Hardness of Approximation for Stochastic Problems via Interactive Oracle Proofs

Gal Arnon
gal.arnon@weizmann.ac.il
Weizmann Institute

Alessandro Chiesa
alessandro.chiesa@epfl.ch
EPFL

Eylon Yogev
eylon.yogev@biu.ac.il
Bar-Ilan University

July 13, 2023

Abstract

Hardness of approximation aims to establish lower bounds on the approximability of optimization problems in NP and beyond. We continue the study of hardness of approximation for problems beyond NP, specifically for *stochastic* constraint satisfaction problems (SCSPs). An SCSP with k alternations is a list of constraints over variables grouped into $2\mathsf{k}$ blocks, where each constraint has constant arity. An assignment to the SCSP is defined by two players who alternate in setting values to a designated block of variables, with one player choosing their assignments uniformly at random and the other player trying to maximize the number of satisfied constraints.

In this paper, we establish hardness of approximation for SCSPs based on interactive proofs. For $\mathsf{k} \leq O(\log n)$, we prove that it is AM[k]-hard to approximate, to within a constant, the value of SCSPs with k alternations and constant arity. Before, this was known only for $\mathsf{k} = O(1)$.

Furthermore, we introduce a natural class of k-round interactive proofs, denoted IR[k] (for *interactive reducibility*), and show that several protocols (e.g., the sumcheck protocol) are in IR[k]. Using this notion, we extend our inapproximability to all values of k: we show that for every k, approximating an SCSP instance with $O(\mathsf{k})$ alternations and constant arity is IR[k]-hard.

While hardness of approximation for CSPs is achieved by constructing suitable PCPs, our results for SCSPs are achieved by constructing suitable IOPs (interactive oracle proofs). We show that every language in AM[$\mathsf{k} \leq O(\log n)$] or in IR[$\mathsf{k}$] has an $O(\mathsf{k})$-round IOP whose verifier has *constant* query complexity (*regardless* of the number of rounds k). In particular, we derive a "sumcheck protocol" whose verifier reads $O(1)$ bits from the entire interaction transcript.

Keywords: hardness of approximation; interactive oracle proofs; stochastic satisfaction problems

Contents

1 Introduction

Many combinatorial optimization problems are NP-hard, and so there is little hope to solve them in polynomial time. This has motivated the study of polynomial-time *approximation algorithms* to solve optimization problems, which has revealed a surprising landscape. While they are equivalent as *decision* problems (under polynomial-time reductions), NP-complete problems behave radically different from the point of view of approximability. Specifically, known approximation algorithms for NP-complete problems sometimes achieve approximations to within any constant, sometimes only to within a certain constant, and sometimes do not even achieve a constant approximation.

This differing behavior is justified via results in the area of *hardness of approximation*. For a given NP-complete problem, the goal is to prove that it is NP-hard to approximate the problem to better than a certain approximation ratio (e.g., better than 1/2). Ideally, this ratio would match the best known approximation algorithm, thereby ruling out better approximation algorithms.

The key tool used to establish hardness of approximation for NP-complete problems are probabilistic proofs [FGLSS96]. For example, the PCP theorem [AS98; ALMSS98] says that every language in NP can be decided via a verifier that reads $O(1)$ bits from a polynomial-length proof, and in turn this implies, e.g., that the value of 3SAT cannot be approximated to within an arbitrary constant.

More generally, improvements in PCP constructions imply hardness results for corresponding *constraint satisfaction problems* (CSPs). Yet, there are numerous problems of interest that are *not* CSPs, and for which we wish to understand their behavior with respect to inapproximability.

Hardness of approximation beyond NP. Prior work has investigated hardness of approximation for natural problems in other complexity classes. In one direction, PCP-like theorems for fine-grained complexity have been used to establish hardness results for problems within the complexity class P (see [AB17; ARW17; AR18; CW19; CGLRR19]). In the other direction, several works study the inapproximability of two-player CSPs. A CSP can be viewed as a one-player game where the player wishes to maximize the number of satisfied constraints; this view naturally leads to two-player CSPs played in moves. Ko and Lin [KL94] proved the inapproximability of two-player CSPs with k moves, based on the hardness of the k-th level of the Polynomial Hierarchy. Haviv, Regev, and Ta-Shma [HRT07] proved that this inapproximability result holds even when each variable occurs $O(1)$ times.

SCSPs and their hardness. In this paper, we study the hardness of approximating a natural class of problems known as *stochastic constraint satisfaction problems* (SCSPs), also known as *games against nature* [Pap83]. Informally, they are two-player CSPs where one player is an adversary and the other player is a (public-coin) referee that plays random moves.

Definition 1.1 (informal)**.** *An* **SCSP** Φ *with* k *alternations is a list of constraints* $C_1, \ldots, C_{\mathsf{m}}$ *over variables that are grouped into* $2\mathsf{k}$ *blocks and take on values over an alphabet* Σ*. The SCSP has* arity q *if each constraint depends on at most* q *variables. An assignment for* Φ *is defined by two players who alternate in setting values to a designated block of variables with one player choosing their assignments uniformly at random and the other player trying to maximize the number of satisfied clauses. The* value *of* Φ *is the expected fraction of clauses satisfied in this process.*

The hardness of approximating the value of SCSPs, to within a constant factor, has been studied in a line of works. Let AM[k] be the class of languages that have a k-round public-coin IP with constant soundness error. Drucker [Dru11] extended the PCP theorem to the stochastic setting, showing that it is AM[1]-complete to approximate the value of SCSPs with one alternation ($\mathsf{k} = 1$) and arity $\mathsf{q} = O(1)$. Subsequently, [ACY22] showed that, for every k, it is AM[k]-complete to

approximate the value of SCSPs with k alternations and arity $\mathsf{q} = O(\mathsf{k})$. In the regime of many alternations, Condon, Feigenbaum, Lund, and Shor [CFLS97] showed that it is PSPACE-complete to approximate the value of SCSPs with $\mathsf{k} = \text{poly}$ alternations and arity $\mathsf{q} = O(1)$. This leaves open the approximation hardness of SCSPs with k alternations and *constant* arity, for general values of k:

How hard is approximating the value of SCSPs with k *alternations and arity* $\mathsf{q} = O(1)$*?*

It seems reasonable to hypothesize that it is AM[k]-hard to approximate SCSPs with k alternations.

Note that Goldreich, Vadhan, and Wigderson [GVW02] showed that AM[k] $\neq$ AM[$o(\mathsf{k})$] (under reasonable hardness assumptions), meaning that increasing the round complexity k adds more power to the complexity class AM[k]. For sufficiently many rounds, we know that IP = PSPACE [LFKN92; Sha92]. Thus, it is imperative to understand the approximation hardness of SCSPs with k alternations while respecting the different regimes for k.

SCSP hardness from IOPs. The above results are derived (implicitly or explicitly) by leveraging the connection between SCSPs and the PCP analog of interactive proofs, called *interactive oracle proofs* (IOPs) [BCS16; RRR16]. A k-round (public-coin) IOP is a k-round (public-coin) IP where the verifier has PCP-like access to each prover message: after the interaction, the verifier probabilistically reads a small number of locations from the interaction transcript and then accepts or rejects.

A k-round public-coin IOP with query complexity q can be viewed as an SCSP with k alternations and arity q (see Section 2.1). Therefore, constructions of IOPs for hard languages imply corresponding hardness of approximation results for the resulting SCSPs. This leads us to ask:

Does every language in AM[k] *have a* k*-round IOP with constant query complexity?*

1.1 Our results

In this paper, we establish hardness of approximation for SCSPs from (general) interactive proofs. Moreover, we also prove that tighter hardness results can be achieved for specific languages that are *interactively reducible* (a notion that we introduce).

On the AM hardness of SCSPs. We prove that it is AM[k]-hard to approximate the value of binary-alphabet SCSPs with k alternations and arity $\max\{O(1), O(\mathsf{k}/\log|\mathbb{x}|)\}$ ($\mathbb{x}$ is the instance).

Theorem 1 (informal)**.** *Let* $L \in$ AM[k] *be a language. There exists a deterministic polynomial-time reduction that maps an instance* $\mathbb{x}$ *for* L *to an* SCSP *instance* Φ *with binary alphabet,* k *alternations, and arity* $\max\{O(1), O(\mathsf{k}/\log|\mathbb{x}|)\}$ *such that:*
- *if* $\mathbb{x} \in L$ *then the value of* Φ *is* 1*;*
- *if* $\mathbb{x} \notin L$ *then the value of* Φ *is at most* 1/2*.*

Our improvement in arity is particularly meaningful for logarithmic round complexity: Theorem 1 establishes that, for $\mathsf{k}(|\mathbb{x}|) = O(\log|\mathbb{x}|)$, approximating the value of an SCSP instance with k alternations and arity $O(1)$ is as hard as deciding all of AM[k]. Previously, this was known only for constant k [Dru11; ACY22].

Many results on the hardness of approximating the value of (standard) CSPs are achieved by constructing suitable PCPs. Similarly (as was noted in [ACY22]), by constructing a suitable k-round IOP for a language L, one can show that approximating the value of SCSPs with k alternations up to a constant factor is as hard as deciding L. We establish Theorem 1 using this framework by providing a transformation that maps a k-round IP into a k-round IOP with small query complexity.

Lemma 1 (informal)**.** *Let L be a language with a k-round public-coin IP. Then L has a k-round public-coin binary IOP where, on input $\mathbb{x}$, the verifier reads $\max\{O(1), O(\mathsf{k}/\log|\mathbb{x}|)\}$ bits of the interaction transcript. All other parameters are polynomially related.*

Lemma 1 is surprising in light of the work of Goldreich, Vadhan, and Wigderson [GVW02], which shows a separation between AM[k] and AM[$o(\mathsf{k})$] (under relatively weak hardness assumptions). Since the query complexity is smaller than the round complexity, Lemma 1 implies that the IOP verifier does not make queries to every round of the protocol. This can be viewed as saying that the power of AM[k] is unchanged even when the verifier only accesses $O(\mathsf{k}/\log|\mathbb{x}|)$ of the k rounds. In other words, reducing round complexity of public-coin protocols reduces their power, but it is nevertheless possible to not read every round of interaction while preserving the power.

Hardness of SCSPs from interactive reducibility. Theorem 1 establishes the hardness of SCSPs with $\mathsf{k} = O(\log|\mathbb{x}|)$ alternations and arity $O(1)$ (over the binary alphabet), but does not work for $O(1)$-arity SCSPs with $\mathsf{k} = \omega(\log|\mathbb{x}|)$ alternations.

We extend this by showing that approximating the value of $O(1)$-arity SCSPs with k alternations is as hard as solving $\#\mathsf{SAT}_{\mathsf{k}}$.[1] In fact, we show this for a more general class of languages that are *interactively reducible*, a notion that we introduce in this work. Informally, the notion requires that it is possible to reduce, via an interactive protocol, multiple transcripts of an IP for the relation into a single transcript. We require that the probability of the verifier accepting conditioned on the reduced transcript be (roughly) the minimum probability of the verifier accepting conditioned on any of the original transcripts. See Section 2.3 for further details and discussion.

Interactive reducibility is a natural property; we show that general interactive proofs can be seen as interactive reductions (albeit ones with bad parameters), and show interactive reductions for the sumcheck protocol [LFKN92] and for Shamir's protocol [Sha92].

The notion of interactive reducibility allows us to get an optimal version of Lemma 1 for additional languages. Let IR[k] be the class of languages that have a (1-round) interactive reduction with k predicates (these predicates roughly align with rounds of an IP).

Lemma 2 (informal)**.** *Let L be a language in* IR[k]*. Then L has an $O(\mathsf{k})$-round public-coin IOP, with polynomial proof length, where the verifier reads $O(1)$ bits of the interaction transcript.*

In particular, applying Lemma 2 to the sumcheck protocol yields the following (perhaps surprising) conclusion: *any k-round sumcheck protocol can be transformed to a $O(\mathsf{k})$-round IOP where the verifier has $O(1)$ query complexity (over the binary alphabet).* Notice that using standard PCP techniques it is only known how to achieve a similar (non-interactive) result with *exponential* proof length.

Using this improved lemma, we immediately get that for every k, deciding whether a binary-alphabet SCSP instance with $O(\mathsf{k})$ alternations and arity $O(1)$ has value 1 or value 1/2 is IR[k]-hard:

Theorem 2 (informal)**.** *Let $L \in$ IR[k] be a language. There exists a deterministic polynomial-time reduction that maps an instance $\mathbb{x}$ for L to an* SCSP *instance Φ with binary alphabet, $O(\mathsf{k})$ alternations, and arity $O(1)$ such that:*

- *if $\mathbb{x} \in L$ then the value of Φ is 1;*
- *if $\mathbb{x} \notin L$ then the value of Φ is at most 1/2.*

Using our interactive reduction for sumcheck, we know that $\#\mathsf{SAT}_{\mathsf{k}} \in$ IR[k]. Thus, by Theorem 2, we establish that approximating the value of SCSPs with $O(\mathsf{k})$ alternations and constant arity is

[1] $\#\mathsf{SAT}_{\mathsf{k}}$ is the restriction of $\#\mathsf{SAT}$ to instances of size n and $\mathsf{k}(n)$ variables.

$\#\mathsf{SAT}_{\mathsf{k}}$-hard. Similarly, using our interactive reduction for Shamir's protocol, we recover the result of [CFLS97], showing that approximating the value of SCSPs with polynomially-many alternations and constant arity is PSPACE-hard.

Summary of results and open questions. We construct $O(1)$-query IOPs for every language in $\mathrm{AM}[\mathsf{k} \leq O(\log |\mathbb{x}|)]$ or in $\mathrm{IR}[\mathsf{k}]$. Moreover, we construct $O(\mathsf{k}/\log |\mathbb{x}|)$-query IOPs for every language in $\mathrm{AM}[\mathsf{k}]$. These results establish approximation hardness for SCSPs as follows: (a) for $\mathsf{k} = O(\log |\mathbb{x}|)$, it is $\mathrm{AM}[\mathsf{k}]$-hard to approximate the value of SCSPs with k alternations and constant arity; and (b) for $\mathsf{k} = \omega(\log |\mathbb{x}|)$, it is $\mathrm{IR}[\mathsf{k}]$-hard to approximate the value of SCSPs with $O(\mathsf{k})$ alternations and constant arity, and $\mathrm{AM}[\mathsf{k}]$-hard when the SCSPs have arity $O(\mathsf{k}/\log |\mathbb{x}|)$. Our results are summarized in Figure 1 together with previously known results.

Our work leaves open the AM-hardness of approximating the value of SCSPs with $\mathsf{k} = \omega(\log |\mathbb{x}|)$ alternations and constant arity. From the perspective of IOPs, we also leave open the basic question that we raised in the introduction: *Does every language in* $\mathrm{AM}[\mathsf{k}]$ *have a* k*-round (public-coin) IOP with constant query complexity over the binary alphabet?* Our results contribute notable progress towards resolving this question (see paragraph above), but answering the question for every regime of k remains a fascinating challenge in the theory of probabilistic proofs.

	hardness	**alternations**	**arity**
[CFLS97]	PSPACE	unbounded	$O(1)$
[this work]	$\mathrm{IR}[\mathsf{k}]$	$O(\mathsf{k})$	$O(1)$
[this work]	$\#\mathsf{SAT}_{\mathsf{k}}$	$O(\mathsf{k})$	$O(1)$
[ACY22]	$\mathrm{AM}[\mathsf{k}]$	k	$O(\mathsf{k})$
[this work]	$\mathrm{AM}[\mathsf{k}]$	k	$\max\{O(1), O(\mathsf{k}/\log |\mathbb{x}|)\}$
[Dru11]	AM[1]	1	$O(1)$
[ALMSS98; AS98; Din07]	NP	n/a	$O(1)$

Figure 1: Summary of results for approximating the value of binary-alphabet SCSPs to within a constant factor. $\mathrm{AM}[\mathsf{k}]$ denotes the class of languages with k-round public-coin interactive proofs. $\mathrm{IR}[\mathsf{k}]$ denotes the class of languages with (1-round) interactive reductions with k predicates. $\#\mathsf{SAT}_{\mathsf{k}}$ is the restriction of $\#\mathsf{SAT}$ to instances of size n and $\mathsf{k}(n)$ variables.

2 Techniques

We outline the main ideas behind our results. In Section 2.1 we explain a generic connection between SCSPs and IOPs: in order to establish the hardness of approximating SCSPs, it suffices to construct IOPs with certain properties. This will be our goal in the remaining sections. In Section 2.2 we show how to transform k-round IPs into k-round IOPs with query complexity $\max\{O(1), O(\mathsf{k}/\log|\mathbb{x}|)\}$. In Section 2.3 we show that for relations that are *interactively reducible* we can construct $O(1)$-query IOPs even when those relations are only known to have IPs with round complexity $\omega(\log|\mathbb{x}|)$.

2.1 On the hardness of approximating SCSPs via IOPs

We review the generic connection between CSPs and PCPs, and then describe the analogous connection between SCSPs and IOPs. In both cases, efficient constructions of PCPs/IOPs imply hardness of approximation results for corresponding CSPs/SCSPs.

CSP hardness from PCPs. A CSP Φ is a list of boolean functions $C_1, \ldots, C_{\mathsf{m}}$ over variables from a bounded alphabet Σ. The CSP has arity q if each constraint depends on at most q variables. The goal is to determine the maximum fraction of constraints that can be satisfied by any assignment.

We can map a non-adaptive PCP verifier $\mathbf{V}$ for a language L and an instance $\mathbb{x}$ into a CSP. The variables represent locations of the PCP string. Each choice of PCP verifier randomness induces a corresponding constraint, whose variables are those that the PCP verifier would have read from the PCP string. The constraint is satisfied if and only if the PCP verifier accepts when it receives the assignment of the variables as its query answers. The CSP's arity equals the PCP's query complexity.

By completeness of the PCP system, if $\mathbb{x} \in L$ then there exists a PCP string that makes the PCP verifier accept with probability 1, which in turn means that there is an assignment that simultaneously satisfies every constraint in the CSP. By the soundness of the PCP system, if $\mathbb{x} \notin L$ then every PCP string makes the PCP verifier accept with at most probability $1/2$, which in turn means that no assignment can satisfy more than half of the constraints of the CSP.

Thus, distinguishing whether the CSP's value is 1 or at most $1/2$ is as hard as deciding L.

SCSP hardness from IOPs. The general connection between SCSPs (Definition 1.1) and IOPs is stated in the lemma below. Recall that a k-round IOP is a k-round IP where the verifier has PCP-like access to each prover message: the prover and verifier interact for k rounds, and after the interaction, the verifier probabilistically reads a small number of bits from each prover message and decides to accept or reject based on the examined locations. The randomness used in the final phase is called *decision randomness* (which we distinguish from the random messages that the verifier sends to the prover during the interaction and may be queried at only few locations).

Lemma 3. *Let L be a language with a non-adaptive k-round public-coin IOP with alphabet Σ, polynomial proof length, query complexity q, decision randomness r_{dc}, and soundness error β.*

Then there exists a deterministic polynomial-time reduction that maps an instance $\mathbb{x}$ for L to an SCSP instance Φ with alphabet Σ, k alternations, arity q, $2^{\mathsf{r}_{\mathsf{dc}}}$ constraints and a polynomial number of variables such that:

- *if $\mathbb{x} \in L$ then the value of Φ is 1;*
- *if $\mathbb{x} \notin L$ then the value of Φ is at most β.*

Proof sketch. Let $\mathbf{V}_{\mathsf{IOP}}$ be the (non-adaptive) IOP verifier for L and let l be the (per-round) proof length of the IOP. Given an instance $\mathbb{x}$, we construct the SCSP instance Φ as follows. The SCSP has

$2\mathsf{k}$ blocks of l variables that align with the interaction transcript between the IOP prover and IOP verifier (the i-th variable of the j-th block corresponds to the i-th symbol of the j-th message of the protocol). The SCSP has a constraint C_ρ for each $\rho \in \{0,1\}^{\mathsf{r_{dc}}}$, whose input variables correspond to the locations of the transcript that $\mathbf{V}_{\mathsf{IOP}}$ queries given instance $\mathbb{x}$ and randomness ρ; the constraint C_ρ is satisfied if and only if $\mathbf{V}_{\mathsf{IOP}}(\mathbb{x};\rho)$ accepts if it reads the symbols assigned to the variables of C_ρ.

The SCSP instance Φ has k alternations (corresponding to the rounds of the IOP) and $\mathsf{l} = \mathrm{poly}(|\mathbb{x}|)$ variables per alternation (corresponding to the message length) that are assigned values in the alphabet Σ (the alphabet of the IOP). Each of its $2^{\mathsf{r_{dc}}}$ constraints has arity q since each constraint has as many inputs as queries made by the IOP verifier $\mathbf{V}_{\mathsf{IOP}}$.

Finally, we analyze the value of the SCSP. By construction, there exists an IOP prover strategy that causes the IOP verifier $\mathbf{V}_{\mathsf{IOP}}$ to accept with probability δ if and only if there exists a strategy for the existential player in the SCSP such that the expected fraction of constraints that are satisfied is δ (i.e., the value of Φ is at least δ). Therefore, by perfect completeness of the IOP, if $\mathbb{x} \in L$ then the value of Φ is 1. Conversely, by soundness of the IOP, if $\mathbb{x} \notin L$ then the value of Φ is at most β. $\square$

2.2 Transforming IPs into IOPs

We outline the proof of Lemma 1 (transforming an IP into an IOP with small query complexity). In Section 2.2.1, we show how to transform a logarithmic-round IP into a $O(1)$-query IOP. Then in Section 2.2.2 we extend this idea to transform a k-round IP into a $O(\mathsf{k}/\log|\mathbb{x}|)$-query IOP.

2.2.1 From $O(\log|\mathbb{x}|)$-round IP to $O(1)$-query IOP

We show how to transform a k-round public-coin IP where $\mathsf{k} = O(\log|\mathbb{x}|)$ into an $O(\mathsf{k})$-round public-coin IOP with the following efficiency: polynomial proof length over the binary alphabet; constant query complexity; and logarithmic decision randomness.

First, we sketch how to transform a k-round public-coin IP $(\mathbf{P}_{\mathsf{IP}}, \mathbf{V}_{\mathsf{IP}})$ into an $O(\mathsf{k})$-round public-coin IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ where the verifier reads $O(1)$ rounds (in their entirety) from the interaction transcript. Then we explain how to ensure that the verifier queries $O(1)$ bits in total.

A strawman protocol. We describe a natural strategy for transforming the IP into an IOP where the verifier reads $O(1)$ rounds, albeit with high soundness error. The IOP prover $\mathbf{P}_{\mathsf{IOP}}$ and IOP verifier $\mathbf{V}_{\mathsf{IOP}}$ respectively simulate the IP prover and verifier ($\mathbf{P}_{\mathsf{IP}}$, $\mathbf{V}_{\mathsf{IP}}$), inducing an interaction transcript $\mathsf{tr} = (\rho_1, a_1, \ldots, \rho_{\mathsf{k}}, a_{\mathsf{k}})$. At this time, however, $\mathbf{V}_{\mathsf{IOP}}$ *does not read any messages from* tr. After this interaction, $\mathbf{P}_{\mathsf{IOP}}$ sends a transcript tr', which is allegedly equal to tr, as a single message. Then $\mathbf{V}_{\mathsf{IOP}}$ reads tr' and checks that tr' is an accepting transcript for the IP verifier $\mathbf{V}_{\mathsf{IP}}$. Moreover, $\mathbf{V}_{\mathsf{IOP}}$ tests consistency between tr (the real interaction) and tr' (the alleged copy of the interaction sent as a single message): $\mathbf{V}_{\mathsf{IOP}}$ samples a random $i \in [\mathsf{k}]$, reads the i-th prover message and i-th verifier message in tr, and checks that these equal the corresponding messages in tr'.

In this IOP, $\mathbf{V}_{\mathsf{IOP}}$ reads $O(1)$ messages from the interaction transcript, but the soundness error of the IOP is large (even when discounting the soundness error of the underlying IP). Indeed, it may be that a cheating IOP prover sends a malicious transcript tr' that is accepting but differs from the real transcript tr in one round only. In this case, $\mathbf{V}_{\mathsf{IOP}}$ catches the inconsistency only with probability $1/\mathsf{k}$, which means that the soundness error could be as large as $1 - 1/\mathsf{k}$.

Note that reducing this soundness error via parallel repetition would increase the number of rounds queried by $\mathbf{V}_{\mathsf{IOP}}$. Achieving constant soundness error would require $O(\mathsf{k})$ repetitions, resulting in an IOP verifier that reads $O(\mathsf{k})$ rounds, taking us back to where we started.

Our transformation. We present a transformation that improves on the above strawman protocol, achieving a constant soundness error for any IP that has a logarithmic number of rounds.

A malicious IOP prover in the strawman protocol has two strategies: the transcript tr' sent as the last message either agrees with the real transcript tr on more than half of the rounds, or it does not. If tr' agrees with tr in less than half of the rounds, then the IOP verifier catches this inconsistency with probability at least $1/2$. Intuitively, the transformation that we sketch below ensures that if tr' is consistent with tr on at least half of the rounds, then the consistent rounds must contain within them a full execution of the underlying IP. Then, since this consistent part was generated interactively in tr, by the soundness property of the IP, this contained transcript will be rejected with high probability. We now describe our transformation in more detail.

Suppose that our public-coin IP has $\mathsf{k}(|\mathbb{x}|) = O(\log|\mathbb{x}|)$ rounds and that the verifier message in each round is r bits long. The IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ on a given instance $\mathbb{x}$ works as follows.

1. $\mathbf{P}_{\mathsf{IOP}}$ sets $S_0 := \{\emptyset\}$ (i.e., S_0 consists of the empty transcript).
2. For $i = 1, \ldots, 2\mathsf{k}$:
   - $\mathbf{P}_{\mathsf{IOP}}$ sends S_{i-1}.
   - $\mathbf{V}_{\mathsf{IOP}}$ sends a random $\rho_i \in \{0,1\}^{\mathsf{r}}$ (corresponding to a message of $\mathbf{V}_{\mathsf{IP}}$).
   - $\mathbf{P}_{\mathsf{IOP}}$ sets $S_i := S_{i-1} \cup \{(\mathsf{tr}||\rho_i||a_{\mathsf{tr},i})\}_{\mathsf{tr}\in S_{i-1}}$ where $a_{\mathsf{tr},i} := \mathbf{P}_{\mathsf{IP}}(\mathbb{x}, \mathsf{tr}||\rho_i)$ for each $\mathsf{tr} \in S_{i-1}$.
3. $\mathbf{P}_{\mathsf{IOP}}$ sends $S_{2\mathsf{k}}$ and, for every $i \in [2\mathsf{k}]$, also sends $T_i := S_i$.
4. In the decision phase, $\mathbf{V}_{\mathsf{IOP}}$ performs the checks below.
   (a) *Subset consistency:* For every $i \in [2\mathsf{k}]$, check that $T_{i-1} \subseteq T_i$.
   (b) *Transcript consistency:* Choose a random $i \in [2\mathsf{k}]$. Check that $S_{i-1} = T_{i-1}$ and $S_i = T_i$. Additionally, check that for every $\mathsf{tr} \in S_{i-1}$ there is a message $a_{\mathsf{tr},i}$ such that $(\mathsf{tr}||\rho_i||a_{\mathsf{tr},i}) \in S_i$, where ρ_i is the verifier message sent during the i-th round of interaction.
   (c) *Membership:* Check that for every transcript $\mathsf{tr} \in T_{2\mathsf{k}}$ that is complete (i.e., contains messages for all k rounds of the IP) it holds that $\mathbf{V}_{\mathsf{IP}}(\mathbb{x}, \mathsf{tr}) = 1$.

Efficiency. We briefly discuss the main efficiency measures of the transformation.

- *Query complexity.* The IOP verifier reads $O(1)$ rounds from the transcript.
- *Communication complexity.* We argue that all messages in the protocol have length $\mathrm{poly}(|\mathbb{x}|)$. For every i, $|S_i| \leq 2|S_{i-1}|$ since S_i contains all transcripts in S_{i-1} and continuations of each of these transcripts. Since $\mathsf{k} = O(\log|\mathbb{x}|)$, $|S_i| = \mathrm{poly}(|\mathbb{x}|)$ for every i. Each transcript within S_i has polynomial length, so the length of these messages is $\mathrm{poly}(|\mathbb{x}|)$. Finally, the sets $T_1, \ldots, T_{2\mathsf{k}}$ have the same sizes as $S_1, \ldots, S_{2\mathsf{k}}$ (respectively) and so the prover's final message has length $\mathrm{poly}(|\mathbb{x}|)$.
- *Decision randomness.* The IOP verifier uses $O(\log \mathsf{k}) = O(\log|\mathbb{x}|)$ bits of decision randomness.

Analysis. In this overview, we discuss soundness only, as completeness follows straightforwardly from the construction. Let β be the soundness error of $(\mathbf{P}_{\mathsf{IP}}, \mathbf{V}_{\mathsf{IP}})$. We show that the IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ has soundness error

$$\beta_{\mathsf{IOP}} = \max\left\{\frac{1}{2}, \binom{2\mathsf{k}}{\mathsf{k}} \cdot \beta\right\} \quad .$$

The above expression can be made constant by applying $\mathrm{poly}(|\mathbb{x}|)$ parallel repetitions to $(\mathbf{P}_{\mathsf{IP}}, \mathbf{V}_{\mathsf{IP}})$ prior to applying our transformations, until it has soundness error $\beta' \leq \left(2 \cdot \binom{2\mathsf{k}}{\mathsf{k}}\right)^{-1}$.

Fix $\mathbb{x} \notin L$ and a cheating IOP prover $\tilde{\mathbf{P}}_{\mathsf{IOP}}$. A fixed transcript of the IOP has the structure:

$$\left(S_0, \rho_1, S_1, \ldots, \rho_{2\mathsf{k}}, S_{2\mathsf{k}}, (T_0, \ldots, T_{2\mathsf{k}})\right) \ .$$

Given such a transcript, we say that an index i is *consistent* if: (a) $S_{i-1} = T_{i-1}$ and $S_i = T_i$; and (b) for every $\mathsf{tr} \in S_{i-1}$ there is a message $a_{\mathsf{tr},i}$ such that $(\mathsf{tr}||\rho_i||a_{\mathsf{tr},i}) \in S_i$.

Conditioned on the event that the transcript generated during the interaction has less than k consistent indices i, $\mathbf{V}_{\mathsf{IOP}}$ rejects with probability at least $1/2$ due to its check in Item 4b. We are thus left to analyze the probability that $\mathbf{V}_{\mathsf{IOP}}$ rejects conditioned on the event that the generated transcript has at least k consistent indices.

Fix indices $i_1 < \cdots < i_{\mathsf{k}}$ and suppose that all these indices are consistent with respect to the transcript. By the definition of consistency, for every $j \in [\mathsf{k}]$, $S_{i_j-1} = T_{i_j-1}$, $S_{i_j} = T_{i_j}$ and $(\mathsf{tr}||\rho_{i_j}||a_{\mathsf{tr},i_j}) \in S_{i_j}$ for every $\mathsf{tr} \in S_{i_j-1}$. This implies that there exist $a_{i_1}, \ldots, a_{i_{\mathsf{k}}}$ such that $(\rho_{i_1}||a_{i_1}||\cdots||\rho_{i_{\mathsf{k}}}||a_{i_{\mathsf{k}}}) \in T_{i_{\mathsf{k}}}$. By the subset consistency check, $\mathbf{V}_{\mathsf{IOP}}$ accepts only if $T_{i_{\mathsf{k}}} \subseteq T_{2\mathsf{k}}$, in which case $(\rho_{i_1}||a_{i_1}||\cdots||\rho_{i_{\mathsf{k}}}||a_{i_{\mathsf{k}}}) \in T_{2\mathsf{k}}$. This transcript was generated interactively by the prover and verifier and hence, by the soundness of the IP, $\mathbf{V}_{\mathsf{IP}}(\mathbb{x}, \rho_{i_1}||a_{i_1}||\cdots||\rho_{i_{\mathsf{k}}}||a_{i_{\mathsf{k}}}) = 1$ with probability at most β. If the transcript is rejecting, then this is detected by $\mathbf{V}_{\mathsf{IOP}}$ in its membership check.

The previous analysis holds for fixed indices $i_1 < \cdots < i_{\mathsf{k}}$. By applying the union bound to all choices of indices, we have that, conditioned on the transcript generated having at least k consistent rounds, $\mathbf{V}_{\mathsf{IOP}}$ accepts with probability at most $\binom{2\mathsf{k}}{\mathsf{k}} \cdot \beta$.

Putting together both (non-intersecting) events of the number of rounds consistent with the generated transcript, we conclude that $\mathbf{V}_{\mathsf{IOP}}$ accepts with probability at most $\max\{\frac{1}{2}, \binom{2\mathsf{k}}{\mathsf{k}} \cdot \beta\}$.

Remark 2.1. As alluded to above, the above transformation can be viewed as two steps: (i) apply a transformation that ensures that (with constant probability) any transcript tr' that agrees with the real interaction transcript tr on at least half of the rounds is rejecting; and (ii) apply the strawman protocol to the new protocol.

We believe that this property of rejecting transcripts that are close to the real interaction transcript, which we call "robust soundness error", is of independent interest and is likely to have further applications. A formal definition of (round-)robust soundness follows.

Definition 2.2 ((Round-)Robust soundness)**.** *Let* IP $= (\mathbf{P}_{\mathsf{IP}}, \mathbf{V}_{\mathsf{IP}})$ *be an IP for a relation* R*.* IP *has* **(round-)robust soundness error** ε **with distance** δ *if for every instance* $\mathbb{x} \notin L(R)$ *and malicious prover* $\tilde{\mathbf{P}}_{\mathsf{IP}}$*:*

$$\Pr_{\rho_1,\ldots,\rho_{\mathsf{k}}}\left[\exists\, \mathsf{tr}' : \Delta_{\mathsf{Round}}(\mathsf{tr}, \mathsf{tr}') \leq \delta \ \wedge\ \mathbf{V}_{\mathsf{IP}}(\mathbb{x}, \mathsf{tr}') = 1 \;\middle|\; \begin{array}{r} a_1 \leftarrow \tilde{\mathbf{P}}_{\mathsf{IP}}(\rho_1) \\ \vdots \\ a_{\mathsf{k}} \leftarrow \tilde{\mathbf{P}}_{\mathsf{IP}}(\rho_1, \ldots, \rho_{\mathsf{k}}) \\ \mathsf{tr} := (\rho_1, a_1, \ldots, \rho_{\mathsf{k}}, a_{\mathsf{k}}) \end{array}\right] \leq \varepsilon \ ,$$

where $\Delta_{\mathsf{Round}}(\mathsf{tr}, \mathsf{tr}')$ *is the fraction of rounds on which* tr *and* tr' *differ.*

Other notions of robustness for IPs can be considered by using different distance measures between tr and tr' (e.g., Hamming distance, distance between groups of rounds, and so on).

Achieving query complexity $O(1)$ over the binary alphabet. The verifier in the IOP described above reads $O(1)$ rounds (in their entirety) from the interaction transcript, rather than $O(1)$ bits in total. We additionally achieve this latter goal by building on a result in [ACY22].

In more detail, [ACY22] transforms a k-round public-coin IP into an $O(\mathsf{k})$-round public-coin IOP with polynomial proof length over the binary alphabet and where the IOP verifier reads $O(1)$ bits from each round. We extend this to transform a k-round IOP whose verifier reads q of the k rounds into a $O(\mathsf{k})$-round IOP whose verifier reads $O(1)$ bits from each of $O(\mathsf{q})$ rounds. See Section 7 for more details.

2.2.2 IOPs from general IPs

The transformation described in the previous section works for $O(\log |\mathbb{x}|)$-round IPs. However, it cannot be directly applied to IPs with more rounds because the proof length (and thus also the verifier running time) would be more than polynomial.

Nevertheless, we extend the transformation to work for any public-coin IP while achieving a moderate improvement on the number of read rounds. In the main loop of the IOP, rather than advancing each IP transcript in S_{i-1} by one round, advance it by $O(\mathsf{k}/\log |\mathbb{x}|)$ rounds before inserting the resulting transcripts into the set S_i. During its decision phase, the IOP verifier chooses an index i and reads the entire $O(\mathsf{k}/\log |\mathbb{x}|)$-round interaction done during this iteration of the IOP. Completeness and soundness of this new transformation are similar to the one presented in the previous section, but now the verifier reads $O(\mathsf{k}/\log |\mathbb{x}|)$ rounds. The rest of the efficiency parameters are similar to the IOP described in the previous section, except that proof length is polynomial regardless of k.

After applying the (adapted) transformation of [ACY22], this process yields a $O(\mathsf{k})$-round IOP with query complexity $O(\mathsf{k}/\log |\mathbb{x}|)$ (over the binary alphabet). This concludes the proof sketch of Lemma 1.

Can we do better? The construction described in this section doubles the number of transcripts stored in the set S_i relative to S_{i-1}. This causes a blow-up in parameters and is the reason why this approach fails in constructing $O(1)$-query IOPs from IPs with super-logarithmic round complexity. Intuitively, if we could reduce this doubling then we may be able to modify the transformation to get $O(1)$-query IOPs. While we do not achieve this for *general* IP, we show that, using this intuition, we can construct $O(1)$-query IOPs for a rich class of relations that are *interactively reducible*. We discuss this notion, and corresponding new IOP constructions, in the following section.

2.3 Interactive reducibility

In Section 2.2.1 we described how to transform an IP into an IOP with $O(1)$ query complexity. The new protocol kept track of a set containing all partial transcripts of the IP generated so far in the protocol. In every round, every partial transcript in the set was advanced by one round, and the newly advanced transcripts were added to the set held previously. This meant that in every round, the set contains twice as many transcripts as in the previous round. As we require polynomial proof length and verifier running time, this technique was unable to allow reading of $O(1)$ rounds for IPs with greater than $O(\log |\mathbb{x}|)$ rounds.

Intuitively, suppose it were possible to take multiple transcripts and reduce them into a single transcript that in some sense preserves soundness of all of the transcripts combined. In that case, this issue could be bypassed, and the protocol would work for IPs with super-logarithmic round complexity. In more detail, suppose we want to reduce transcripts $\mathsf{tr}_1, \ldots, \mathsf{tr}_\mathsf{t}$ into a new transcript tr'. The *acceptance probability* of a transcript prefix tr_i is the maximum over all prover strategies of the probability that the verifier will end up accepting when continuing interaction with the prover

from transcript tr_i. Roughly, we require that: (a) if the acceptance probability of every tr_i is 1, then so is the acceptance probability of tr'; and (b) if there exists some transcript tr_i with small acceptance probability, then (with high probability) the acceptance probability of tr' is also small.

In this section, we introduce the concept of *interactive reducibility*, which formally captures this intuition. We exemplify this in Section 2.3.1 by describing how to reduce multiple transcripts of the sumcheck protocol into a single transcript. Then, in Section 2.3.2, we formally define interactive reducibility. In Section 2.3.3, we show how to adapt the protocol described in Section 2.2.1 to work with interactive reductions and bypass the blow-up in the original protocol. Finally, in Section 2.3.4, we discuss relations known to have interactive reducibility.

2.3.1 An interactive reduction for sumcheck

We exemplify the notion of interactive reducibility in the case of the sumcheck protocol [LFKN92]. Below we review this protocol, and then explain how to reduce multiple transcripts into one transcript via an interactive reduction.

The sumcheck protocol. The verifier has query access to a n-variate polynomial p of individual degree d over some field $\mathbb{F}$. The goal of the verifier is to test, for a given field element γ, whether

$$\sum_{\alpha_1,\ldots,\alpha_n\in\{0,1\}} p(\alpha_1,\ldots,\alpha_n) = \gamma \ .$$

The protocol begins with the prover sending a polynomial $\tilde{p}_1$ of degree d, claimed to equal $p_1(X) := \sum_{\alpha_2,\ldots,\alpha_n\in\{0,1\}} p(X,\alpha_2,\ldots,\alpha_n)$. The verifier checks that $\tilde{p}_1(0)+\tilde{p}_1(1)=\gamma$ (rejecting if not), samples a random field element r_1, and sends it to the prover. Both parties define $\gamma_1 := \tilde{p}_1(r_1)$.

This one-round interaction leads to a new sumcheck claim

$$\sum_{\alpha_2,\ldots,\alpha_n\in\{0,1\}} p(r_1,\alpha_2,\ldots,\alpha_n) = \gamma_1 \ ,$$

that has the following properties: (a) if the original claim is true then the new claim is also true; and (b) if the original claim is false then with high probability the new claim is also false.

Next, the prover sends $\tilde{p}_2$ claimed to equal $p_2(X) := \sum_{\alpha_3,\ldots,\alpha_n\in\{0,1\}} p(r_1,X,\alpha_3,\ldots,\alpha_n)$ and the protocol repeats as before. This process continues until the n variables are fixed to some field elements $(r_1,\ldots,r_n)$, and the problem has been reduced to checking that $p(r_1,\ldots,r_n)=\gamma_n$, which the verifier can check via one query to the polynomial p.

One can associate a round j of the protocol with a list of field elements $(r_1,\ldots,r_j)$ and a claimed sum γ_j, and think of that round as "reducing" a claim $\mathbb{z} = ((r_1,\ldots,r_j),\gamma_j)$ that $\sum_{\alpha_{j+1},\ldots,\alpha_n\in\{0,1\}} p(r_1,\ldots,r_j,\alpha_{j+1},\ldots,\alpha_n) = \gamma_j$ into a new claim $\mathbb{z}' = ((r_1,\ldots,r_{j+1}),\gamma_{j+1})$ that $\sum_{\alpha_{j+2},\ldots,\alpha_n\in\{0,1\}} p(r_1,\ldots,r_{j+1},\alpha_{j+2},\ldots,\alpha_n) = \gamma_{j+1}$.

Reducing multiple sumcheck claims. We are given claims $\mathbb{z}_1,\ldots,\mathbb{z}_{\mathsf{t}}$ where each $\mathbb{z}_i$ consists of $(r_{i,1},\ldots,r_{i,j})$ and a claimed sum $\gamma_{i,j}$. We seek to reduce these t claims into a single claim $\mathbb{z}' = ((r'_1,\ldots,r'_{j+1}),\gamma'_{j+1})$ such that: (a) If each $\mathbb{z}_i$ is a true statement, then $\mathbb{z}'$ is a true statement; and (b) If there is some $\mathbb{z}_i$ that is a false statement, then with high probability $\mathbb{z}'$ is a false statement. Notice that in order to merge multiple sumcheck transcripts it suffices to merge multiple sumcheck claims $\mathbb{z}_1,\ldots,\mathbb{z}_{\mathsf{t}}$. Thus we will focus on merging such claims.

Before describing the reduction, we define a polynomial $I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathsf{t}}}\colon \mathbb{F}\to\mathbb{F}^j$ that represents a curve through the t points described by the instances $\mathbb{z}_1,\ldots,\mathbb{z}_{\mathsf{t}}$. That is, $I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathsf{t}}}(i) = (r_{i,1},\ldots,r_{i,j})$ for

every $i \in [\mathsf{t}]$ (here we implicitly associate the set $[\mathsf{t}]$ with an arbitrary set $S \subseteq \mathbb{F}$ of size t, known to all parties). By interpolation, the degree of $I_{\mathbb{z}_1,\ldots,\mathbb{z}_\mathsf{t}}$ is less than t.

Given this definition, we describe the interactive reduction for the sumcheck protocol.

- Prover: Send the polynomial $g \in \mathbb{F}[X_1, X_2]$ defined as:

$$g(X_1, X_2) := \sum_{\alpha_{j+2},\ldots,\alpha_n \in \{0,1\}} p(I_{\mathbb{z}_1,\ldots,\mathbb{z}_\mathsf{t}}(X_1), X_2, \alpha_{j+2}, \ldots, \alpha_n) \ . \tag{1}$$

- Verifier: Receive a bivariate polynomial $\tilde{g} \in \mathbb{F}[X_1, X_2]$ of degree at most $j \cdot d \cdot (\mathsf{t} - 1)$ in X_1 and degree at most d in X_2.
  1. *Consistency:* Check that for every $i \in [\mathsf{t}]$ it holds that $\sum_{\alpha \in \{0,1\}} \tilde{g}(i, \alpha) = \gamma_{i,j}$. (Reject if not.)
  2. *Generate new instance:*
     (a) Sample uniformly random field elements $\rho, r^* \leftarrow \mathbb{F}$ and send them to the prover.
     (b) Set $(r'_1, \ldots, r'_j) := I_{\mathbb{z}_1,\ldots,\mathbb{z}_\mathsf{t}}(\rho)$ and $\gamma_{j+1} := \tilde{g}(\rho, r^*)$ and output the new instance

$$\mathbb{z}' := \Big((r'_1, \ldots, r'_j, r^*), \gamma_{j+1}\Big) \ .$$

Analysis. It follows straightforwardly from the protocol that, if $\mathbb{z}_1, \ldots, \mathbb{z}_\mathsf{t}$ are all true statements and the prover acts honestly, then $\mathbb{z}'$ is a true statement. We show that if any one of the statements $\mathbb{z}_1, \ldots, \mathbb{z}_\mathsf{t}$ is false then with high probability so is $\mathbb{z}'$.

Let g be as defined in Equation (1) with respect to $\mathbb{z}_1, \ldots, \mathbb{z}_\mathsf{t}$. Suppose that $\mathbb{z}_i$ is a false claim (i.e., $\sum_{\alpha_{j+1},\ldots,\alpha_n \in \{0,1\}} p(r_{i,1}, \ldots, r_{i,j}, \alpha_{j+1}, \ldots, \alpha_n) \neq \gamma_{i,j}$). Then, by definition,

$$\sum_{\alpha \in \{0,1\}} g(i, \alpha) = \sum_{\alpha_{j+1},\ldots,\alpha_n \in \{0,1\}} p(r_{i,1}, \ldots, r_{i,j}, \alpha_{j+1}, \ldots, \alpha_n) \neq \gamma_{i,j} \ .$$

During its consistency check, the verifier checks that $\sum_{\alpha \in \{0,1\}} \tilde{g}(i, \alpha) = \gamma_{i,j}$. Thus, in order for the verifier to not reject, a cheating prover must send $\tilde{g} \neq g$. By the Schwartz–Zippel lemma, since g and $\tilde{g}$ are low-degree polynomials (provided that the degree d and the number of instances being reduced t are small with respect to $|\mathbb{F}|$), the probability that the uniformly chosen ρ and r^* are such that $\tilde{g}(\rho, r^*) = g(\rho, r^*)$ is small. Whenever $\tilde{g}(\rho, r^*) \neq g(\rho, r^*)$ we have that

$$\sum_{\alpha_{j+2},\ldots,\alpha_n \in \{0,1\}} p(r'_1, \ldots, r'_j, r^*, \alpha_{j+2}, \ldots, \alpha_n) = g(\rho, r^*) \neq \tilde{g}(\rho, r^*) = \gamma_{j+1} \ ,$$

and so the resulting statement $\mathbb{z}' := ((r'_1, \ldots, r'_j, r^*), \gamma_{j+1})$ is false.

2.3.2 Defining interactive reducibility

We define interactive reducibility, which captures the capability of merging multiple transcripts/instances into a single transcript/instance while preserving correctness and soundness.

Definition 1. *An ℓ-round public-coin protocol $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ where $\mathbf{V}_{\mathsf{IR}}$ runs in polynomial time is an* **interactive reduction** *for a relation R with k predicates and soundness error ε if there exists a sequence of predicates $f_0, f_1, \ldots, f_\mathsf{k}$ such that the following holds.*

- **Completeness:** *For every* $(\mathbb{x}, \mathbb{w}) \in R$, $j \in [\mathsf{k}]$, *and* $\mathbb{z}_1, \ldots, \mathbb{z}_\mathsf{t}$ *such that* $f_{j-1}(\mathbb{x}, \mathbb{z}_i) = 1$ *for every* $i \in [\mathsf{t}]$, *it holds that:*

$$\Pr\left[\ f_j(\mathbb{x}, \mathbb{z}') = 1 \ \middle|\ \mathbb{z}' \leftarrow \langle \mathbf{P}_{\mathsf{IR}}(\mathbb{x}, \mathbb{w}, \mathbb{z}_1, \ldots, \mathbb{z}_\mathsf{t}), \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_1, \ldots, \mathbb{z}_\mathsf{t})\rangle\ \right] = 1 \ .$$

- **Soundness:** *For every* $\mathbb{x} \notin L(R)$, $j \in [\mathsf{k}]$, *and* $\mathbb{z}_1, \ldots, \mathbb{z}_\mathsf{t}$, *if there exists* $i \in [\mathsf{t}]$ *where* $f_{j-1}(\mathbb{x}, \mathbb{z}_i) = 0$ *then for every (computationally unbounded)* $\tilde{\mathbf{P}}_{\mathsf{IR}}$ *it holds that:*

$$\Pr\left[\ f_j(\mathbb{x}, \mathbb{z}') = 1 \ \middle|\ \mathbb{z}' \leftarrow \langle \tilde{\mathbf{P}}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_1, \ldots, \mathbb{z}_\mathsf{t})\rangle\ \right] \leq \varepsilon(\mathbb{x}, \mathsf{t}) \ .$$

- **Relation identity:** $f_0(\mathbb{x}, \mathbb{z}) = 1$ *if and only if* $\mathbb{x} \in L(R)$.
- **Triviality:** $f_\mathsf{k}(\mathbb{x}, \mathbb{z})$ *can be computed in time* $\mathrm{poly}(|\mathbb{x}|, |\mathbb{z}|)$.

We call $\mathbb{x}$ *the* base instance *and* $\mathbb{z}_1, \ldots, \mathbb{z}_\mathsf{t}$ round instances.

An interactive reduction $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ has (polynomially) **bounded output length** if there exists $c \in \mathbb{N}$ such that, for every base instance $\mathbb{x}$, witness $\mathbb{w}$, and round instances $\mathbb{z}_1, \ldots, \mathbb{z}_\mathsf{t}$, the new instance $\mathbb{z}'$ output by $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ on inputs $(\mathbb{x}, \mathbb{w}, \mathbb{z}_1, \ldots, \mathbb{z}_\mathsf{t})$ has length at most $|\mathbb{x}|^c$.

2.3.3 IOPs from interactive reducibility

We show that any relation with an ℓ-round interactive reduction with k predicates (and bounded output length) has a $(\ell \cdot \mathsf{k})$-round public-coin IOP with query complexity $O(\mathsf{k})$. This is a variation of the protocol described in Section 2.2, adapted to work with interactive reducibility. For simplicity, in this overview, we present the protocol only for the case $\ell = 1$ (such as the sumcheck protocol).

To aid with notation, in the description of the protocol, we replace the set S_i (which in Section 2.2.1 contained the set of all transcripts generated up until the i-th iteration) with an array A_i where $A_i[j]$ contains all of the round instances generated in the i-th iteration that are associated with the j-th predicate of the interactive reduction. In iteration i, the interactive reduction will be run k times in parallel, where for every $j \in [\mathsf{k}]$ we run given the round instances stored in $A_{i-1}[j]$.

The protocol. Let $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ be a one-round interactive reduction for R with k predicates. The IOP prover $\mathbf{P}_{\mathsf{IOP}}$ receives as input an instance $\mathbb{x}$ and witness $\mathbb{w}$, and the IOP verifier $\mathbf{V}_{\mathsf{IOP}}$ receives as input the instance $\mathbb{x}$. They interact as follows.

1. For every $i \in \{0, \ldots, 2\mathsf{k}\}$, $\mathbf{P}_{\mathsf{IOP}}$ defines the $(\mathsf{k}+1)$-entry array A_i as follows

$$A_i[j] := \begin{cases} \{\bot\} & \text{if } j = 0 \\ \emptyset & \text{if } j \in \{1, \ldots, \mathsf{k}\} \end{cases} \ .$$

   The set $A_i[j]$ will store all instances corresponding to f_j collected by iteration i of the protocol.

2. For $i = 1, \ldots, 2\mathsf{k}$:
   (a) $\mathbf{P}_{\mathsf{IOP}}$ sends A_{i-1} to $\mathbf{V}_{\mathsf{IOP}}$.
   (b) $\mathbf{V}_{\mathsf{IOP}}$ sends a random $\rho_i \leftarrow \{0,1\}^\mathsf{r}$ (this corresponds to a message of $\mathbf{V}_{\mathsf{IR}}$).
   (c) $\mathbf{P}_{\mathsf{IOP}}$ sends $a_{i,j} := \mathbf{P}_{\mathsf{IR}}(\mathbb{x}, \mathbb{w}, A_{i-1}[j-1], \rho_i)$ for all $j \in [\mathsf{k}]$, and sets $A_i[j] := A_{i-1}[j] \cup \{\mathbb{z}_{i,j}\}$ where $\mathbb{z}_{i,j} := \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, A_{i-1}[j-1], \rho_i, a_{i,j})$ is the output of the interactive reduction verifier given base instance $\mathbb{x}$, round instances $A_{i-1}[j-1]$, verifier randomness ρ_i, and prover reply $a_{i,j}$.

3. $\mathbf{P}_{\mathsf{IOP}}$ sends $A_{2\mathsf{k}}$ and, for every $i \in \{0, \ldots, 2\mathsf{k}\}$, sends $B_i := A_i$. This concludes the interaction.
4. In the decision phase, $\mathbf{V}_{\mathsf{IOP}}$ is given oracle access to a transcript with the following structure:

$$(A_0, \rho_1, (a_{1,1}, \ldots, a_{1,\mathsf{k}}), A_1, \ldots, \rho_{2\mathsf{k}}, (a_{2\mathsf{k},1}, \ldots, a_{2\mathsf{k},\mathsf{k}}), A_{2\mathsf{k}}, (B_0, \ldots, B_{2\mathsf{k}})) \quad .$$

$\mathbf{V}_{\mathsf{IOP}}$ performs the checks below.

(a) *Subset consistency.* Read the arrays $B_0, B_1, \ldots, B_{2\mathsf{k}}$ in their entirety. For every $i \in [2\mathsf{k}]$ and $j \in \{0, \ldots, \mathsf{k}\}$ check that $B_{i-1}[j] \subseteq B_i[j]$.

(b) *Transcript consistency.* Sample a random $i \in [2\ell]$. Read the arrays A_{i-1} and A_i sent by $\mathbf{P}_{\mathsf{IOP}}$ and the interaction ρ_i and $(a_{i,1}, \ldots, a_{i,\ell})$.

i. Check that $A_{i-1} = B_{i-1}$ and $A_i = B_i$.

ii. For every $j \in [\mathsf{k}]$, check that $A_i[j] = A_{i-1}[j] \cup \{\mathbb{z}'_{i,j}\}$ where

$$\mathbb{z}'_{i,j} := \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, A_{i-1}[j-1], \rho_i, a_{i,j}) \ ,$$

is the output of the interactive reduction verifier given base instance $\mathbb{x}$, round instances $A_{i-1}[j-1]$, verifier randomness ρ_i, and prover reply $a_{i,j}$. (Reject if $\mathbf{V}_{\mathsf{IR}}$ rejects.)

(c) *Final predicate holds.* Check that $f_{\mathsf{k}}(\mathbb{x}, \mathbb{z}) = 1$ for every $\mathbb{z} \in B_{2\mathsf{k}}[\mathsf{k}]$.

Analysis. The protocol has perfect completeness and soundness error $\max\{\frac{1}{2}, \binom{2\cdot\mathsf{k}}{\mathsf{k}} \cdot \mathsf{k} \cdot \epsilon\}$ where k is the number of predicates and ϵ is the soundness of the interactive reduction respectively. This can be shown in a similar manner to that described in Section 2.2.1. The main difference between the two protocols is in the analysis of the proof length. In the protocol of Section 2.2.1 the number of sent transcripts doubled in each round. In contrast, in the new protocol, one round instance is added for each predicate. In more detail, for every $i \in [2\mathsf{k}]$ and $j \in [\mathsf{k}]$, we have $|A_i[j]| = |A_{i-1}[j]| + 1$. Since $\mathsf{k} = \text{poly}(|\mathbb{x}|)$, the total number of round instances generated and sent is polynomial in $|\mathbb{x}|$. If the interactive reduction has bounded output length, then each of these round instances has polynomially-bounded length. We can therefore conclude that the overall proof length is $\text{poly}(|\mathbb{x}|)$.

2.3.4 Relations with interactive reducibility

Several relations of interest have interactive reductions.

General IPs. We show that any relation with a k-round interactive proof has an m-round interactive reduction with k/m predicates (for any m that divides k). To see this, consider round instances $\mathbb{z}$ that are sets of j-message partial transcripts of the IP. The interactive reduction advances each of the transcripts in the set $\mathbb{z}$ by m rounds, as in the IP. The predicates are defined with respect to the "state function" of the IP, which roughly denotes whether the prover has an accepting strategy with respect to this transcript or whether no strategy will cause the verifier to accept with high probability (over the remaining interaction). See Section 3.2 for a formal definition of the state function of an IP through the concept of round-by-round soundness.

Notice that if this interactive reduction is used in the protocol of Section 2.3.3, this yields the protocol described in Section 2.2.1. This interactive reduction does not have bounded output length since the new round instance stores all of the previous transcripts and their continuations. Therefore the resulting IOP does not achieve $O(1)$ total query complexity.

Sumcheck protocol. Using the ideas described in Section 2.3.2, we show that any relation that can be reduced into k-variate sumcheck has a one-round interactive reduction with k predicates and

bounded output length. As a result, any relation that can be reduced into k-variate sumcheck has a $O(\mathsf{k})$-round public-coin IOP with query complexity $O(1)$.

Shamir's protocol. Shamir's protocol [Sha92] gives an IP for all of PSPACE, thereby showing the IP = PSPACE theorem. Extending the ideas developed in Section 2.3.2, we show a one-round interactive reduction with bounded output length and polynomially-many predicates for Shamir's protocol. This establishes that every language in PSPACE has a poly($|\mathbb{x}|$)-round public-coin IOP with query complexity $O(1)$.

Future directions. We leave the exploration of what other relations have interactive reductions to future work. Following the extensive use of polynomials in both the sumcheck protocol and Shamir's protocol, it seems likely that these techniques can be adapted to also work for low-depth circuits through the delegation protocol in [GKR15].

3 Preliminaries

We consider proof systems for *binary* relations. A binary relation R is a set of tuples $(\mathbb{x}, \mathbb{w})$ where $\mathbb{x}$ is the instance and $\mathbb{w}$ the witness. The corresponding language $L(R)$ is the set of $\mathbb{x}$ for which there exists $\mathbb{w}$ such that $(\mathbb{x}, \mathbb{w}) \in R$. In some theorems we highlight important parameters with a yellow background.

3.1 Interactive oracle proofs

Interactive Oracle Proofs (IOPs) [BCS16; RRR16] are information-theoretic proof systems that combine aspects of Interactive Proofs [Bab85; GMR89] and Probabilistically Checkable Proofs [BFLS91; FGLSS91; AS98; ALMSS98], and also generalize the notion of Interactive PCPs [KR08]. Below we describe *public-coin* IOPs.

A $\mathsf{k}_{\mathsf{IOP}}$-round public-coin IOP works as follows. For each round $i \in [\mathsf{k}_{\mathsf{IOP}}]$, the verifier sends a uniformly random message $\boldsymbol{\rho}_i$ to the prover; then the prover sends a proof string Π_i to the verifier. After $\mathsf{k}_{\mathsf{IOP}}$ rounds of interaction, the verifier makes some queries to the proof strings $\Pi_1, \ldots, \Pi_{\mathsf{k}_{\mathsf{IOP}}}$ sent by the prover, and then decides if to accept or to reject.

In more detail, let $\mathsf{IOP} = (\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ be a tuple where $\mathbf{P}_{\mathsf{IOP}}$ (the prover) is an interactive algorithm, and $\mathbf{V}_{\mathsf{IOP}}$ (the verifier) is an interactive oracle algorithm. We say that IOP is a *public-coin IOP* for a binary relation R with $\mathsf{k}_{\mathsf{IOP}}$ rounds and soundness error β_{IOP} if the following holds.

- **Completeness.** For every $(\mathbb{x}, \mathbb{w}) \in R$,

$$\Pr_{\rho_1,\ldots,\rho_{\mathsf{k}_{\mathsf{IOP}}},\rho_{\mathsf{dc}}}\left[\mathbf{V}_{\mathsf{IOP}}^{\Pi_1,\ldots,\Pi_{\mathsf{k}_{\mathsf{IOP}}},\rho_1,\ldots,\rho_{\mathsf{k}_{\mathsf{IOP}}}}(\mathbb{x};\rho_{\mathsf{dc}}) = 1 \;\middle|\; \begin{array}{r} \Pi_1 \leftarrow \mathbf{P}_{\mathsf{IOP}}(\mathbb{x},\mathbb{w},\rho_1) \\ \vdots \\ \Pi_{\mathsf{k}_{\mathsf{IOP}}} \leftarrow \mathbf{P}_{\mathsf{IOP}}(\mathbb{x},\mathbb{w},\rho_1,\ldots,\rho_{\mathsf{k}_{\mathsf{IOP}}}) \end{array}\right] = 1 \ .$$

- **Soundness.** For every $\mathbb{x} \notin L(R)$ and unbounded malicious prover $\tilde{\mathbf{P}}_{\mathsf{IOP}}$,

$$\Pr_{\rho_1,\ldots,\rho_{\mathsf{k}_{\mathsf{IOP}}},\rho_{\mathsf{dc}}}\left[\mathbf{V}_{\mathsf{IOP}}^{\tilde{\Pi}_1,\ldots,\tilde{\Pi}_{\mathsf{k}_{\mathsf{IOP}}},\rho_1,\ldots,\rho_{\mathsf{k}_{\mathsf{IOP}}}}(\mathbb{x};\rho_{\mathsf{dc}}) = 1 \;\middle|\; \begin{array}{r} \tilde{\Pi}_1 \leftarrow \tilde{\mathbf{P}}_{\mathsf{IOP}}(\rho_1) \\ \vdots \\ \tilde{\Pi}_{\mathsf{k}_{\mathsf{IOP}}} \leftarrow \tilde{\mathbf{P}}_{\mathsf{IOP}}(\rho_1,\ldots,\rho_{\mathsf{k}_{\mathsf{IOP}}}) \end{array}\right] \leq \beta_{\mathsf{IOP}}(|\mathbb{x}|) \ .$$

Complexity measures. We consider several complexity measures beyond soundness error. All of these complexity measures are implicitly functions of the instance $\mathbb{x}$.

- *proof length* $\mathsf{l}_{\mathsf{IOP}}$: the total number of bits in $\Pi_1, \ldots, \Pi_{\mathsf{k}_{\mathsf{IOP}}}$.
- *queries* $\mathsf{q}_{\mathsf{IOP}}$: the number of bits read by the verifier from $\rho_1, \Pi_1, \ldots, \rho_{\mathsf{k}_{\mathsf{IOP}}}, \Pi_{\mathsf{k}_{\mathsf{IOP}}}$.
- *interaction randomness length* $\mathsf{r}_{\mathsf{int}}$: the total number of bits in $\rho_1, \ldots, \rho_{\mathsf{k}_{\mathsf{IOP}}}$.
- *decision randomness length* r_{dc}: The number of bits in ρ_{dc}.
- *prover time* $\mathsf{pt}_{\mathsf{IOP}}$: $\mathbf{P}_{\mathsf{IOP}}$ runs in time $\mathsf{pt}_{\mathsf{IOP}}$.
- *verifier time* $\mathsf{vt}_{\mathsf{IOP}}$: $\mathbf{V}_{\mathsf{IOP}}$ runs in time $\mathsf{vt}_{\mathsf{IOP}}$.
- *decision complexity* $\mathsf{dt}_{\mathsf{IOP}}$: Following the choice of queries, $\mathbf{V}_{\mathsf{IOP}}$ runs in time $\mathsf{dt}_{\mathsf{IOP}}$ to decide whether to accept or reject.

Round-query IOPs and IPs. A round-query IOP with round-query complexity $\mathsf{q}_{\mathsf{rnd}}$ is an IOP in which the verifier reads every symbol of $\mathsf{q}_{\mathsf{rnd}}$ rounds from the interaction. An interactive proof (IP) is

a round-query IOP with round-query complexity equal to the number of rounds in the protocol (i.e., the verifier reads every symbol in the interaction). Unless explicitly stated otherwise, we assume that IPs have no decision randomness.

3.2 Round-by-round soundness

Definition 3.1. *Let* $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ *be an IOP for a relation* R*. A* **state function** *for* $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ *with decision error* δ_{dc} *is a (possibly inefficient) Boolean function that receives as input an instance* $\mathbb{x}$ *and a transcript* tr *and outputs a bit for which the following holds.*

- *Empty transcript: if* $\mathsf{tr} = \emptyset$ *is the empty transcript then* $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 0$ *if and only if* $\mathbb{x} \notin L(R)$*.*
- *Prover moves: If* $\mathbb{x} \notin L(R)$*,* tr *is a transcript where the prover is about to move and* $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 0$*, then for any potential prover message* a*,* $\mathsf{state}(\mathbb{x}, \mathsf{tr}||a) = 0$*.*
- *Full transcript: if* tr *is a full transcript and* $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 0$ *then* $\Pr_{\rho_{\mathsf{dc}}}[\mathbf{V}^{\mathsf{tr}}_{\mathsf{IOP}}(\mathbb{x}; \rho_{\mathsf{dc}}) = 1] \leq \delta_{\mathsf{dc}}$*.*

Definition 3.2. *An IOP* $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ *with* $\mathsf{k}_{\mathsf{IOP}}$ *rounds for a relation* R *has* $(\beta_{\mathrm{rbr}}, \delta_{\mathsf{dc}})$**-round-by-round soundness** *if there exists a state function* state *with decision error* δ_{dc} *such that for all* $\mathbb{x} \notin L(R)$*, every* $i \in [\mathsf{k}_{\mathsf{IOP}}]$*, and every transcript* tr *of the first* i *rounds where the verifier is about to move and* $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 0$ *it holds that*

$$\Pr_{\rho}[\mathsf{state}(\mathbb{x}, \mathsf{tr}||\rho) = 1] \leq \beta_{\mathrm{rbr}} \ .$$

We call β_{rbr} *the interaction error and* δ_{dc} *the decision error. If* $\delta_{\mathsf{dc}} = 0$*, we omit* δ_{dc} *and simply say that* IOP *has round-by-round soundness* β_{rbr} *which aligns with the standard definition of round-by-round soundness for IPs.*

Notice that if a k-round IOP has $(\beta_{\mathrm{rbr}}, \delta_{\mathsf{dc}})$-round-by-round soundness, then it has soundness error at most $\mathsf{k} \cdot \beta_{\mathrm{rbr}} + \delta_{\mathsf{dc}}$. Additionally, if a k-round IOP has constant soundness error, then it has $(c^{1/\mathsf{k}}, O(1))$-round-by-round soundness error for some constant $0 < c < 1$.

The following lemma is a generalization of [CCHLRR18], Corollary 5.7, and shows that one can achieve small round-by-round soundness errors for IOPs with relatively little overhead.

Lemma 3.3. *Let* $m, t \in \mathbb{N}$ *be parameters and* R *be a relation with a non-adaptive IOP* $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ *with soundness error* β_{IOP}*. Then* R *has an IOP* $(\mathbf{P}'_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}})$ *with* $(\beta_{\mathsf{IOP}}^{m/2\mathsf{k}_{\mathsf{IOP}}}, \beta_{\mathsf{IOP}}^{t/2})$*-round-by-round soundness and the following parameters:*

IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ for R	
Rounds	$\mathsf{k}_{\mathsf{IOP}}$
Alphabet size	λ_{IOP}
Proof length	$\mathsf{l}_{\mathsf{IOP}}$
Round queries	$\mathsf{q}_{\mathsf{rnd}}$
Queries per round	$\mathsf{q}_{\mathsf{IOP}}$
Total interaction randomness	$\mathsf{r}_{\mathsf{int}}$
Decision randomness	r_{dc}
Soundness	β_{IOP}
Verifier running time	$\mathsf{vt}_{\mathsf{IOP}}$

$\longrightarrow$

IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ for R	
Rounds	$\mathsf{k}_{\mathsf{IOP}}$
Alphabet size	$\lambda_{\mathsf{IOP}}^{m}$
Proof length	$\mathsf{l}_{\mathsf{IOP}}$
Round queries	$t \cdot \mathsf{q}_{\mathsf{rnd}}$
Queries per round	$t \cdot \mathsf{q}_{\mathsf{IOP}}$
Total interaction randomness	$m \cdot \mathsf{r}_{\mathsf{int}}$
Decision randomness	$t \cdot \mathsf{r}_{\mathsf{dc}}$
Round-by-round soundness	$(\beta_{\mathsf{IOP}}^{m/2\mathsf{k}_{\mathsf{IOP}}}, \beta_{\mathsf{IOP}}^{t/2})$
Verifier running time	$m \cdot t \cdot \mathsf{vt}_{\mathsf{IOP}}$

Proof. We augment the IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ as follows: first augment the protocol by making m repetitions of the interaction phase and changing the decision phase to check every one of the

repetitions simultaneously (at the same locations) and accept if and only if the verifier would have accepted in each execution. This IOP is then augmented by running the decision phase for t different times. Let $(\mathbf{P}'_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}})$ be the final IOP.

Perfect completeness and the efficiency parameters of the IOP follow straightforwardly from the construction.

We turn to showing round-by-round soundness. Fix $\mathbb{x} \notin R(L)$ and let β_{IOP} be the soundness error of $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$. Then, we claim that for every malicious prover $\tilde{\mathbf{P}}_{\mathsf{IOP}}$:

$$\Pr_{\rho_1,\ldots,\rho_{\mathsf{k}_{\mathsf{IOP}}}}\left[\Pr_{\rho_{\mathsf{dc}}}\left[\mathbf{V}_{\mathsf{IOP}}^{\tilde{\Pi}_1,\ldots,\tilde{\Pi}_{\mathsf{k}_{\mathsf{IOP}}},\rho_1,\ldots,\rho_{\mathsf{k}_{\mathsf{IOP}}}}(\mathbb{x};\rho_{\mathsf{dc}})=1\right]\geq\beta_{\mathsf{IOP}}^{1/2}\,\middle|\,\begin{array}{r}\tilde{\Pi}_1\leftarrow\tilde{\mathbf{P}}_{\mathsf{IOP}}(\rho_1)\\ \vdots\\ \tilde{\Pi}_{\mathsf{k}_{\mathsf{IOP}}}\leftarrow\tilde{\mathbf{P}}_{\mathsf{IOP}}(\rho_1,\ldots,\rho_{\mathsf{k}_{\mathsf{IOP}}})\end{array}\right]\leq\beta_{\mathsf{IOP}}^{1/2}\ .$$

Indeed, suppose towards contradiction that there exist some $\tilde{\mathbf{P}}_{\mathsf{IOP}}$ such that:

$$\Pr_{\rho_1,\ldots,\rho_{\mathsf{k}_{\mathsf{IOP}}}}\left[\Pr_{\rho_{\mathsf{dc}}}\left[\mathbf{V}_{\mathsf{IOP}}^{\tilde{\Pi}_1,\ldots,\tilde{\Pi}_{\mathsf{k}_{\mathsf{IOP}}},\rho_1,\ldots,\rho_{\mathsf{k}_{\mathsf{IOP}}}}(\mathbb{x};\rho_{\mathsf{dc}})=1\right]\geq\beta_{\mathsf{IOP}}^{1/2}\,\middle|\,\begin{array}{r}\tilde{\Pi}_1\leftarrow\tilde{\mathbf{P}}_{\mathsf{IOP}}(\rho_1)\\ \vdots\\ \tilde{\Pi}_{\mathsf{k}_{\mathsf{IOP}}}\leftarrow\tilde{\mathbf{P}}_{\mathsf{IOP}}(\rho_1,\ldots,\rho_{\mathsf{k}_{\mathsf{IOP}}})\end{array}\right]>\beta_{\mathsf{IOP}}^{1/2}\ .$$

In this case, $\tilde{\mathbf{P}}_{\mathsf{IOP}}$ causes $\mathbf{V}_{\mathsf{IOP}}$ to accept at least whenever the verifier chooses interaction randomness for which the internal probability is true (which happens with probability greater than $\beta_{\mathsf{IOP}}^{1/2}$), and where the decision randomness chosen is chosen causes the verifier to accept (which happens with probability $\beta_{\mathsf{IOP}}^{1/2}$). All together, $\tilde{\mathbf{P}}_{\mathsf{IOP}}$ causes the verifier to accept with probability greater than $\beta_{\mathsf{IOP}}^{1/2}\cdot\beta_{\mathsf{IOP}}^{1/2}=\beta_{\mathsf{IOP}}$ in contradiction to the soundness error of the IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$.

Now notice that in order for in order for the verifier $\mathbf{V}'_{\mathsf{IOP}}$ to accept given proofs, it must succeed in all of its t checks of the decision phase. Moreover, these, t checks must succeed for each one of the m independent repetitions. Hence for every $\tilde{\mathbf{P}}'_{\mathsf{IOP}}$:

$$\Pr_{\rho'_1,\ldots,\rho'_{\mathsf{k}_{\mathsf{IOP}}}}\left[\Pr_{\rho'_{\mathsf{dc}}}\left[\mathbf{V}_{\mathsf{IOP}}^{\prime\tilde{\Pi}'_1,\ldots,\tilde{\Pi}'_{\mathsf{k}_{\mathsf{IOP}}},\rho'_1,\ldots,\rho'_{\mathsf{k}_{\mathsf{IOP}}}}(\mathbb{x};\rho'_{\mathsf{dc}})=1\right]\geq\beta_{\mathsf{IOP}}^{t/2}\,\middle|\,\begin{array}{r}\tilde{\Pi}'_1\leftarrow\tilde{\mathbf{P}}'_{\mathsf{IOP}}(\rho'_1)\\ \vdots\\ \tilde{\Pi}'_{\mathsf{k}_{\mathsf{IOP}}}\leftarrow\tilde{\mathbf{P}}'_{\mathsf{IOP}}(\rho'_1,\ldots,\rho'_{\mathsf{k}_{\mathsf{IOP}}})\end{array}\right]\leq\beta_{\mathsf{IOP}}^{m/2}\ .$$

We now describe a $(\beta_{\mathsf{IOP}}^{m/2\mathsf{k}_{\mathsf{IOP}}}, \beta_{\mathsf{IOP}}^{t/2})$-state function state for $(\mathbf{P}'_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}})$. It is defined as follows for instance $\mathbb{x}$ and transcript tr:

- If $\mathsf{tr}=(\rho'_1,\tilde{\Pi}'_1,\ldots,\rho'_{\mathsf{k}_{\mathsf{IOP}}},\tilde{\Pi}'_{\mathsf{k}_{\mathsf{IOP}}})$ is a full transcript, then $\mathsf{state}(\mathbb{x},\mathsf{tr})=1$ if and only if
$$\Pr_{\rho'_{\mathsf{dc}}}\left[\ \mathbf{V}_{\mathsf{IOP}}^{\prime\mathsf{tr}}(\mathbb{x};\rho'_{\mathsf{dc}})=1\ \right]\geq\beta_{\mathsf{IOP}}^{t/2}\ .$$
- We define $\mathsf{state}(\mathbb{x},\mathsf{tr})$ inductively for transcripts tr that are not full and contain i rounds:
  - If $\mathsf{tr}=(\rho'_1,\tilde{\Pi}'_1,\ldots,\rho'_i,\tilde{\Pi}'_i)$ ends in a prover message. Then $\mathsf{state}(\mathbb{x},\mathsf{tr})=1$ if and only if
  $$\Pr_{\rho'_{i+1}}\left[\ \mathsf{state}(\mathbb{x},\mathsf{tr}||\rho'_{i+1})=1\ \right]\geq\beta_{\mathsf{IOP}}^{m/2\mathsf{k}_{\mathsf{IOP}}}\ .$$
  .
  - If $\mathsf{tr}=(\rho'_1,\tilde{\Pi}'_1,\ldots,\rho'_{i-1},\tilde{\Pi}'_{i-1},\rho'_i)$ ends with a verifier message, then $\mathsf{state}(\mathbb{x},\mathsf{tr})=1$ if and only if there exists a proof $\tilde{\Pi}'_i$ such that $\mathsf{state}(\mathbb{x},\mathsf{tr}||\tilde{\Pi}'_i)=1$.

It follows from the construction that state has decision error $\beta_{\mathsf{IOP}}^{t/2}$. The interaction error of $\beta_{\mathsf{IOP}}^{m/2\mathsf{k}_{\mathsf{IOP}}}$ can be shown by a similar argument to that of [CCHLRR18, Proposition 5.5]. □

3.3 Extractors

Definition 3.4. *The* **min-entropy** *of a random variable X is*

$$H_{\mathsf{min}}(X) = \min_{x \in \mathsf{supp}(X)} -\log \Pr[X = x]$$

Definition 3.5. *A function* $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k,ε)**-extractor** *if for every X with min-entropy at least k,* $\mathsf{SD}(\mathsf{Ext}(X, U_d), U_m) \leq \varepsilon$ *(where* SD *is the statistical distance). An extractor is explicit if it is computable in polynomial time.*

We use the following explicit construction of extractors with tight parameters.

Theorem 3.6 ([GUV09])**.** *For every constant $\alpha > 0$, and all positive integers n, k and all $\varepsilon > 0$, there is an explicit construction of a (k,ε)-extractor* $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *with $d = O(\log n + \log(1/\varepsilon))$, and $m \geq (1-\alpha)k$.*

Setting specific parameters, we will use this simpler version of the theorem.

Theorem 3.7. *For all positive integers m, and $\ell \geq \log m$ there is an explicit construction of a $(2m, 2^{-\ell})$-extractor* $\mathsf{Ext}\colon \{0,1\}^{3m} \times \{0,1\}^d \to \{0,1\}^m$ *with $d = O(\ell)$.*

Fact 3.8. *For all $n \in \mathbb{N}$, all $x \in \{0,1\}^n$ and $0 < \gamma < 1$ we have that*

$$|\{x' \in \{0,1\}^n : \Delta(x, x') \leq \gamma)\}| \leq 2^{n \cdot H(\gamma)} \ .$$

(here H is the entropy function $H(p) = -p\log(p) - (1-p)\log(1-p)$).

4 Interactive reducibility

We define the notion of interactive reducibility. Then we show basic properties of interactive reducibility: (a) in Section 4.1 we show that if a relation has an interactive reduction then it also has an IP; and (b) in Section 4.2 we show how to reduce the soundness error of an interactive reduction. Finally, in Section 4.3, we show an interactive reduction for any public-coin IP and bounded-output-length interactive reductions for the sumcheck protocol and Shamir's protocol.

Definition 4.1. *An* **interactive reduction** *for a relation R with k_{IR} predicates and soundness error $\varepsilon_{\mathsf{IR}}$ is a ℓ_{IR}-round public-coin protocol $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ (where $\mathbf{V}_{\mathsf{IR}}$ runs in polynomial time) for which there exists a list of predicates $f_0, f_1, \ldots, f_{\mathsf{k}_{\mathsf{IR}}}$ such that the following holds.*

- **Completeness.** *For every $(\mathbb{x}, \mathbb{w}) \in R$, $j \in [\mathsf{k}_{\mathsf{IR}}]$, and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$ such that $f_{j-1}(\mathbb{x}, \mathbb{z}_i) = 1$ for every $i \in [\mathsf{t}]$, it holds that*

$$\Pr\left[\, f_j(\mathbb{x}, \mathbb{z}') = 1 \,\middle|\, \mathbb{z}' \leftarrow \langle \mathbf{P}_{\mathsf{IR}}(\mathbb{x}, \mathbb{w}, \mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}), \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}) \rangle \,\right] = 1 \ .$$

- **Soundness.** *For every $\mathbb{x} \notin L(R)$, $j \in [\mathsf{k}_{\mathsf{IR}}]$, and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$, if there exists $i \in [\mathsf{t}]$ such that $f_{j-1}(\mathbb{x}, \mathbb{z}_i) = 0$ then for every (computationally unbounded) $\tilde{\mathbf{P}}_{\mathsf{IR}}$ it holds that*

$$\Pr\left[\, f_j(\mathbb{x}, \mathbb{z}') = 1 \,\middle|\, \mathbb{z}' \leftarrow \langle \tilde{\mathbf{P}}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}) \rangle \,\right] \leq \varepsilon_{\mathsf{IR}}(\mathbb{x}, \mathsf{t}) \ .$$

- **Relation identity.** *$f_0(\mathbb{x}, \mathbb{z}) = 1$ if and only if $\mathbb{x} \in L(R)$.*
- **Triviality.** *$f_{\mathsf{k}_{\mathsf{IR}}}(\mathbb{x}, \mathbb{z})$ can be computed in time $\mathrm{poly}(|\mathbb{x}|, |\mathbb{z}|)$.*

We call $\mathbb{x}$ the base instance *and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$* round instances.

A relation R is *interactively reducible* with k_{IR} predicates and ℓ_{IR} rounds if R has an interactive reduction with k_{IR} predicates and k_{IR}-rounds. An interactive reduction has soundness error that is *well-behaved* if $\varepsilon_{\mathsf{IR}}$ is (weakly) monotonically increasing with t and grows at most polynomially with t. All interactive reductions described in this paper have well-behaved soundness error.

Definition 4.2. *An interactive reduction has (polynomially)* **bounded output length** *if there exists $c \in \mathbb{N}$ such that, for every base instance $\mathbb{x}$, witness $\mathbb{w}$, and round instances $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$, the new instance output by $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ on inputs $(\mathbb{x}, \mathbb{w}, \mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}})$ has length at most $|\mathbb{x}|^c$.*

4.1 Interactive proofs from interactive reductions

We construct an IP for any relation that is interactively reducible.

Theorem 4.3. *If a relation R has an interactive reduction $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ then R has an IP $(\mathbf{P}_{\mathsf{IP}}, \mathbf{V}_{\mathsf{IP}})$ with the parameters indicated below.*

Interactive reduction for R	
Number of predicates	k_{IR}
Rounds	ℓ_{IR}
Communication	l_{IR}
Randomness	r_{IR}
Soundness error	$\varepsilon_{\mathsf{IR}}$
Verifier running time	$\mathsf{vt}_{\mathsf{IR}}$
Final predicate time	$\mathsf{ft}_{\mathsf{IR}}$

$\longrightarrow$

IP for R	
Rounds	$\mathsf{k}_{\mathsf{IR}} \cdot \ell_{\mathsf{IR}}$
Communication	$\mathsf{k}_{\mathsf{IR}} \cdot \mathsf{l}_{\mathsf{IR}}$
Randomness	$\mathsf{k}_{\mathsf{IR}} \cdot \mathsf{r}_{\mathsf{IR}}$
Soundness error	$\mathsf{k}_{\mathsf{IR}} \cdot \varepsilon_{\mathsf{IR}}(\mathbb{x}, 1)$
Verifier running time	$\mathsf{k}_{\mathsf{IR}} \cdot \mathsf{vt}_{\mathsf{IR}} + \mathsf{ft}_{\mathsf{IR}}$

Moreover, if $\ell_{\mathsf{IR}} = 1$ then the IP for R has round-by-round soundness error $\varepsilon_{\mathsf{IR}}(\mathbb{x}, 1)$.

Construction 4.4. The IP prover $\mathbf{P}_{\mathsf{IP}}$ receives as input an instance $\mathbb{x}$ and a witness $\mathbb{w}$, and the IP verifier $\mathbf{V}_{\mathsf{IP}}$ receives as input the instance $\mathbb{x}$. Letting $\mathbb{z}_0 := \mathbb{x}$, they interact as follows. For $j = 1, \ldots, \mathsf{k}_{\mathsf{IR}}$, $\mathbf{P}_{\mathsf{IP}}$ and $\mathbf{V}_{\mathsf{IP}}$ run the interactive reduction $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ to obtain a new instance

$$\mathbb{z}_j \leftarrow \langle \mathbf{P}_{\mathsf{IR}}(\mathbb{x}, \mathbb{w}, \mathbb{z}_{j-1}), \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_{j-1}) \rangle \ .$$

Finally, after the interaction, $\mathbf{V}_{\mathsf{IP}}$ checks that $f_{\mathsf{k}_{\mathsf{IR}}}(\mathbb{x}, \mathbb{z}_{\mathsf{k}_{\mathsf{IR}}}) = 1$.

Proof of Theorem 4.3. We prove completeness, then prove soundness (and round-by-round soundness), and finally analyze complexity measures.

Completeness. Fix $(\mathbb{x}, \mathbb{w}) \in R$. By completeness of $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$, for every $j \in [\mathsf{k}_{\mathsf{IR}}]$, if $f_{j-1}(\mathbb{x}, \mathbb{z}_{j-1}) = 1$, then

$$\Pr\left[\ f_j(\mathbb{x}, \mathbb{z}_j) = 1 \ \middle| \ \mathbb{z}_j \leftarrow \langle \mathbf{P}_{\mathsf{IR}}(\mathbb{x}, \mathbb{w}, \mathbb{z}_{j-1}), \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_{j-1}) \rangle \ \right] = 1 \ .$$

Since $f_0(\mathbb{x}, \mathbb{z}_0) = f_0(\mathbb{x}, \mathbb{x}) = 1$, by induction it holds that $f_{\mathsf{k}_{\mathsf{IR}}}(\mathbb{x}, \mathbb{z}_{\mathsf{k}_{\mathsf{IR}}}) = 1$ with probability 1. We conclude that $\Pr[\langle \mathbf{P}_{\mathsf{IP}}(\mathbb{x}, \mathbb{w}), \mathbf{V}_{\mathsf{IP}}(\mathbb{x}) \rangle = 1] = 1$.

Soundness. Fix $\mathbb{x} \notin L(R)$ and a malicious IP prover $\tilde{\mathbf{P}}_{\mathsf{IP}}$ and let $\varepsilon_{\mathsf{IR}} := \varepsilon_{\mathsf{IR}}(\mathbb{x}, 1)$. Let $\tilde{\mathbf{P}}^{(j)}_{\mathsf{IR}}$ be the machine that describes $\tilde{\mathbf{P}}_{\mathsf{IP}}$ during the j-th execution of the interactive reduction. Each $\tilde{\mathbf{P}}^{(j)}_{\mathsf{IR}}$ can be seen as a (potentially malicious) prover for the interactive reduction. We show that $f_{\mathsf{k}_{\mathsf{IR}}}(\mathbb{x}, \mathbb{z}_{\mathsf{k}_{\mathsf{IR}}}) = 1$ (a necessary condition for the IP verifier $\mathbf{V}_{\mathsf{IP}}$ to accept) with probability at most $\mathsf{k}_{\mathsf{IR}} \cdot \varepsilon_{\mathsf{IR}}$.

We show by induction that $f_j(\mathbb{x}, \mathbb{z}_j) = 1$ with probability at most $j \cdot \varepsilon_{\mathsf{IR}}$. For $j = 0$ this holds because $f_0(\mathbb{x}, \mathbb{z}_0) = f_0(\mathbb{x}, \mathbb{x}) = 0$. For $j > 0$, suppose that $f_{j-1}(\mathbb{x}, \mathbb{z}_{j-1}) = 1$ with probability at most $(j-1) \cdot \varepsilon_{\mathsf{IR}}$. It may be that whenever $f_{j-1}(\mathbb{x}, \mathbb{z}_{j-1}) = 1$ it holds holds that $f_j(\mathbb{x}, \mathbb{z}_j) = 1$. However, whenever $f_{j-1}(\mathbb{x}, \mathbb{z}_{j-1}) = 0$, we can invoke the soundness property of $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ to deduce that

$$\Pr\left[\ f_j(\mathbb{x}, \mathbb{z}_j) = 1 \ \middle| \ \mathbb{z}_j \leftarrow \langle \tilde{\mathbf{P}}^{(j)}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_{j-1}) \rangle \ \right] \leq \varepsilon_{\mathsf{IR}} \ .$$

By applying the union bound, the probability that $f_j(\mathbb{x}, \mathbb{z}_j) = 1$ is at most $(j-1) \cdot \varepsilon_{\mathsf{IR}} + \varepsilon_{\mathsf{IR}} = j \cdot \varepsilon_{\mathsf{IR}}$.

Round-by-round soundness. Suppose that $\ell_{\mathsf{IR}} = 1$ (that is, $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ has one round) and let $\varepsilon_{\mathsf{IR}} := \varepsilon_{\mathsf{IR}}(\mathbb{x}, 1)$. We focus on the MA case (the prover moves first in the interactive reduction); the AM case can be shown in a similar manner. To show round-by-round soundness we first define a state function state. Let $\mathbb{x}$ be an instance and tr be a transcript of the protocol up to round j (i.e., j full executions of the interactive reduction), and let $(\mathbb{z}_0, \ldots, \mathbb{z}_j)$ be the round instances output by the reduction verifier $\mathbf{V}_{\mathsf{IR}}$ in an execution of the protocol according to tr (if the transcript ends with a prover message, then $\mathbb{z}_j$ is the last round instance specified according to the transcript). Then we set $\mathsf{state}(\mathbb{x}, \mathsf{tr}) := 1$ if and only $f_j(\mathbb{x}, \mathbb{z}_j) = 1$. We show that this is indeed a state function.

- *Empty transcript.* If $\mathsf{tr} = \emptyset$ is the empty transcript then by definition $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 0$ if and only if $f_0(\mathbb{x}, \emptyset) = 0$, which occurs if and only if $\mathbb{x} \notin L(R)$.
- *Prover moves.* Given $\mathbb{x} \notin L(R)$ and a transcript tr. As in the definition of the state function, let $\mathbb{z}_0, \ldots, \mathbb{z}_j$ be the instances generated by $\mathbf{V}_{\mathsf{IR}}$. If $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 0$ then by definition $f_j(\mathbb{x}, \mathbb{z}_j) = 0$. Since no new instance is generated given only the prover's message, $\mathsf{state}(\mathbb{x}, \mathsf{tr} || a) = 0$ for every a.
- *Full transcript.* If tr is a full transcript and $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 0$, this implies that $f_{\mathsf{k}_{\mathsf{IR}}}(\mathbb{x}, \mathbb{z}_{\mathsf{k}_{\mathsf{IR}}}) = 0$. If this is the case then the IP verifier $\mathbf{V}_{\mathsf{IP}}$ rejects.

We argue that $(\mathbf{P}_{\mathsf{IP}}, \mathbf{V}_{\mathsf{IP}})$ has round-by-round soundness $\varepsilon_{\mathsf{IR}}$ with respect to this state function. Fix an instance $\mathbb{x}$, a round number $j \in [\mathsf{k}_{\mathsf{IR}}]$, and a transcript tr that contains messages up to round j (where the next message is a verifier message). Let $(\mathbb{z}_0, \ldots, \mathbb{z}_j)$ be the instances implied by tr. Suppose that $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 0$. Since the next message is a verifier message, and the interactive reduction is public-coin, we have that the verifier's next message ρ is a uniformly random string. Thus, we show that

$$\Pr_{\rho}\left[\, \mathsf{state}(\mathbb{x}, \mathsf{tr}||\rho) = 1 \,\right] \leq \varepsilon_{\mathsf{IR}} \ .$$

Since $\mathsf{state}(\mathsf{tr}) = 0$, we have that $f_j(\mathbb{x}, \mathbb{z}_j) = 0$. The claim, then, is implied by soundness of $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$, stating that for any malicious prover $\tilde{\mathbf{P}}_{\mathsf{IR}}$:

$$\Pr\left[\, f_{j+1}(\mathbb{x}, \mathbb{z}_{j+1}) = 1 \,\middle|\, \mathbb{z}_{j+1} \leftarrow \langle \tilde{\mathbf{P}}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_j) \rangle \,\right] \leq \varepsilon_{\mathsf{IR}} \ .$$

Complexity measures. We discuss complexity measures of $(\mathbf{P}_{\mathsf{IP}}, \mathbf{V}_{\mathsf{IP}})$.

- *Round complexity.* The round complexity of the IP is $\mathsf{k}_{\mathsf{IR}} \cdot \ell_{\mathsf{IR}}$ because the IP consists of k_{IR} interactive reductions, each with ℓ_{IR} rounds, ran in sequence.
- *Communication complexity.* The communication complexity of the IP is $\mathsf{k}_{\mathsf{IR}} \cdot \mathsf{l}_{\mathsf{IR}}$, because each of the k_{IR} interactive reductions has communication complexity l_{IR}.
- *Randomness complexity.* The IP verifier uses $\mathsf{k}_{\mathsf{IR}} \cdot \mathsf{r}_{\mathsf{IR}}$ random bits because each of the k_{IR} invocations of the reduction verifier uses r_{IR} random bits.
- *Verifier running time.* The IP verifier runs in time $\mathsf{k}_{\mathsf{IR}} \cdot \mathsf{vt}_{\mathsf{IR}} + \mathsf{ft}_{\mathsf{IR}}$ because it runs k_{IR} invocations of the reduction verifier (whose time complexity is $\mathsf{vt}_{\mathsf{IR}}$) and also runs the predicate $f_{\mathsf{k}_{\mathsf{IR}}}$ (whose time complexity is $\mathsf{ft}_{\mathsf{IR}}$).

□

4.2 Error reduction for interactive reducibility

We show that the soundness error of an interactive reduction can be amplified. In order to improve soundness from $\varepsilon_{\mathsf{IR}}$ to $\varepsilon_{\mathsf{IR}}^{\tau}$ we have each "new" round instance contain τ "original" round instances for the original protocol. Then, when reducing t separate new round instances (each one containing τ old round instances), we run the original interactive reduction on all of the old round instances contained within the new round instances.

Theorem 4.5. *Let R be a relation and $\tau \in \mathbb{N}$ a parameter. If R has an interactive reduction $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ then R has an interactive reduction $(\mathbf{P}_{\mathsf{IR}}^{(\tau)}, \mathbf{V}_{\mathsf{IR}}^{(\tau)})$ with the parameters indicated below.*

Interactive reduction for R	
Number of predicates	k_{IR}
Rounds	ℓ_{IR}
Output length	s_{IR}
Communication	l_{IR}
Randomness	r_{IR}
Soundness error	$\varepsilon_{\mathsf{IR}}$
Verifier running time	$\mathsf{vt}_{\mathsf{IR}}$
Final predicate time	$\mathsf{ft}_{\mathsf{IR}}$

$\longrightarrow$

Interactive reduction for R with reduced error	
Number of predicates	k_{IR}
Rounds	$\ell_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t})$
Output length	$\tau \cdot \mathsf{s}_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t})$
Communication	$\tau \cdot \mathsf{l}_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t})$
Randomness	$\tau \cdot \mathsf{r}_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t})$
Soundness error	$\varepsilon_{\mathsf{IR}}^{\tau}(\mathbb{x}, \tau \cdot \mathsf{t})$
Verifier running time	$\tau \cdot \mathsf{vt}_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t})$
Final predicate time	$\tau \cdot \mathsf{ft}_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t})$

Construction 4.6. We define the new predicates and then describe the new interactive reduction.

The predicates. Let f_j be the j-th predicate in $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$. Define the new predicate $f_j^{(\tau)}$ for $(\mathbf{P}_{\mathsf{IR}}^{(\tau)}, \mathbf{V}_{\mathsf{IR}}^{(\tau)})$ so that $f_j^{(\tau)}(\mathbb{x}, \mathbb{z}) = 1$ if and only if $\mathbb{z} = (\mathbb{z}_1, \ldots, \mathbb{z}_\tau)$ and $f_j(\mathbb{x}, \mathbb{z}_m) = 1$ for all $m \in [\tau]$.

The reduction. The reduction prover $\mathbf{P}_{\mathsf{IR}}^{(\tau)}$ and reduction verifier $\mathbf{V}_{\mathsf{IR}}^{(\tau)}$ receive as input a base instance $\mathbb{x}$ and round instances $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$ where $\mathbb{z}_i := (\mathbb{z}_{i,1}, \ldots, \mathbb{z}_{i,\tau})$. They interact as follows.

- They run the interactive reduction $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ with base instance $\mathbb{x}$ and round instances $(\mathbb{z}_{1,1}, \ldots, \mathbb{z}_{1,\tau}, \ldots, \mathbb{z}_{\mathsf{t},1}, \ldots, \mathbb{z}_{\mathsf{t},\tau})$ for τ times in parallel with fresh randomness in each execution.
- Output $\mathbb{z}' := (\mathbb{z}'_1, \ldots, \mathbb{z}'_\tau)$, where $\mathbb{z}'_i$ is the result of the i-th independent execution of $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$.

Proof of Theorem 4.5. First we argue completeness, then soundness of the interactive reduction and triviality of the predicate $f_{\mathsf{k}_{\mathsf{IR}}}$, and finally analyze the other complexity measures of the protocol.

Completeness. Fix $(\mathbb{x}, \mathbb{w}) \in R$ and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$ with $f_j(\mathbb{x}, \mathbb{z}_1) = \ldots = f_j(\mathbb{x}, \mathbb{z}_{\mathsf{t}}) = 1$. Completeness of the interactive reduction $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ implies that with probability 1 every output $\mathbb{z}'_i$ satisfies $f_{j+1}(\mathbb{x}, \mathbb{z}'_i) = 1$; in turn, by definition of the new predicates, this means that $\mathbb{z}' := (\mathbb{z}'_1 \ldots, \mathbb{z}'_\tau)$ is such that $f_j^{(\tau)}(\mathbb{x}, \mathbb{z}') = 1$.

Soundness. Fix $\mathbb{x}$ and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$ where $f_j^{(\tau)}(\mathbb{x}, \mathbb{z}_i) = 0$ for some $i \in [\mathsf{t}]$. This implies that there exists $\mathbb{z}_{i,m} \in \mathbb{z}_i$ such that $f_j^{(\tau)}(\mathbb{x}, \mathbb{z}_{i,m}) = 0$. This is one of the instances in the list $(\mathbb{z}_{1,1}, \ldots, \mathbb{z}_{1,\tau}, \ldots, \mathbb{z}_{\mathsf{t},1}, \ldots, \mathbb{z}_{\mathsf{t},\tau})$ given as input to the interactive reduction $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$. Thus, for each execution in the protocol we have, by the definition of reducible soundness that for every $\tilde{\mathbf{P}}_{\mathsf{IR}}$:

$$\Pr\left[f_{j+1}(\mathbb{x}, \mathbb{z}) = 1 \;\middle|\; \mathbb{z} \leftarrow \langle \tilde{\mathbf{P}}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_{1,1}, \ldots, \mathbb{z}_{1,\tau}, \ldots, \mathbb{z}_{\mathsf{t},1}, \ldots, \mathbb{z}_{\mathsf{t},\tau}) \rangle \right] \leq \varepsilon_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t}) \ .$$

Since each of the τ executions is independent, with probability at least $1 - \varepsilon_{\mathsf{IR}}^{\tau}(\mathbb{x}, \tau \cdot \mathsf{t})$ there exists $i \in [\tau]$ with $f_{j+1}(\mathbb{x}, \mathbb{z}'_i) = 0$. This implies that $f_{j+1}^{(\tau)}(\mathbb{x}, \mathbb{z}') = 0$ with probability at least $1 - \varepsilon_{\mathsf{IR}}^{\tau}(\mathbb{x}, \tau \cdot \mathsf{t})$.

Triviality. Triviality of $f_{\mathsf{k}_{\mathsf{IR}}}^{(\tau)}$ follows directly from the triviality of $f_{\mathsf{k}_{\mathsf{IR}}}$.

Complexity measures. We discuss complexity measures of the new interactive reduction $(\mathbf{P}_{\mathsf{IR}}^{(\tau)}, \mathbf{V}_{\mathsf{IR}}^{(\tau)})$. Running the new interactive reduction with t round instances is essentially τ executions of the interactive reduction $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ with $\tau \cdot \mathsf{t}$ round instances each. Thus all complexity measures depend on the complexity of $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ with $\tau \cdot \mathsf{t}$ round instances.

- *Output length.* The output of the protocol has length $\tau \cdot \mathsf{s}_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t})$.
- *Round complexity.* The protocol has $\ell_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t})$ rounds.
- *Communication complexity.* The communication complexity of the protocol is $\tau \cdot \mathsf{l}_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t})$.
- *Randomness complexity.* The verifier of the new interactive reduction uses $\tau \cdot \mathsf{r}_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t})$ random bits.
- *Verifier running time.* The verifier of the new interactive reduction runs in time $\tau \cdot \mathsf{vt}_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t})$.
- *Final predicate time.* The final predicate of the new interactive reduction runs in time $\tau \cdot \mathsf{ft}_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{t})$.

□

4.3 Relations with interactive reducibility

We present interactive reductions for various relations. In Section 4.3.1 we show that every language with an IP has an interactive reduction, albeit not one with bounded output length. Then we show how to achieve bounded output length for specific classes of relations: 1. the sumcheck protocol in Section 4.3.2; and 2. Shamir's protocol in Section 4.3.3;

4.3.1 Interactive reductions for any IP

We show that any IP can be transformed into an IP with interactive reducibility where the base instance is the instance for the IP, and the round instances are sets of partial transcripts. The length of a merged round instance grows with the number of partial transcripts in all of the reduction instances combined, and so this reduction does not have bounded output-length.

Theorem 4.7. *If a relation R has a k_{IP}-round public-coin IP $(\mathbf{P}_{\mathsf{IP}}, \mathbf{V}_{\mathsf{IP}})$ then R has an interactive reduction $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ with the parameters below for every $m > 0$ that divides k_{IP}.*

IP for R	
Rounds	k_{IP}
Per-round communication	l_{IP}
Per-round randomness	r_{IP}
Round-by-round soundness error	β_{rbr}
Per-round verifier running time	vt
Decision time	d_{IP}

$\longrightarrow$

Interactive reduction for R for round j given $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$	
Number of predicates	$\mathsf{k}_{\mathsf{IP}}/m$
Rounds	m
Output length	$\mathsf{t} \cdot m \cdot \mathsf{l}_{\mathsf{IP}} + \sum_{i \in [\mathsf{t}]} \lvert\mathbb{z}_i\rvert$
Communication	$\mathsf{t} \cdot m \cdot \mathsf{l}_{\mathsf{IP}}$
Randomness	$m \cdot \mathsf{r}_{\mathsf{IP}}$
Soundness error	$m \cdot \beta_{\mathrm{rbr}}$
Verifier running time	$\mathsf{t} \cdot m \cdot \mathsf{vt}_{\mathsf{IP}}$
Final predicate time	d_{IP}

Construction 4.8. Let state be the state function of $(\mathbf{P}_{\mathsf{IP}}, \mathbf{V}_{\mathsf{IP}})$. We define the predicates and then describe the interactive reduction for the relation described by this IP.

The predicates. For each $j \in [\mathsf{k}_{\mathsf{IP}}/m]$, we define $f_j(\mathbb{x}, \mathbb{z}) := 1$ if and only if the following are true:
1. $\mathbb{z}$ is a set of $(j \cdot m)$-round partial transcript ending with a prover message.
2. For every $\mathsf{tr} \in \mathbb{z}$, $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 1$.
3. If $\mathbb{x} \in L(R)$ and $j \neq \mathsf{k}_{\mathsf{IP}}/m$, then we require that very $\mathsf{tr} \in \mathbb{z}$ is consistent with the honest prover: Suppose $\mathsf{tr} = (a_1, \rho_1, \ldots, a_{j \cdot m})$ then $a_1 := \mathbf{P}_{\mathsf{IP}}(\mathbb{x})$ and for every $1 < k \leq j \cdot m$, $a_k := \mathbf{P}_{\mathsf{IP}}(\mathbb{x}, a_1, \rho_1, \ldots, a_{k-1}, \rho_{k-1})$.

The reduction. The reduction prover $\mathbf{P}_{\mathsf{IR}}$ and reduction verifier $\mathbf{V}_{\mathsf{IR}}$ receive as input a base instance $\mathbb{x}$ and round instances $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$ where each $\mathbb{z}_i$ is a set of j-round partial transcripts ending in a prover message. The reduction prover $\mathbf{P}_{\mathsf{IR}}$ additionally receives as input a witness $\mathbb{w}$. The protocol is as follows:

1. For $k = 1$ to m:
   (a) $\mathbf{V}_{\mathsf{IR}}$: Send a uniformly random message ρ_k corresponding to the IP verifier's $(j \cdot m + k)$-th message.
   (b) $\mathbf{P}_{\mathsf{IR}}$: For every $\mathsf{tr} \in \bigcup_{i \in [\mathsf{t}]} \mathbb{z}_i$, send the prover message $a_{\mathsf{tr},k} := \mathbf{P}_{\mathsf{IP}}(\mathbb{x}, \mathbb{w}, \mathsf{tr}, \rho_1, \ldots, \rho_k)$.
2. $\mathbf{V}_{\mathsf{IR}}$: Output $\mathbb{z}' := \{\mathsf{tr} || \rho_1 || a_{\mathsf{tr},1} || \ldots || \rho_m || a_{\mathsf{tr},m}\}_{\mathsf{tr} \in \cup_{i \in [\mathsf{t}]} \mathbb{z}_i}$.

Proof of Theorem 4.7. First we argue completeness, then soundness and then show triviality of the predicate $f_{\mathsf{k}_{\mathsf{IP}}/m}$.

Completeness. Fix $(\mathbb{x}, \mathbb{w}) \in R$, $j \in [\mathsf{k}_{\mathsf{IP}}/m]$, and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$ such that $f_{j-1}(\mathbb{x}, \mathbb{z}_1) = \cdots = f_{j-1}(\mathbb{x}, \mathbb{z}_{\mathsf{t}}) = 1$. This implies that $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 1$ for every $\mathsf{tr} \in \bigcup_{i\in[\mathsf{t}]} \mathbb{z}_i$ and that every transcript is consistent with the honest prover. We now show that for every such transcript tr and $\rho_1, \ldots, \rho_m$, setting $a_{\mathsf{tr},k} := \mathbf{P}_{\mathsf{IP}}(\mathbb{x}, \mathbb{w}, \mathsf{tr}, \rho_1, a_{\mathsf{tr},1}, \ldots, a_{\mathsf{tr},k-1}, \rho_k)$, we have

$$\mathsf{state}(\mathbb{x}, \mathsf{tr}||\rho_1||a_{\mathsf{tr},1}|| \ldots ||\rho_m||a_{\mathsf{tr},m}) = 1 \ .$$

This implies that $f_{j+1}(\mathbb{x}, \mathbb{z}') = 1$, since $f_{j+1}(\mathbb{x}, \mathbb{z}') = 1$ if and only if for every tr the state of the resulting transcript after running the protocol is 1.

Notice that any for any transcript tr that is consistent with the honest prover and any ρ: $\mathsf{state}(\mathbb{x}, \mathsf{tr}||\rho||a_{\mathsf{tr}}) = 1$ where $a_{\mathsf{tr}} := \mathbf{P}_{\mathsf{IP}}(\mathbb{x}, \mathsf{tr}||\rho)$. Suppose towards contradiction of perfect completeness of the underlying IP that this was not the case. Then for some choice of ρ the honest prover sends a message that sets the state to 0. Since the protocol has non-zero round-by-round soundness error, once the state is set to zero, there is a non-zero probability that the final transcript has state 0, which will imply that the verifier rejects. Since tr was consistent with the honest prover, there is a non-zero probability that it is generated in an honest execution of the protocol. Therefore, there is non-zero probability that the honest prover will not convince the verifier in a correct execution, in contradiction to perfect completeness of the IP.

The above argument shows that for every transcript, its state will remain 1. By construction, every new transcript $(\mathsf{tr}||\rho_1||a_{\mathsf{tr},1}|| \ldots ||\rho_m||a_{\mathsf{tr},m})$ is consistent with the honest prover. We therefore conclude that $f_j(\mathbb{x}, \mathbb{z}') = 1$.

Soundness. Fix $\mathbb{x} \notin L(R)$, $j \in [\mathsf{k}_{\mathsf{IP}}/m]$, and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$ and a (computationally unbounded) malicious reduction prover $\tilde{\mathbf{P}}_{\mathsf{IR}}$. Before we begin our analysis, recall that round-by-round soundness of the IP, for any transcript tr with $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 0$:

$$\Pr_{\rho}[\mathsf{state}(\mathbb{x}, \mathsf{tr}||\rho) = 1] \leq \beta_{\mathrm{rbr}} \ .$$

Moreover, by the properties of the state function, since $\mathbb{x} \notin L(R)$, if ρ is such that $\mathsf{state}(\mathbb{x}, \mathsf{tr}||\rho) = 0$ then for any a we have that $\mathsf{state}(\mathbb{x}, \mathsf{tr}||\rho_j||a) = 0$. Using induction, we can conclude that for any tr with $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 0$:

$$\Pr_{\rho_1,\ldots,\rho_m}\left[\mathsf{state}(\mathbb{x}, \mathsf{tr}||\rho_1||a_{\mathsf{tr},1}|| \ldots ||\rho_m||a_{\mathsf{tr},m}) = 1 \;\middle|\; \begin{array}{r} a_{\mathsf{tr},1} := \tilde{\mathbf{P}}_{\mathsf{IR}}(\mathsf{tr}, \rho_1) \\ \vdots \\ a_{\mathsf{tr},m} := \tilde{\mathbf{P}}_{\mathsf{IR}}(\mathsf{tr}, \rho_1, a_{\mathsf{tr},1}, \ldots, a_{\mathsf{tr},m-1}, \rho_m) \end{array}\right] \leq m \cdot \beta_{\mathrm{rbr}} \ . \tag{2}$$

Suppose that there exists $i \in [\mathrm{t}]$ such that $f_{j-1}(\mathbb{x}, \mathbb{z}_i) = 0$. This implies that there exists a transcript $\mathsf{tr} \in \bigcup_{i\in[\mathsf{t}]} \mathbb{z}_i$ with $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 0$ (notice that the requirement of consistency with the honest prover described in Item 3 is only for $\mathbb{x} \in L(R)$). By Equation (2) and since $(\mathsf{tr}||\rho_1||a_{\mathsf{tr},1}|| \ldots ||\rho_m||a_{\mathsf{tr},m})$ is added to the new instance $\mathbb{z}'$, this implies that with probability at least $1 - m \cdot \beta_{\mathrm{rbr}}$, $f_j(\mathbb{x}, \mathbb{z}') = 0$ as required.

Triviality. $f_{\mathsf{k}_{\mathsf{IP}}/m}(\mathbb{x}, \mathbb{z}) = 1$ if and only if all of the transcripts in $\mathbb{z}$ are complete transcripts and have state 1 (notice that since this is the last round, we do not have the requirement of consistency with the honest prover described in Item 3). This can be verified by running $\mathbf{V}_{\mathsf{IP}}$ on every $\mathsf{tr} \in \mathbb{z}$, since $\mathbf{V}_{\mathsf{IP}}(\mathbb{x}, \mathsf{tr}) = \mathsf{state}(\mathbb{x}, \mathsf{tr})$.

Complexity measures. We discuss complexity measures of $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$.

- *Output length.* The new instance size is $\mathsf{t} \cdot m \cdot \mathsf{l}_{\mathsf{IP}} + \sum_{i \in [\mathsf{t}]} |\mathbb{z}_i|$.
- *Round complexity.* The protocol has m rounds.
- *Communication complexity.* The communication complexity of the protocol is $\mathsf{t} \cdot m \cdot \mathsf{l}_{\mathsf{IP}}$ where l_{IP} is the per-round communication complexity of $(\mathbf{P}_{\mathsf{IP}}, \mathbf{V}_{\mathsf{IP}})$.
- *Randomness complexity.* $\mathbf{V}_{\mathsf{IR}}$ uses r_{IP} random bits where $m \cdot \mathsf{r}_{\mathsf{IP}}$ is the per-round randomness complexity of $\mathbf{V}_{\mathsf{IP}}$.
- *Verifier running time.* $\mathbf{V}_{\mathsf{IR}}$ runs the IP verifier for $\mathsf{t} \cdot m$ times in total time $\mathsf{t} \cdot m \cdot \mathsf{vt}_{\mathsf{IP}}$.
- *Final predicate time.* The final predicate can be checked in time d_{IP}.

□

4.3.2 Sumcheck language

Definition 4.9. *A* **sumcheck instance** *has the form $(\mathbb{F}, H, n, d, p, \gamma)$ where $\mathbb{F}$ is a field, $H \subseteq \mathbb{F}$, $n \in \mathbb{N}$ is a number of variables, $d \in \mathbb{N}$ is a degree bound, $p\colon \mathbb{F}^n \to \mathbb{F}$ is an n-variate polynomial of individual degree at most d (given to the verifier as an oracle) and $\gamma \in \mathbb{F}$ is a claimed sum. We define $(\mathbb{F}, H, n, d, p, \gamma) \in L_\Sigma$ if and only if*

$$\sum_{\alpha_1, \ldots, \alpha_n \in H} p(\alpha_1, \ldots, \alpha_n) = \gamma \ .$$

Theorem 4.10. *The sumcheck language L_Σ has an interactive reduction that, for instances $\mathbb{x} = (\mathbb{F}, H, n, d, p, \gamma)$, has the following parameters.*

Interactive reduction for L_Σ	
Number of predicates	n
Messages	2
Output length	$\lvert\mathbb{x}\rvert$
Communication	$O(n\mathsf{t}d^2 \log \lvert\mathbb{F}\rvert)$
Randomness	$2 \log \lvert\mathbb{F}\rvert$
Soundness error	$O\left(n\mathsf{t}d/\lvert\mathbb{F}\rvert\right)$
Verifier running time	$\mathrm{poly}(\lvert\mathbb{x}\rvert, \mathsf{t})$
Final predicate time	1 call to p

Construction 4.11. We define the predicates and then describe the interactive reduction.

The predicates. For a base instance $\mathbb{x} = (\mathbb{F}, H, n, d, p, \gamma)$, for each $j \in [n]$, we define $f_j(\mathbb{x}, \mathbb{z}) = 1$ if and only if, parsing $\mathbb{z} = (r_1, \ldots, r_j, \gamma_j) \in \mathbb{F}^{j+1}$, we have

$$\sum_{\alpha_{j+1}, \ldots, \alpha_n \in H} p(r_1, \ldots, r_j, \alpha_{j+1}, \ldots, \alpha_n) = \gamma_j \ .$$

The reduction. The reduction prover $\mathbf{P}_{\mathsf{IR}}$ and reduction verifier $\mathbf{V}_{\mathsf{IR}}$ receive as input a base instance $\mathbb{x}$ and round instances $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$. Parse each $\mathbb{z}_i := (r_1^{(i)}, \ldots, r_j^{(i)}, \gamma_j^{(i)})$. Let $I_{\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}}\colon \mathbb{F} \to \mathbb{F}^j$ be the polynomial of degree less than t such that $I_{\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}}(i) = (r_1^{(i)}, \ldots, r_j^{(i)})$ for every $i \in [\mathsf{t}]$;[2] for any $\gamma \in \mathbb{F}$, $I_{\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}}(\gamma)$ can be computed in $\mathrm{poly}(\mathsf{t})$ operations. The interactive reduction is as follows:

[2]Here we implicitly associate the set $[\mathsf{t}]$ with an arbitrary set $S \subseteq \mathbb{F}$ of size t.

- $\mathbf{P}_{\mathsf{IR}}$: Send the polynomial $g \in \mathbb{F}[X_1, X_2]$ defined as:

$$g(X_1, X_2) := \sum_{\alpha_{j+2},\ldots,\alpha_n \in H} p(I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathsf{t}}}(X_1), X_2, \alpha_{j+2}, \ldots, \alpha_n) \ . \tag{3}$$

- $\mathbf{V}_{\mathsf{IR}}$: Receive a bivariate polynomial $\tilde{g} \in \mathbb{F}[X_1, X_2]$ of degree at most $j \cdot d \cdot (\mathsf{t}-1)$ in X_1 and degree at most d in X_2.
  1. *Consistency:* Check that for every $i \in [\mathsf{t}]$ it holds that $\sum_{\alpha \in H} \tilde{g}(i, \alpha) = \gamma_j^{(i)}$. (Reject if not.)
  2. *Generate new instance:*
     (a) Sample uniformly random field elements $\rho, r^* \leftarrow \mathbb{F}$ and send them to $\mathbf{P}_{\mathsf{IR}}$.
     (b) Output the new instance $\mathbb{z}' := (r'_1, \ldots, r'_j, r^*, \gamma_{j+1})$ where $(r'_1, \ldots, r'_j) := I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathsf{t}}}(\rho)$ and $\gamma_{j+1} := \tilde{g}(\rho, r^*)$.

Proof of Theorem 4.10. We argue completeness, then soundness, and then triviality of the predicate f_n.

Completeness. Fix $\mathbb{x} = (\mathbb{F}, H, n, d, p, \gamma) \in L_\Sigma$ and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$ such that $f_j(\mathbb{x}, \mathbb{z}_1) = \cdots = f_j(\mathbb{x}, \mathbb{z}_{\mathsf{t}}) = 1$ where $\mathbb{z}_i := (r_1^{(i)}, \ldots, r_j^{(i)}, \gamma_j^{(i)})$. We argue that

$$\Pr\left[\ f_j(\mathbb{x}, \mathbb{z}') = 1 \,\middle|\, \mathbb{z}' \leftarrow \langle \mathbf{P}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}), \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}) \rangle \ \right] = 1 \ .$$

Let g be defined as in Equation (3). Notice that for every $i \in [\mathsf{t}]$,

$$g(i, X_2) = \sum_{\alpha_{j+2},\ldots,\alpha_n \in H} p(r_1^{(i)}, \ldots, r_j^{(i)}, X_2, \alpha_{j+2}, \ldots, \alpha_n) \ ,$$

which, since $f_j(\mathbb{x}, \mathbb{z}_i) = 1$, implies that $\sum_{\alpha \in H} g(i, \alpha) = \gamma_j^{(i)}$. Therefore the reduction verifier $\mathbf{V}_{\mathsf{IR}}$ does not reject in the consistency test (in Item 1). Moreover, by the definition of g, for every ρ and r^*,

$$\begin{aligned}\sum_{\alpha_{j+2},\ldots,\alpha_n \in H} p(r'_1, \ldots, r'_j, r^*, \alpha_{j+2}, \ldots, \alpha_n) &= \sum_{\alpha_{j+2},\ldots,\alpha_n \in H} p(I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathsf{t}}}(\rho), r^*, \alpha_{j+2}, \ldots, \alpha_n) \\ &= g(\rho, r^*) \ .\end{aligned}$$

Therefore we have $f_i(\mathbb{x}, \mathbb{z}') = 1$ with probability 1.

Soundness. Fix $\mathbb{x} = (\mathbb{F}, H, n, d, p, \gamma) \notin L_\Sigma$ and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$ where $\mathbb{z}_i := (r_1^{(i)}, \ldots, r_j^{(i)}, \gamma_j^{(i)})$. Suppose that $f_j(\mathbb{x}, \mathbb{z}_i) = 0$ for some $i \in [\mathsf{t}]$. Fix a malicious reduction prover $\tilde{\mathbf{P}}_{\mathsf{IR}}$. We show that

$$\Pr\left[\ f_j(\mathbb{x}, \mathbb{z}') = 1 \,\middle|\, \mathbb{z}' \leftarrow \langle \tilde{\mathbf{P}}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}) \rangle \ \right] \leq \frac{d \cdot (1 + n \cdot (\mathsf{t}-1))}{|\mathbb{F}|} \ .$$

If $\tilde{g}$ does not pass the consistency test (in Item 1) then the reduction verifier outputs $\perp$, and we have $f_{j+1}(\mathbb{x}, \perp) = 0$. Thus, we can assume that $\tilde{g}$ passes the consistency test (that is, for every $i \in [\mathsf{t}]$ it holds that $\sum_{\alpha \in H} \tilde{g}(i, \alpha) = \gamma_j^{(i)}$). Let g be defined as in Equation (3) with respect to $\mathbb{x}$ and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$.

We have that $g \not\equiv \tilde{g}$ due to the fact that $f_j(\mathbb{x}, \mathbb{z}_i) = 0$, which implies that

$$\sum_{\alpha \in H} g(i, \alpha) = \sum_{\alpha_{j+1},\ldots,\alpha_n \in H} p(r_1^{(i)}, \ldots, r_j^{(i)}, \alpha_{j+1}, \ldots, \alpha_n) \neq \gamma_j^{(i)} = \sum_{\alpha \in H} \tilde{g}(i, \alpha) \ ,$$

Thus, by applying the Schwartz–Zippel lemma, and recalling that g and $\tilde{g}$ both have degree $j \cdot d \cdot (\mathtt{t}-1)$ in their first variable and d in their second variable, we get:

$$\Pr_{\rho,r^*}[\, g(\rho, r^*) = \tilde{g}(\rho, r^*) \,] \leq \frac{d \cdot (1 + j \cdot (\mathtt{t}-1))}{|\mathbb{F}|} \leq \frac{d \cdot (1 + n \cdot (\mathtt{t}-1))}{|\mathbb{F}|} \ .$$

For any choice of ρ and r^* such that $g(\rho, r^*) \neq \tilde{g}(\rho, r^*)$, we have that

$$\sum_{\alpha_{j+2},\ldots,\alpha_n \in H} p(r'_1, \ldots, r'_j, r^*, \alpha_{j+2}, \ldots, \alpha_n) = g(\rho, r^*) \neq \tilde{g}(\rho, r^*) = \gamma_{j+1} \ ,$$

and so $f_{j+1}(\mathbb{x}, (r'_1, \ldots, r'_j, r^*, \gamma_{j+1})) = 0$ with probability at least $1 - \frac{d \cdot (1+n \cdot (\mathtt{t}-1))}{|\mathbb{F}|}$.

Triviality. By definition, $f_n(\mathbb{x}, \mathbb{z}) = f_n((\mathbb{F}, H, n, d, p, \gamma), (r_1, \ldots, r_n, \gamma_n)) = 1$ if and only if $p(r_1, \ldots, r_n) = \gamma_n$, which can be efficiently verified by querying p once.

Complexity measures. We discuss complexity measures of the protocol $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$.

- *Output length.* The output is a list of $j + 2 \leq n + 2$ field elements. Thus, the maximal length output is at most $O(n \cdot \log |\mathbb{F}|)$, which is a fixed polynomial in the input length.
- *Rounds.* The interactive reduction has one round.
- *Communication complexity.* The prover sends g which is bivariate and has degree at most $n \cdot d \cdot (\mathtt{t}-1)$ in its first variable and d in its second variable. This requires sending $O(n\mathtt{t}d^2)$ field elements. The verifier sends two field elements. Thus the communication complexity of the protocol is $O(n\mathtt{t}d^2 \log |\mathbb{F}|)$ bits.
- *Randomness complexity.* $\mathbf{V}_{\mathsf{IR}}$ uses $2 \log |\mathbb{F}|$ random bits.
- *Verifier running time.* $\mathbf{V}_{\mathsf{IR}}$ runs in time $\mathrm{poly}(n, \log |\mathbb{F}|) = \mathrm{poly}(|\mathbb{x}|)$.
- *Final predicate time.* Computing the final predicate requires one call to p.

□

4.3.3 Shamir's protocol

We show an interactive reduction for a class of languages that contains all of PSPACE. For notational convenience, we define an operator that acts either as a sum or product, depending on the index of the variable it is applied over. For every $j \in \mathbb{N}$ let:

$$\Psi_{\alpha_j \in H} := \begin{cases} \sum_{\alpha_j \in H} & j \text{ is odd} \\ \prod_{\alpha_j \in H} & j \text{ is even} \end{cases} \ .$$

Given polynomials $p_0, \ldots, p_n$ where each p_j is a j-variate polynomial over a field $\mathbb{F}$, we derive polynomials $h_0, \ldots, h_n$ where $h_j \colon \mathbb{F}^j \to \mathbb{F}$ is defined recursively as follows:

- $h_n(X_1, \ldots, X_n) := p_n(X_1, \ldots, X_n)$.
- $h_j(X_1, \ldots, X_j) := p_j(X_1, \ldots, X_j) \cdot \Psi_{\alpha_{j+1}} \Big(h_{j+1}(X_1, \ldots, X_j, \alpha_{j+1}) \Big)$.

Observe that using this notation:

$$h_0 = p_0 \cdot \sum_{\alpha_1 \in H} \left[p_1(\alpha_1) \cdot \prod_{\alpha_2 \in H} \left[p_2(\alpha_1, \alpha_2) \cdot \sum_{\alpha_3 \in H} \left[\dots \left[p_{n-1}(\alpha_1, \dots, \alpha_{n-1}) \cdot \Psi_{\alpha_n \in H} \, p_n(\alpha_1, \dots, \alpha_n) \right] \dots \right] \right] \right] .$$

Definition 4.12. *A* **Shamir instance** *has the form* $(\mathbb{F}, H, n, d, p_0, \dots, p_n, \gamma)$ *where* $\mathbb{F}$ *is a field,* $H \subseteq \mathbb{F}$*,* $n \in \mathbb{N}$ *is a number of variables,* $d \in \mathbb{N}$ *is a degree bound, each* $p_j \colon \mathbb{F}^j \to \mathbb{F}$ *is an* j*-variate polynomial where the derived polynomials* $h_0, \dots, h_n$ *have individual degree at most* d *and* $\gamma \in \mathbb{F}$ *is a claimed value. We define* $(\mathbb{F}, H, n, d, p_0, \dots, p_n, \gamma) \in L_{\mathsf{Shmr}}$ *if and only if* $h_0 = \gamma$*.*

Note that a sumcheck instance (as defined in Section 4.3.2) is a Shamir instance where $p_0, \dots, p_{n-1}$ are equal to the constant 1 polynomial, and $p_n(X_1, \dots, X_n) \triangleq p'_n(X_1, X_3, \dots, X_{n-1})$ (assuming n is even) for some polynomial p'_n of individual degree at most d.

Theorem 4.13. *The Shamir language* L_{Shmr} *has an interactive reduction that, for instances* $\mathbb{x} = (\mathbb{F}, H, n, d, p_0, \dots, p_n, \gamma)$*, has the following parameters.*

Interactive reduction for L_{Shmr}	
Number of predicates	n
Messages	2
Output length	$\lvert\mathbb{x}\rvert$
Communication	$O(n\mathtt{t}d^2 \log \lvert\mathbb{F}\rvert)$
Randomness	$2 \log \lvert\mathbb{F}\rvert$
Soundness error	$O\left(n\mathtt{t}d/\lvert\mathbb{F}\rvert\right)$
Verifier running time	$\mathrm{poly}(\lvert\mathbb{x}\rvert, \mathtt{t})$
Final predicate time	1 call to p_n

Shamir [Sha92] showed how deciding every language in PSPACE can be reduced to computing the value of an arithmetic expression of the form of Shamir instances. Specifically, Shamir's reduction takes a TQBF formula and transforms it into a "simple" TQBF formula, roughly defined as one in which every occurrence of every variable is separated from its quantification point by at most one universal quantifier (and arbitrarily many other symbols). This ensures that when arithmetizing the formula, one ends up with an expression in which the degree of every variable is polynomial.

Using the above theorem, and Shamir's reduction we get the following corollary showing that every language in PSPACE has an interactive reduction with bounded output length albeit with polynomially-many predicates.

Corollary 4.14. *Every* $L \in$ PSPACE *has an interactive reduction that, for instances* $\mathbb{x}$*, has the following parameters:*

Interactive reduction for L	
Number of predicates	$\mathrm{poly}(\lvert\mathbb{x}\rvert)$
Messages	2
Output length	$\mathrm{poly}(\lvert\mathbb{x}\rvert)$
Communication	$\mathrm{poly}(\lvert\mathbb{x}\rvert)$
Randomness	$\mathrm{poly}(\lvert\mathbb{x}\rvert)$
Soundness error	$O\left(\mathtt{t}/2^{\lvert\mathbb{x}\rvert}\right)$
Verifier running time	$\mathrm{poly}(\lvert\mathbb{x}\rvert, \mathtt{t})$
Final predicate time	$\mathrm{poly}(\lvert\mathbb{x}\rvert)$

Construction 4.15. We define the predicates and then describe the interactive reduction.

The predicates. For a base instance $\mathbb{x} = (\mathbb{F}, H, n, d, p_0, \ldots, p_n, \gamma)$, for each $j \in [n]$, we define $f_j(\mathbb{x}, \mathbb{z}) = 1$ if and only if, parsing $\mathbb{z} = (r_1, \ldots, r_j, \gamma_j) \in \mathbb{F}^j$, we have $h_j(r_1, \ldots, r_j) = \gamma_j$.

The reduction. The reduction prover $\mathbf{P}_{\mathsf{IR}}$ and reduction verifier $\mathbf{V}_{\mathsf{IR}}$ receive as input a base instance $\mathbb{x}$ and round instances $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathtt{t}}$. Parse each $\mathbb{z}_i := (r_1^{(i)}, \ldots, r_j^{(i)}, \gamma_j^{(i)})$. Let $I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathtt{t}}} \colon \mathbb{F} \to \mathbb{F}^j$ be the polynomial of degree less than $\mathtt{t}$ such that $I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathtt{t}}}(i) = (r_1^{(i)}, \ldots, r_j^{(i)})$ for every $i \in [\mathtt{t}]$ (we implicitly associate the set $[\mathtt{t}]$ with an arbitrary set $S \subseteq \mathbb{F}$ of size $\mathtt{t}$.); for any $\gamma \in \mathbb{F}$, $I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathtt{t}}}(\gamma)$ can be computed in $\mathrm{poly}(\mathtt{t})$ operations. The interactive reduction is as follows:

- $\mathbf{P}_{\mathsf{IR}}$: Send the polynomial $g \in \mathbb{F}[X_1, X_2]$ defined as:

$$g(X_1, X_2) := h_{j+1}(I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathtt{t}}}(X_1), X_2) \ . \tag{4}$$

- $\mathbf{V}_{\mathsf{IR}}$: Receive a bivariate polynomial $\tilde{g} \in \mathbb{F}[X_1, X_2]$ of degree at most $j \cdot d \cdot (\mathtt{t} - 1)$ in X_1 and degree at most d in X_2.
  1. *Consistency:* Check that for every $i \in [\mathtt{t}]$ it holds that

$$p_j(I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathtt{t}}}(i)) \cdot \sum_{\alpha_{j+1} \in H} \tilde{g}(i, \alpha_{j+1}) = \gamma_j^{(i)} \ .$$

     (Reject if not.)
  2. *Generate new instance:*
     (a) Sample uniformly random field elements $\rho, r^* \leftarrow \mathbb{F}$ and send them to $\mathbf{P}_{\mathsf{IR}}$.
     (b) Output the new instance $\mathbb{z}' := (r'_1, \ldots, r'_j, r^*, \gamma_{j+1})$ where $(r'_1, \ldots, r'_j) := I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathtt{t}}}(\rho)$ and $\gamma_{j+1} := \tilde{g}(\rho, r^*)$.

Proof of Theorem 4.10. We argue completeness, then soundness, and then triviality of the predicate f_n.

Completeness. Fix $\mathbb{x} = (\mathbb{F}, H, n, d, p_0, \ldots, p_n, \gamma) \in L_{\mathsf{Shmr}}$ and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathtt{t}}$ such that $f_j(\mathbb{x}, \mathbb{z}_1) = \cdots = f_j(\mathbb{x}, \mathbb{z}_{\mathtt{t}}) = 1$ where $\mathbb{z}_i := (r_1^{(i)}, \ldots, r_j^{(i)}, \gamma_j^{(i)})$. We argue that

$$\Pr\left[\ f_j(\mathbb{x}, \mathbb{z}') = 1 \ \middle| \ \mathbb{z}' \leftarrow \langle \mathbf{P}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_1, \ldots, \mathbb{z}_{\mathtt{t}}), \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_1, \ldots, \mathbb{z}_{\mathtt{t}}) \rangle \ \right] = 1 \ .$$

For every $i \in [\mathtt{t}]$, since $f_j(\mathbb{x}, \mathbb{z}_i) = 1$, we have that

$$\begin{aligned} p_j(I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathtt{t}}}(i)) \cdot \sum_{\alpha_{j+1} \in H} g(i, \alpha_{j+1}) &= p_j(r_1^{(i)}, \ldots, r_j^{(i)}) \cdot \sum_{\alpha_{j+1} \in H} h_{j+1}(r_1^{(i)}, \ldots, r_j^{(i)}, \alpha_{j+1}) \\ &= h_j(r_1^{(i)}, \ldots, r_j^{(i)}) \\ &= \gamma_j^{(i)} \ . \end{aligned}$$

Therefore the reduction verifier $\mathbf{V}_{\mathsf{IR}}$ does not reject in the consistency test (in Item 1). Moreover, by the definition of g, for every ρ and r^*,

$$\begin{aligned} h_{j+1}(r'_1, \ldots, r'_j, r^*) &= h_{j+1}(I_{\mathbb{z}_1,\ldots,\mathbb{z}_{\mathtt{t}}}(\rho), r^*) \\ &= g(\rho, r^*) \end{aligned}$$

$$= \gamma_{j+1} \ .$$

Therefore we have $f_i(\mathbb{x}, \mathbb{z}') = 1$ with probability 1.

Soundness. Fix $\mathbb{x} = (\mathbb{F}, H, n, d, p_0, \ldots, p_n, \gamma) \notin L_{\mathsf{Shmr}}$ and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$ where $\mathbb{z}_i := (r_1^{(i)}, \ldots, r_j^{(i)}, \gamma_j^{(i)})$. Suppose that $f_j(\mathbb{x}, \mathbb{z}_i) = 0$ for some $i \in [\mathsf{t}]$. Fix a malicious reduction prover $\tilde{\mathbf{P}}_{\mathsf{IR}}$. We show that

$$\Pr\left[\ f_j(\mathbb{x}, \mathbb{z}') = 1 \ \middle| \ \mathbb{z}' \leftarrow \langle \tilde{\mathbf{P}}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}) \rangle \ \right] \leq \frac{d \cdot (1 + n \cdot (\mathsf{t} - 1))}{|\mathbb{F}|} \ .$$

If $\tilde{g}$ does not pass the consistency test (in Item 1) then the reduction verifier rejects. Thus, we can assume that $\tilde{g}$ passes the consistency test. That is, for every $i \in [\mathsf{t}]$ it holds that

$$p_j(I_{\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}}(i)) \cdot \sum_{\alpha_{j+1} \in H} \tilde{g}(i, \alpha_{j+1}) = \gamma_j^{(i)} \ .$$

Let g be defined as in Equation (4) with respect to $\mathbb{x}$ and $\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}$.

We have that $g \not\equiv \tilde{g}$ due to the fact that $f_j(\mathbb{x}, \mathbb{z}_i) = 0$, which implies that

$$\begin{aligned} p_j(I_{\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}}(i)) \cdot \sum_{\alpha_{j+1} \in H} g(i, \alpha_{j+1}) &= h_j(r_1^{(i)}, \ldots, r_j^{(i)}) \\ &\neq \gamma_j^{(i)} \\ &= p_j(I_{\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}}}(i)) \cdot \sum_{\alpha_{j+1} \in H} \tilde{g}(i, \alpha_{j+1}) \ . \end{aligned}$$

Thus, by applying the Schwartz–Zippel lemma, and recalling that g and $\tilde{g}$ both have degree $j \cdot d \cdot (\mathsf{t} - 1)$ in their first variable and d in their second variable, we get:

$$\Pr_{\rho, r^*}[\ g(\rho, r^*) = \tilde{g}(\rho, r^*) \ ] \leq \frac{d \cdot (1 + j \cdot (\mathsf{t} - 1))}{|\mathbb{F}|} \leq \frac{d \cdot (1 + n \cdot (\mathsf{t} - 1))}{|\mathbb{F}|} \ .$$

For any choice of ρ and r^* such that $g(\rho, r^*) \neq \tilde{g}(\rho, r^*)$, we have that

$$h_{j+1}(r'_1, \ldots, r'_j, r^*) = g(\rho, r^*) \neq \tilde{g}(\rho, r^*) = \gamma_{j+1} \ .$$

and so $f_{j+1}(\mathbb{x}, (r'_1, \ldots, r'_j, r^*, \gamma_{j+1})) = 0$ with probability at least $1 - \frac{d \cdot (1 + n \cdot (\mathsf{t} - 1))}{|\mathbb{F}|}$.

Triviality. By definition, $f_n(\mathbb{x}, \mathbb{z}) = f_n((\mathbb{F}, H, n, d, p_0, \ldots, p_n, \gamma), (r_1, \ldots, r_n, \gamma_n)) = 1$ if and only if

$$h_j(r_1, \ldots, r_n) \triangleq p_n(r_1, \ldots, r_n) = \gamma \ ,$$

which can be efficiently verified by querying p_n once.

Complexity measures. We discuss complexity measures of the protocol $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$.

- *Output length.* The output is a list of $j + 1 \leq n + 1$ field elements. Thus, the output length is at most $O(n \cdot \log |\mathbb{F}|)$, which is a fixed polynomial in the input length.
- *Rounds.* The interactive reduction has one round.

- *Communication complexity.* The prover sends g which is bivariate and has degree at most $n \cdot d \cdot (\mathtt{t}-1)$ in its first variable and d in its second variable. This requires sending $O(n\mathtt{t}d^2)$ field elements. The verifier sends two field elements. Thus the communication complexity of the protocol is $O(n\mathtt{t}d^2 \log |\mathbb{F}|)$ bits.
- *Randomness complexity.* $\mathbf{V}_{\mathsf{IR}}$ uses $2 \log |\mathbb{F}|$ random bits.
- *Verifier running time.* $\mathbf{V}_{\mathsf{IR}}$ runs in time $\mathrm{poly}(n, \log |\mathbb{F}|) = \mathrm{poly}(|\mathbb{x}|)$.
- *Final predicate time.* Computing the final predicate requires one call to p_n.

□

5 IOPs from interactive reducibility

We show how to use interactive reducibility to construct IOPs with small query complexity. In Section 5.1, we show that every language with a bounded-output-length interactive reduction (with good enough soundness) has a round-query IOP with round-query complexity $O(1)$. Recall that a round-query IOP with round-query complexity $\mathsf{q}_{\mathsf{rnd}}$ is an IOP in which the verifier reads $\mathsf{q}_{\mathsf{rnd}}$ rounds (prover and verifier messages) in their entirety. In Section 5.2, we show that, while the interactive reduction for any IP (in Section 4.3.1) does not have bounded output length, a similar claim can be made for any k-round IP; in this case, the resulting verifier queries $\max\{O(1), O(\mathsf{k}/\log|\mathbb{x}|)\}$ rounds.

Combining Theorem 5.3 and Theorem 7.1 (for transforming round-query IOPs into binary IOPs), we obtain the following corollary showing that languages with interactive reducibility have binary IOPs with constant query complexity.

Corollary 5.1 (restatement of Lemma 2)**.** *Let R be a relation with a bounded-output-length interactive reduction with well-behaved soundness error, k_{IR} predicates, and ℓ_{IR} rounds. Then R has a non-adaptive public-coin IOP with the parameters below.*

IOP for R	
Rounds	$O(\ell_{\mathsf{IR}} \cdot \mathsf{k}_{\mathsf{IR}})$
Proof length	$\mathrm{poly}(\lvert\mathbb{x}\rvert)$
Alphabet size	2
Queries	$O(\ell_{\mathsf{IR}})$
Interaction randomness	$\mathrm{poly}(\lvert\mathbb{x}\rvert)$
Decision randomness	$O(\log\lvert\mathbb{x}\rvert)$
Soundness error	$O(1)$
Verifier running time	$\mathrm{poly}(\lvert\mathbb{x}\rvert)$

Proof. We first amplify the soundness error of the interactive reduction using Theorem 4.5 until we have $\binom{2\cdot\mathsf{k}_{\mathsf{IR}}}{\mathsf{k}_{\mathsf{IR}}} \cdot \mathsf{k}_{\mathsf{IR}} \cdot \varepsilon_{\mathsf{IR}}(\mathbb{x}, 2\cdot\mathsf{k}_{\mathsf{IR}}) < \frac{1}{2}$. Since $\varepsilon_{\mathsf{IR}}$ is well-behaved and $\binom{2\cdot\mathsf{k}_{\mathsf{IR}}}{\mathsf{k}_{\mathsf{IR}}} < 2^{2\mathsf{k}_{\mathsf{IR}}}$, this incurs only a polynomial overhead. We then apply Theorem 5.3 with parameter $\tau = 2$ to get an IOP with $O(\ell_{\mathsf{IR}} \cdot \mathsf{k}_{\mathsf{IR}})$ rounds in which the verifier reads only $O(\ell_{\mathsf{IR}})$ of the rounds with alphabet size $2^{\mathrm{poly}(|\mathbb{x}|)}$. We then apply the transformation described Theorem 7.1 to transform this IOP into a binary one in which the verifier queries $O(\ell_{\mathsf{IR}})$ rounds and makes $O(1)$ queries to each round (including its own random messages). □

Similarly, we get the following corollary for any public-coin IP.

Corollary 5.2 (restatement of Lemma 1)**.** *Let R be a relation with a non-adaptive k_{IP}-round public-coin IP. Then R has an IOP with the parameters below.*

IOP for R	
Rounds	k_{IP}
Proof length	$\mathrm{poly}(\lvert\mathbb{x}\rvert)$
Alphabet size	2
Queries	$\max\{O(1), O(\mathsf{k}_{\mathsf{IP}}/\log\lvert\mathbb{x}\rvert)\}$
Interaction randomness	$\mathrm{poly}(\lvert\mathbb{x}\rvert)$
Decision randomness	$O(\log\lvert\mathbb{x}\rvert)$
Soundness error	$O(1)$
Verifier running time	$\mathrm{poly}(\lvert\mathbb{x}\rvert)$

Proof. Given an IP for the relation R, we first apply the Babai–Moran transformation [BM88] to reduce the number of rounds by a constant fraction (enough to counteract the constant added to the number of rounds by the next two transformations). Next, we apply Theorem 5.7 when setting $m := \max\{O(1), O(\mathsf{k}'_{\mathsf{IP}}/\log|\mathbb{x}|)\}$ (where $\mathsf{k}'_{\mathsf{IP}}$ is the number of rounds of the IP following the Babai–Moran transformation) and then apply Theorem 7.1 to the resulting IOP. □

5.1 Round-query IOPs from bounded-output-length interactive reductions

We show that any relation with a bounded-output-length interactive reduction has a round-query IOP with polynomial proof length with small round-query complexity.

Theorem 5.3. *Let $\tau \in \mathbb{N}$ and let R be a relation with a bounded-output-length interactive reduction $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ with well-behaved soundness error. Then R has a public-coin round-query IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ with the parameters indicated below.*

Interactive reduction for R	
Number of predicates	k_{IR}
Rounds	ℓ_{IR}
Communication	l_{IR}
Output length	s_{IR}
Randomness	r_{IR}
Soundness error	$\varepsilon_{\mathsf{IR}}$
Verifier running time	$\mathsf{vt}_{\mathsf{IR}}$
Final predicate time	$\mathsf{ft}_{\mathsf{IR}}$

$\longrightarrow$

Round-query IOP for R	
Rounds	$\tau \cdot \mathsf{k}_{\mathsf{IR}} \cdot \ell_{\mathsf{IR}}$
Proof length (per round)	$\tau \cdot \mathsf{k}_{\mathsf{IR}}^2 \cdot (\mathsf{s}_{\mathsf{IR}} + \mathsf{l}_{\mathsf{IR}})$
Round queries	$O(\ell_{\mathsf{IR}})$
Interaction randomness	$O(\tau \cdot \mathsf{k}_{\mathsf{IR}} \cdot \mathsf{r}_{\mathsf{IR}})$
Decision randomness	$O(\log(\tau \cdot \mathsf{k}_{\mathsf{IR}}))$
Soundness error	$\max\{1/\tau, \binom{\tau \cdot \mathsf{k}_{\mathsf{IR}}}{\mathsf{k}_{\mathsf{IR}}} \cdot \mathsf{k}_{\mathsf{IR}} \cdot \varepsilon_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{k}_{\mathsf{IR}})\}$
Verifier running time	$\mathrm{poly}(\tau, \mathsf{k}_{\mathsf{IR}}, \mathsf{s}_{\mathsf{IR}}, \mathsf{l}_{\mathsf{IR}}, \mathsf{vt}_{\mathsf{IR}}, \mathsf{ft}_{\mathsf{IR}})$

Construction 5.4. Let $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ be an interactive reduction for R with k_{IR} predicates.

The IOP prover $\mathbf{P}_{\mathsf{IOP}}$ receives as input an instance $\mathbb{x}$ and witness $\mathbb{w}$, and the IOP verifier $\mathbf{V}_{\mathsf{IOP}}$ receives as input the instance $\mathbb{x}$. They interact as follows.

1. For every $i \in \{0, \ldots, \tau \cdot \mathsf{k}_{\mathsf{IR}}\}$, $\mathbf{P}_{\mathsf{IOP}}$ defines the $(\mathsf{k}_{\mathsf{IR}} + 1)$-entry array A_i as follows

$$A_i[j] := \begin{cases} \{\perp\} & \text{if } j = 0 \\ \emptyset & \text{if } j \in \{1, \ldots, \mathsf{k}_{\mathsf{IR}}\} \end{cases} .$$

   The set $A_i[j]$ will be used to store all of the instances corresponding to f_j collected by iteration i of the protocol.

2. For $i = 1, \ldots, \tau \cdot \mathsf{k}_{\mathsf{IR}}$:
   (a) $\mathbf{P}_{\mathsf{IOP}}$ sends A_{i-1} to $\mathbf{V}_{\mathsf{IOP}}$.
   (b) Do in parallel for every $j \in [\mathsf{k}_{\mathsf{IR}}]$ such that $A_{i-1}[j-1] \neq \emptyset$:
      i. Execute the *interaction phase* of:

$$\mathbb{z}'_{i,j} \leftarrow \langle \mathbf{P}_{\mathsf{IR}}(\mathbb{x}, \mathbb{w}, A_{i-1}[j-1]), \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, A_{i-1}[j-1]) \rangle .$$

         Here the verifier only sends messages and does not verify the correctness of the execution. Hence $\mathbb{z}'_{i,j}$ is computed only by the prover. Moreover, the verifier's random messages are shared among the parallel executions of $\langle \mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}} \rangle$ over each choice of $j \in [\mathsf{k}_{\mathsf{IR}}]$.
      ii. $\mathbf{P}_{\mathsf{IOP}}$ sets $A_i[j] := A_{i-1}[j] \cup \{\mathbb{z}'_{i,j}\}$.

3. $\mathbf{P}_{\mathsf{IOP}}$ sends $A_{\tau\cdot\mathsf{k}_{\mathsf{IR}}}$. Additionally, for every $i \in \{0,\ldots,\tau\cdot\mathsf{k}_{\mathsf{IR}}\}$, it sends $B_i := A_i$. This concludes the interaction phase.

4. In the decision phase, $\mathbf{V}_{\mathsf{IOP}}$ is given oracle access to a transcript with the following structure:

$$\left(A_0, \{\mathsf{tr}_{1,j}\}_{j\in[\mathsf{k}_{\mathsf{IR}}]}, A_1, \cdots, \{\mathsf{tr}_{\tau\cdot\mathsf{k}_{\mathsf{IR}},j}\}_{j\in[\mathsf{k}_{\mathsf{IR}}]}, A_{\tau\cdot\mathsf{k}_{\mathsf{IR}}}, (B_0,\cdots,B_{\tau\cdot\mathsf{k}_{\mathsf{IR}}})\right) \ , \tag{5}$$

   where $\mathsf{tr}_{i,j}$ corresponds to the interaction between the prover and verifier in the j-th parallel execution of $(\mathbf{P}_{\mathsf{IR}},\mathbf{V}_{\mathsf{IR}})$ on the i-th iteration. $\mathbf{V}_{\mathsf{IOP}}$ performs the checks below.

   (a) *Subset consistency.* Read the arrays $B_0, B_1, \ldots, B_{\tau\cdot\mathsf{k}_{\mathsf{IR}}}$ in their entirety. For every $i \in [\tau\cdot\mathsf{k}_{\mathsf{IR}}]$ and $j \in \{0,\ldots,\mathsf{k}_{\mathsf{IR}}\}$ check that $B_{i-1}[j] \subseteq B_i[j]$.

   (b) *Transcript consistency.* Sample a random $i \in [\tau\cdot\mathsf{k}_{\mathsf{IR}}]$. Read the arrays A_{i-1} and A_i sent by $\mathbf{P}_{\mathsf{IOP}}$ and the entire ℓ_{IR}-round interaction $\{\mathsf{tr}_{i,j}\}_{j\in[\mathsf{k}_{\mathsf{IR}}]}$ between the prover and the verifier done during the i-th iteration of the the interaction phase.

      i. Check that $A_{i-1} = B_{i-1}$ and $A_i = B_i$.

      ii. For every $j \in [\mathsf{k}_{\mathsf{IR}}]$, check that $A_i[j] = A_{i-1}[j] \cup \{\mathbb{z}'_{i,j}\}$ where

$$\mathbb{z}'_{i,j} := \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, A_{i-1}[j-1]; \mathsf{tr}_{i,j}) \ .$$

      (If $\mathbf{V}_{\mathsf{IR}}$ rejects then $\mathbf{V}_{\mathsf{IOP}}$ immediately rejects.)

   (c) *Final predicate holds.* Check that $f_{\mathsf{k}_{\mathsf{IR}}}(\mathbb{x},\mathbb{z}) = 1$ for every $\mathbb{z} \in B_{\tau\cdot\mathsf{k}_{\mathsf{IR}}}[\mathsf{k}_{\mathsf{IR}}]$.

Proof of Theorem 5.3. We prove completeness and soundness; then we analyze the complexity measures of the protocol.

Completeness. Fix $(\mathbb{x},\mathbb{w}) \in R$. We consider each of $\mathbf{V}_{\mathsf{IOP}}$'s checks, and show that $\mathbf{P}_{\mathsf{IOP}}$'s messages satisfy each check with probability 1.

(a) *Subset consistency.* $\mathbf{P}_{\mathsf{IOP}}$ sets, in each round i and for every j, $A_i[j] := A_{i-1}[j] \cup \{\mathbb{z}'_{i,j}\}$ where $\mathbb{z}'_{i,j}$ is the output of the interactive reduction. Since $\mathbf{P}_{\mathsf{IOP}}$ is honest, $B_{i-1} = A_{i-1}$ and $B_i = A_i$. Together this implies that $B_{i-1}[j] \subseteq B_i[j]$ and so $\mathbf{V}_{\mathsf{IOP}}$'s subset consistency check passes.

(b) *Transcript consistency.* The conditions established in the previous item also directly imply that $\mathbf{V}_{\mathsf{IOP}}$'s transcript consistency check passes.

(c) *Final predicate holds.* Since $B_i := A_i$, if A_i contains only round instances for which the predicate holds, then so does B_i, and $\mathbf{V}_{\mathsf{IOP}}$ accepts. We prove by induction on $i \in \{0,\ldots,\tau\cdot\mathsf{k}_{\mathsf{IR}}\}$ that for every $j \in \{0,\ldots,\mathsf{k}_{\mathsf{IR}}\}$ and $\mathbb{z} \in A_i[j]$ it holds that $f_j(\mathbb{x},\mathbb{z}) = 1$.

   First we consider the case when $j = 0$ (and any i). For every i, we have that $A_i[0] = \{\perp\}$. Since $\mathbb{x} \in L(R)$, we have that $f_j(\mathbb{x},\perp) = 1$.

   Next we consider the case when $j > 0$.

   - $i = 0$: For every $j \in [\mathsf{k}_{\mathsf{IR}}]$ we set $A_0[j] = \emptyset$ and so the property holds (trivially) for these values of j.
   - $i > 0$: Suppose that, with probability 1, for every $j \in [\mathsf{k}_{\mathsf{IR}}]$ and $\mathbb{z}_i \in A_i[j]$ it holds that $f_j(\mathbb{x},\mathbb{z}_i) = 1$. We argue that, with probability 1, for every $j \in [\mathsf{k}_{\mathsf{IR}}]$ and $\mathbb{z}_{i+1} \in A_{i+1}[j]$, it holds that $f_j(\mathbb{x},\mathbb{z}_{i+1}) = 1$. Fix $j > 0$. Since for every $\mathbb{z}_i \in A_i[j]$, $f_j(\mathbb{x},\mathbb{z}_i) = 1$ and $A_i[j] := A_{i-1}[j] \cup \{\mathbb{z}'_{i,j}\}$, we need only show that $f(\mathbb{x},\mathbb{z}'_{i,j}) = 1$. $\mathbb{z}'_{i,j}$ is generated by applying

the interactive reduction with inputs $\mathbb{x}$ and $\{\mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}_{i,j}}\} := A_{i-1}[j-1]$. Since for every $\mathbb{z}' \in A_{i-1}[j-1]$, $f_{j-1}(\mathbb{x}, \mathbb{z}') = 1$, by completeness of the interactive reduction we have that

$$\Pr\left[\ f_j(\mathbb{x}, \mathbb{z}'_{i,j}) = 1 \,\middle|\, \mathbb{z}'_{i,j} \leftarrow \langle \mathbf{P}_{\mathsf{IR}}(\mathbb{x}, \mathbb{w}, \mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}_{i,j}}), \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, \mathbb{z}_1, \ldots, \mathbb{z}_{\mathsf{t}_{i,j}})\rangle\ \right] = 1\ .$$

Soundness. Fix $\mathbb{x} \notin R(L)$ and an IOP prover $\tilde{\mathbf{P}}_{\mathsf{IOP}}$. We show that $\mathbf{V}_{\mathsf{IOP}}$ accepts with probability at most $\max\{1/\tau, \binom{\tau \cdot \mathsf{k}_{\mathsf{IR}}}{\mathsf{k}_{\mathsf{IR}}} \cdot \mathsf{k}_{\mathsf{IR}} \cdot \varepsilon_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{k}_{\mathsf{IR}})\}$. Consider a transcript tr of the interaction of the IOP with the structure described in Equation (5). For fixed tr we say that an index $i \in [\tau \cdot \mathsf{k}_{\mathsf{IR}}]$ is *consistent* if:

- $A_{i-1} = B_{i-1}$ and $A_i = B_i$ and,
- For every $j \in [\mathsf{k}_{\mathsf{IR}}]$: $A_i[j] = A_{i-1}[j] \cup \{\mathbb{z}'_{i,j}\}$ where $\mathbb{z}'_{i,j} := \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, A_{i-1}[j]; \mathsf{tr}_{i,j})$ (if $\mathbf{V}_{\mathsf{IR}}$ rejects then i is not consistent).

We give two claims analyzing the probability that the verifier accepts relative to how many indices of the protocol are consistent. Claim 5.5 shows that in any execution of the protocol with less than k_{IR} consistent indices, $\mathbf{V}_{\mathsf{IOP}}$ accepts with probability at most $1/\tau$. By Claim 5.6, for any fixed set of k_{IR} indices, conditioned on the interaction outputting a transcript in which these indices are consistent, $\mathbf{V}_{\mathsf{IOP}}$ accepts with probability at most $\mathsf{k}_{\mathsf{IR}} \cdot \varepsilon_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{k}_{\mathsf{IR}})$. Since there are $\binom{\tau \cdot \mathsf{k}_{\mathsf{IR}}}{\mathsf{k}_{\mathsf{IR}}}$ choices of k_{IR} indices, we can conclude that, conditioned on the transcript having at least k_{IR} consistent indices, $\mathbf{V}_{\mathsf{IOP}}$ accepts with probability at most $\binom{\tau \cdot \mathsf{k}_{\mathsf{IR}}}{\mathsf{k}_{\mathsf{IR}}} \cdot \mathsf{k}_{\mathsf{IR}} \cdot \varepsilon_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{k}_{\mathsf{IR}})$. Putting the two cases together, we conclude that $\mathbf{V}_{\mathsf{IOP}}$ accepts with probability at most $\max\{1/\tau, \binom{\tau \cdot \mathsf{k}_{\mathsf{IR}}}{\mathsf{k}_{\mathsf{IR}}} \cdot \mathsf{k}_{\mathsf{IR}} \cdot \varepsilon_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{k}_{\mathsf{IR}})\}$.

Claim 5.5. *Conditioned on the transcript* tr *generated by* $\langle \tilde{\mathbf{P}}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}}(\mathbb{x})\rangle$ *being consistent with less than* k_{IR} *indices,* $\mathbf{V}_{\mathsf{IOP}}$ *accepts with probability at most* $1/\tau$ *(over its decision randomness).*

Proof. $\mathbf{V}_{\mathsf{IOP}}$ accepts only if its choice of $i \in [\tau \cdot \mathsf{k}_{\mathsf{IR}}]$ in Item 4b is consistent. Since there are at most k_{IR} consistent indices, the probability of the verifier sampling one of these indices is at most $1/\tau$. □

Claim 5.6. *Fix indices* $1 \le i_1 < \cdots < i_{\mathsf{k}_{\mathsf{IR}}} \le \tau \cdot \mathsf{k}_{\mathsf{IR}}$. *Then, conditioned on the transcript* tr *generated by* $\langle \tilde{\mathbf{P}}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}}(\mathbb{x})\rangle$ *being consistent with respect to indices* $i_1, \ldots, i_{\mathsf{k}_{\mathsf{IR}}}$, $\mathbf{V}_{\mathsf{IOP}}$ *accepts with probability at most* $\mathsf{k}_{\mathsf{IR}} \cdot \varepsilon_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{k}_{\mathsf{IR}})$ *(over its interaction randomness).*

Proof. Let $i_0 := 0$. We show by induction that for every $j \in \{0, \ldots, \mathsf{k}_{\mathsf{IR}}\}$ there exists $\mathbb{z} \in A_{i_j}[j]$ such that $f_j(\mathbb{x}, \mathbb{z}) = 0$ except with probability $j \cdot \varepsilon_{\mathsf{IR}}$. This implies the statement, since if $f_{\mathsf{k}_{\mathsf{IR}}}(\mathbb{x}, \mathbb{z}) = 0$, then $\mathbf{V}_{\mathsf{IOP}}$ rejects in Item 4c.

This is immediate for $j = 0$, since $\mathbb{x} \notin L(R)$ and so $f_0(\mathbb{x}, \mathbb{z}) = 0$ for any $\mathbb{z}$. Fix some $j > 0$ and suppose that there exists $\mathbb{z} \in A_{i_{(j-1)}}[j-1]$ such that $f_{j-1}(\mathbb{x}, \mathbb{z}) = 0$ (which happens with probability $1 - (j-1) \cdot \varepsilon_{\mathsf{IR}}$). We first show that $\mathbb{z} \in A_{i_j}[j-1]$. This can be seen by the following:

$$\mathbb{z} \in A_{i_{j-1}}[j-1] = B_{i_{j-1}}[j-1] \subseteq B_{i_j-1}[j-1] = A_{i_j-1}[j-1]\ ,$$

where the equalities are due to the fact that the indices $i_{(j-1)}$ and i_j are consistent. The containment of $B_{i_{(j-1)}}[j-1]$ in $B_{i_j-1}[j-1]$ is due to the fact that $i_{(j-1)} \le i_j - 1$ and that, in order for the verifier to accept the transcript, the check in Item 4a must pass.

This implies that for any $\tilde{\mathbf{P}}_{\mathsf{IR}}$ (and in particular for whatever $\tilde{\mathbf{P}}_{\mathsf{IOP}}$ does in this stage):

$$\Pr\left[\ f_j(\mathbb{x}, \mathbb{z}'_{i_j,j}) = 1 \,\middle|\, \mathbb{z}'_{i_j,j} \leftarrow \langle \tilde{\mathbf{P}}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}}(\mathbb{x}, A_{i_j-1}[j-1])\rangle\ \right] \le \varepsilon_{\mathsf{IR}}(\mathbb{x}, |A_{i_j-1}[j-1]|) \le \varepsilon_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{k}_{\mathsf{IR}})\ ,$$

where the final inequality is true since $|A_{i_j-1}[j-1]| \le \tau \cdot \mathsf{k}_{\mathsf{IR}}$, when additionally noting that $\varepsilon_{\mathsf{IR}}$ is well-behaved. Using again the fact that index i_j is consistent, we have that $A_{i_j}[j] := A_{i_j-1}[j] \cup \{\mathbb{z}'_{i_j,j}\}$

where $\mathbb{z}'_{i_j,j}$ is a round instance generated by the actual interaction between $\tilde{\mathbf{P}}_{\mathsf{IOP}}$ and $\mathbf{V}_{\mathsf{IOP}}$ during the j-th parallel execution of the interactive reduction in iteration i_j. Therefore, we have that, conditioned there existing $\mathbb{z} \in A_{i_{(j-1)}}[j-1]$ where $f_{j-1}(\mathbb{x}, \mathbb{z}) = 0$, there exists $\mathbb{z}'_{i_j,j} \in A_{i_j}[j]$ where $f_j(\mathbb{x}, \mathbb{z}'_{i_j,j}) = 0$ except with probability $\varepsilon_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{k}_{\mathsf{IR}})$. By the induction hypothesis, the condition happens with probability at least $1 - (j-1) \cdot \varepsilon_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{k}_{\mathsf{IR}})$ and so we have that with probability at least $1 - j \cdot \varepsilon_{\mathsf{IR}}(\mathbb{x}, \tau \cdot \mathsf{k}_{\mathsf{IR}})$, there exists $\mathbb{z}'_{i_j,j} \in A_{i_j}[j]$ where $f_j(\mathbb{x}, \mathbb{z}'_{i_j,j}) = 0$. □

Complexity measures. We analyze the efficiency parameters of the IOP:

- *Proof length.* The largest message sent by $\mathbf{P}_{\mathsf{IOP}}$ is the final round, which contains all of the instances generated in the previous interaction. In each of the $\tau \cdot \mathsf{k}_{\mathsf{IR}}$ rounds, one instance is added to the list sent to the verifier for every $j \in [\mathsf{k}_{\mathsf{IR}}]$. The output of the interactive reduction has length at most s_{IR}. Thus this requires sending $O(\tau \cdot \mathsf{k}_{\mathsf{IR}}^2 \cdot \mathsf{s}_{\mathsf{IR}})$ bits. In every execution of the interactive reduction an additional l_{IR} bits are sent, for a total of $\mathsf{k}_{\mathsf{IR}} \cdot \mathsf{l}_{\mathsf{IR}}$ per round. Thus the final proof is $O(\tau \cdot \mathsf{k}_{\mathsf{IR}}^2 \cdot (\mathsf{s}_{\mathsf{IR}} + \mathsf{l}_{\mathsf{IR}}))$ bits long.
- *Round complexity.* The IOP contains $\tau \cdot \mathsf{k}_{\mathsf{IR}}$ executions of the ℓ_{IR}-round interactive reduction $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$. Therefore the IOP has $\tau \cdot \mathsf{k}_{\mathsf{IR}} \cdot \ell_{\mathsf{IR}}$ rounds.
- *Round queries.* $\mathbf{V}_{\mathsf{IOP}}$ reads in its entirety the last message in the protocol, reads A_{i-1} and A_i, and reads the execution of the interactive reduction between blocks $i-1$ and i. Therefore it reads $O(\ell_{\mathsf{IR}})$ rounds.
- *Randomness complexity.* $\mathbf{V}_{\mathsf{IOP}}$ generates r_{IR} bits for every separate $i \in [\tau \cdot \mathsf{k}_{\mathsf{IR}}]$ since we share verifier randomness between each of the parallel executions of $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$. The interaction randomness is therefore $O(\tau \cdot \mathsf{k}_{\mathsf{IR}} \cdot \mathsf{r}_{\mathsf{IR}})$ bits. There are $O(\log(\tau \cdot \mathsf{k}_{\mathsf{IR}}))$ bits of decision randomness because $\mathbf{V}_{\mathsf{IOP}}$ samples a random $i \in [\tau \cdot \mathsf{k}_{\mathsf{IR}}]$.
- *Verifier running time.* $\mathbf{V}_{\mathsf{IOP}}$ checks all of the arrays $B_1, \ldots, B_{\tau \cdot \mathsf{k}_{\mathsf{IR}}}$ for consistency, executes the interactive reduction k_{IR} times (each in time $\mathsf{vt}_{\mathsf{IR}}$), and verifies the final predicate of at most $\tau \cdot \mathsf{k}_{\mathsf{IR}}$ elements in $B_{\tau \cdot \mathsf{k}_{\mathsf{IR}}}$ (each in time $\mathsf{ft}_{\mathsf{IR}}$). Overall $\mathbf{V}_{\mathsf{IOP}}$ runs in time $\mathrm{poly}(\tau, \mathsf{k}_{\mathsf{IR}}, \mathsf{s}_{\mathsf{IR}}, \mathsf{l}_{\mathsf{IR}}, \mathsf{vt}_{\mathsf{IR}}, \mathsf{ft}_{\mathsf{IR}})$.
- *Adaptivity.* $\mathbf{V}_{\mathsf{IOP}}$ is non-adaptive.

□

5.2 Round-query IOPs from public-coin IPs

The interactive reduction for any public-coin IP described in Section 4.3.1 does not have bounded output length. Hence we cannot use it directly in Theorem 5.3. Nonetheless, the output length grows slowly enough to achieve weaker results. While the transformation is essentially identical to the protocol described in Construction 5.4, we describe it explicitly.

Theorem 5.7. *Let R be a relation with a public-coin IP $(\mathbf{P}_{\mathsf{IP}}, \mathbf{V}_{\mathsf{IP}})$ with k_{IP} rounds. Let $\tau, m \geq 1$ be parameters where m divides k_{IP}. Then R has a round-query IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ with the parameters indicated below.*

IP for R	
Rounds	k_{IP}
Total communication	l_{IP}
Randomness per round	r_{IP}
Round-by-round soundness	β_{rbr}
Verifier running time	$\mathsf{vt}_{\mathsf{IP}}$

$\longrightarrow$

Round-query IOP for R	
Rounds	$\tau \cdot \mathsf{k}_{\mathsf{IP}}$
Proof length (per round)	$O(\mathsf{l}_{\mathsf{IP}} \cdot 2^{\tau \cdot \mathsf{k}_{\mathsf{IP}}/m})$
Round queries	$O(m)$
Interaction randomness	$O(\tau \cdot \mathsf{r}_{\mathsf{IP}} \cdot \mathsf{k}_{\mathsf{IP}})$
Decision randomness	$O(\log(\tau \cdot \mathsf{k}_{\mathsf{IP}}/m))$
Soundness error	$\max\{1/\tau, \binom{\tau \cdot \mathsf{k}_{\mathsf{IP}}/m}{\mathsf{k}_{\mathsf{IP}}/m} \cdot m \cdot \mathsf{k}_{\mathsf{IP}} \cdot \beta_{\mathrm{rbr}}\}$
Verifier running time	$\mathrm{poly}(2^{\tau \cdot \mathsf{k}_{\mathsf{IP}}/m}, \mathsf{l}_{\mathsf{IP}}, \mathsf{vt}_{\mathsf{IR}})$

Construction 5.8. Let $(\mathbf{P}_{\mathsf{IP}}, \mathbf{V}_{\mathsf{IP}})$ be a public-coin IP with k_{IP} rounds and where the verifier message at each round is r_{IP} bits long. Let $\tau > 1$ be a parameter. On input $\mathbb{x}$ and with witness $\mathbb{w}$ the IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ is as follows:

- Prover sets $S_0 := \{\emptyset\}$ (i.e., S_0 contains only the empty transcript).
- For $i = 1$ to $\tau \cdot \mathsf{k}_{\mathsf{IP}}/m$:
  - Prover sends S_{i-1}.
  - For $j \in [m]$:
    1. Verifier chooses and sends $\rho_{i,j} \leftarrow \{0,1\}^{\mathsf{r}_{\mathsf{IP}}}$ uniformly at random.
    2. For every $\mathsf{tr} \in S_{i-1}$, prover sends $a_{\mathsf{tr},j} := \mathbf{P}_{\mathsf{IP}}(\mathbb{x}, \mathbb{w}, \mathsf{tr}||\rho_{i,1}||a_{\mathsf{tr},1}|| \ldots ||\rho_{i,j})$.
  - Prover sets $S_i := S_{i-1} \cup \{(\mathsf{tr}||\rho_{i,1}||a_{\mathsf{tr},1}|| \ldots ||\rho_{i,m}||a_{i,m})\}_{\mathsf{tr} \in S_{i-1}}$.
- For every $i \in [\tau \cdot \mathsf{k}_{\mathsf{IP}}/m]$, the Prover sends $T_i := S_i$.
- Verifier accepts if and only if:
  1. *Subset consistency:* For every $i \in [\tau \cdot \mathsf{k}_{\mathsf{IP}}/m]$, $T_{i-1} \subseteq T_i$.
  2. *Transcript consistency:* Choose a random $i \in [\tau \cdot \mathsf{k}_{\mathsf{IP}}/m]$. Check that $S_{i-1} = T_{i-1}$ and $S_i = T_i$. Additionally, assert that for every $\mathsf{tr} \in S_{i-1}$, there are some messages $a_{\mathsf{tr},1} \ldots, a_{\mathsf{tr},m}$ such that $(\mathsf{tr}||\rho_{i,1}||a_{\mathsf{tr},1}|| \ldots ||\rho_{i,m}||a_{i,m}) \in S_i$, where $\rho_{i,1} \ldots, \rho_{i,m}$ are the verifier messages sent during the i-th round of interaction.
  3. *Membership:* Every transcript $\mathsf{tr} \in T_{\tau \cdot \mathsf{k}_{\mathsf{IP}}/m}$ that is complete (i.e., contains messages for all k_{IP} rounds of the IP) has $\mathbf{V}_{\mathsf{IP}}(\mathbb{x}, \mathsf{tr}) = 1$.

Proof of Theorem 5.7. The proofs of completeness and soundness are identical to those for Theorem 5.3, using the interactive reduction for IPs described in Theorem 4.7. We therefore only need to analyze the complexity measures of the resulting IOP.

- *Proof length.* The proof length of the protocol is $O(\mathsf{l}_{\mathsf{IP}} \cdot 2^{\tau \cdot \mathsf{k}_{\mathsf{IP}}/m})$ bits.
- *Rounds.* The protocol has $\tau \cdot \mathsf{k}_{\mathsf{IP}}$ rounds.
- *Round queries.* The verifier reads $O(m)$ rounds.
- *Randomness complexity.* The IOP verifier uses $O(\tau \cdot \mathsf{r}_{\mathsf{IP}} \cdot \mathsf{k}_{\mathsf{IP}})$ bits of interaction randomness and $O(\log(\tau \cdot \mathsf{r}_{\mathsf{IP}}/m))$ bits of decision randomness .
- *Verifier running time.* The verifier running time is $\mathrm{poly}(2^{\tau \cdot \mathsf{k}_{\mathsf{IP}}}, \mathsf{l}_{\mathsf{IP}}, \mathsf{vt}_{\mathsf{IR}})$.
- *Adaptivity.* $\mathbf{V}_{\mathsf{IOP}}$ is non-adaptive.

□

6 Hardness of approximation for stochastic problems

We use the transformations described in Section 5 to construct IOPs with small query complexity and show that they imply hardness of approximation results for certain stochastic problems. In Section 6.1 we define stochastic constraint satisfaction problems (SCSP) and adapt a theorem of [ACY22] to this setting. Then in Section 6.2, we establish hardness of approximation results for SCSPs.

6.1 Stochastic constraint satisfaction problems

In a constraint satisfaction problem (CSP) with l variables over an alphabet Σ, we are given a set of m constraints, each of which takes as input q variables (out of the l) and outputs a bit denoting whether the constraint is satisfied. Works on hardness of approximation of CSPs generally study the complexity of distinguishing whether there exists an assignment to the variables that satisfies all of the constraints or whether every assignment satisfies at most a constant fraction of the constraints.

We extend CSPs to the stochastic setting, where the variables are split into two types: variables chosen randomly; and variables chosen existentially.

Definition 6.1. *An* $(\Sigma, \mathsf{k}, \mathsf{l}, \mathsf{m}, \mathsf{q})$-SCSP *(***stochastic CSP***) instance* Φ *with alphabet* Σ*,* k *alternations,* l *variables per quantifier,* m *constraints and* q *constraint arity is a list of* m *constraints* $C_1, \ldots, C_{\mathsf{m}}$*. Each constraint consists of* q *tuples* $(i_1, j_1, k_1), \ldots, (i_{\mathsf{q}}, j_{\mathsf{q}}, k_{\mathsf{q}}) \subseteq [\mathsf{k}] \times \{a, \rho\} \times [\mathsf{l}]$ *and a function* $f \colon \Sigma^{\mathsf{q}} \to \{0, 1\}$*.*

An assignment $z \colon [\mathsf{k}] \times \{a, \rho\} \times [\mathsf{l}] \to \Sigma$ *satisfies* a constraint C if

$$f\big(z(i_1, j_1, k_1), \ldots, z(i_{\mathsf{q}}, j_{\mathsf{q}}, k_{\mathsf{q}})\big) = 1 \ .$$

An instance Φ is in the language $(\Sigma, \mathsf{k}, \mathsf{l}, \mathsf{m}, \mathsf{q})$-SCSP if for uniformly random $z_{1,\rho} \colon [\mathsf{l}] \to \Sigma$ there is a choice of $z_{1,a} \colon [\mathsf{l}] \to \Sigma$ such that for random $z_{2,\rho} \colon [\mathsf{l}] \to \Sigma$, and so on, the probability that the assignment $z(i, j, k) := z_{i,j}(k)$ satisfies all of the constraints in Φ is greater than $1/2$.

Definition 6.2. *The* **value** *of a* $(\Sigma, \mathsf{k}, \mathsf{l}, \mathsf{m}, \mathsf{q})$-SCSP *instance* Φ *is the expected fraction of satisfied constraints if the existential variables are chosen to maximize the number of satisfied constraints.*

Theorem 6.3 (implied by [ACY22])**.** *Let* R *be a relation with a non-adaptive* k*-round public-coin IOP with alphabet* Σ*, per-round message length* l *(for both the prover and verifier messages), query complexity* q*, decision randomness* r_{dc}*, decision complexity* d*, and soundness error* β*.*

Then there exists a deterministic polynomial-time reduction that maps an instance $\mathtt{x}$ *for* L *to an instance* Φ *for* $(\Sigma, \mathsf{k}, \mathsf{l}, 2^{\mathsf{r}_{\mathsf{dc}}}, \mathsf{q})$-SCSP *such that:*

- *if* $\mathtt{x} \in L$ *then the value of* Φ *is* 1*;*
- *if* $\mathtt{x} \notin L$ *then the value of* Φ *is at most* β*.*

Moreover, each constraint in Φ *has circuit complexity* $O(\mathsf{d})$*.*

6.2 Hardness of approximation for SCSP

By combining Corollary 5.1 with Theorem 6.3, we show that approximating the value of SCSPs to within a constant factor is at least as hard as solving all languages with interactive reducibility.

Theorem 6.4 (restatement of Theorem 2)**.** *Let R be a relation with a bounded-output-length interactive reduction with well-behaved soundness error, k_{IR} predicates and ℓ_{IR} rounds. Then there exists a deterministic polynomial-time reduction that maps an instance $\mathbb{x}$ for L to an* SCSP *instance Φ such that:*

- *if $\mathbb{x} \in L$ then the value of Φ is 1;*
- *if $\mathbb{x} \notin L$ then the value of Φ is at most $1/2$.*

Moreover, Φ has a binary alphabet, $\ell_{\mathsf{IR}}(|\mathbb{x}|) \cdot \mathsf{k}_{\mathsf{IR}}(|\mathbb{x}|)$ alternations, $\mathrm{poly}(|\mathbb{x}|)$ *variables per quantifier,* $\mathrm{poly}(|\mathbb{x}|)$ *constraints and constraint arity $O(1)$.*

Similarly, by combining Corollary 5.2 with Theorem 6.3, we achieve a theorem for any IP.

Theorem 6.5 (restatement of Theorem 1)**.** *Let $L \in \mathrm{AM}[\mathsf{k}]$ be a language. Then there exists a deterministic polynomial-time reduction that maps an instance $\mathbb{x}$ for L to an* SCSP *instance Φ such that:*

- *if $\mathbb{x} \in L$ then the value of Φ is 1;*
- *if $\mathbb{x} \notin L$ then the value of Φ is at most $1/2$.*

Moreover, Φ has a binary alphabet, $\mathsf{k}(|\mathbb{x}|)$ alternations, $\mathrm{poly}(|\mathbb{x}|)$ *variables per quantifier,* $\mathrm{poly}(|\mathbb{x}|)$ *constraints and constraint arity* $\max\{O(1), \frac{\mathsf{k}(|\mathbb{x}|)}{\log|\mathbb{x}|}\}$.

7 From round-query IOPs to binary IOPs

We show how to transform a round-query IOP with round-query complexity $\mathsf{q}_{\mathsf{rnd}}$ into a binary IOP with query complexity $O(\mathsf{q}_{\mathsf{rnd}})$. This adapts a result of [ACY22] that transforms an IP into a binary IOP to also apply to round-query IOPs.

Theorem 7.1. *Let* IOP *be a non-adaptive public-coin round-query IOP for a relation $R = \{(\mathbb{x}, \mathbb{w})\}$. Then there exists a non-adaptive public-coin IOP for R with the parameters below.*

Round-query IOP for R	
Rounds	$\mathsf{k}_{\mathsf{IOP}}$
Proof length	$\mathsf{l}_{\mathsf{IOP}}$
Round queries	$\mathsf{q}_{\mathsf{rnd}}$
Interaction randomness	$\mathsf{r}_{\mathsf{int}}$
Decision randomness	r_{dc}
Soundness error	$O(1)$
Verifier running time	$\mathsf{vt}_{\mathsf{IOP}}$

$\longrightarrow$

IOP for R	
Rounds	$O(\mathsf{k}_{\mathsf{IOP}})$
Alphabet size	2
Proof length	$\mathrm{poly}(\lvert\mathbb{x}\rvert, \mathsf{l}_{\mathsf{IOP}})$
Queries	$O(\mathsf{q}_{\mathsf{rnd}})$
Interaction randomness	$\mathrm{poly}(\lvert\mathbb{x}\rvert, \mathsf{r}_{\mathsf{int}})$
Decision randomness	$O(\log \lvert\mathbb{x}\rvert + \mathsf{r}_{\mathsf{dc}})$
Soundness error	$O(1)$
Verifier running time	$\mathrm{poly}(\mathsf{vt}_{\mathsf{IOP}})$

The transformation of [ACY22] is obtained via two transformations:

- *Local access to interaction randomness.* The IP is transformed into an IOP in which the verifier reads only $O(1)$ bits from each of its random messages but still reads the prover's messages in their entirety. The transformation described in [ACY22] does not immediately apply to our setting (i.e., if we start out with a round-query IOP rather than an IP). In Section 7.1 we show how to adapt their proof to our setting.
- *Local access to prover messages.* The IOP where the verifier has local access to its interaction randomness is transformed into an IOP in which the verifier queries only a few bits of the entire transcript. In our case, we begin with an IOP in which the verifier has local access to its interaction randomness and only reads a small number of rounds overall. We discuss why the transformation of [ACY22] is sufficient for our setting in Section 7.2.

Putting together the two (modified) transformations yields Theorem 7.1.

7.1 Local access to interaction randomness

This transformation maps an IP into an IOP where the verifier reads the prover's messages exactly in their entirety but reads only a few bits of each one of its own random messages. In more detail, we prove that a $\mathsf{k}_{\mathsf{IOP}}$-round round-query public-coin IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ with round-query complexity $\mathsf{q}_{\mathsf{rnd}}$ and constant soundness error can be transformed into a $O(\mathsf{k}_{\mathsf{IOP}})$-round round-query public-coin IOP $(\mathbf{P}'_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}})$ with round-query complexity $O(\mathsf{q}_{\mathsf{rnd}})$ and constant soundness error, in which the verifier reads $O(\mathsf{q}_{\mathsf{rnd}})$ *bits* from its interaction randomness, but still reads $O(\mathsf{q}_{\mathsf{rnd}})$ messages sent by the prover in their entirety.

Construction 7.2. On input $\mathbb{x}$, with parameters $n_z, n_s \in \mathbb{N}$ and (constant) $\gamma \in (0,1)$ the protocol $(\mathbf{P}'_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}})$ works as follows, given an extractor $\mathsf{Ext}\colon \{0,1\}^{n_z} \times \{0,1\}^{n_s} \to \{0,1\}^{\mathsf{r}_{\mathsf{IOP}}}$ with error $\varepsilon_{\mathsf{Ext}}$. The parameters n_z, n_s, $\varepsilon_{\mathsf{Ext}}$ and γ will be fixed during the analysis.

1. The IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ has constant soundness error. Using Lemma 3.3, augment $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ such that it has $(1/(|\mathbb{x}| + \mathsf{k}_{\mathsf{IOP}}^2), \gamma/4)$-round-by-round soundness error, and interaction randomness complexity $\mathsf{r}'_{\mathsf{IOP}} = \mathrm{poly}(|\mathbb{x}| + \mathsf{k}_{\mathsf{IOP}})$ and decision randomness complexity $\mathsf{r}'_{\mathsf{dc}} = O(\mathsf{r}_{\mathsf{dc}})$, where r_{dc} is the decision randomness of $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$. Since γ is a constant, the augmented IOP has round-query complexity $O(\mathsf{q}_{\mathsf{rnd}})$.

2. For $j = 1, \ldots, \mathsf{k}_{\mathsf{IOP}}$:

   (a) $\mathbf{V}'_{\mathsf{IOP}}$: Send to the prover a random string $z_j \leftarrow \{0,1\}^{n_z}$.

   (b) $\mathbf{P}'_{\mathsf{IOP}}(\mathbb{x}, \mathbb{w}, z_1, s_1, \ldots, z_j)$: Respond with $z'_j \in \{0,1\}^{n_z}$ where (honestly) $z'_j := z_j$.

   (c) $\mathbf{V}'_{\mathsf{IOP}}$: Send to the prover a random seed $s_j \leftarrow \{0,1\}^{n_s}$.

   (d) $\mathbf{P}'_{\mathsf{IP}}(\mathbb{x}, \mathbb{w}, z_1, s_1, \ldots, z_j, s_j)$:

      i. Compute $\rho_j := \mathsf{Ext}(z'_j, s_j)$.

      ii. Compute $a_j \leftarrow \mathbf{P}_{\mathsf{IOP}}(\mathbb{x}, \mathbb{w}, \rho_1, \ldots, \rho_j)$.

      iii. Send (s_j, a_j) to the verifier.

3. $\mathbf{V}'^{\mathsf{tr}}_{\mathsf{IOP}}(\mathbb{x})$ where $\mathsf{tr} = \left(z_1, z'_1, s_1, s'_1, a_1, \ldots, z_{\mathsf{k}_{\mathsf{IOP}}}, z'_{\mathsf{k}_{\mathsf{IOP}}}, s_{\mathsf{k}_{\mathsf{IOP}}}, s'_{\mathsf{k}_{\mathsf{IOP}}}, a_{\mathsf{k}_{\mathsf{IOP}}}\right)$. Do the following:

   (a) Sample decision randomness $\rho_{\mathsf{dc}} \leftarrow \{0,1\}^{\mathsf{r}_{\mathsf{dc}}}$ for $\mathbf{V}_{\mathsf{IOP}}$ and sample random indices $m_z \leftarrow [n_z]$ and $m_s \leftarrow [n_s]$.

   (b) Emulate $\mathbf{V}_{\mathsf{IOP}}^{\rho_1, a_1, \ldots, \rho_{\mathsf{k}_{\mathsf{IOP}}}, a_{\mathsf{k}_{\mathsf{IOP}}}}(\mathbb{x}; \rho_{\mathsf{dc}})$ where queries are answered as follows:

   - Any query made by $\mathbf{V}_{\mathsf{IOP}}$ to a prover message a_i is answered by querying the corresponding prover message sent during interaction.
   - If $\mathbf{V}_{\mathsf{IOP}}$ attempts to read ρ_j then: Check that $z_j[m_z] = z'_j[m_z]$ and $s_j[m_s] = s'_j[m_s]$. If the check fails then reject. Otherwise, pass the string $\rho_j := \mathsf{Ext}(z'_j, s'_j)$ to $\mathbf{V}_{\mathsf{IOP}}$.

   (c) Accept if and only if $\mathbf{V}_{\mathsf{IOP}}^{\rho_1, a_1, \ldots, \rho_{\mathsf{k}_{\mathsf{IOP}}}, a_{\mathsf{k}_{\mathsf{IOP}}}}(\mathbb{x}; \rho_{\mathsf{dc}}) = 1$.

Completeness. Fix $(\mathbb{x}, \mathbb{w}) \in R$. Let tr be a transcript generated in a random execution of the protocol. Since the prover is honest, we have that $z'_j = z_j$ and $s'_j = s_j$, and so the verifier's checks in Item 3b pass with probability 1. Moreover, since the original IP has perfect completeness, and for every j, $a_j := \mathbf{P}_{\mathsf{IOP}}(\mathbb{x}, \mathbb{w}, \rho_1, \ldots, \rho_j)$, we have that (always) $\mathbf{V}_{\mathsf{IOP}}^{\rho_1, a_1, \ldots, \rho_{\mathsf{k}_{\mathsf{IOP}}}, a_{\mathsf{k}_{\mathsf{IOP}}}}(\mathbb{x}) = 1$. Therefore, $\mathbf{V}'^{\mathsf{tr}}_{\mathsf{IOP}}(\mathbb{x})$ accepts with probability 1.

Soundness. Fix $\mathbb{x} \notin L(R)$ and a malicious prover $\tilde{\mathbf{P}}_{\mathsf{IOP}}$. Let E be the event over the verifier's random coins, both interaction and decision, $(z_1, s_1, \ldots, z_{\mathsf{k}_{\mathsf{IOP}}}, s_{\mathsf{k}_{\mathsf{IOP}}}, \rho_{\mathsf{dc}})$, that there exists some j where the emulated the verifier $\mathbf{V}_{\mathsf{IOP}}$ reads ρ_j and at least one of the following is true: (i) $\Delta(z'_j, z_j) \geq \gamma$ or; (ii) $\Delta(s'_j, s_j) \geq \gamma$, where $z'_j := \tilde{\mathbf{P}}_{\mathsf{IOP}}(z_1, s_1, \ldots, z_{j-1}, s_{j-1}, z_j)$ and $(s'_j, a_j) := \tilde{\mathbf{P}}_{\mathsf{IOP}}(z_1, s_1, \ldots, z_{j-1}, s_{j-1}, z_j, s_j)$.

We first show that the probability that the verifier accepts and the event E happens is small:

Claim 7.3. *We have that:*

$$\Pr\left[\, \langle \tilde{\mathbf{P}}_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}}(\mathbb{x}) \rangle = 1 \wedge (z_1, s_1, \ldots, z_{\mathsf{k}_{\mathsf{IOP}}}, s_{\mathsf{k}_{\mathsf{IOP}}}, \rho_{\mathsf{dc}}) \in \mathsf{E} \,\right] \leq 1 - \gamma \ .$$

Proof. For every choice of verifier randomness $(z_1, s_1, \ldots, z_{\mathsf{k}_{\mathsf{IOP}}}, s_{\mathsf{k}_{\mathsf{IOP}}}, \rho_{\mathsf{dc}}) \in \mathsf{E}$, there exists some j where the emulated the verifier $\mathbf{V}_{\mathsf{IOP}}$ reads ρ_j in which either $\Delta(z_j, z'_j) \geq \gamma$ or $\Delta(s_j, s'_j) \geq \gamma$. As a result, one of the checks made by $\mathbf{V}'_{\mathsf{IOP}}$ in Item 3b, causes the verifier to reject with probability at least γ. We conclude the claim by noting that

$$\begin{aligned}
&\Pr\left[\ \langle \tilde{\mathbf{P}}_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}}(\mathbb{x})\rangle = 1 \wedge (z_1, s_1, \ldots, z_{\mathsf{k}_{\mathsf{IOP}}}, s_{\mathsf{k}_{\mathsf{IOP}}}, \rho_{\mathsf{dc}}) \in \mathsf{E}\ \right] \\
&\quad \leq \Pr\left[\ \langle \tilde{\mathbf{P}}_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}}(\mathbb{x})\rangle = 1 \mid (z_1, s_1, \ldots, z_{\mathsf{k}_{\mathsf{IOP}}}, s_{\mathsf{k}_{\mathsf{IOP}}}, \rho_{\mathsf{dc}}) \in \mathsf{E}\ \right] \leq 1 - \gamma\ .
\end{aligned}$$

□

We introduce a new malicious prover $\tilde{\mathbf{P}}^*_{\mathsf{IOP}}$ for the IOP system that "corrects" any z'_j and s'_j messages sent by the prover that are γ-far from what they are claimed to be. $\tilde{\mathbf{P}}^*_{\mathsf{IOP}}$ has the following next-message function:

- $\tilde{\mathbf{P}}^*_{\mathsf{IOP}}(z_1, s_1, \ldots, z_{j-1}, s_{j-1}, z_j)$: Compute $z''_j := \tilde{\mathbf{P}}_{\mathsf{IOP}}(z_1, s_1, \ldots, z_{j-1}, s_{j-1}, z_j)$. If $\Delta(z''_j, z_j) < \gamma$ then output z''_j and otherwise output z_j.
- $\tilde{\mathbf{P}}^*_{\mathsf{IOP}}(z_1, s_1, \ldots, z_{j-1}, s_{j-1}, z_j, s_j)$: Compute $(s''_j, a_j) := \tilde{\mathbf{P}}_{\mathsf{IOP}}(z_1, s_1, \ldots, z_{j-1}, s_{j-1}, z_j, s_j)$. If $\Delta(s''_j, s_j) < \gamma$ then output (s''_j, a_j) and otherwise output (s_j, a_j).

Notice that $\tilde{\mathbf{P}}^*_{\mathsf{IOP}}$ and $\tilde{\mathbf{P}}_{\mathsf{IOP}}$ send exactly the same messages in rounds where $\tilde{\mathbf{P}}_{\mathsf{IOP}}$ outputs z'_j and s'_j with $\Delta(z'_j, z_j) < \gamma$ and $\Delta(s'_j, s_j) < \gamma$. Therefore we can reinterpret the event E relative to $\tilde{\mathbf{P}}^*_{\mathsf{IOP}}$ as follows: $(z_1, s_1, \ldots, z_{\mathsf{k}_{\mathsf{IOP}}}, s_{\mathsf{k}_{\mathsf{IOP}}}, \rho_{\mathsf{dc}}) \notin \mathsf{E}$ if for every j where $\mathbf{V}_{\mathsf{IOP}}$ reads ρ_j and $\tilde{\mathbf{P}}^*_{\mathsf{IOP}}$ has not "corrected" z'_j or s'_j (by sending z_j or s_j respectively). Whenever the verifier does not read a corrected round, it acts identically to an interaction with $\tilde{\mathbf{P}}_{\mathsf{IP}}$ where it does not read a round that was far from the messages sent by the verifier. Therefore, using this interpretation we have:

$$\begin{aligned}
&\Pr\left[\ \langle \tilde{\mathbf{P}}_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}}(\mathbb{x})\rangle = 1 \wedge (z_1, s_1, \ldots, z_{\mathsf{k}_{\mathsf{IOP}}}, s_{\mathsf{k}_{\mathsf{IOP}}}, \rho_{\mathsf{dc}}) \notin \mathsf{E}\ \right] \\
&\quad = \Pr\left[\ \langle \tilde{\mathbf{P}}^*_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}}(\mathbb{x})\rangle = 1 \wedge (z_1, s_1, \ldots, z_{\mathsf{k}_{\mathsf{IOP}}}, s_{\mathsf{k}_{\mathsf{IOP}}}, \rho_{\mathsf{dc}}) \notin \mathsf{E}\ \right] \\
&\quad \leq \Pr\left[\ \langle \tilde{\mathbf{P}}^*_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}}(\mathbb{x})\rangle = 1\ \right]
\end{aligned}$$

We now show that the probability of $\mathbf{V}'_{\mathsf{IOP}}$ accepting when interacting with $\tilde{\mathbf{P}}^*_{\mathsf{IOP}}$ is small. We begin by showing that $\tilde{\mathbf{P}}^*_{\mathsf{IOP}}$-s messages z_j have high min-entropy.

Claim 7.4. *For every j: $H_{\mathsf{min}}(Z'_j \mid \neg\mathsf{E}) \geq 0.5n_z$, where Z'_j is the random variable describing the output z'_j of $\tilde{\mathbf{P}}^*_{\mathsf{IP}}$ in a random execution of $\langle \tilde{\mathbf{P}}^*_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}}(\mathbb{x})\rangle$.*

Proof. Fix a round number j and some string z^*_j. We have that

$$\begin{aligned}
\Pr\left[\ Z'_j = z^*_j\ \right] &\leq \Pr_{z_j}\left[\ \Delta(z^*_j, z_j) < \gamma\ \right] && (6)\\
&= |\{\ x' \in \{0,1\}^{n_z} : \Delta(x, x') \leq \gamma\ \}|/2^{n_z} \\
&\leq 2^{-n_z + n_z H(\gamma)} && (7)\\
&< 2^{-0.5n_z}\ . && (8)
\end{aligned}$$

Above, Equation (6) is due to the fact the output z'_j of $\tilde{\mathbf{P}}^*_{\mathsf{IP}}$ always has Hamming distance at less than γ from z_j. Equation (7) true due to Fact 3.8 and Equation (8) is true for a small enough constant γ. Then, we get that

$$H_{\mathsf{min}}(Z'_j) \triangleq \min_{z^*_j} -\log \Pr[\ Z'_j = z^*_j\ ] > 0.5n_z \ .$$

□

Claim 7.5. *Suppose that, after amplification in Item 1, the IOP* $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ *has* $(\beta_{\mathrm{rbr}}, \delta_{\mathsf{dc}})$*-round-by-round soundness with state function* state*. Then for every transcript* tr *generated by an interaction between* $\tilde{\mathbf{P}}^*_{\mathsf{IOP}}$ *and* $\mathbf{V}'_{\mathsf{IOP}}$ *where the verifier is about to make its* j*-th move such that* $\mathsf{state}(\mathbb{x}, \mathsf{tr}) = 0$*:*

$$\Pr[\ \mathsf{state}(\mathbb{x}, \mathsf{tr}||\rho_j) = 1\ ] \leq (\beta_{\mathrm{rbr}} + \varepsilon_{\mathsf{Ext}}) \cdot 2^{n_s \cdot H(\gamma)} \ ,$$

where ρ_j *is drawn as in the protocol description.*

Proof. Fix some j and a transcript as in the claim statement. By Claim 7.4, we have that z'_j has min-entropy at least $0.5n_z$. Thus, by definition of the extractor,

$$|\Pr[\ \mathsf{state}(\mathbb{x}, \mathsf{tr}||\mathsf{Ext}(z'_j, U_{n_s})) = 1\ ] - \Pr[\ \mathsf{state}(\mathbb{x}, \mathsf{tr}||U_{\mathsf{r}'_{\mathsf{IOP}}}) = 1\ ]| \leq \varepsilon_{\mathsf{Ext}} \ ,$$

where U_{n_s} and $U_{\mathsf{r}'_{\mathsf{IOP}}}$ are the uniform distributions over bit strings of length n_s and $\mathsf{r}'_{\mathsf{IOP}}$ respectively. Furthermore, by $(\beta_{\mathrm{rbr}}, \delta_{\mathsf{dc}})$-round-by-round soundness of $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$, we have that

$$\Pr[\ \mathsf{state}(\mathbb{x}, \mathsf{tr}||U_{\mathsf{r}'_{\mathsf{IOP}}}) = 1\ ] < \beta_{\mathrm{rbr}} \ .$$

Therefore, the fraction of seeds that cause the state function to change from 0 to 1 is at most $\varepsilon_{\mathsf{Ext}} + \beta_{\mathrm{rbr}}$. Recall that in the protocol, $\rho_j := \mathsf{Ext}(z'_j, s'_j)$, i.e., the seed of the extractor is s'_j rather than a uniformly random seed. Recall that $\tilde{\mathbf{P}}^*_{\mathsf{IOP}}$ only outputs messages s'_j with $\Delta(s'_j, s_j) < \gamma$. We say that a seed s_j is *bad* if there exists some s'_j with $\Delta(s'_j, s_j) < \gamma$ such that $\mathsf{state}(\mathbb{x}, \mathsf{tr}||\mathsf{Ext}(z'_j, s'_j)) = 1$. Every point s'_j that inhibits changing of the state function has a ball of size $2^{n_s \cdot H(\gamma)}$ of random seeds that have distance at most γ from it (see Fact 3.8). The total probability of landing on a bad seed is at most the probability that a random seed s_j falls within one of these balls. Therefore the probability that s_j bad is at most $(\varepsilon_{\mathsf{Ext}} + \beta_{\mathrm{rbr}}) \cdot 2^{n_s \cdot H(\gamma)}$. □

Let β_{rbr} be the round-by-round interaction error of the IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ following the augmentation in Item 1. Then $\log 1/\beta_{\mathrm{rbr}} = O(\log(|\mathbb{x}| + \mathsf{k}^2_{\mathsf{IOP}})) > O(\log(|\mathbb{x} + \mathsf{k}_{\mathsf{IOP}}|)) = \log \mathsf{r}'_{\mathsf{IOP}}$. Therefore, setting $n_z = 4\mathsf{r}'_{\mathsf{IOP}}$, by Theorem 3.7, there exists an extractor with error $\varepsilon_{\mathsf{Ext}} = \beta_{\mathrm{rbr}}$, on a source with min entropy $0.5n_z = 2\mathsf{r}'_{\mathsf{IOP}}$ which extracts $\mathsf{r}'_{\mathsf{IOP}}$ bits of randomness. The seed length is $n_s = O(\log 1/\varepsilon_{\mathsf{Ext}}) = O(\log(1/\beta_{\mathrm{rbr}}))$. We therefore have that:

$$\begin{aligned}
\Pr[\langle \tilde{\mathbf{P}}^*_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IP}}(\mathbb{x})\rangle = 1] &\leq \delta_{\mathsf{dc}} + \Pr[\ \exists j : \mathsf{state}(\mathbb{x}, \mathsf{tr}||\rho_j) = 1\ ] && (9)\\
&\leq \delta_{\mathsf{dc}} + \mathsf{k}_{\mathsf{IOP}} \cdot (\beta_{\mathrm{rbr}} + \varepsilon_{\mathsf{Ext}}) \cdot 2^{n_s \cdot H(\gamma)} && (10)\\
&\leq \delta_{\mathsf{dc}} + \mathsf{k}_{\mathsf{IOP}} \cdot 2\beta_{\mathrm{rbr}} \cdot 2^{O(\log(1/\beta_{\mathrm{rbr}})) \cdot H(\gamma)} && (11)\\
&\leq \delta_{\mathsf{dc}} + \mathsf{k}_{\mathsf{IOP}} \cdot \sqrt{\beta_{\mathrm{rbr}}} && (12)\\
&\leq \gamma/4 + \mathsf{k}_{\mathsf{IOP}}/\sqrt{|\mathbb{x}| + \mathsf{k}^2_{\mathsf{IOP}}} < \gamma/2 \ . && (13)
\end{aligned}$$

Equation (9) follows from the fact that the verifier $\mathbf{V}'_{\mathsf{IP}}$ accepts only if $\mathbf{V}_{\mathsf{IP}}$ accepts given $\mathbb{x}$, prover messages $a_1, \ldots, a_{\mathsf{k_{IOP}}}$ and verifier randomness $\rho_1, \ldots . \rho_{\mathsf{k_{IOP}}}$. By the round-by-round soundness of the original IP, since $\mathsf{state}(\mathbb{x}, \emptyset) = 0$ (which follows from the fact that $\mathbb{x} \notin L$), in order for the verifier to accept, it must be that the value of the state function changed from 0 to 1 in some round. Equation (10) is true by applying the union bound and Claim 7.5. We have Equation (11) by noting that we set $\varepsilon_{\mathsf{Ext}} = \beta_{\mathrm{rbr}}$ and $n_s = O(\log(1/\beta_{\mathrm{rbr}}))$. Equation (12) holds for a small enough constant $\gamma > 0$, and, finally, Equation (13) holds by the definitions of δ_{dc} and β_{rbr} and for large enough values of $|\mathbb{x}|$.

Thus we have that

$$\Pr\left[\langle \tilde{\mathbf{P}}_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}}(\mathbb{x}) \rangle = 1 \wedge (z_1, s_1, \ldots, z_{\mathsf{k_{IOP}}}, s_{\mathsf{k_{IOP}}}, \rho_{\mathsf{dc}}) \notin \mathsf{E} \right] \leq \Pr\left[\langle \tilde{\mathbf{P}}^*_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}}(\mathbb{x}) \rangle = 1 \right] \leq \gamma/2 \ .$$

Putting this fact together with Claim 7.3, we have that

$$\begin{aligned}
\Pr\left[\langle \tilde{\mathbf{P}}_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}}(\mathbb{x}) \rangle = 1 \right] &= \Pr\left[\langle \tilde{\mathbf{P}}_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}}(\mathbb{x}) \rangle = 1 \wedge (z_1, s_1, \ldots, z_{\mathsf{k_{IOP}}}, s_{\mathsf{k_{IOP}}}, \rho_{\mathsf{dc}}) \in \mathsf{E} \right] \\
&\quad + \Pr\left[\langle \tilde{\mathbf{P}}_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}}(\mathbb{x}) \rangle = 1 \wedge (z_1, s_1, \ldots, z_{\mathsf{k_{IOP}}}, s_{\mathsf{k_{IOP}}}, \rho_{\mathsf{dc}}) \notin \mathsf{E} \right] \\
&\leq 1 - \gamma + \gamma/2 = 1 - \gamma/2 \ .
\end{aligned}$$

Thus the verifier accepts with constant probability.

Complexity measures. We analyze the efficiency parameters of the resulting round-query IOP:

- *Rounds.* The round-query IOP has $2\mathsf{k_{IOP}}$ rounds.
- *Proof length.* We first amplify the protocol, giving polynomial overhead to all messages. In addition to the original prover messages, the prover also sends z'_j and s'_j. Therefore the proof length is $\mathrm{poly}(|\mathbb{x}|, \mathsf{l_{IOP}})$.
- *Round queries.* $\mathbf{V}'_{\mathsf{IOP}}$ queries $O(\mathsf{q_{rnd}})$ rounds.
- *Query complexity to randomness.* The verifier queries s_j and z_j in $O(1)$ locations.
- *Randomness complexity.* $\mathbf{V}'_{\mathsf{IP}}$ generates $n_z + n_s = \mathrm{poly}(\mathsf{r_{IP}}, |\mathbb{x}|)$ bits in every round.
- *Decision randomness.* $\mathbf{V}'_{\mathsf{IP}}$ chooses decision randomness $\mathsf{r}'_{\mathsf{dc}} = O(\mathsf{r_{dc}})$ and then uses $\log n_z + \log n_s = O(\log |\mathbb{x}| + \log \mathsf{k_{IOP}})$ bits of decision randomness. All-in-all $O(\mathsf{r_{dc}} + \log |\mathbb{x}|)$.
- *Verifier running time.* $\mathbf{V}'_{\mathsf{IP}}$ runs the original IOP verifier for polynomially many repetitions, generates a few random strings and runs the extractor. Its running time is therefore polynomially related to the running time of $\mathbf{V}_{\mathsf{IP}}$.
- *Adaptivity.* $\mathbf{V}'_{\mathsf{IP}}$ makes non-adaptive queries to its interaction randomness. Therefore, if $\mathbf{V}_{\mathsf{IOP}}$ was non-adaptive, then so is $\mathbf{V}'_{\mathsf{IOP}}$.

7.2 Local access to prover messages

In this transformation, we take a round-query IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ with round-query complexity $\mathsf{q_{rnd}}$ where the verifier reads $O(1)$ bits from the interaction randomness of the rounds that it queries

but reads the entire prover messages, and transform it into a binary IOP $(\mathbf{P}'_{\mathsf{IOP}}, \mathbf{V}'_{\mathsf{IOP}})$ with query complexity $O(\mathsf{q}_{\mathsf{rnd}})$. We sketch this transformation, assuming familiarity with [ACY22].

The transformation involves simulating the round-query IOP, except that $\mathbf{P}'_{\mathsf{IOP}}$ encodes the messages of $\mathbf{P}_{\mathsf{IOP}}$ using an "index-decodable" PCP (ID-PCP). In the final round, $\mathbf{P}'_{\mathsf{IOP}}$ sends, for every choice of decision randomness of $\mathbf{V}_{\mathsf{IOP}}$, a proof convincing that $\mathbf{V}_{\mathsf{IOP}}$ would have accepted, having chosen this decision randomness. The new IOP verifier $\mathbf{V}'_{\mathsf{IOP}}$ then chooses decision randomness and runs the ID-PCP verifier to verify the proof. Executing the ID-PCP verifier involves reading the relevant bits of the verifier interaction randomness, $O(1)$ bits of the final prover message, and $O(1)$ bits from the encoding of each prover message that $\mathbf{V}_{\mathsf{IOP}}$ would have queried using this decision randomness. Hence, $\mathbf{V}'_{\mathsf{IOP}}$ has (total) query complexity $O(\mathsf{q}_{\mathsf{rnd}})$.

Acknowledgments

Gal Arnon is supported in part by a grant from the Israel Science Foundation (no. 2686/20) and by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness. Alessandro Chiesa is funded by the Ethereum Foundation. Eylon Yogev is supported by the BIU Center for Research in Applied Cryptography and Cyber Security in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office, and by the Alter Family Foundation.

References

[AB17] Amir Abboud and Arturs Backurs. "Towards Hardness of Approximation for Polynomial Time Problems". In: *Proceedings of the 8th Innovations in Theoretical Computer Science Conference*. ITCS '17. 2017, 11:1–11:26.

[ACY22] Gal Arnon, Alessandro Chiesa, and Eylon Yogev. *A PCP Theorem for Interactive Proofs*. Cryptology ePrint Archive, Report 2021/915. To appear at EUROCRYPT 2022. 2022.

[ALMSS98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. "Proof verification and the hardness of approximation problems". In: *Journal of the ACM* 45.3 (1998). Preliminary version in FOCS '92., pp. 501–555.

[AR18] Amir Abboud and Aviad Rubinstein. "Fast and Deterministic Constant Factor Approximation Algorithms for LCS Imply New Circuit Lower Bounds". In: *Proceedings of the 9th Innovations in Theoretical Computer Science Conference*. ITCS '18. 2018, 35:1–35:14.

[ARW17] Amir Abboud, Aviad Rubinstein, and R. Ryan Williams. "Distributed PCP Theorems for Hardness of Approximation in P". In: *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '17. 2017, pp. 25–36.

[AS98] Sanjeev Arora and Shmuel Safra. "Probabilistic checking of proofs: a new characterization of NP". In: *Journal of the ACM* 45.1 (1998). Preliminary version in FOCS '92., pp. 70–122.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. "Interactive Oracle Proofs". In: *Proceedings of the 14th Theory of Cryptography Conference*. TCC '16-B. 2016, pp. 31–60.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. "Checking computations in polylogarithmic time". In: *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*. STOC '91. 1991, pp. 21–32.

[BM88] László Babai and Shlomo Moran. "Arthur-Merlin Games: A Randomized Proof System, and a Hierarchy of Complexity Classes". In: *Journal of Computer and System Sciences* 36.2 (1988), pp. 254–276.

[Bab85] László Babai. "Trading group theory for randomness". In: *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*. STOC '85. 1985, pp. 421–429.

[CCHLRR18] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, and Ron D. Rothblum. *Fiat–Shamir From Simpler Assumptions*. Cryptology ePrint Archive, Report 2018/1004. 2018.

[CFLS97] Anne Condon, Joan Feigenbaum, Carsten Lund, and Peter W. Shor. "Random Debaters and the Hardness of Approximating Stochastic Functions". In: *SIAM Journal on Computing* 26.2 (1997), pp. 369–400.

[CGLRR19] Lijie Chen, Shafi Goldwasser, Kaifeng Lyu, Guy N. Rothblum, and Aviad Rubinstein. "Fine-grained Complexity Meets IP = PSPACE". In: *Proceedings of the 30th Annual Symposium on Discrete Algorithms*. SODA '19. 2019, pp. 1–20.

[CW19] Lijie Chen and Ryan Williams. "An Equivalence Class for Orthogonal Vectors". In: *Proceedings of the 30th Annual Symposium on Discrete Algorithms*. SODA '19. 2019, pp. 21–40.

[Din07] Irit Dinur. "The PCP theorem by gap amplification". In: *Journal of the ACM* 54.3 (2007), p. 12.

[Dru11] Andrew Drucker. "A PCP Characterization of AM". In: *Proceedings of the 38th International Colloquium on Automata, Languages and Programming*. ICALP '11. 2011, pp. 581–592.

[FGLSS91] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. "Approximating clique is almost NP-complete (preliminary version)". In: *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*. SFCS '91. 1991, pp. 2–12.

[FGLSS96] Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. "Interactive proofs and the hardness of approximating cliques". In: *Journal of the ACM* 43.2 (1996), pp. 268–292.

[GKR15] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. "Delegating Computation: Interactive Proofs for Muggles". In: *Journal of the ACM* 62.4 (2015), 27:1–27:64.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The knowledge complexity of interactive proof systems". In: *SIAM Journal on Computing* 18.1 (1989). Preliminary version appeared in STOC '85., pp. 186–208.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. "Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes". In: *Journal of the ACM* 56.4 (2009), 20:1–20:34.

[GVW02] Oded Goldreich, Salil Vadhan, and Avi Wigderson. "On interactive proofs with a laconic prover". In: *Computational Complexity* 11.1/2 (2002), pp. 1–53.

[HRT07] Ishay Haviv, Oded Regev, and Amnon Ta-Shma. "On the Hardness of Satisfiability with Bounded Occurrences in the Polynomial-Time Hierarchy". In: *Theory of Computing* 3.1 (2007), pp. 45–60.

[KL94] Ker-I Ko and Chih-Long Lin. "Non-approximability in the polynomial-time hierarchy". In: *Technical Report 94-2, Dept. of Computer Science, SUNY at Stony Brook* (1994).

[KR08] Yael Kalai and Ran Raz. "Interactive PCP". In: *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*. ICALP '08. 2008, pp. 536–547.

[LFKN92] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. "Algebraic Methods for Interactive Proof Systems". In: *Journal of the ACM* 39.4 (1992), pp. 859–868.

[Pap83] Christos H. Papadimitriou. "Games Against Nature (Extended Abstract)". In: *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*. STOC '83. 1983, pp. 446–450.

[RRR16] Omer Reingold, Ron Rothblum, and Guy Rothblum. "Constant-Round Interactive Proofs for Delegating Computation". In: *Proceedings of the 48th ACM Symposium on the Theory of Computing*. STOC '16. 2016, pp. 49–62.

[Sha92] Adi Shamir. "IP = PSPACE". In: *Journal of the ACM* 39.4 (1992), pp. 869–877.