# On Doubly-Efficient Interactive Proof Systems

**Oded Goldreich**
Weizmann Institute of Science
oded.goldreich@weizmann.ac.il

the essence of knowledge
Boston — Delft

# Foundations and Trends® in Theoretical Computer Science

*Published, sold and distributed by:*
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

*Outside North America:*
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

O. Goldreich. *On Doubly-Efficient Interactive Proof Systems*. Foundations and Trends® in Theoretical Computer Science, vol. 13, no. 3, pp. 158–246, 2018.

*This Foundations and Trends® issue was typeset in LaTeX using a class file designed by Neal Parikh. Printed on acid-free paper.*

ISBN: 978-1-68083-408-6

**Foundations and Trends® in Theoretical Computer Science**
Volume 13, Issue 3, 2018

# Editorial Board

# Editorial Scope

## Topics

Foundations and Trends® in Theoretical Computer Science publishes surveys and tutorials on the foundations of computer science. The scope of the series is broad. Articles in this series focus on mathematical approaches to topics revolving around the theme of efficiency in computing. The list of topics below is meant to illustrate some of the coverage, and is not intended to be an exhaustive list.

- Algorithmic game theory
- Computational algebra
- Computational aspects of combinatorics and graph theory
- Computational aspects of communication
- Computational biology
- Computational complexity
- Computational geometry
- Computational learning
- Computational Models and Complexity
- Computational Number Theory
- Cryptography and information security
- Data structures
- Database theory
- Design and analysis of algorithms
- Distributed computing
- Information retrieval
- Operations research
- Parallel algorithms
- Quantum computation
- Randomness in computation

## Information for Librarians

Foundations and Trends® in Theoretical Computer Science, 2018, Volume 13, 4 issues. ISSN paper version 1551-305X. ISSN online version 1551-3068. Also available as a combined paper and online subscription.

Foundations and Trends® in
Theoretical Computer Science
Vol. 13, No. 3 (2018) 158–246

DOI: 10.1561/2300000057

now
the essence of knowledge

# On Doubly-Efficient Interactive Proof Systems

Oded Goldreich
Weizmann Institute of Science
oded.goldreich@weizmann.ac.il

# Contents

## Abstract

An interactive 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. Such proof systems, introduced by Goldwasser, Kalai, and Rothblum (*JACM*, 2015), make the benefits of interactive proof system available to real-life agents who are restricted to polynomial-time computation.

We survey some of the known results regarding doubly-efficient interactive proof system. We start by presenting two simple constructions for $t$-no-`CLIQUE` (due to Goldreich and Rothblum (*ECCC*, TR17-018 and TR18-046)), where the first construction offers the benefit of being generalized to any "locally characterizable" set, and the second construction offers the benefit of preserving the combinatorial flavor of the problem. We then turn to two more general constructions of doubly-efficient interactive proof system: the proof system for sets having (uniform) bounded-depth circuits (due to Goldwasser, Kalai and Rothblum (*JACM*, 2015)), and the proof system for sets that are recognized in polynomial-time and small space (due to Reingold, Rothblum, and Rothblum (*STOC*, 2016)). Our presentation of the GKR construction is quite complete (and is somewhat different from the original presentation), but for the RRR construction we only provide an overview.

O. Goldreich. *On Doubly-Efficient Interactive Proof Systems*. Foundations and Trends® in Theoretical Computer Science, vol. 13, no. 3, pp. 158–246, 2018.
DOI: 10.1561/2300000057.

# 1

# Introduction

The notion of interactive proof systems, put forward by Goldwasser, Micali, and Rackoff [28], and the demonstration of their power by Lund, Fortnow, Karloff, and Nisan [35] and Shamir [44] 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 $V$ always accept any input in $S$, but no strategy can make $V$ accept an input outside of $S$, except with negligible probability. (See Appendix A.1 for a formal definition of interactive proofs, and [19, Chap. 9] for a wider perspective.)

The original definition does not restrict the complexity of the strategy of the prescribed prover and the constructions of [35, 44] 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; this is beneficial when the proof system offers an advantage (over NP-proof systems) such as being zero-knowledge [28, 22].)

## 1.1 The notion of doubly-efficient interactive proof systems

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* [27] and *interactive proofs for delegating computation* [42]). 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. That is, doubly-efficient interactive proof systems are restricted by two additional efficiency requirements:

> *Prover's efficiency requirement*: The prescribed prover strategy (referred to in the completeness condition) should be implemented in polynomial-time.
>
> *Verifier's efficiency requirement*: The verifier strategy should be implemented in almost-linear time.

(We stress that unlike in *argument systems*, the soundness condition holds for all possible cheating strategies (not only for feasible ones).)[1]

Restricting the prescribed prover to run in polynomial-time implies that such systems may exist only for sets in $\mathcal{BPP}$, whereas 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 [27], who constructed such proof systems for any set that has log-space uniform circuits of small depth (e.g., log-space uniform $\mathcal{NC}$). A recent work of Reingold, Rothblum, and Rothblum [42] provided doubly-efficient (constant-round) proof systems for any set that can be decided in polynomial-time and small amount of space (e.g., for all sets in $\mathcal{SC}$). These two results are actually incomparable. The two constructions will be reviewed in Chapters 3 and 4, respectively, but before doing so we shall consider simpler constructions (see Section 1.2 and Chapter 2).

[1] See further discussion in Section 1.5.

**Terminology:** We keep the term *almost linear* vague on purpose, but whenever appropriate we shall spell-out a specific interpretation of it. The most strict interpretation is that a function $f : \mathbb{N} \to \mathbb{N}$ is almost linear if $f(n) = \widetilde{O}(n) = \text{poly}(\log n) \cdot n$. This interpretation is suitable for Chapter 3 and most of Chapter 2 (i.e., for the results of [27] and [24, 25], resp). In contrast, Chapter 4 (following [42]) uses a much more liberal interpretation by which a sequence of functions of the form $f_\epsilon : \mathbb{N} \to \mathbb{N}$ (representing the complexities of a sequence of constructions) is almost linear if $f_\epsilon(n) = O(n^{1+\epsilon})$ for every $\epsilon > 0$. We mention that these interpretations have been used in the past also in other settings (see discussion in [20, Sec. 13.3.3]).

## 1.2 Doubly-efficient NP-proof systems

For starters, we mention that doubly-efficient interactive proof systems exist for some sets in $\mathcal{P}$ that are believed not to have almost-linear decision procedures (or at least are not known to have such procedures). Examples include `perfect matching` and $t$-`CLIQUE`, for any constant $t \geq 3$. In these cases, the proof systems are actually of the NP-type. The point is that in each of these cases, an easily verified NP-witness (i.e., one that can be verified in almost-linear time) can be found in polynomial-time.

The foregoing assertion is quite evident for `perfect matching`, by virtue of the known matching algorithms and the fact that checking whether a set of edges is a perfect matching in a given graph can be done in linear time. The same hold for the maximum flow problem, by virtue of the min-cut dual. More generally, morally, the same holds for linear programming (again by duality; i.e., by checking the feasibility of the dual program).[2] Turning to fixed-parameter problems, such as $t$-`CLIQUE`, let us now consider the (popular) sets

$t$-`CLIQUE`**:** The set of $n$-vertex graphs that contain a clique of size $t$.

[2] Formally, issues may arise with respect to the length of the description of the solutions to the primal and dual programs, but these issues can be resolved by padding the input to that length (which seems quite natural in this context).

$t$-SUM: The set of $n$-long sequences of integers, say, in $[-n^{t+3}, n^{t+3}]$, that contain $t$ elements that sum-up to zero.

$t$-OV: The set of $n$-long sequences of vectors over $\mathrm{GF}(2)^{\log^2 n}$ that contain $t$ vectors such that their coordinate-wise multiplication yields the all-zero vector.

In all cases, the NP-witness is a set of $t$ elements, which can be found in time $\binom{n}{t}$ and verified in $\mathrm{poly}(t \cdot \log n)$-time. Recall that $t$-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]), and that 3-SUM is conjectured to require almost quadratic time (see, e.g., [40], which promoted it), whereas $t$-OV generalizes the Orthogonal Vectors problem (see [6]).[3] Furthermore, $t$-CLIQUE is $\mathcal{W}[1]$-complete [15], solving it in time $n^{o(t)}$ refutes the ETH [13], and lower bounds for $t$-SUM are known to follow from lower bounds for $t$-CLIQUE (see [2]).

In general, the class of sets having doubly-efficient NP-proof systems is a subclass of $\mathcal{P} \cap \mathrm{Ntime}(\tilde{L})$, where $\tilde{L}$ denotes the set of almost-linear functions. (Indeed, the class of sets having doubly-efficient NP-proof systems consists of all sets $S \in \mathrm{Ntime}(\tilde{L})$ associated with a witness relation $R$ such that given $x \in S$ we can find $y \in R(x) \stackrel{\mathrm{def}}{=} \{w : (x,w) \in R\}$ in polynomial-time.)[4] As in the case of $\mathcal{IP}$-versus-$\mathcal{NP}$, the question is what can be gained by allowing the verifier to be interactive, toss coins, and rule by statistical evidence. That is, moving

[3] Indeed, the Orthogonal Vectors problem corresponds to the case of $t = 2$. Our formulation of $t$-OV is different but equivalent to the one in [6], where the sequence of vectors is partitioned into $t$ equal parts and a YES-instance has to take a single vector from each part.

[4] Formally, $S$ has a doubly-efficient NP-proof system if and only if there exists a relation $R$ such that

1. if $(x,y) \in R$, then $|y|$ is almost linear in $|x|$;
2. $S = \{x : \exists w \text{ s.t. } (x,w) \in R\}$;
3. there exists a polynomial-time algorithm that, on input $x \in S$, outputs an element of $\{w : (x,w) \in R\}$;
4. there exists an almost-linear time algorithm that, on input $(x,y)$, decides whether or not $(x,y) \in R$.

beyond doubly-efficient NP-proof systems, we focus on the power of doubly-efficient *interactive* proof systems.

## 1.3 The power of doubly-efficient interactive proof systems

The bulk of this survey is devoted to demonstrating the power of doubly-efficient interactive proof systems. We shall start by presenting two different (doubly-efficient interactive) proof systems for $t$-no-CLIQUE (i.e., the complement of $t$-CLIQUE). These proof systems (presented in Sections 2.1 and 2.3) are considerably simpler than the proof systems that can be derived from the general results captured by Theorems 1.1 and 1.2 (and presented in Chapters 3 and 4).[5]

Before turning to these more general results, we briefly discuss the two doubly-efficient interactive proof systems for $t$-no-CLIQUE . One of these systems (i.e., the one presented in Section 2.3) proceeds in $t$ rounds such that, *in the* $i^{\text{th}}$ *round, a claim regarding the number of* $(t-i+1)$*-cliques in a graph is reduced to a claim regarding the number of* $(t-i)$*-cliques in a related graph*. Hence, in each iteration, a natural computational problem is reduced to a closely related computational problem, while preserving the combinatorial flavor of the original problem. This proof system can also handle varying $t$, yielding an alternative interactive proof system for $\#\mathcal{P}$.

The idea that underlies the other proof system for $t$-no-CLIQUE (presented in Section 2.1) can be applied to a natural class of "locally characterizable" sets (defined in Section 2.2, following [24, Sec. 5]). This class, which contains $t$-no-CLIQUE, is a subclass of $\mathcal{NC} \cap \mathcal{SC}$. This means that doubly-efficient interactive proof systems for locally characterizable sets can be obtained from either Theorem 1.1 or Theorem 1.2, but the point of presenting the direct proof systems for locally characterizable sets (in Section 2.2) is that they are considerably simpler than those obtained by either Theorem 1.1 or Theorem 1.2. Still, the latter theorems yield the most general results known regarding doubly-efficient interactive proof systems.

[5] The two simple proof systems for $t$-no-CLIQUE are due to [24, Sec. 3] and [25, Sec. 2], resp., whereas Theorems 1.1 and 1.2 are due to [27] and [42], resp.

**Theorem 1.1** (doubly-efficient interactive proof systems for log-space uniform $\mathcal{NC}$ [27])**.** Every set that is decidable by a family of log-space uniform circuits of depth $d$ such that $d(n) = \widetilde{O}(n)$, has a doubly-efficient interactive proof system. Furthermore, the proof system uses $O(\log n) \cdot d(n)$ rounds, and the verifier runs in $\widetilde{O}(n + d(n))$-time.

Although circuit size and depth is related to time and space [10], this relation is not tight enough to relate $\mathcal{NC}$ and $\mathcal{SC}$.[6] Hence, Theorem 1.1 is incomparable to the following

**Theorem 1.2** (doubly-efficient interactive proof systems for $\mathcal{SC}$ [42])**.** Every set that is decidable by an algorithm that runs in polynomial time and has space complexity $s$ such that $s(n) < \sqrt{n}$, has a doubly-efficient interactive proof system. Furthermore, for any constant $\delta > 0$, the proof system uses $\exp(\widetilde{O}(1/\delta))$ rounds, and verifier runs in $(\widetilde{O}(n) + s(n)^2 \cdot n^{\delta})$-time.

As noted in [42], Theorem 1.2 can be extended to randomized algorithms by first reducing their randomness complexity to linear (using adequate pseudorandom generators), and then letting the verifier toss coins for the derived algorithm and send the outcomes to the prover (asking it to prove membership in the corresponding residual set).[7] A begging open problem is whether the upper bound of $s(n)^2$ can be replaced by $s(n)$; that is,

**Problem 1.3** (a possible quantitative improvement of Theorem 1.2)**.** Does every set that is decidable by an algorithm that runs in polynomial time and linear space have a doubly-efficient interactive proof system?

[6] The point is that the translations between depth and space do not preserve polynomial bounds on the size and time, respectively. Specifically, $\mathcal{SC}$ can be emulated by uniform circuits of polylogarithmic depth (and quasi-polynomial size), whereas log-space uniform $\mathcal{NC}$ can be emulated by algorithms of polylogarithmic space complexity (that run in quasi-polynomial time).

[7] When applied to a randomized algorithm with two-sided error this yields an interactive proof system with two-sided error (a.k.a imperfect completeness (see Appendix A.1)). Recall, however, that we our focus is on interactive proof systems with perfect completeness (as in Definition A.1); such proof systems are obtained here when applying the foregoing reduction to a randomized algorithm that always accepts YES-instances. A similar comment holds with respect to all subsequent statements about $\mathcal{BPP}$ (see, e.g., Theorem 1.4 and Section 1.5); that is, when considering proof systems of perfect completeness, one may replace $\mathcal{BPP}$ by co$\mathcal{RP}$.

Another intriguing question is whether the round complexity in Theorem 1.2 can be reduced to $\text{poly}(1/\delta)$.

We mention that all the aforementioned proof systems, which will be surveyed in the subsequent chapters, are of the public-coin type.[8] Note that the known transformation of general interactive proof systems to public-coin ones does not apply to doubly-efficient interactive proof systems, since the resulting prover strategy is not efficient and this seems inherent [48]. This begs the question of whether general systems (i.e., ones that are not of the public-coin type) can offer some advantages in the context of doubly-efficient interactive proof systems.

## 1.4 An upper bound on doubly-efficient interactive proof systems

As stated upfront, doubly-efficient interactive proof systems exists only for sets in $\mathcal{BPP}$. This is the case, since a decision procedure can just emulate the interaction between the prescribed polynomial-time prover (and the polynomial-time verifier). Using the hypothesis that the verifier runs in almost linear time, it follows that such sets are decidable in almost linear space. This is the case since the hypothesis implies that the communication and the verifier's randomness are (at most) almost-linear (in the length of the input), and the same holds for the space complexity of the verifier. Hence, a machine of almost linear space complexity can decide membership in the set by emulating all possible interactions. For future reference, let us state the conclusion of the foregoing discussion.

**Theorem 1.4** (upper bound)**.** Every set that has a doubly-efficient interactive proof system can be decided in $\mathcal{BPP} \cap \text{Dspace}(\tilde{\ell})$, where $\tilde{\ell}$ is an almost linear function.

Note that even if Theorem 1.2 is improved as suggested in Problem 1.3, a gap will remain between it and Theorem 1.4, since $\text{TiSp}(T, s)$ is not necessarily equal $\text{Time}(T) \cap \text{Space}(s)$. Hence, another begging

[8] In the public coin model, at each round, the verifier tosses a predetermined number of coins and sends the outcome to the prover (see discussion at the end of Appendix A.1).

open problem is whether the power of doubly-efficient interactive proof systems is captured by $\mathrm{TiSp}(\mathrm{poly}, s)$ or by $\mathcal{BPP} \cap \mathrm{Space}(s)$ (or by neither).

**Problem 1.5** (on the gap between Theorems 1.2 and 1.4)**.** Prove or provide evidence against at least one of the following conjectures:

1. For $s(n) = \widetilde{O}(n)$, every set in $\mathcal{P} \cap \mathrm{Dspace}(s)$ has a doubly-efficient interactive proof system. Even establishing the claim for some $s(n) = \omega(\log n)$ would be interesting.[9]

2. Every set that has a doubly-efficient interactive proof system can be decided by a probabilistic algorithm that runs in polynomial time and almost linear space.

We mention that the second conjecture contradicts the conjecture that *log-space uniform circuits of linear depth and polynomial size cannot be emulated by polynomial-time algorithms of almost-linear space complexity*, since Theorem 1.1 provides doubly-efficient interactive proof systems for the former. Can stronger evidence be brought against the second conjecture? On the other hand, we propose a milder form of the second conjecture asserting that every set that has a *constant-round* doubly-efficient interactive proof system can be decided by a probabilistic algorithm that runs in polynomial time and almost linear space.

## 1.5 On doubly-efficient argument systems

Recall that argument systems are defined as interactive proof systems with the exception that the soundness condition is replaced by a computational soundness condition. That is, while the (standard) soundness condition requires that no cheating strategy can make the prover accept false assertions, except with negligible probability, the computational soundness condition only requires the infeasibility of cheating (i.e., that

[9] Note that, by Theorem 1.1, the claim holds for any $s(n) = O(\log n)$, since $\mathcal{L}$ is contained in log-space uniform $\mathcal{NC}$.

cheating strategies that can be implemented by polynomial size circuits may only fool the verifier with negligible probability).[10]

Doubly-efficient argument systems for any set in $\mathcal{BPP}$ are implicit in Kilian's argument system for sets in $\mathcal{NP}$, which relies on collision resistance hashing [32].[11] This system uses a constant number of rounds. Furthermore, assuming the existence of computational PIR schemes [33], every set in $\mathcal{BPP}$ has a two-message doubly-efficient argument system [30, 31, 11].[12] On the other hand, any set having a doubly-efficient argument system is in $\mathcal{BPP}$ (since we can decide membership in such a set by emulating the interaction between the prescribed prover and verifier strategies).[13]

The fact that argument systems are always asserted by relying on an intractability assumption is no coincidence, since these asserted systems do not satisfy the information theoretic soundness requirement. In fact, the existence of an argument system that is not an interactive proof system (i.e., does not satisfy standard soundness) implies a complexity separation (which is not known unconditionally). Specifically:

[10] Polynomial size circuits are preferred over probabilistic polynomial-time algorithms in order to account for auxiliary information that may be available to the prover (esp., when used as a subroutine inside a higher-level application).

[11] The claim is evident for sets in $\mathcal{P}$, whereas proving membership in $\mathcal{BPP}$-sets can be reduced to proving membership in $\mathcal{P}$-set as follows. First note that the hypothesis (i.e., the existence of collision resistance hashing) implies the existence of one-way functions, and hence of pseudorandom generators [29]. Using such a generator, we can reduce the randomness complexity of the decision procedures for $\mathcal{BPP}$ to linear, and let the verifier send a random-tape for such a procedure (as part of its first message). Hence, it suffices to verify a claim that refers to the residual set, which is in $\mathcal{P}$.

[12] The result of [31], which builds on [30], uses a computational PIR of quasi-polynomial security. The assumption was weakened to standard (polynomial security) by [11]. The original results that are stated for $\mathcal{P}$ can be extended to $\mathcal{BPP}$ (cf. Footnote 11).

[13] The validity of this decision procedure refers only to the probability that the prescribed prover convinces the (prescribed) verifier.

**Theorem 1.6** (arguments that are not proofs imply separations)**.** Let $V$ be a verifier strategy for an argument system for a set $S$, and suppose that $V$ does not satisfy the (information theoretic) soundness requirement. Then, $\mathcal{PSPACE}$ is not contained in $\mathcal{P}/\text{poly}$. In particular, $\mathcal{BPP} \neq \mathcal{PSPACE}$.[14]

In other words, Theorem 1.6 asserts that a gap between information theoretic soundness and computational soundness means a gap between computationally unbounded (prover) strategies and computationally bounded (prover) strategies. Recalling that, in the current setting (of fooling a probabilistic polynomial-time verifier), the former can be implemented in $\mathcal{PSPACE}$, whereas the computational restriction refers to $\mathcal{P}/\text{poly}$, the main claim follows (and $\mathcal{BPP} \neq \mathcal{PSPACE}$ follows, since $\mathcal{BPP} \subset \mathcal{P}/\text{poly}$).

**Proof:** Assume, for simplicity and without loss of generality, that in the said argument system each message of the prover consists of a single bit, and let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ denote an optimal prover strategy with respect to this system (i.e., $f(x,\gamma)$ is the message sent by the prover on common input $x$ after receiving the sequence of verifier messages described in the communication transcript $\gamma$). Then, $f \in \mathcal{PSPACE}$, since an optimal prover strategy (w.r.t a probabilistic polynomial-time verifier) can be implemented in polynomial space. On the other hand, $f \notin \mathcal{P}/\text{poly}$, since otherwise a polynomial size circuit could implement the optimal strategy (for convincing $V$), and so there will be no gap between the information theoretic soundness and the computational soundness of this proof system. ■

## 1.6 Preliminaries: The Sum-Check protocol

The Sum-Check protocol, designed by Lund, Fortnow, Karloff, and Nisan [35], is a key ingredient in some of the constructions that we present. In particular, it will used in Sections 2.1 and 2.2 as well as in Chapter 3.

[14] Note that $\mathcal{BPP} \neq \mathcal{PSPACE}$ implies $\mathcal{BPP} \subset \mathcal{PSPACE}$, since $\mathcal{BPP} \subseteq \mathcal{PSPACE}$.

Fixing a finite field $\mathcal{F}$ and a set $H \subset \mathcal{F}$ (e.g., $H$ may be a two-element set), 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

$$\sum_{\sigma_1,\ldots,\sigma_m \in H} P(\sigma_1, \ldots, \sigma_m) = v, \tag{1.1}$$

assuming that the verifier can evaluate $P$ by itself. The Sum-Check protocol proceeds in $m$ iterations, such that in the $i^{\text{th}}$ iteration the number of summations (over $H$) decreases from $m - i + 1$ to $m - i$. Specifically, the $i^{\text{th}}$ iteration starts with a claim of the form $\sum_{\sigma_i,\ldots,\sigma_m \in H} P(r_1, \ldots, r_{i-1}, \sigma_i, \ldots, \sigma_m) = v_{i-1}$, where $r_1, \ldots, r_{i-1}$ and $v_{i-1}$ are as determined in prior iterations (with $v_0 = v$), and ends with a claim of the form $\sum_{\sigma_{i+1},\ldots,\sigma_m \in H} P(r_1, \ldots, r_i, \sigma_{i+1}, \ldots, \sigma_m) = v_i$, where $r_i$ and $v_i$ are determined in the $i^{\text{th}}$ iteration. Initializing the process with $v_0 = v$, in the $i^{\text{th}}$ iteration the parties act as follows.

**Prover's move:** The prover computes a univariate polynomial of degree $d$ over $\mathcal{F}$

$$P_i(z) \stackrel{\text{def}}{=} \sum_{\sigma_{i+1},\ldots,\sigma_m \in H} P(r_1, \ldots, r_{i-1}, z, \sigma_{i+1}, \ldots, \sigma_m), \tag{1.2}$$

where $r_1, \ldots, r_{i-1}$ are as determined in prior iterations, and sends $P_i$ to the verifier (claiming that $\sum_{\sigma \in H} P_i(\sigma) = v_{i-1}$).

**Verifier's move:** Upon receiving a degree $d$ polynomial, denoted $\widetilde{P}$, the verifier checks that $\sum_{\sigma \in H} \widetilde{P}(\sigma) = v_{i-1}$ and rejects if inequality holds. Otherwise, it selects $r_i$ uniformly in $\mathcal{F}$, and sends it to the prover, while setting $v_i \leftarrow \widetilde{P}(r_i)$.

If all $m$ iterations are completed successfully (i.e., without the verifier rejecting in any of them), the verifier conducts a final check. It computes the value of $P(r_1, \ldots, r_m)$ and accepts if and only if this value equals $v_m$.

Clearly, if Eq. (1.1) 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 $m \cdot d/|\mathcal{F}|$, because in each iteration if the prover provides the correct

polynomial, then the verifier rejects (since $\sum_{\sigma\in H} P_i(\sigma) = P_{i-1}(r_{i-1}) \neq v_{i-1}$), and otherwise the (degree $d$) polynomial sent agrees with $P_i$ on at most $d$ points.[15]

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 $(d+1)\cdot|H|^{m-i}$ points, for each $i \in [m]$. (That is, the polynomial $P_i$ is computed by obtaining its values at $d+1$ points, where the value of $P_i$ at each point is obtained by summing the values of $P$ at $|H|^{m-i}$ points.)[16]

## 1.7 Organization

In Chapter 2 we present simple constructions of doubly-efficient interactive proof systems for $t$-no-CLIQUE, as well as for a natural class that contains it and is contained in uniform $\mathcal{NC}$ (and also in $\mathcal{SC}$). These proof systems are due to Goldreich and Rothblum [24, 25]. In Chapter 3 we present the proof systems of Goldwasser, Kalai and Rothblum [27], which are applicable to sets that are recognized by small depth circuits (e.g., uniform $\mathcal{NC}$). In Chapter 4 we provide an outline of the proof systems of Reingold, Rothblum, and Rothblum [42], which are applicable to sets recognized in polynomial-time and small space (e.g., sets in $\mathcal{SC}$).

As hinted in the foregoing paragraph, Chapter 2 is much easier to read than Chapters 3 and 4. Furthermore, while Chapters 2 and 3

[15] If $P_i$ does not satisfy the current claim (i.e., $\sum_{\sigma\in H} P_i(\sigma) \neq$ ï£ $\cdot\, v_{i-1}$), then the prover can avoid upfront rejection only if it sends $\widetilde{P} \neq P_i$. But in such a case, $\widetilde{P}$ and $P_i$ (both being degree $d$ polynomials) may agree on at most $d$ points. Hence, if the chosen $r_i \in \mathcal{F}$ is not one of these points, it holds that $v_i = \widetilde{P}(r_i) \neq P_i(r_i)$, which means that the next iteration will also start with a false claim. Hence, starting with a false claim (i.e., $\sum_{\sigma\in H} P_1(\sigma) \neq$ ï£ $\cdot\, v_0$ since Eq. (1.1) does not hold), with probability at least $1 - m\cdot d/|\mathcal{F}|$, after $m$ iterations we reach a false claim regarding the value of $P$ at a single point.

[16] Specifically, the value of $P_i$ at $p$ is obtained from the values of $P$ at the points $(r_1, \ldots, r_{i-1}, p, \sigma)$, where $\sigma \in H^{m-i}$.

provide full expositions of the claimed proof systems, Chapter 4 provides only an overview of the claimed proof system (while referring the interested reader to the 70-page description in the original work [42]).

Each of the following three chapters starts with several overview paragraphs that outline the contents of the chapter. Furthermore, Chapters 3 and 4 proceed with overview sections (see Sections 3.1 and 4.1, respectively).

We conclude (in Chapter 5) with an attempt to provide a high-level digest of the four fundamentally different proof systems reviewed above and with some speculations regarding the study of interactive proof systems at large.

**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 explicitly stated differently, all logarithms are to base 2.

# 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, Johan 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] Laszlo Babai, Lance Fortnow, Leonid Levin, and Mario Szegedy. Checking Computations in Polylogarithmic Time. In *23rd ACM Symposium on the Theory of Computing*, pages 21–31, 1991.

[6] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant N. Vasudevan. Average-Case Fine-Grained Hardness. In *48th ACM Symposium on the Theory of Computing*, pages 483–496, 2017.

[7] Mihir Bellare, Russell Impagliazzo, and Moni Naor. Does Parallel Repetition Lower the Error in Computationally Sound Protocols? In *38th IEEE Symposium on Foundations of Computer Science*, pages 374–383, 1997.

[8] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding. *SIAM Journal on Computing*, Vol. 36 (4), pages 889–974, 2006. Extended abstract in *36th STOC*, 2004.

[9] 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.

[10] Allan Borodin. On Relating Time and Space to Size and Depth. *SIAM Journal on Computing*, Vol. 6 (4), pages 733–744, 1977.

[11] Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. Non-interactive delegation and batch NP verification from standard computational assumptions. In *49th ACM Symposium on the Theory of Computing*, pages 474–482, 2017.

[12] 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.

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

[14] Irit Dinur and Omer Reingold. Assignment-testers: Towards a combinatorial proof of the PCP-Theorem. *SIAM Journal on Computing*, Vol. 36 (4), pages 975–1024, 2006. Extended abstract in *45th FOCS*, 2004.

[15] 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.

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

[17] Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Approximating Clique is almost NP-complete. *Journal of the ACM*, Vol. 43, pages 268–292, 1996. Preliminary version in *32nd FOCS*, 1991.

[18] Oded Goldreich. *Foundations of Cryptography: Basic Tools.* Cambridge University Press, 2001.

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

[20] Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.

[21] Oded Goldreich. On the doubly-efficient interactive proof systems of GKR. *ECCC*, TR17-101, 2017.

[22] 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.

[23] Oded Goldreich and Or Meir. Input-Oblivious Proof Systems and a Uniform Complexity Perspective on P/poly. *TOCT*, Vol. 7 (4), pages 16:1–16:13, 2015.

[24] Oded Goldreich and Guy N. Rothblum. Simple doubly-efficient interactive proof systems for locally-characterizable sets. *ECCC*, TR17-018, 2017.

[25] Oded Goldreich and Guy N. Rothblum. Counting $t$-cliques: Worst-case to average-case reductions and direct interactive proof systems. *ECCC*, TR18-046, 2018.

[26] Oded Goldreich, Salil Vadhan, and Avi Wigderson. On interactive proofs with a laconic provers. *Computational Complexity*, Vol. 11, pages 1–53, 2002.

[27] 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.

[28] 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.

[29] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A Pseudorandom Generator from any One-way Function. *SIAM Journal on Computing*, Volume 28, Number 4, pages 1364–1396, 1999.

[30] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. Delegation for bounded space. In *45th ACM Symposium on the Theory of Computing*, pages 565–574, 2013.

[31] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In *46th ACM Symposium on the Theory of Computing*, pages 485–494, 2014.

[32] Joe Kilian. A Note on Efficient Zero-Knowledge Proofs and Arguments. In *24th ACM Symposium on the Theory of Computing*, pages 723–732, 1992.

[33] Eyal Kushilevitz and Rafail Ostrovsky. Replication is NOT Needed: SINGLE Database, Computationally-Private Information Retrieval. In *38th IEEE Symposium on Foundations of Computer Science*, pages 364–373, 1977.

[34] Maya Leshkowitz. Round Complexity Versus Randomness Complexity in Interactive Proofs. *ECCC*, TR17-055, 2017.

[35] 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.

[36] Or Meir. IP = PSPACE Using Error-Correcting Codes. *SIAM Journal on Computing*, Vol. 42 (1), pages 380–403, 2013.

[37] Silvio Micali. Computationally Sound Proofs. *SIAM Journal on Computing*, Vol. 30 (4), pages 1253–1298, 2000. Preliminary version in *35th FOCS*, 1994.

[38] 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.

[39] Jaroslav Nesetril and Svatopluk Poljak. On the complexity of the subgraph problem. *Commentationes Mathematicae Universitatis Carolinae*, Vol. 26, No. 2, pages 415–419, 1985.

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

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

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

[43] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Efficient Batch Verification for UP. *ECCC*, TR18-022, 2018.

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

[45] 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.

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

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

[48] Salil P. Vadhan. On transformation of interactive proofs that preserve the prover's complexity. In *32nd ACM Symposium on the Theory of Computing*, pages 200–207, 2000.

[49] Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, Vol. 8, pages 189–201, 1979.

[50] 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.

[51] 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.