Hardness of Approximation for Stochastic Problems via Interactive Oracle Proofs

Gal Arnon
Weizmann Institute of Science, Rehovot, Israel

Alessandro Chiesa
EPFL, Lausanne, Switzerland

Eylon Yogev
Bar-Ilan University, Ramat-Gan, Israel

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 $\mathrm{AM}[\mathsf{k}]$-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 $\mathrm{IR}[\mathsf{k}]$ (for *interactive reducibility*), and show that several protocols (e.g., the sumcheck protocol) are in $\mathrm{IR}[\mathsf{k}]$. 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 $\mathrm{IR}[\mathsf{k}]$-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 $\mathrm{AM}[\mathsf{k} \leq O(\log n)]$ or in $\mathrm{IR}[\mathsf{k}]$ 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.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography; Theory of computation → Interactive proof systems

Keywords and phrases hardness of approximation, interactive oracle proofs, stochastic satisfaction problems

Digital Object Identifier 10.4230/LIPIcs.CCC.2022.24

Related Version *Full Version*: `https://eprint.iacr.org/2022/168`

Funding *Gal Arnon*: 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: Funded by the Ethereum Foundation.
Eylon Yogev: 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.

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 [13]. For example, the PCP theorem [6, 5] 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 [1, 3, 2, 9, 8]). 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 [17] 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 [16] 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* [19]. 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** (informal). *An* **SCSP** Φ *with* k *alternations is a list of constraints* $C_1, \dots, C_\mathsf{m}$ *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 [12] 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, [4] 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 [10] 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 [14] 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 [18, 21]. 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) [7, 20]. 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).

1.1.1 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 2** (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 2 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 [12, 4].

Many results on the hardness of approximating the value of (standard) CSPs are achieved by constructing suitable PCPs. Similarly (as was noted in [4]), 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 2 using this framework by providing a transformation that maps a k-round IP into a k-round IOP with small query complexity.

▶ **Lemma 3** (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 3 is surprising in light of the work of Goldreich, Vadhan, and Wigderson [14], 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 3 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.

1.1.2 Hardness of SCSPs from interactive reducibility

Theorem 2 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 [18] and for Shamir's protocol [21].

The notion of interactive reducibility allows us to get an optimal version of Lemma 3 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 4** (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 4 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.

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

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 $\mathrm{IR}[\mathsf{k}]$-hard:

▶ **Theorem 5** (informal). *Let $L \in \mathrm{IR}[\mathsf{k}]$ 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 $\#\mathsf{SAT}_\mathsf{k} \in \mathrm{IR}[\mathsf{k}]$. Thus, by Theorem 5, we establish that approximating the value of SCSPs with $O(\mathsf{k})$ alternations and constant arity is $\#\mathsf{SAT}_\mathsf{k}$-hard. Similarly, using our interactive reduction for Shamir's protocol, we recover the result of [10], showing that approximating the value of SCSPs with polynomially-many alternations and constant arity is PSPACE-hard.

1.1.3 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**
[10]	PSPACE	unbounded	$O(1)$
[this work]	$\mathrm{IR}[\mathsf{k}]$	$O(\mathsf{k})$	$O(1)$
[this work]	$\#\mathsf{SAT}_\mathsf{k}$	$O(\mathsf{k})$	$O(1)$
[4]	$\mathrm{AM}[\mathsf{k}]$	k	$O(\mathsf{k})$
[this work]	$\mathrm{AM}[\mathsf{k}]$	k	$\max\{O(1), O(\mathsf{k}/\log|\mathbb{x}|)\}$
[12]	$\mathrm{AM}[1]$	1	$O(1)$
[5, 6, 11]	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.

2.1.1 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.

2.1.2 SCSP hardness from IOPs

The general connection between SCSPs (Definition 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 6.** *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*,* $2^{\mathsf{r_{dc}}}$ *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. ◀

2.2 Transforming IPs into IOPs

We outline the proof of Lemma 3 (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.

2.2.1.1 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.

2.2.1.2 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 $\mathbf{V}_{\mathsf{IP}}(\mathbb{x}, \mathsf{tr}) = 1$.

2.2.1.3 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.

2.2.1.4 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

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

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:

$$\big(S_0, \rho_1, S_1, \ldots, \rho_{2\mathsf{k}_{\mathsf{IP}}}, S_{2\mathsf{k}}, (T_0, \ldots, T_{2\mathsf{k}})\big) .$$

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 $\max\{\frac{1}{2}, \binom{2\mathsf{k}}{\mathsf{k}} \cdot \beta\}$.

2.2.1.5 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 [4].

In more detail, [4] 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 the full version of this paper 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 [4], 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 3.

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 [18]. Below we review this protocol, and then explain how to reduce multiple transcripts into one transcript via an interactive reduction.

2.3.1.1 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

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

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

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

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 $\sum_{\alpha_{j+2},\ldots,\alpha_n \in \{0,1\}} p(r_1,\ldots,r_{j+1},\alpha_{j+2},\ldots,\alpha_n) = \gamma_{j+1}$.

2.3.1.2 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 $\mathbb{z}_1, \dots, \mathbb{z}_\mathsf{t}$. Thus we will focus on merging such claims.

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

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

- 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:*
     - **(i)** Sample uniformly random field elements $\rho, r^* \leftarrow \mathbb{F}$ and send them to the prover.
     - **(ii)** Set $(r'_1, \dots, r'_j) := I_{\mathbb{z}_1,\dots,\mathbb{z}_\mathsf{t}}(\rho)$ and $\gamma_{j+1} := \tilde{g}(\rho, r^*)$ and output the new instance

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

Analysis. It follows straightforwardly from the protocol that, if $\mathbb{z}_1, \dots, \mathbb{z}_\mathsf{t}$ 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 $\mathbb{z}_1, \dots, \mathbb{z}_\mathsf{t}$ is false then with high probability so is $\mathbb{z}'$.

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

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

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

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

and so the resulting statement $\mathbb{z}' := ((r'_1, \dots, r'_j, r^*), \gamma_{j+1})$ 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 7.** *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 $f_0, f_1, \dots, f_{\mathsf{k}}$ such that the following holds.*

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

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

- **Soundness:** *For every $\mathbb{x} \notin L(R)$, $j \in [\mathsf{k}]$, and $\mathbb{z}_1, \dots, \mathbb{z}_{\mathsf{t}}$, 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:*

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

- **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 $\mathbb{z}_1, \dots, \mathbb{z}_{\mathsf{t}}$* 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 $\mathbb{z}_1, \dots, \mathbb{z}_{\mathsf{t}}$, the new instance $\mathbb{z}'$ output by $(\mathbf{P}_{\mathsf{IR}}, \mathbf{V}_{\mathsf{IR}})$ on inputs $(\mathbb{x}, \mathbb{w}, \mathbb{z}_1, \dots, \mathbb{z}_{\mathsf{t}})$ 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 $A_{i-1}[j]$.

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

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

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

2. For $i = 1, \dots, 2\mathsf{k}$:
   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:

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

$\mathbf{V}_{\mathsf{IOP}}$ 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 $\mathbb{z} \in B_{2\mathsf{k}}[\mathsf{k}]$.

2.3.3.2 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 $\mathsf{k} = \mathrm{poly}(|\mathbb{x}|)$, 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 $\mathrm{poly}(|\mathbb{x}|)$.

2.3.4 Relations with interactive reducibility

Several relations of interest have interactive reductions.

2.3.4.1 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 the full version of this paper 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.

2.3.4.2 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 $O(1)$.

2.3.4.3 Shamir's protocol

Shamir's protocol [21] 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 $O(1)$.

2.3.4.4 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 [15].

References

1 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, pages 11:1–11:26, 2017.

2 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, pages 35:1–35:14, 2018.

3 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, pages 25–36, 2017.

4 Gal Arnon, Alessandro Chiesa, and Eylon Yogev. A PCP theorem for interactive proofs. Cryptology ePrint Archive, Report 2021/915, 2022. To appear at EUROCRYPT 2022.

5 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. Preliminary version in FOCS '92.

6 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. Preliminary version in FOCS '92.

7 Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In *Proceedings of the 14th Theory of Cryptography Conference*, TCC '16-B, pages 31–60, 2016.

8 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, pages 1–20, 2019.

9 Lijie Chen and Ryan Williams. An equivalence class for orthogonal vectors. In *Proceedings of the 30th Annual Symposium on Discrete Algorithms*, SODA '19, pages 21–40, 2019.

10 Anne Condon, Joan Feigenbaum, Carsten Lund, and Peter W. Shor. Random debaters and the hardness of approximating stochastic functions. *SIAM Journal on Computing*, 26(2):369–400, 1997.

11 Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3):12, 2007.

12 Andrew Drucker. A PCP characterization of AM. In *Proceedings of the 38th International Colloquium on Automata, Languages and Programming*, ICALP '11, pages 581–592, 2011.

13 Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996.

14 Oded Goldreich, Salil Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. *Computational Complexity*, 11(1/2):1–53, 2002.

15 Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *Journal of the ACM*, 62(4):27:1–27:64, 2015.

16 Ishay Haviv, Oded Regev, and Amnon Ta-Shma. On the hardness of satisfiability with bounded occurrences in the polynomial-time hierarchy. *Theory of Computing*, 3(1):45–60, 2007.

17 Ker-I Ko and Chih-Long Lin. Non-approximability in the polynomial-time hierarchy. *Technical Report 94-2, Dept. of Computer Science, SUNY at Stony Brook*, 1994.

18 Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

19 Christos H. Papadimitriou. Games against nature (extended abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, STOC '83, pages 446–450, 1983.

20 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, pages 49–62, 2016.

21 Adi Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, 1992.