\widetilde{O}(n^t)$ rather than $O(|E| \cdot n^{t-2})$.

---

[3] Alternatively, we may consider sequences of positive integers and ask if they contain a t-subset that sums-up to target integer, which is also given as part of the input.

[4] We remark that the protocol of [25] operates in a more challenging streaming setting, which we do not consider or elaborate on in this work.

Comparison to [8, 28, 7]

Several recent works [8, 28, 7] constructed *non-interactive* proof systems for problems in $\mathcal{P}$. The main distinction with our work is that we focus on *interactive* proofs, and obtain proof systems with *faster verification* (and smaller amounts of communication). Similarly to our work, the proofs in the systems of [8, 28] can be produced in polynomial-time.

Carmosino *et al.* [8] construct $\mathcal{NP}$ certificates for 3-no-SUM. The certificates are of length $\widetilde{O}(n^{1.5})$ and can be verified in deterministic $\widetilde{O}(n^{1.5})$-time. Our proof systems for this problem are interactive, and the verification is probabilistic, but the communication is only polylogarithmic, and the verifier's work is almost-linear (indeed, this remains true for the t-no-SUM problem, for any constant t).

Williams [28] constructs $\mathcal{MA}$ proof systems for problems in $\mathcal{P}$. An $\mathcal{MA}$ proof system is one in which the prover sends a single message to the verifier, who runs a probabilistic verification procedure. Focusing on problems that have been at the center of a recent theory of "hardness within $\mathcal{P}$" [27], he constructs $\mathcal{MA}$ proof systems for counting the number of orthogonal vectors within a collection of n input vectors and for counting the number of t-cliques in a given graph. The $\mathcal{MA}$ proof for counting the number of t-cliques has length and verification time $\widetilde{O}(n^{\lfloor t/2 \rfloor + 2})$. Björklund and Kaski [7] construct an $\mathcal{MA}$ proof with length and verification time $\widetilde{O}(n^{(\omega+\epsilon)t/6})$, where ω is the exponent of square matrix multiplication over the integers and $\epsilon > 0$ is an arbitrarily small constant. The time to construct their proof is $\widetilde{O}(n^{(\omega+\epsilon)t/3})$, matching the best sequential algorithm known for solving the problem. Comparing these works with our *interactive* proof for t-no-CLIQUE (which can also be used to verify the number of cliques), the interactive proof has poly-logarithmic communication, the verifier's work is almost-linear, and the prover's work is $\widetilde{O}(n^t)$.

Inspired by Williams [28] and using one of his results, we also present an $\mathcal{MA}$ proof system of verification complexity $\widetilde{O}(n^{(c+1)/2})$ for every locally-characterizable set (i.e., the class considered in Theorem 1). This proof system is presented in the appendix. For a more restricted subclass of locally-characterizable sets, which includes the c-no-CLIQUE problem, we construct an $\mathcal{MA}$ proof with improved length and verification time $\widetilde{O}(n^{c/2})$.

Relation to [6]

Subsequently to our work, Ball *et al.* [6] propose an *interactive* proof system for the generalized orthogonal vectors problem (a problem in $\mathcal{BPP}$), where the verification can be performed in nearly-linear time. They note that their approach can be extended to other problems studied in [5], as well as to our notion of locally-characterizable sets and a more general algebraic class of problems. We remark that the generalized orthogonal vectors problem lies in the class of locally-characterizable sets, and thus our general construction also applies to this problem, and achieves similar performance to the protocol of [6]. We note also that their earlier independent work [5] constructs (building on [28]) an $\mathcal{MA}$ proof system for the generalized orthogonal vectors problem, with polynomial communication and verification time.

1.3 Organization and conventions

Section 3 presents our simplest proof system, which is for t-no-CLIQUE. Our generic construction for any locally-characterizable set is presented in Section 4: The corresponding class is defined in Section 4.1, the basic construction is in Section 4.2, extensions and ramifications are in Sections 4.3 and 4.4. We present our proof system for t-no-SUM in Section 5. This

proof system is not significantly simpler than the generic construction but uses an idea that may be of independent interest.

Sections 3, 4 and 5 can be read independently of one another, and without reading Section 2 in which we merely review the celebrated sum-check protocol. (We assume that the reader is familiar with the definition of an interactive proof system.) Brief conclusions are in Section 6.

Conventions

We assume that the verifier (resp., prover) has *direct access* to the common input; that is, each bit in the input can be read in unit cost. Unless stated explicitly differently, all logarithms are to base 2.

2 Preliminaries: The sum-check protocol

Fixing a finite field $\mathcal{F}$ and a set $H \subset \mathcal{F}$ (e.g., H may consist of the 0 and 1 elements of $\mathcal{F}$), we consider an m-variate polynomial $P : \mathcal{F}^m \to \mathcal{F}$ of individual degree d. Given a value v, the sum-check protocol is used to prove that

$$\sum_{\sigma_1,...,\sigma_m \in H} P(\sigma_1, ..., \sigma_m) = v. \tag{1}$$

The sum-check protocol (of [18]) proceeds in m iterations, starting with $v_0 = v$, such that in the i^{th} iteration the parties act as follows.

Prover's move: The prove computes a univariate polynomial of degree d

$$P_i(z) \stackrel{\text{def}}{=} \sum_{\sigma_{i+1},...,\sigma_m \in H} P(r_1, ..., r_{i-1}, z, \sigma_{i+1}, ..., \sigma_m) \tag{2}$$

where $r_1, ..., r_{i-1}$ are as determined in prior iterations, and sends P_i to the verifier.

Verifier's move: The verifier checks that $\sum_{\sigma \in H} P_i(\sigma) = v_{i-1}$ and rejects if inequality holds.

Otherwise, it selects r_i uniformly in $\mathcal{F}$, and sends it to the prover, while setting $v_i \leftarrow P_i(r_i)$. If all iteration are completed, the verifier conducts a final check. It computes the value of $P(r_1, ..., r_m)$ and accepts if and only if this value equals v_m.

Clearly, if (1) holds (and the prover acts according to the protocol), then the verifier accepts with probability 1. Otherwise, no matter what the prover does, the verifier accepts with probability at most $m \cdot d/|\mathcal{F}|$. The complexity of verification is dominated by the complexity of evaluating P (on a single point). As for the prescribed prover, it may compute the relevant P_i's by interpolation, which is based on computing the value of P at $(d+1) \cdot 2^{m-i}$ points, for $i \in [m]$.

3 The case of t-CLIQUE

For a parameter $t \in \mathbb{N}$, given a graph $G = ([n], E)$, the task is determining whether there exist t vertices such that the subgraph induced by them is a clique; that is, does there exist distinct $v_1, ..., v_t \in [n]$ such that $\bigwedge_{j,k \in [t]: j<k} \chi_{v_j,v_k}$, where $\chi_{u,v} = 1$ if and only if $\{u, v\} \in E$. (Our focus is on simple graphs, and so we assume that $\chi_{v,v} = 0$ for every $v \in [n]$.)

The set of YES-instances (i.e., having a t-clique) has an NP-proof system that uses proofs of length $t \log n$. We shall present a doubly-efficient interactive proof for the set of NO-instances. This system is based on the observation that membership of an n-vertex

graph in t-no-CLIQUE can be captured by an explicit low degree polynomial having n^t terms. Furthermore, each term of this polynomial can be evaluated in almost-linear time (in the size of the graph). Hence, applying the Sum-Check protocol (reviewed in Section 2) yields the desired proof system.

Letting $\ell = \log n$, consider a finite field $\mathcal{F}$ of prime size greater than n^t, and identify $\{0,1\}$ with the set H containing the zero and one elements of $\mathcal{F}$. Using this identification, we define a polynomial $P : (\mathcal{F}^\ell)^t \to \mathcal{F}$ such that

$$P(z^{(1)},...,z^{(t)}) = \prod_{j,k\in[t]:j<k} \sum_{\alpha,\beta\in H^\ell} \mathtt{EQ}(\alpha\beta, z^{(j)}z^{(k)}) \cdot \chi_{\alpha,\beta} \tag{3}$$

$$\text{where } \mathtt{EQ}(\overline{\gamma},\overline{z}) = \textstyle\prod_{i\in[2\ell]}(z_i\gamma_i + (1-z_i)(1-\gamma_i)). \tag{4}$$

(There is some abuse of notation in (3): In the first two occurrences, α and β are viewed as an elements of $H^t \subset \mathcal{F}^\ell$, whereas in the last occurrence they viewed as elements of $[n] \equiv \{0,1\}^\ell$.)

Note that P has individual degree $O(t^2)$, and that it is straightforward to evaluate it in time $O(t^2 \cdot 2^{2\ell} \cdot \ell) = \widetilde{O}(t^2 \cdot n^2)$. Also, for $\overline{\gamma},\overline{z} \in H^{2\ell}$, it holds that $\mathtt{EQ}(\overline{\gamma},\overline{z}) = 1$ if $\overline{\gamma} = \overline{z}$ and $\mathtt{EQ}(\overline{\gamma},\overline{z}) = 0$ otherwise. Hence, for $\overline{z} \in H^{t\ell}$, it holds that $P(\overline{z}) = \prod_{j,k\in[t]:j<k} \chi_{z^{(j)},z^{(k)}}$.

The key observation is that the graph G is a NO-instances if and only if for all $\overline{z} \in (H^\ell)^t$ it holds that $P(\overline{z}) = 0$. (This holds since, for distinct $v^{(1)},...,v^{(t)} \in H^\ell$, it holds that $P(v^{(1)},...,v^{(t)}) = 1$ if the subgraph induced by $v^{(1)},...,v^{(t)}$ is a clique, and $P(v^{(1)},...,v^{(t)}) = 0$ otherwise.)[5] Hence, *we obtain an interactive proof system* (for the set of NO-instances) *by applying the sum-check protocol to the claim* $\sum_{\overline{z}\in(H^\ell)^t} P(\overline{z}) = 0$.

The complexity of the verifier's strategy is dominated by the evaluation of P (as defined in (3)), which reduces to $\binom{t}{2}$ computations of sums having the form

$$\sum_{\alpha,\beta\in H^\ell} \mathtt{EQ}(\alpha\beta, r^{(j)}r^{(k)}) \cdot \chi_{\alpha,\beta} \tag{5}$$

where $j < k \in [t]$ and $r^{(1)} \cdots r^{(t)} \in (\mathcal{F}^\ell)^t$ are determined in the execution of the sum-check protocol. Writing (5) as $\sum_{(\alpha,\beta)\in H^{2\ell}\cap E} \mathtt{EQ}(\alpha\beta, r^{(j)}r^{(k)})$, we can evaluate this sum in time $O(|E| \cdot \ell) = \widetilde{O}(|E| + n)$.

Turning our attention to the complexity of proving, we observe that the prover has to compute the polynomials that arise in each of the iterations of the sum-check protocol. The prover can do so by computing partial sums with at most $|H|^{t\ell} = n^t$ terms, where computing each such term amounts to $\mathrm{poly}(t)$ evaluations of P. Hence, the prover's complexity is definitely bounded by $\widetilde{O}(n^t \cdot |E|)$. A closer inspection reveals that we can do better. Specifically, in the i^{th} iteration of the sum-check, the prover has to provide the univariate polynomial (in z)

$$\sum_{\overline{s}\in H^{t\ell-i}} P(\overline{r}, z, \overline{s}) \tag{6}$$

where $\overline{r} \in \mathcal{F}^{i-1}$ was determined in the previous iteration. This univariate polynomial can be found by interpolation, and so the complexity of finding it is $\mathrm{poly}(t) \cdot 2^{t\ell-i} \cdot O(|E| \cdot \ell)$, which is $\widetilde{O}(n^t)$ for $i \geq 2\ell$. Thus, we focus on the case of $i \in [2\ell - 1]$. In that case, the complexity is $n^t/2^i$ times the complexity of evaluating P on $\overline{r}u\overline{s}$, where $(\overline{r},u) \in \mathcal{F}^{i-1} \times \mathcal{F}$ and $\overline{s} \in H^{t\ell-i}$ (for $O(t^2)$ values of $u \in \mathcal{F}$). So the question is what is the complexity of evaluating P on such a $t\ell$-element argument (which has a $(t\ell - i)$-long suffix in $H^{t\ell-i}$).

[5] If $v^{(j)} = v^{(k)}$ for some $j < k$, then $P(v^{(1)},...,v^{(t)}) = 0$, since $\chi_{v^{(j)},v^{(k)}} = 0$.

Focusing on evaluating each of the inner sums (captured by (5)), we consider evaluating the sum $\sum_{\alpha,\beta\in H^\ell} \mathtt{EQ}(\alpha\beta, v)\cdot\chi_{\alpha,\beta}$, when given $v = uw \in \mathcal{F}^{2\ell}$ such that $w \in H^{2\ell-p}$, where p is determined by i and j,k (indeed, $p = i \in [2\ell]$ if $(j,k) = (1,2)$, but $p < i$ otherwise). Letting $\chi_\gamma = \chi_{\alpha,\beta}$, where $\gamma = \alpha\beta$ such that $|\alpha| = |\gamma|/2$, we have

$$\begin{aligned}\sum_{\gamma\in H^{2\ell}} \mathtt{EQ}(\gamma, uw)\cdot\chi_\gamma &= \sum_{(\gamma',\gamma'')\in H^p\times H^{2\ell-p}} \mathtt{EQ}(\gamma', u)\cdot\mathtt{EQ}(\gamma'', w)\cdot\chi_{\gamma'\gamma''}\\ &= \sum_{\gamma'\in H^p} \mathtt{EQ}(\gamma', u)\cdot\chi_{\gamma' w}\end{aligned}$$

where the second equality holds because for $\gamma'', w \in H^{2\ell-p}$ it holds that $\mathtt{EQ}(\gamma'', w) = 1$ if $\gamma'' = w$ and $\mathtt{EQ}(\gamma'', w) = 0$ otherwise. Hence, evaluating this sum takes time $O(2^p\cdot\ell)$. The final observation is that, in our application, it holds that $p \le i$, since the p values in $\mathcal{F}\setminus H$ can only arise from the values determined in the previous $i-1$ iterations and the single value used for interpolation in the current iteration. It follows that the complexity of implementing the prover's strategy in the i^{th} iteration is $\mathrm{poly}(t)\cdot 2^{t\ell-i}\cdot O(2^p\cdot\ell) = \widetilde{O}(n^t)$.

4 The general result

In this section we prove Theorem 1; that is, we present a simple doubly-efficient interactive proof system for any locally-characterizable set. The latter class is defined next.

4.1 A natural class: locally-characterizable sets

The following definition is related but different from the definition of "local characterization" that is often used in the property testing literature (see, Sudan's survey [24]). Most importantly, the latter definitions do not specify the complexity of the functions ϕ_n and $\pi_{n,j}$, and typically take p to be a constant.[6]

▶ **Definition 2** (locally-characterizable sets)**.** A set S is locally-characterizable if there exist a constant c, a polynomial p and a polynomial-time algorithm that on input n outputs $\mathrm{poly}(\log n)$-sized formulae $\phi_n : [n]^{p(\log n)}\times\{0,1\}^{p(\log n)}\to\{0,1\}$ and $\pi_{n,1},...,\pi_{n,p(\log n)} : \{0,1\}^{c\log n}\to[n]$ such that, for every $x\in\{0,1\}^n$, it holds that $x\in S$ if and only if for all $w\in\{0,1\}^{c\log n}$

$$\Phi_x(w) \stackrel{\text{def}}{=} \phi_n(\pi_{n,1}(w),...,\pi_{n,p(\log n)}(w), x_{\pi_{n,1}(w)},...,x_{\pi_{n,p(\log n)}(w)}) \tag{7}$$

equals 0.[7]

That is, each value of $w\in\{0,1\}^{c\log n}$ yields a local condition that refers to polylogarithmically many locations in the input (i.e., the locations $\pi_{n,1}(w),...,\pi_{n,p(\log n)}(w)\in[n]$). This local condition is captured by ϕ_n, and in its general form it depends both on the selected locations and on the value on the input in these locations. A simplified form, which suffices in many case, uses a local condition that only depends on the values of the input in these locations (i.e., $\phi_n : [n]^{p(\log n)}\times\{0,1\}^{p(\log n)}\to\{0,1\}$ only depends on the $p(\log n)$-bit suffix).

[6] In addition, the notion used in property testing does not restrict the domain of Φ_x to have size $\mathrm{poly}(|x|)$, although this can be assumed without loss of generality.

[7] To simplify our exposition, we require that in case of inputs in S, the predicate ϕ_n evaluates to 0 (rather than to 1).

The simplified form (in which $\phi_n : \{0,1\}^{p(\log n)} \to \{0,1\}$) suffices for capturing the specific problems studied in the previous two sections. Specifically, for fixed $t \in \mathbb{N}$, when representing n-vertex graphs by their adjacency matrix, denoted $x = (x_{r,c})_{r,c\in[n]}$, the t-CLIQUE problem is captured by $\Phi_x(i_1,...,i_t) = \wedge_{j<k} x_{i_j,i_k}$; that is, we use $\phi_{n^2} : \{0,1\}^{t^2} \to \{0,1\}$ and $\pi_{n^2,(j,k)} : \{0,1\}^{t\log n} \to [n^2]$ (for $j,k \in [t]$) such that $\phi_{n^2}(\sigma_{1,1},...,\sigma_{t,t}) = \wedge_{j<k}\sigma_{j,k}$ and $\pi_{n^2,(j,k)}(i_1,...,i_t) = (i_j,i_k)$. Likewise, with some abuse of notation, the t-SUM problem over $[m]$, where $x = (a_1,...,a_n,b) \in [m]$ (and $m = \mathrm{poly}(n)$), is captured by $\Phi_x(i_1,...,i_t) = 1$ if and only if $\sum_{j\in[t]} x_{i_j} = x_{n+1}$; that is, we use $\phi_n : [m]^{t+1} \to \{0,1\}$ such that $\phi_n(z_1,...,z_t,z_{t+1}) = \mathtt{TruthValue}(\sum_{j\in[t]} z_j \neq z_{t+1})$ and $\pi_{n,j} : \{0,1\}^{t\log n} \to [n+1]$ such that $\pi_{n,j}(i_1,...,i_t) = i_j$ if $j \in [t]$ and $\pi_{n,t+1}(i_1,...,i_t) = n+1$.

Note that the complement of every locally-characterizable set has a doubly-efficient interactive proof system. In this proof system, on input $x \in \{0,1\}^n$, letting $\ell' = c\ell = c\log n$, the prover finds an adequate $w \in \{0,1\}^{\ell'}$, sends it to the verifier, who retrieves the bits $x_{\pi_{n,1}(w)},...,x_{\pi_{n,p(\ell)}(w)}$ and evaluates ϕ_n on them. Indeed, in this NP-proof system, the prover runs in time $2^{\ell'}\cdot\widetilde{O}(|x|) = \mathrm{poly}(|x|)$, whereas the verifier runs in $\mathrm{poly}(\log|x|)$-time (given direct access to the input). On the other hand, the set of YES-instances of this set has a doubly-efficient interactive proof systems (since computing the function $\sum_{w\in\{0,1\}^{\ell'}}\Phi_x(w)$, where Φ_x is as in (7), is in $\mathcal{NC}$, and so the proof system of [15] can be used).[8] Here we present a direct construction.

4.2 Proof of Theorem 1

Letting $\ell = \log n$, we associate $[n]$ with $\{0,1\}^\ell$, and derive from each $\pi_{n,j} : \{0,1\}^{c\ell} \to [n]$ Boolean formulae $\pi_{n,j,1},...,\pi_{n,j,\ell} : \{0,1\}^{c\ell} \to \{0,1\}$ such that $\pi_{n,j,k}(w)$ is the k^{th} bit of $\pi_{n,j}(w)$. We may assume, without loss of generality, that the depth of each of the formulae (i.e., ϕ_n and the $\pi_{n,j,k}$'s) is logarithmic in their size (which is $\mathrm{poly}(\ell)$).[9] Next, for a finite field $\mathcal{F}$ of size $\mathrm{poly}(n)$, we construct arithmetic formula $\widehat{\phi}_n : \mathcal{F}^{(\ell+1)\cdot p(\ell)} \to \mathcal{F}$ and $\widehat{\pi}_{n,j,k} : \mathcal{F}^{c\ell} \to \mathcal{F}$ such that $\widehat{\phi}_n$ (resp., $\widehat{\pi}_{n,j,k}$) agrees with ϕ_n on $H^{\ell\cdot p(\ell)+p(\ell)}$ (resp., with $\pi_{n,j,k}$ on $H^{c\ell}$). The crucial point is that these arithmetic formulae preserve the depth of the Boolean counterparts, and so the degrees of the functions that they compute is upper bounded by $D = \mathrm{poly}(\log n) \ll |\mathcal{F}|$. (Note: $\mathcal{F}$ is chosen so that the latter inequality holds.) Now, letting $\widehat{\pi}_{n,j} : \mathcal{F}^{c\ell} \to \mathcal{F}^\ell$ such that $\widehat{\pi}_{n,j}(\overline{z}) = (\widehat{\pi}_{n,j,1}(\overline{z}),...,\widehat{\pi}_{n,j,\ell}(\overline{z}))$, we consider the function $\widehat{\Phi}_x : \mathcal{F}^{c\ell} \to \mathcal{F}$ (i.e., an extension of Φ_x) such that

$$\widehat{\Phi}_x(\overline{z}) \stackrel{\text{def}}{=} \widehat{\phi}_n(\widehat{\pi}_{n,1}(\overline{z}),...,\widehat{\pi}_{n,p(\ell)}(\overline{z}),X_1,...,X_{p(\ell)}) \tag{8}$$

$$\text{where } X_i = \textstyle\sum_{\alpha\in\{0,1\}^\ell} \mathtt{EQ}(\widehat{\pi}_{n,i}(\overline{z}),\alpha)\cdot x_\alpha \tag{9}$$

and, as in (4), $\mathtt{EQ}(\overline{y},\alpha) = \prod_{i\in[\ell]}(y_i\alpha_i + (1-y_i)(1-\alpha_i))$. That is, the value of $\widehat{\Phi}_x(\overline{z})$ is obtained by feeding to $\widehat{\phi}_n : \mathcal{F}^{(\ell+1)\cdot p(\ell)} \to \mathcal{F}$ the $(p(\ell)\cdot\ell + p(\ell))$-sequence consisting of $(\widehat{\pi}_{n,1}(\overline{z}),...,\widehat{\pi}_{n,p(\log n)}(\overline{z})) \in (\mathcal{F}^\ell)^{p(\ell)}$ and the $p(\ell)$-long sequence whose j^{th} location contains the field element $\sum_{\alpha\in\{0,1\}^\ell}\mathtt{EQ}(\widehat{\pi}_{n,j}(\overline{z}),\alpha)\cdot x_\alpha$.

We invoke the sum-check protocol on the claim that the sum $\sum_{w\in\{0,1\}^{c\ell}}\widehat{\Phi}_x(w)$ equals 0, relying on the fact that $\mathcal{F}$ is larger than $2^{c\ell}$. This protocol performs $c\ell$ iterations (i.e., it is applied only to the outer sum), and the verifier evaluates the residual expression, which

[8] Alternatively, one observes that this computation is in $\mathcal{SC}$ and use the proof system of [22].

[9] Note that the transformation to this form can be performed in polynomial time, whereas the relevant formulae are of $\mathrm{poly}(\log n)$-size.

amounts to evaluating the $\widehat{\pi}_{n,j}$'s and $\widehat{\phi}_n$ as well as computing $\text{poly}(\ell)$ sums of 2^ℓ terms each. The prover's computation is dominated by computing a sum of $2^{c\ell}$ terms, where each term requires a computation of the type conducted by the verifier. Recalling that $2^\ell = n$, it follows that the verifier runs in almost-linear-time (i.e., it runs in $\widetilde{O}(n)$-time), whereas the prover runs in polynomial-time (i.e., it runs in $\widetilde{O}(n^{c+1})$-time). ◀

Comment

Applied to the t-no-CLIQUE problem, the foregoing generic construction yields a verifier that runs in time that is almost-linear in the size of the adjacency matrix, whereas the tailored proof system (presented in Section 3) yields a verifier that runs in time that is almost linear in the number of edges. Furthermore, the prover's complexity in this generic construction is n times slower than that of the tailored proof system. Looking ahead, we note that when applied to the t-no-SUM problem, the generic construction yields a proof system of complexity that is comparable to that of the tailored proof system presented in Section 5.

▶ **Remark 3** (a relaxation of Definition 2)**.** *Theorem 1 holds also when relaxing the notion of locally-characterizable sets such that the* $\text{poly}(\log n)$*-sized formulae are required to be generated in* $\widetilde{O}(n)$*-time, rather than in* $\text{poly}(\log n)$*-time. Actually, the foregoing proof remains intact even if the said formulae may depend on* x *itself, but we consider the class as defined in Definition 2 more natural.*

▶ **Remark 4** (counting the number of violated condition)**.** *The interactive proof system presented above is applicable also to the task of verifying the number of violated conditions. The same applies also to the problem-tailored proof systems presented in Section 3 and (looking ahead) in Section 5. Note that the number of violated conditions in* t-no-CLIQUE *is the number of* t*-cliques in the graph. Similarly, the number of violated conditions for* t-no-SUM *is the number of* t*-tuples that sum to the target.*

4.3 Generalization: round versus computation trade-off

By using a set H of arbitrary size (rather than $H \equiv \{0,1\}$), we obtain a general trade-off between the computational complexity and the number of communication rounds. Specifically, the computational complexity increases by a factor of $\widetilde{O}(|H|)$, whereas the number of rounds is decaresed by a factor of $\log |H|$.

▶ **Theorem 5** (main result, restated)**.** *Every locally-characterizable set has a simple doubly-efficient interactive proof system. Specifically, on input of length* n *and using a parameter* $h \leq n$*, we get public-coin interactive proof systems of round-complexity* $O(\log_h n)$*, verification time* $\widetilde{O}(h \cdot n)$*, and proving time* $\widetilde{O}(h \cdot n^{c+1})$*.*

In particular, setting $h = n^\epsilon$ for any constant $\epsilon > 0$, yields a constant-round interactive proof of verification time $\widetilde{O}(n^{1+\epsilon})$. On the other hand, using $h = \log n$ maintains the computational complexity bounds of Theorem 1 (i.e., $\widetilde{O}(n)$-time verification and $\widetilde{O}(n^{c+1})$-time prover strategy) while using $o(\log n)$ rounds of communication.

Proof. We use $H = [h]$ but maintain $\ell = \log n$. Defining the $\pi_{n,j,k}$'s as above and letting $d = \log h$, we associate $\{0,1\}^{c\ell}$ with $H^{c\ell/d}$ and consider $\pi_{n,j,k} : H^{c\ell/d} \to \{0,1\}$. Now, we let $\widehat{\pi}_{n,j,k} : \mathcal{F}^{c\ell/d} \to \mathcal{F}$ be the low degree extension of $\pi_{n,j,k}$, and define $\widehat{\pi}_{n,j} : \mathcal{F}^{c\ell/d} \to \mathcal{F}^\ell$ as

before. The definition of $\widehat{\Phi}_x : \mathcal{F}^{c\ell/d} \to \mathcal{F}$ is analogous to (8) except that (9) is replaced by

$$X_i = \sum_{\alpha \in H^{\ell/d}} \left(\prod_{i \in [\ell/d]} \prod_{\beta \in H \setminus \{\alpha_i\}} \frac{\beta - z_i}{\beta - \alpha_i} \right) \cdot x_\alpha \tag{10}$$

Invoking the sum-check protocol (w.r.t the non-binary H) on the claim that the sum $\sum_{w \in H^{c\ell/d}} \widehat{\Phi}_x(w)$ equals 0, yields a protocol that performs $c\ell/d$ iterations. Again, the verifier evaluates the residual expression, which amounts to evaluating the $\widehat{\pi}_{n,j}$'s and $\widehat{\phi}_n$ as well as computing $\mathrm{poly}(\ell)$ sums of $h^{\ell/d} = 2^\ell$ terms each, but here each term calls for evaluating a polynomial that has an arithmetic formula of size $O(\ell \cdot h/d)$. The prover's computation is dominated by computing a sum of $h^{c\ell/d} = 2^{c\ell}$ terms, where each term requires a computation of the type conducted by the verifier. ◀

4.4 Extension to a wider class

As a motivation towards the following extension, we mention the problem of finding a dominating set of constant size t. This problem does not seem to be locally-characterizable in the sense of Definition 2 (cf. [21]), but it definitely resides in the following class.

▶ **Definition 6** (locally ∀∃-characterizable sets)**.** A set S is locally ∀∃-characterizable if there exist constants c, c', a polynomial p and an almost-linear time algorithm that on input 1^n outputs $\mathrm{poly}(\log n)$-sized formulae $\phi_n : [n]^{p(\log n)} \times \{0,1\}^{p(\log n)} \to \{0,1\}$ and $\pi_{n,1}, ..., \pi_{n,p(\log n)} : \{0,1\}^{(c+c')\log n} \to [n]$ such that, for every $x \in \{0,1\}^n$, it holds that $x \in S$ if and only if for all $w \in \{0,1\}^{c \log n}$ there exists $w' \in \{0,1\}^{c' \log n}$ such that

$$\Phi_x(w, w') \stackrel{\text{def}}{=} \phi_n(\pi_{n,1}(w, w'), ..., \pi_{n,p(\log n)}(w, w'), x_{\pi_{n,1}(w,w')}, ..., x_{\pi_{n,p(\log n)}(w,w')}) \tag{11}$$

equals 0.

Like in Definition 2, sometimes one may use a simplified form in which ϕ_n only depends on its $p(\log n)$-bit long suffix. This simplification suffices for the charcaterizing the set of graphs not having a dominating set of size $t = O(1)$. Specifically, when representing n-vertex graphs by their adjacency matrix x (augmented with 1's on the diagonal), the t-dominating set problem is captured by $\Phi_x(i_1, ..., i_t, i_{t+1}) = \vee_{j \in [t]} x_{i_j, i_{t+1}}$ (i.e., x has no t-dominating set iff for every $w = (i_1, ..., i_t) \in [n]^t$ there exists $w' = i_{t+1} \in [n]$ such that $\Phi_x(w, w') = 0$); that is, we used $\phi_{n^2} : \{0,1\}^t \to \{0,1\}$ and $\pi_{n^2,j} : \{0,1\}^{(t+1)\log n} \to [n^2]$ such that $\phi_{n^2}(\sigma_1, ..., \sigma_t) = \vee_j \sigma_j$ and $\pi_{n^2,j}(i_1, ..., i_t, i_{t+1}) = (i_j, i_{t+1})$ for $j \in [t]$.

Every locally ∀∃-characterizable set is in $\mathcal{NC}$ (resp., in $\mathcal{SC}$) and so the existence of a doubly-efficient interactive proof system for it (and its complement) is guaranteed by [15] (resp., [22]). Here we present a direct construction.

▶ **Theorem 7** (main result, extended)**.** *Every locally ∀∃-characterizable set has a simple doubly-efficient interactive proof system. Specifically, on input of length n, the verifier runs in $\widetilde{O}(n)$-time and the strategy of the prescribed prover can be implemented in $\widetilde{O}(n^{c+c'+1})$-time, where $n^{c+c'}$ denotes the number of local conditions in the characterization. Furthermore, the interactive proof system uses public coins and has logarithmic round complexity.*

Note that the complement of every locally ∀∃-characterizable set also has a doubly-efficient interactive proof system. In this proof system, on input $x \in \{0,1\}^n$, letting $\ell = \log n$, the prover finds an adequate $w \in \{0,1\}^{c\ell}$, sends it to the verifier, and the parties engage in a doubly-efficient interactive proof of the residual claim that *for every $w \in \{0,1\}^{c'\ell}$ it holds that $\Phi_{x,w}(w') \stackrel{\text{def}}{=} \Phi_x(w, w')$ evaluates to 0.* (Such a proof system is provided by Theorem 1.)

Proof Sketch. We extend the proof of Theorem 1, as presented in Section 4.2. Specifically, letting $\ell = \log n$, we derive a low degree extension, $\widehat{\Phi}_x : \mathcal{F}^{(c+c')\ell} \to \mathcal{F}$, of Φ_x by following the same steps as in the former proof.[10] Here we have to provide a proof system for establishing that for every w there exists a w' such that $\widehat{\Phi}_x(w, w')$ equals 0, and the problem is converting this claim to one that can be handled by the sum-check protocol. We do so by mimicking the proof of Toda's Theorem [26]. Specifically, for $\ell' = O(\ell)$ and $H \equiv \{0,1\}$, we consider the following arithmetic expression

$$\sum_{w \in H^{c\ell}} r_w \cdot \prod_{i \in [\ell']} \left(1 - \sum_{w' \in H^{c'\ell}} r^{(i)}_{w'} \cdot (1 - \widehat{\Phi}_x(w, w')) \right) \tag{12}$$

where the r_w's and $r^{(i)}_{w'}$'s are selected at random by the verifier at the very beginning of the interaction. Actually, the verifier will select the sequence $(r_w)_{w \in H^{c\ell}}$ from a small biased sample space (over GF(2))[11], and ditto for the sequences $(r^{(i)}_{w'})_{w' \in H^{c'\ell}}$ (which are selected independently for each $i \in [\ell']$). In particular, we shall use $\mathcal{F}$ that is an extension field of GF(2), and view the r_w's (resp., $r^{(i)}_{w'}$'s) as elements of the base field GF(2).

Note that if x does not belong to the set (i.e., $\exists w \forall w' \; \Phi_x(w, w') = 1$), then there exists a w such that for every $i \in [\ell']$ and for every choice of the $r^{(i)}_{w'}$'s it holds that $\Psi^{(i)}_x(w) = 1$, where

$$\Psi^{(i)}_x(z) \stackrel{\text{def}}{=} 1 - \sum_{w' \in H^{c'\ell}} r^{(i)}_{w'} \cdot (1 - \widehat{\Phi}_x(z, w')). \tag{13}$$

Hence, when the r_w's are selected from a small bias set, it holds that

$$\Pr\left[\sum_{w \in H^{c\ell}} r_w \cdot \prod_{i \in [\ell']} \Psi^{(i)}_x(w) = 0 \right] \approx 1/2.$$

On the other hand, if x belongs to the set (i.e., $\forall w \exists w'$ s.t. $\Phi_x(w, w') = 0$), then for every w and i it holds that $\Pr[\Psi^{(i)}_x(w) = 1] < 0.51$, where the probability is taken over the choice of the $r^{(i)}_{w'}$'s (since $(1 - \widehat{\Phi}_x(w, w')) = 1$ for some w' and it follows that $\Pr[\sum_{w' \in H^{c'\ell}} r^{(i)}_{w'} \cdot (1 - \widehat{\Phi}_x(w, w'))) = 0]$ is approximately $1/2$). Hence, $\Pr[\prod_{i \in [\ell']} \Psi^{(i)}_x(w) = 0] < 0.51^{\ell'}$ and so, for every choice of r_w's, it holds that

$$\Pr\left[\sum_{w \in H^{c\ell}} r_w \cdot \prod_{i \in [\ell']} \Psi^{(i)}_x(w) = 0 \right] > 1 - 2^{c\ell} \cdot 0.51^{\ell'} \approx 1,$$

where the probability is taken over the choice of the $r^{(i)}_{w'}$'s (and the approximation assumes $\ell' \gg c \log \ell$ (e.g., $\ell' = 2c\ell$ will do)).

In order to prepare for an application of the sum-check, we need to replace the sequences $(r_w)_{w \in H^{c\ell}}$ and $(r^{(i)}_{w'})_{w' \in H^{c'\ell}}$ (for each $i \in [\ell']$) by the evaluation of low degree polynomials in w (resp., w') (which are defined over $\mathcal{F}^{c\ell}$ (resp., over $\mathcal{F}^{c'\ell}$) and agree with the said sequences on $H^{c\ell}$ (resp., on $H^{c'\ell}$)). (That is, for each fixing of the seed for a small bias generator,

[10] We stress that we use $H \equiv \{0,1\}$ and the corresponding function `EQ` as in Section 4.2 (rather than the settings used in Section 4.3).

[11] See [19] or [12, Sec. 8.5.2]. A seed length of $O(\ell)$ will do.

we consider the function that maps a location in the output sequence to a value of the corresponding bit.) Fortunately, the LFSR construction of [3] is suitable for that purpose, since the j^{th} bit in the corresponding sequence is produced by raising a matrix R to the power j and multiplying the first row of the resulting matrix by a vector s, where R and s are determined by the seed of this pseudorandom generator.[12] Specifically, the j^{th} bit is the top element of the vector $R^j s$, where matrix R and the vector s have dimension that is linear in the seed length (which in turn is logarithmic in the length of the produced sequence). Hence, we may replace r_w, where $w \equiv (w_1, ..., w_{c\ell})$, by a polynomial that computes the top bit of the vector $R^{\sum_{j\in[c\ell]} w_j 2^{j-1}} s$, by precomputing $R_j = R^{2^{j-1}}$ and using

$$R^{\sum_{j\in[c\ell]} w_j 2^{j-1}} = \prod_{j\in[c\ell]} R_j^{w_j} = \prod_{j\in[c\ell]} (w_j R_j + (1-w_j)I),$$

where $I = R^0$ is the identity matrix. Thus, r_w will be replaced by $\widehat{r}(w)$, where $\widehat{r} : \mathcal{F}^{c\ell} \to \mathcal{F}$ is such that $\widehat{r}(z)$ equals the top element of $(\prod_{j\in[c\ell]}(z_j R_j + (1-z_j)I))s$, and ditto for each $(r^{(i)}_{w'})_{w'\in H^{c'\ell}}$ (via the corresponding $\widehat{r}^{(i)} : \mathcal{F}^{c'\ell} \to \mathcal{F}$). We stress that $\widehat{r}$ (resp., $\widehat{r}^{(i)}$) is a polynomial of degree $c\ell$ (resp., $c'\ell$) and it can be evaluated in time $\text{poly}(\ell)$. Hence, the claim that (12) evaluates to 0 can be replaced by the claim

$$\sum_{w\in H^{c\ell}} \widehat{r}(w) \cdot \prod_{i\in[\ell']} \left(1 - \sum_{w'\in H^{c'\ell}} \widehat{r}^{(i)}(w') \cdot (1 - \widehat{\Phi}_x(w,w'))\right) = 0 \tag{14}$$

We outline two ways of handling this claim. The first way consists of invoking the generalized sum-check protocol, which can also handle products, on (14). Pursuing this approach requires identifying $[\ell']$ with $H^{\log \ell'}$ and introducing a low degree polynomial $\widehat{r}' : \mathcal{F}^{(\log \ell')+c'\ell} \to \mathcal{F}$ such that for every $i \in [\ell']$ it holds that $\widehat{r}'(i, z') = \widehat{r}^{(i)}(z')$.

Alternatively, we can apply the sum-check protocol to the claim $\sum_{w\in H^{c\ell}} \widehat{r}(w) \cdot \Psi_x(w) = 0$, where $\Psi_x(w) \stackrel{\text{def}}{=} \prod_{i\in[\ell']} \Psi^{(i)}_x(w)$. This involves $c\ell$ rounds of interactions, and leaves us with verifying a claim of the form $\Psi_x(r) = v$, where $r \in \mathcal{F}^{c\ell}$ and $v \in \mathcal{F}$ are determined by the said execution. At this point, the prover is asked to present the values of $\Psi^{(i)}_x(r)$ for each $i \in [\ell']$, the verifier checks that their products equals v, and the parties involve the sum-check protocol to each of the claimed values. That is, in the i^{th} execution, the prover proves that $1 - \sum_{w'\in H^{c'\ell}} \widehat{r}^{(i)}(w') \cdot (1 - \widehat{\Phi}_x(r, w'))$ equals v_i, where v_i is the value provided for $\Psi^{(i)}_x(r)$. (Note that these ℓ' executions can be performed in parallel.)[13]

This protocol performs $c\ell + c'\ell$ iterations, and the verifier evaluates the residual expression, which (as in the proof of Theorem 1) amounts to evaluating the $\widehat{\pi}_{n,j}$'s and $\widehat{\phi}_n$ as well as computing $\text{poly}(\ell)$ sums of 2^ℓ terms each. The prover's computation is dominated by computing a sum of $2^{c\ell+c'\ell}$ terms, where each term requires a computation of the type conducted by the verifier. ◀

[12] Alternatively, we can use the "powering" (in finite field) construction of [3].

[13] Alternatively, the verifier can select at random $r'_1, .., r'_{\ell'} \in \mathcal{F}$, and ask the prover to prove that $\sum_{i\in[\ell']} r'_i \cdot \Psi^{(i)}_x(r)$ equals $\sum_{i\in[\ell']} r'_i \cdot v_i$. Note that $\sum_{i\in[\ell']} r'_i \cdot \Psi^{(i)}_x(r) = \sum_{i\in[\ell']} r'_i - \sum_{w'\in H^{c'\ell}} \sum_{i\in[\ell']} r'_i \cdot \widehat{r}^{(i)}(w') \cdot (1 - \widehat{\Phi}_x(r, w'))$, so we can apply the sum-check to the outer sum (of $w' \in H^{c'\ell}$) and let the verifier evaluate the residual expression (which has ℓ' terms) by itself.

Digest

The interactive proof presented in the proof of Theorem 7 uses a randomized reduction of evaluating (11) to evaluating (12). In a straightforward implementation, this reduction calls upon the verifier to toss $\text{poly}(n)$ coins and send the outcome to the prover, whereas we aim at verifiers that run in time $\widetilde{O}(n)$. Hence, we use an adequate pseudorandom generator, and let the verifier select a (much shorter) random seed and send it to the prover. For this to work, we need the function that describes the pseudorandom sequence that corresponds to a fixed seed to have low complexity (in an adequate sense). That is, the relevant complexity measure here refers to the function that maps possible locations in a fixed sequence to the value of the corresponding bits, whereas the standard complexity measures refer to the mapping of possible seeds to the value of a fixed location in the corresponding output sequence.

▶ **Remark 8** (beyond $\forall\exists$-characterization)**.** *The notion of a locally $\forall\exists$-characterizable set can be further extended to allow a constant number of* (alternating) *quantifiers; for example, a $\forall\exists\forall$-characterization corresponds to the case that for all $w \in \{0,1\}^{c \log n}$ there exists $w' \in \{0,1\}^{c' \log n}$ such that for all $w'' \in \{0,1\}^{c'' \log n}$ it holds that $\Phi_x(w,w',w'') = 0$. The proof of Theorem 7 extends naturally to that case* (cf. the proof of Toda's Theorem [26]).

Lastly, we note the correspondence between the foregoing local characterizations and the levels of a known hierarchy of *parameterized complexity classes* [11]. In particular, Definition 2 corresponds to a class denoted $\mathcal{W}[1]$, and Definition 6 corresponds to $\mathcal{W}[2]$. (In terms of the $\mathcal{W}$-hierarchy, our definitions are restricted in requiring that the "defining circuits" be more uniform.)

5 The case of t-SUM

For a parameter $t \in \mathbb{N}$, given $(a_1, ..., a_n, b) \in [m]^{n+1}$, the problem is determining whether there exists t indices $i_1, ..., i_t \in [n]$ such that $\sum_{j\in[t]} a_{i_j} = b$. We shall assume, without loss of generality, that $m = \text{poly}(n)$ and that if $\sum_{j\in[t]} a_{i_j} = b$ then $|\{i_j : j \in [t]\}| = t$. These assumptions are justified as follows.

- Given an arbitrary instance $(a_1, ..., a_n, b)$, we consider the instance $(a'_{1,1}, ..., a'_{n,t}, b')$ such that $a'_{i,j} = (t+1)^t \cdot a_i + (t+1)^{j-1}$ and $b' = (t+1)^t \cdot b + \sum_{j\in[t]}(t+1)^{j-1}$. Hence, if $\sum_{(i,j)\in T} a'_{i,j} = b'$ for some $|T| \leq t$, then for every $j \in [t]$ there exists a unique $i \in [n]$ such that $(i,j) \in T$.
- Starting with the case of $m = \exp(\text{poly}(n))$, we reduce to the case of $m = \text{poly}(n)$ as follows. We pick uniformly at random a prime p in $S \stackrel{\text{def}}{=} [O(n^t \cdot \log m)]$ and reduces all integers modulo p.
  Observe that if $\sum_{j\in[t]} a_{i_j} \neq b$, then equality modulo p may hold for at most $\frac{\log tm}{\log\log tm}$ primes $p > \log m$, whereas the number of primes in S is n^t times larger.
  To get back to a problem over the integers (rather than over $\mathbb{Z}_p$), we reduce the modular problem to t instances of the integral problem. Specifically, we use the fact that $\sum_{j\in[t]} a_{i_j} \equiv b \pmod p$ if and only if for some $i \in [t]$ it holds that $\sum_{j\in[t]} a_{i_j} = b + (i-1)\cdot p$.

Our goal is to present an interactive proof for proving that for every $i_1, ..., i_t \in [n]$ it holds that $\sum_{j\in[t]} a_{i_j} \neq b$.

Letting $\chi\colon \mathbb{Z} \to \{0,1\}$ denote the predicate that returns 0 only on 0, we wish to prove that for all $i_1, ..., i_t \in [n]$ it holds that $\chi(b - \sum_{j\in[t]} a_{i_j}) = 1$. Letting B denote the set of primes in $[m']$, where $m' \stackrel{\text{def}}{=} \log(tm)$, we may prove instead that for all $i_1, ..., i_t \in [n]$ it holds that $\prod_{p\in B}\left(1 - \chi\left(b - \sum_{j\in[t]} a_{i_j} \bmod p\right)\right) = 0$, since $|b - \sum_{j\in[t]} a_{i_j}| < tm$ and $\prod_{p\in B} p > tm$.

Letting $[a]_p$ denote the value of a mod p, we can rewrite the above as

$$\prod_{p\in B}\left(1-\chi\left([b]_p-\sum_{j\in[t]}[a_{i_j}]_p \bmod p\right)\right)=0. \tag{15}$$

Observing that $a \stackrel{\text{def}}{=} [b]_p-\sum_{j\in[t]}[a_{i_j}]_p$ resides in $[-tp+t,p-1]$, it follows that $\sum_{i=0}^{t-1}\chi(a+i\cdot p) = t-(1-\chi(a \bmod p))$. Hence, we replace $1-\chi(a \bmod p)$ by $t-\sum_{i=0}^{t-1}\chi(a+i\cdot p)$, and rewrite (15) as

$$\prod_{p\in B}\left(t-\left(\sum_{i=0}^{t-1}\chi\left([b]_p+ip-\sum_{j\in[t]}[a_{i_j}]_p\right)\right)\right)=0. \tag{16}$$

Since the arguments to χ resides in $[-tp+t,tp-1]\subset[-tm'+1,tm'-1]$, reducing it modulo any prime $q>tm'$ does not change the outcome. We shall do so next, while replacing the condition that all (0-1) terms (which correspond to the various $(i_1,...,i_t)\in[n]^t$) evaluate to 1 by the condition that for a random prime $q\in[O(\log n)]$ it holds that

$$\sum_{i_1,...,i_t\in[n]}\ \prod_{p\in B}\left(t-\left(\sum_{i=0}^{t-1}\chi\left([b]_p+ip-\sum_{j\in[t]}[a_{i_j}]_p \bmod q\right)\right)\right)\equiv 0 \pmod q. \tag{17}$$

(Recall that if each of the terms equals 0 then (17) holds, whereas otherwise with high probability over the choice of q (say $q\in[2\log(n^tm),3\log(n^tm)]$) (17) does not holds.) At this point we can implement χ arithmetically (by just raising the argument to power $q-1$). This yields the condition

$$\sum_{i_1,...,i_t\in[n]}\ \prod_{p\in B}\left(t-\left(\sum_{i=0}^{t-1}\left([b]_p+ip-\sum_{j\in[t]}[a_{i_j}]_p \bmod q\right)^{q-1}\right)\right)\equiv 0 \pmod q. \tag{18}$$

Towards applying the sum-check protocol, using $\ell=\log n$ and $\mathcal{F}=\mathrm{GF}(q)$, we define $P:(\mathcal{F}^\ell)^t\to\mathcal{F}$ such that

$$P(z^{(1)},...,z^{(t)})=\prod_{p\in B}\left(t-\left(\sum_{i=0}^{t-1}\left(b'_{p,i}-\sum_{j\in[t]}\sum_{\alpha\in H^\ell}\mathtt{EQ}(\alpha,z^{(j)})\cdot a'_{\alpha,p}\right)^{q-1}\right)\right) \tag{19}$$

where $\{0,1\}\equiv H\subset\mathcal{F}$, $\mathtt{EQ}:\mathcal{F}^\ell\times\mathcal{F}^\ell\to\{0,1\}$ is the indentity indicator (i.e., $\mathtt{EQ}(\overline{\gamma},\overline{z})=\prod_{i\in[\ell]}(z_i\gamma_i+(1-z_i)(1-\gamma_i))$), and $b'_{p,i}=[b]_p+ip$ and $a'_{\alpha,p}=[a_\alpha]_p$.

We wish to use the sum-check protocol in order to verify that $\sum_{\overline{z}\in(H^\ell)^t}P(\overline{z})$ equals 0 mod q, but the problem is that P is a ($t\ell$-variate) polynomial over $\mathcal{F}=\mathrm{GF}(q)$ with individual degree $|B|\cdot(q-1)$. This is a problem because, when running the sum-check protocol, the field size must be larger than the product of the individual degree of the polynomial and the number of variables in the polynomial. The solution is to run the sum-check protocol over an extension field. Specifically, it suffices to use the extension field $\mathcal{K}=\mathcal{F}^3$, since in this case we have $t\ell\cdot(q-1)|B|<q^3/4$, provided that $q\geq\log(n^tm)$ (whereas $|B|<\log(tm)$ and $\ell=\log n$). We thus consider (19) as an expression over $\mathcal{K}$, while noting that its value is in the base field $\mathcal{F}$, and that this value indicates whether the original instance is a YES-instance or a NO-instance (provided that we were not unlucky in our choice of the random prime $q\in[O(\log n)]$).

To wrap-up. The interactive proof starts with the verifier selecting uniformly a random prime $q \in [2\log(n^t m), 3\log(n^t m)]$, and expecting the prover to prove that $\sum_{\overline{z}\in(H^\ell)^t} P(\overline{z}) = 0$, where this expression as well as the definition of P (in (19)) are considered over the extension field $\mathcal{K} = \mathrm{GF}(q)^3 = \mathrm{GF}(q^3)$. The parties then run the sum-check protocol for $t \cdot \ell$ rounds. At the end of the interaction, the verifier evaluates the residual condition (i.e., evaluates P on a single point). Hence, the verifier's computation is dominated by the evaluation of the multi-linear polynomial $\mathtt{EQ}$ on $t \cdot 2^\ell = tn$ points, which means that its complexity is $\widetilde{O}(tn)$. The prover's complexity is $2^{t\ell} = n^t$ times larger.

Digest

One interesting aspect of the foregoing proof system is that it applies to asserting the value of $\sum_{\overline{z}\in H^{t\ell}} P(\overline{z})$, where $P : \mathcal{F}^{t\ell} \to \mathcal{F}$ is a polynomial over $\mathcal{F}$. But since we had no good upper bound on the degree of P, the sum-check was invoked over an extension field of $\mathcal{F}$, denoted $\mathcal{K}$. That is, we actually considered a polynomial over $\mathcal{K}$ that agrees with P on inputs that reside in $\mathcal{F}^{t\ell}$. We note that a similar idea was used by Gur and Raz [17] in their Arthur-Merlin streaming algorithm.

6 Conclusions

Our goal in this work was identifying structures and patterns that facilitate the design of efficient proof systems. Towards this goal, we view the identification of the class of locally-characterizable sets as one of our primary contributions. This is a large and natural class that permits simple and efficient interactive proof systems. Building on the identification of this class and its proof systems, our subsequent work [14], which also builds on the work of [5], shows worst-case to average-case reductions between problems in a closely related (and also natural) class (see the discussion following Theorem 1). We hope that future work will further explore classes and problems that permit efficient proof systems, and that this exploration will contribute to our understanding of these problems' computational complexity.

References

1. Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. If the Current Clique Algorithms are Optimal, So is Valiant's Parser. In *46th IEEE Symposium on Foundations of Computer Science*, pages 98–117, 2015.
2. Amir Abboud, Kevin Lewi, and Ryan Williams. Losing Weight by Gaining Edges. In *22nd ESA*, pages 1–12, 2014
3. Noga Alon, Oded Goldreich, Joahn Håstad, and Rene Peralta. Simple Constructions of Almost k-wise Independent Random Variables. *Journal of Random Structures and Algorithms*, Vol. 3, No. 3, pages 289–304, 1992. Preliminary version in *31st FOCS*, 1990.
4. Laszlo Babai. Trading Group Theory for Randomness. In *17th ACM Symposium on the Theory of Computing*, pages 421–429, 1985.
5. Marshall Ball, Alon Rosen, Manuel Sabin and Prashant Nalini Vasudevan. Average-case fine-grained hardness. In *49th ACM Symposium on the Theory of Computing*, pages 483–496, 2017.
6. Marshall Ball, Alon Rosen, Manuel Sabin and Prashant Nalini Vasudevan. Proofs of Useful Work. IACR Cryptology ePrint Archive, Report 2017/203, 2017.
7. Andreas Björklund and Petteri Kaski. How Proofs are Prepared at Camelot. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, pages 391–400, 2016.

8 Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic Extensions of the Strong Exponential Time Hypothesis and Consequences for Non-reducibility. In *2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 261–270, 2016.

9 Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput.*, Vol. 201 (2), pages 216–231, 2005.

10 Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science A*, Vol. 141 (1–2), pages 109–131, 1995.

11 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Springer-Verlag Monographs in Computer Science, 1999.

12 Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.

13 Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that Yield Nothing but their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *Journal of the ACM*, Vol. 38, No. 3, pages 691–729, 1991. Preliminary version in *27th FOCS*, 1986.

14 Oded Goldreich and Guy N. Rothblum. Worst-case to Average-case reductions for subclasses of P. *ECCC* TR17-130, 2017.

15 Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating Computation: Interactive Proofs for Muggles. *Journal of the ACM*, Vol. 62(4), Art. 27:1-27:64, 2015. Extended abstract in *40th STOC*, pages 113–122, 2008.

16 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, Vol. 18, pages 186–208, 1989. Preliminary version in *17th STOC*, 1985. Earlier versions date to 1982.

17 Tom Gur and Ran Raz. Arthur-Merlin streaming complexity. *Information and Computation*, Vol. 243, pages 145–165, 2015.

18 Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, Vol. 39, No. 4, pages 859–868, 1992. Extended abstract in *31st FOCS*, 1990.

19 Joseph Naor and Moni Naor. Small-bias Probability Spaces: Efficient Constructions and Applications. *SIAM Journal on Computing*, Vol 22, pages 838–856, 1993. Preliminary version in *22nd STOC*, 1990.

20 Mihai Patrascu. Towards polynomial lower bounds for dynamic problems. In *42nd ACM Symposium on the Theory of Computing*, pages 603–610, 2010.

21 Mihai Patrascu and Ryan Williams. On the Possibility of Faster SAT Algorithms. In *21st SODA*, pages 1065–1075, 2010.

22 Omer Reingold, Guy N. Rothblum, Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In *48th ACM Symposium on the Theory of Computing*, pages 49–62, 2016.

23 Adi Shamir. IP = PSPACE. *Journal of the ACM*, Vol. 39, No. 4, pages 869–877, 1992. Preliminary version in *31st FOCS*, 1990.

24 Madhu Sudan. Invariances in Property Testing. In *Property Testing: Current Research and Surveys*. Springer, Lecture Notes in Computer Science (Vol. 6390), pages 211–227, 2010.

25 Justin Thaler. Semi-Streaming Algorithms for Annotated Graph Streams. In *43rd International Colloquium on Automata, Languages, and Programming*, pages 59:1–59:14, 2016.

26 Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, Vol. 20 (5), pages 865–877, 1991.

27 Virginia Vassilevska Williams. Hardness of Easy Problems: Basing Hardness on Popular Conjectures such as the Strong Exponential Time Hypothesis. In *10th International Symposium on Parameterized and Exact Computation*, pages 17–29, 2015.

28 Ryan Williams. Strong ETH Breaks With Merlin and Arthur: Short Non-Interactive Proofs of Batch Evaluation. In *31st Conference on Computational Complexity*, pages 2:1–2:17, 2016.

Appendix: An $\mathcal{MA}$ proof system for locally-chracterizable sets

We present first an $\mathcal{MA}$ proof system of verification complexity $\widetilde{O}(n^{(c+1)/2})$ for every locally-characterizable set, where n denotes the input length and the constant $c \geq 1$ is as in Definition 2. Recall that in the case of t-no-CLIQUE, the input length (for n-vertex graphs) is n^2 and $c = t/2$.

Our starting point is the claim $\sum_{w\in\{0,1\}^{c\ell}} \widehat{\Phi}_x(w) = 0$, where $\widehat{\Phi}_x$ is as in the proof of Theorem 1. Letting $\ell' = (c+1)\ell/2$ and $\ell'' = (c-1)\ell/2$, we write the claim $\sum_{w\in\{0,1\}^{c\ell}} \widehat{\Phi}_x(w) = 0$ as $\sum_{w'\in\{0,1\}^{\ell'}} P(w') = 0$, where

$$P(\overline{z}') = \sum_{w''\in\{0,1\}^{\ell''}} \widehat{\Phi}_x(\overline{z}'w'') \tag{20}$$

The key observation is that P is a multi-variate polynomial of degree $\mathrm{poly}(\ell)$ that can be computed by an arithmetic circuit of size $\widetilde{O}(2^{\ell''+\ell}) = \widetilde{O}(2^{\ell'})$. The size bound is due to summing over $2^{\ell''}$ summands in (20), where the summands are given by Eq. (8)-(9), and each summand is computed using a circuit of size $\widetilde{O}(2^{\ell})$ (the dominant part in computing each summand is computing the terms X_i). Thus, our $\mathcal{MA}$ proof system proceeds as follows.

1. The prover provides the verifier with $v_{w'} \leftarrow P(w')$, for every $w' \in \{0,1\}^{\ell'}$.
2. Using the $\mathcal{MA}$ system for "batch evaluation" of Williams [28], the prover proves to the verifier that $P(w') = v_{w'}$ for every $w' \in \{0,1\}^{\ell'}$.
   Recall that this $\mathcal{MA}$-proof can be verified in time that is almost linear in the sum of the number of evaluation points and the size of the circuit, where in our case each of these quantities is $\widetilde{O}(2^{\ell'})$. (The complexity is also linear in the degree of the computed polynomial, which is our case adds another $\mathrm{poly}(\ell)$ factor, and requires that the field is large enough (which holds too).)
3. Finally, the verifier checks that $\sum_{w'\in\{0,1\}^{\ell'}} v_{w'} = 0$.

Indeed, the non-obvious part is the $\mathcal{MA}$ system for "batch evaluation" of Williams [28], which is employed in Step 2.

Improvement for the case of $\pi_{n,i}$'s that are projections

We say that $\pi : \{0,1\}^{c\ell} \to [n]$ is a projection if there exists an ℓ-subset $I \subseteq [c\ell]$ such that $\pi(w) = w_I$ (where, as usual, $\{0,1\}^{\ell}$ is associated with $[n]$). For $c \geq 2$, in the special case that the $\pi_{n,i}$'s in Definition 2 are projections, we improve the verification time by a $\sqrt{n}$ factor (and the claim regarding t-no-CLIQUE follows). Letting $\ell' = \ell'' = c\ell/2$, observe that the polynomial P of (20) can be written as

$$P(\overline{z}') = \sum_{w''\in\{0,1\}^{\ell''}} Q(\overline{z}'w'', A_1(\overline{z}'w''), ..., A_{p(\ell)}(\overline{z}'w'')), \tag{21}$$

where $Q : \mathcal{F}^{c\ell+p(\ell)\cdot\ell} \to \mathcal{F}$ and $A_1, ..., A_{p(\ell)} : \mathcal{F}^{c\ell} \to \mathcal{F}$ are defined as

$$Q(\overline{z}, \overline{a}) = \widehat{\phi}_n(\widehat{\pi}_{n,1}(\overline{z}), ..., \widehat{\pi}_{n,p(\ell)}(\overline{z}), \overline{a}) \tag{22}$$

$$A_i(\overline{z}) = \sum_{\alpha\in\{0,1\}^\ell} \mathtt{EQ}(\widehat{\pi}_{n,i}(\overline{z}), \alpha) \cdot x_\alpha. \tag{23}$$

Observe that Q is a multi-variate polynomial of degree $\mathrm{poly}(\ell)$ that can be computed by an arithmetic of size $\mathrm{poly}(\ell)$, whereas the A_i's are multi-linear polynomials that can be computed by circuits of size $\widetilde{O}(2^\ell)$. Combining these circuits and summing over all $w'' \in \{0,1\}^{\ell''}$, as done above, yields a circuit of size $\widetilde{O}(2^{\ell''+\ell})$, whereas we aim at a circuit of size $\widetilde{O}(2^{\ell''} + 2^\ell)$. Towards this end, we use the hypothesis that the $\pi_{n,i}$'s are projections. Specifically, denoting the corresponding projections by I_i's, we observe that $A_i(\overline{z})$ actually depends only on $\overline{z}_{I_i}$. Furthermore, letting $I''_i = \{j - \ell' : j \in I_i \setminus [\ell']\}$ and $I'_i = I_i \cap [\ell']$, we can replace $A_i(\overline{z}'w'')$ by $C_{w''_{I''_i},i}(\overline{z}')$, where

$$C_{s,i}(\overline{z}') = \sum_{\alpha\in\{0,1\}^\ell} \mathtt{EQ}(\overline{z}'_{I'_i} s, \alpha) \cdot x_\alpha. \tag{24}$$

Hence, we obtain the circuit

$$P(\overline{z}') = \sum_{w''\in\{0,1\}^{\ell''}} Q(\overline{z}'w'', C_{w''_{I''_1},1}(\overline{z}'), ..., C_{w''_{I''_{p(\ell)}},p(\ell)}(\overline{z}')), \tag{25}$$

which has size $\widetilde{O}(2^{\ell'} + 2^{2\ell})$, where the size bound is due to the number of different circuits $C'_{s,i}$: for each $i \in [p(\ell)]$, there are $2^{|I''_i|} \leq 2^\ell$ possible values for s, and each circuit $C_{s,i}$ has size $\widetilde{O}(2^\ell)$. The key observation here is that the $2^{\ell''}$ terms in the main sum can reuse the values computed by the $\widetilde{O}(2^\ell)$ smaller circuits such that each term is fed by $p(\ell)$ small circuits (which are determined by its identity).

A closer inspection of these smaller circuits allows to upper bound their total size by $\widetilde{O}(2^\ell)$, instead of by $\widetilde{O}(2^{2\ell})$. Specifically, for each $i \in [p(\ell)]$, we have $2^{|I''_i|}$ different circuits but each of these circuits is a multilinear circuit in $|I'_i|$ bits (i.e., the bits $\overline{z}'_{I'_i}$ (see (24))), and so has size $\widetilde{O}(2^{|I'_i|})$. Hence, the circuit captured by (25) has size $\widetilde{O}(2^{\ell'}) + \widetilde{O}(2^\ell) = \widetilde{O}(n^{c/2} + n)$. Applying the foregoing $\mathcal{MA}$ proof system to the circuit captured by (25) (rather than to the circuit captured by (20) and Eq. (8)-(9)), yields a system with verification time $\widetilde{O}(n^{c/2} + n)$.