Non-Cooperative Rational Interactive Proofs

Jing Chen
Stony Brook University, Stony Brook, NY 11794-4400, USA
jingchen@cs.stonybrook.edu

Samuel McCauley
Williams College, Williamstown, MA 01267, USA
sam@cs.williams.edu

Shikha Singh
Williams College, Williamstown, MA 01267, USA
shikha@cs.williams.edu

Abstract

Interactive-proof games model the scenario where an honest party interacts with powerful but strategic provers, to elicit from them the correct answer to a computational question. Interactive proofs are increasingly used as a framework to design protocols for computation outsourcing.

Existing interactive-proof games largely fall into two categories: either as games of cooperation such as multi-prover interactive proofs and cooperative rational proofs, where the provers work together as a team; or as games of conflict such as refereed games, where the provers directly compete with each other in a zero-sum game. Neither of these extremes truly capture the strategic nature of service providers in outsourcing applications. How to design and analyze non-cooperative interactive proofs is an important open problem.

In this paper, we introduce a mechanism-design approach to define a multi-prover interactive-proof model in which the provers are *rational* and *non-cooperative* – they act to maximize their expected utility given others' strategies. We define a strong notion of backwards induction as our solution concept to analyze the resulting extensive-form game with imperfect information.

We fully characterize the complexity of our proof system under different *utility gap* guarantees. (At a high level, a utility gap of u means that the protocol is robust against provers that may not care about a utility loss of $1/u$.) We show, for example, that the power of non-cooperative rational interactive proofs with a polynomial utility gap is exactly equal to the complexity class $\mathsf{P}^{\mathsf{NEXP}}$.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory and mechanism design; Theory of computation → Interactive proof systems; Theory of computation → Computational complexity and cryptography

Keywords and phrases non-cooperative game theory, extensive-form games with imperfect information, refined sequential equilibrium, rational proofs, interactive proofs

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.29

Related Version A full version of the paper is available at `https://arxiv.org/abs/1708.00521`.

Funding This work has been partially supported by NSF CAREER Award CCF 1553385, CNS 1408695, CCF 1439084, IIS 1247726, IIS 1251137, CCF 1217708, by Sandia National Laboratories, by the European Research Council under the European Union's 7th Framework Programme (FP7/2007-2013) / ERC grant agreement no. 614331, and a Zuckerman STEM Fellowship.

1 Introduction

Game theory has played a central role in analyzing the conflict and cooperation in interactive proof games. These games model the scenario where an honest party interacts with powerful but strategic agents, to elicit from them the correct answer to a computational question. The extensive study of these games over decades has fueled our understanding of important

complexity classes (e.g., [4, 16, 22–24, 26, 27, 37]). From a modern perspective, these games capture the essence of computation outsourcing – the honest party is a client outsourcing his computation to powerful rational service providers in exchange for money.

In this paper, we consider a natural type of interactive-proof game. For the moment, let us call our client Arthur. Arthur hires a service provider Merlin to solve a computational problem for him, and hires a second service provider Megan to cross-check Merlin's answer. Arthur wants the game (and associated payments) to be designed such that if Merlin gives the correct answer, Megan agrees with him; however, if Merlin cheats and gives a wrong answer, Megan is incentivized to contradict him, informing Arthur of Merlin's dishonesty. This means that Merlin and Megan are not purely cooperative nor purely competitive. Each is simply a rational agent who wants to maximize their own utility.

This is a mechanism design problem – how can Arthur incentivize non-cooperative rational agents (Merlin and Megan) to give truthful answers to his questions, helping him solve a computational problem? This problem is the focus of our paper.

Structure of the game

We borrow the structure and terminology of interactive proofs [3,6,29], as was done in previous work on rational proofs [1, 2, 11, 12, 17–19, 31, 32] and refereed games [16, 22, 24–26, 35, 40]. We call Arthur the **verifier** and assume that he is computationally bounded (he may be probabilistic, but must run in polynomial time). Arthur's coin flips are treated as Nature moves in the game. We call Merlin and Megan the **provers**; they have unbounded computational power.

The verifier exchanges messages with the provers in order to determine the answer to a decision problem. The exchange proceeds in rounds: in a round, either a verifier sends a message to all provers or receives a response from each. The provers cannot observe the messages exchanged between the verifier and other provers.

At the end, the verifier gives a payment to *each* prover. Our goal is to design protocols and payments such that, under an appropriate solution concept of the resulting game, the provers' best strategies lead the verifier to the correct answer.

The interactive protocols described above form an extensive-form game of imperfect information. To analyze them, we essentially use a strong notion of backward induction as our solution concept. We refine it further by eliminating strategies that are weakly dominated on "subgames" within the entire game. We define the solution concept formally in Section 2.1.

Comparison to previous work

The model of our games is based on interactive proof systems [3, 29], in which a verifier exchanges messages with untrustworty provers and at the end either accepts or rejects their claim. Interactive proofs guarantee that, roughly speaking, the verifier accepts a truthful claim with probability at least 2/3 (**completeness**) and no strategy of the provers can make the verifier accept a false claim with probability more than 1/3 (**soundness**).

The study of interactive proofs has found extensive applications in both theory and practice. Classical results on IPs have led us to better understand complexity classes through characterizations such as $\mathsf{IP} = \mathsf{PSPACE}$ [37, 43] and $\mathsf{MIP} = \mathsf{NEXP}$ [4, 23, 27], and later led to the important area of probabilistically checkable proofs [44]. More recently, the study of IPs has resulted in extremely efficient (e.g., near linear or even logarithmic time) protocols for delegation of computation [7, 9, 15, 30, 41]. Such super-efficient IPs have brought theory closer to practice, resulting in "nearly practical" systems (e.g., see [8, 13, 45, 47]).

Indeed, interactive proofs are not only a fundamental theoretical concept but an indispensable framework to design efficient computation-outsourcing protocols.

Existing interactive-proof games

Interactive-proof systems with multiple provers have largely been studied as games that fall into two categories: either as games of cooperation such as MIP [6], cooperative multi-prover rational proofs (MRIP) [18], and variants [4, 10, 27, 30, 33], where the provers work together to convince the verifier of their joint claim; or as games of conflict such as refereed games [14–16, 22, 24, 26, 34], where the provers directly compete with each other to convince the verifier of their conflicting claims.

Both of these categories have limitations. In a game of cooperation, provers cannot be leveraged directly against each other. That is, the verifier cannot directly ask one prover if another prover is lying. On the other hand, in a game of conflict, such as refereed games, one prover must "win" the zero-sum game. Thus, such games need to assume that at least one prover – who must be the winning prover in a correct protocol – can be trusted to always tell the truth. Despite their limitations, both models have proved to be fundamental constructs to understand and characterize important complexity classes [4, 16, 18, 22, 26], and to design efficient computation outsourcing protocols [7, 8, 14, 15, 30].

1.1 Contributions and Results

In this paper, we introduce a new interactive-proof game, *non-cooperative rational interactive proofs (ncRIP)*. This model generalizes multi-prover rational proofs [17–19].

Solution concept for ncRIP

We define a refinement of sequential equilibrium [36], **strong sequential equilibrium** (SSE), that essentially says that players' beliefs about the histories that led them to an unreachable information set should be irrelevant to their best response. From a mechanism-design perspective, we want to design the protocols and payments that allow this strong guarantee to hold – letting the players' best responses be unaffected by their beliefs.[1]

Finally, we eliminate SSE strategies that are suboptimal within "subgames" by defining and enforcing a backward-induction-compatible notion of dominance. Roughly speaking, we say a protocol is a ncRIP if there exists a strategy profile of the provers that is a dominant SSE among the *subforms* of the extensive form game, and under this strategy the provers' lead the verifier to the correct answer. We define the model formally in Section 2.

Utility gap for non-cooperative provers

Utility gap is a fundamental concept for rational proofs [2, 18, 19, 31] which is analogous to *soundness gap* in interactive proofs. It measures how robust a protocol is against the provers' possible deviations from the desired strategy.

This notion is straightforward to define for cooperative rational protocols – they have a utility gap of u if the *total* expected payment decreases by $1/u$ whenever the provers report the wrong answer. In non-cooperative protocols, however, it is not a priori clear how to define such a payment loss or to choose which prover should incur the loss. A payment loss

[1] We believe that SSE is of independent interest as a solution concept for designing extensive-form mechanisms (e.g. [21, 28, 46]). In the full version of the paper, we prove important properties of SSE that may prove useful in future studies.

solely imposed on the total payment may not prevent some provers from deviating, and a loss solely imposed on the provers' final payments may not prevent them from deviating within subgames.

We define a meaningful notion of utility gap for ncRIP that is naturally incorporated in a backward-induction-compatible way to the dominant SSE concept.

Tight characterizations of ncRIP classes

In this paper, we completely characterize the power of non-cooperative rational proofs under different utility-gap guarantees.

We construct ncRIP protocols with constant, polynomial, and exponential utility gaps for powerful complexity classes, demonstrating the strength of our solution concept. Our protocols are simple and intuitive (requiring only a few careful tweaks from their cooperative counterparts), and are thus easy to explain and implement. However, proving their correctness involves analyzing the extensive-game (including subtleties in the incentives and beliefs of each player at each round) to show that the protocol meets the strong solution-concept and utility-gap requirements.

We then prove tight upper bounds for all three ncRIP classes. Proving tight upper bounds is the most technically challenging part of the paper. We prove the upper bounds by simulating the decisions of the verifier and provers with a Turing Machine. However, there are several obstacles to attain the correct bounds. For example, the polynomial randomness of the verifier can induce an exponential-sized game tree, which is too large to be verified by the polynomial-time machine in Theorems 1 and 2. Furthermore, an NEXP oracle cannot itself verify whether a strategy profile is a dominant SSE. The key lemma that helps us overcome these challenges is the pruning lemma (Lemma 13). At a high level, it shows that we can prune the nature moves of the verifier in the resulting game tree, while preserving the dominant-SSE and utility-gap guarantees.

Our results are summarized in Figure 1, where we use $\mathsf{O}(1)$-ncRIP, $\mathsf{poly}(n)$-ncRIP and $\mathsf{exp}(n)$-ncRIP to denote ncRIP classes with constant, polynomial and exponential utility gaps respectively. The notations are analogous for MRIP [17] (the cooperative variant). We characterize ncRIP classes via oracle Turing machines. In particular, $\mathsf{P}^{\mathsf{NEXP}[\mathsf{O}(1)]}$ is the class of languages decided by a polynomial-time Turing machine that makes $O(1)$ queries to an NEXP oracle, and $\mathsf{EXP}^{\mathsf{poly\text{-}NEXP}}$ is the class decided by an exponential-time Turing machine with polynomial-length queries to an NEXP oracle.

Note that lower and upper bounds for the case of exponential utility gap (that is, Theorem 3 and Corollary 6) are deferred to the full version of the paper.

▶ **Theorem 1.** $\mathsf{O}(1)$-ncRIP $= \mathsf{P}^{\mathsf{NEXP}[\mathsf{O}(1)]}$	▶ **Corollary 4.** $\mathsf{O}(1)$-ncRIP $= \mathsf{O}(1)$-MRIP
▶ **Theorem 2.** $\mathsf{poly}(n)$-ncRIP $= \mathsf{P}^{\mathsf{NEXP}}$	▶ **Corollary 5.** $\mathsf{poly}(n)$-ncRIP $\supseteq \mathsf{poly}(n)$-MRIP
▶ **Theorem 3.** $\mathsf{exp}(n)$-ncRIP $= \mathsf{EXP}^{\mathsf{poly\text{-}NEXP}}$	▶ **Corollary 6.** $\mathsf{exp}(n)$-ncRIP $= \mathsf{exp}(n)$-MRIP

Figure 1 Summary of our results.

Power of non-cooperative vs. cooperative and competitive provers

Interestingly, in the case of constant and exponential utility gap, the power of ncRIP and MRIP coincide. This can be explained by the power of adaptive versus non-adaptive queries in oracle Turing machines.

Indeed, our results reveal the main difference between non-cooperative and cooperative provers: the former can be used to handle adaptive oracle queries, the latter cannot (see [17, 18]). Intuitively, this makes sense – cooperative provers may collude across adaptive queries, answering some of them incorrectly to gain on future queries. On the other hand, non-cooperativeness allows us to treat the subgame involving the oracle queries as a separate game from the rest.

Our results also show that non-cooperative provers are more powerful than competing provers. Feige and Kilian [22] proved that the power of refereed games with imperfect information and perfect recall is equal to EXP.

2 Non-Cooperative Rational Interactive Proofs

In this section we introduce the model for ncRIP.

Notation

First, we review the structure of ncRIP protocols and related notation; this is largely the same as [18].

The decision problem being solved by an interactive proof is modeled as whether a given string x is in language L. An interactive protocol is a pair $(V, \vec{P})$, where V is the **verifier**, $\vec{P} = (P_1, \ldots, P_{p(n)})$ is the vector of $p(n)$ **provers**, where $p(n)$ is polynomial in $n = |x|$. The verifier runs in polynomial time and flips private coins. Each P_i is computationally unbounded. The verifier and provers are given the input x. Similar to classical multi-prover interactive proofs, the verifier can communicate with each prover privately, but no two provers can communicate with each other once the protocol begins.

In a **round**, either each prover sends a message to V, or V sends a message to each prover, and these two cases alternate. The length of each message $\ell(n)$, and the number of rounds $k(n)$ are both polynomial in n. The final transcript $\vec{m}$ of the protocol is a random variable depending on r, the random string used by V. At the end of the communication, the verifier computes an **answer bit** $c \in \{0, 1\}$ for the membership of x in L based on x, r, and $\vec{m}$. V also computes a payment vector $\vec{R} = (R_1, R_2, \ldots, R_{p(n)})$, where R_i is the payment given to P_i, $R_i \in [-1, 1]$, and the total $\sum_{i=1}^{p(n)} R_i \in [-1, 1]$ as well.[2] The protocol and the payment function $\vec{R}$ are public knowledge.

Each prover P_i's **strategy** at round j maps the transcript seen at the beginning of round j to the message he sends in that round. Let $s_i = (s_{i1}, \ldots, s_{ik(n)})$ be the strategy of prover P_i, and $s = (s_1, \ldots, s_{p(n)})$ be the **strategy profile** of the provers. Given input x, and strategy profile s, let $u_k(x, s, (V, \vec{P}))$ denote the expected payment of prover P_k in the protocol $(V, \vec{P})$ based on randomness r, input x and s; if $(V, \vec{P})$ is clear from context, we shorten this to $u_k(x, s)$ or $u_k(s)$.

The protocol forms an extensive-form game with imperfect information and should be designed such that the provers are incentivized to reach an equilibrium that leads V to the correct answer bit c. We formalize the solution concept next.

[2] Negative payments are used to reflect punishment. The individual payments and the total payment can be shifted and scaled to lie in $[0, 1]$.

2.1 Solution concept for ncRIP

We want the solution concept for ncRIP to satisfy a strong notion of backward induction [39], a standard criterion applied to extensive-form games based on the common knowledge of rationality. Backwards induction refers to the condition of being "sequentially rational" in an extensive-form game, that is, each player must play his best response at each node where he has to move, even if his rationality implies that such a node will not be reached.

If an interactive protocol forms an extensive-form game of perfect information, it is easy to formalize this condition. A strategy s is **sequentially rational** or satisfies **backward induction**, if for every player i and every decision node of i, conditioned on reaching the decision node, s_i is a best response to s_{-i}, that is, $u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i})$ for any strategy s'_i of prover i. In other words, s induces a best response at every subgame.[3]

In a game of imperfect information, the decision nodes corresponding to a player's turn are partitioned into **information sets**, where the player is unable to distinguish between the possible histories within an information set. To reason about sequential rationality we need a probability distribution u_I on each information set I, so as to determine the players' expected utility conditioned on reaching I and thus their best response at I. The probability distribution μ_I is referred to as the player's **beliefs** about the potential histories leading to I.

Given a strategy profile s, beliefs u_I at **reachable information sets** (reached with non-zero probability under s) are derived from s using Bayes' rule; this is a standard derivation used in most solution concepts for extensive-form games [39]. We sometimes write μ_I^s to emphasize that the beliefs depend on s.

Past work has introduced a variety of methods for defining the beliefs u_I^s at **unreachable information sets** I (i.e. information sets reached with probability zero under s); see e.g. [20, 36, 38, 42]. The most well-known is sequential equilibrium [36], which demands an explicit system of beliefs that satisfies a (somewhat artificial) consistency condition. Other equilibria, like trembling hand [42], reason implicitly about beliefs at unreachable information sets by assigning a negligible probability with which the player's hand "trembles," and reaches an otherwise-unreachable information set. Further refinements of these take the structure and payoffs of the game into account [5, 20, 38].

The treatment of beliefs at unreachable information sets in these solution concepts is often focused on ensuring that they can be used to analyze *every* extensive-form game. From a mechanism-design perspective, our focus is different – we want to design mechanisms in such a way that they admit much stronger equilibrium requirements, even if such an equilibrium cannot be used to analyze every game.

At a high-level, we want the players' beliefs to be irrelevant in determining their best response at unreachable information sets. We call this notion **strong sequential rationality**. A strategy profile s is **strongly sequentially rational** if for every information set I, conditioned on reaching I, s_i is a best response to s_{-i} with respect to μ_I^s, where

- μ_I^s is derived using Bayes's if I is reachable under s, and
- μ_I^s is *any* arbitrary probability distribution if I is unreachable under s.

In the full version of the paper, we show that this requirement is equivalent to saying that, at an unreachable information set I, s_i must be a best response to s_{-i} conditioned on reaching each history $h \in I$. In other words, at an unreachable information set I, each player must have a *single* action that is the best response to every possible history in I. We say a strategy profile is a **strong sequential equilibrium** (SSE) if it satisfies strong sequential rationality.

[3] A subgame is a subtree that can be treated as a separate well-defined game. In a perfect-information game, every node starts a new subgame. "Backward induction" and "subgame-perfect equilibrium" are used interchangeably in the literature [28].

We refine our solution concept further to eliminate strategies that are weakly dominated within "subgames" of the entire game. This is crucial to deal with equilibrium selection, in particular, because the players' cannot unilaterally deviate out of a suboptimal equilibria. We say an SSE s **weakly dominates** another SSE s' if, for any player i, $u_i(s) \geq u_i(s')$. A strategy s is **weakly dominant** if it dominates all SSEs. Next we eliminate SSEs that are weakly dominated in subgames of the entire game. We use the generalized notion of subgames, called **subforms**, defined by Kreps and Wilson [36] for extensive-form games with imperfect information.

To review the definition of subforms, we need further notation. Let H be the set of histories of the game. Recall that a history is a sequence $(a^1, \ldots, a^K)$ of actions taken by the players.[4] For histories $h, h' \in H$, we say h has h' as a **prefix** if there exists some sequence of actions $b^1, \ldots, b^L$ (possibly empty) such that $h = (h', b^1, \ldots, b^L)$. For a history $h \in H$, let $I(h)$ be the unique information set containing h.

For an information set I, let H_I be the set of all histories following I, that is, H_I is the set of all histories $h \in H$ such that h has a prefix in I. We say that H_I is a **subform *rooted at*** I if for every information set I' such that $I' \cap H_I \neq \emptyset$, it holds that $I' \subseteq H_I$. Roughly speaking, a subform H_I "completely contains" all histories of the information sets following I, so there is no information asymmetry between the players acting within H_I.

Thus, given a strategy profile, the subform H_I together with the probability distribution μ_I^s on I, can be treated as a well-defined game.

We say an SSE s **weakly dominates** SSE s' **on a subform** H_I if, for any player j acting in H_I, the expected utility of j under s_I in the game (H_I, μ_I^s) is greater than or equal to their utility under s'_I in the game $(H_I, \mu_I^{s'})$.

We eliminate weakly dominated strategies by imposing this dominance condition in a backward-induction-compatible way on the subforms as follows.

▶ **Definition 7** (Dominant Strong Sequential Equilibrium). *A strategy profile s is a dominant strong sequential equilibrium if s is an SSE and*

- *for every subform H_I of height 1: s weakly dominates s' on H_I for any SSE s'*
- *for every subform H_I subgame of height > 1: s weakly dominates s' on H_I for any SSE s' that is a dominant SSE in all subforms of height at most $h - 1$.*

We are ready to define non-cooperative rational interactive proofs.

▶ **Definition 8** (Non-Cooperative Rational Interactive Proof). *Fix an arbitrary string x and language L. An interactive protocol $(V, \vec{P})$ is a* non-cooperative rational interactive proof *(ncRIP) protocol for L if there exists a strategy profile s of the provers that is a dominant SSE in the resulting extensive-form game, and under any dominant SSE, the answer bit c output by the verifier is correct (i.e., $c = 1$ iff $x \in L$) with probability 1, where the probability is taken over the verifier's randomness.*

2.2 Utility Gap in ncRIP Protocols

In game theory, players are assumed to be perfectly rational and "sensitive" to arbitrarily small utility losses. In reality, some provers may not care about small losses. Such provers may not have sufficient incentive to reach a dominant SSE, and could end up leading the verifier to the wrong answer. To design ncRIP protocols that are robust against such "insensitive" provers, we define the notion of **utility gap**.

[4] In the full version of the paper, we present a more formal treatment of the underlying extensive-form game, based on [39].

Informally, a utility gap of u means that if a strategy profile s leads the verifier to the wrong answer, there must exist a subform, such that some provers must lose at least a $1/u$ amount in their final individual payments (compared to their optimal strategy in that subform). As a consequence, these provers will not deviate to s, as long as they care about $1/u$ payment losses. We formalize this notion below. (We say a subform H_I is reachable under s if the information set I is reached under s with non-zero probability.)

▶ **Definition 9** (Utility Gap). *Let $(V, \vec{P})$ be an ncRIP protocol for a language L and s^* be a dominant SSE of the resulting game. The protocol $(V, \vec{P})$ has an $\alpha(n)$-*utility gap *or $\alpha(n)$-gap, if for any strategy profile s' under which the answer bit c' is wrong, there exists a subform H_I reachable under s', and a prover P_j acting in H_I who has deviated from s^* such that*

$$u_j(x, (s'_{-I}, s^*_I), (V, \vec{P})) - u_j(x, (s'_{-I}, s'_I), (V, \vec{P})) > 1/\alpha(n),$$

where s'_{-I} denotes the strategy profile s' outside subform H_I, that is, $s'_{-I} = s' \setminus s'_I$.

The class of languages that have an ncRIP protocol with **constant**, **polynomial** and **exponential** utility gap, are denoted by $\mathsf{O}(1)$-ncRIP, $\mathsf{poly}(n)$-ncRIP, and $\mathsf{exp}(n)$-ncRIP respectively.[5] Note that $\alpha(n)$ gap corresponds to a payment loss of $1/\alpha(n)$, so an exponential utility gap is the weakest guarantee.

3 Lower Bounds: ncRIP Protocols with Utility Gap

In this section, we give an $O(1)$-utility gap ncRIP protocol for the class NEXP and use it to give an $O(\alpha(n))$-utility gap ncRIP protocol for the class $\mathsf{P}^{\mathsf{NEXP}[\alpha(n)]}$. Setting $\alpha(n)$ to be a constant or polynomial in n gives us $\mathsf{P}^{\mathsf{NEXP}[\mathsf{O}(1)]} \subseteq \mathsf{O}(1)$-ncRIP and $\mathsf{P}^{\mathsf{NEXP}} \subseteq \mathsf{poly}(n)$-ncRIP respectively.

Here we argue correctness of our protocols at a high level; see the full version of the paper for formal proofs.

A constant-gap ncRIP protocol for NEXP

The ncRIP protocol for any language in NEXP is in Figure 2. The protocol uses the 2-prover 1-round MIP for NEXP [23] as a blackbox.[6] The protocol in Figure 2 essentially forces the non-cooperative provers to coordinate by giving them identical payments. As a result, it is almost identical to the MRIP protocol for NEXP [18].

While the payment scheme is simple, in the analysis we have to open up the black-box MIP. In particular, if P_1 sends $c = 0$ in round 1, all the information sets of P_1 and P_2 in round 3 become unreachable. To show that an SSE exists, we show that the provers have a best response at these unreachable sets, which is argued based on the messages exchanged in the MIP protocol.

▶ **Lemma 10.** *Any language $L \in$ NEXP has a 2-prover 3-round $6/5$-gap ncRIP protocol.*

[5] These classes are formally defined by taking the union over languages with $\alpha(n)$ utility gap, for every $\alpha(n)$ that is constant, polynomial and exponential in n respectively.

[6] It is also possible to give a scoring-rule based ncRIP protocol for NEXP, similar to MRIP [18]. However, such a protocol has an exponential utility gap.

For any input x and language $L \in \mathsf{NEXP}$, the protocol (V, P_1, P_2) for L is:

1. P_1 sends a bit c to V. V outputs c at the end of the protocol.
2. If $c = 0$, then the protocol ends and the payments are $R_1 = R_2 = 1/2$.
3. Otherwise, V runs the classic 2-prover 1-round MIP protocol for NEXP [23] with P_1 and P_2 to prove if $x \in L$. If the MIP protocol accepts then $R_1 = 1$, $R_2 = 1$; else, $R_1 = -1$, $R_2 = -1$.

Figure 2 A simple $O(1)$-utility gap ncRIP protocol for NEXP.

An $O(\alpha(n))$-gap ncRIP protocol for $\mathsf{P}^{\mathsf{NEXP}[\alpha(n)]}$

Using the above NEXP protocol as a subroutine, we give an ncRIP protocol with $O(\alpha(n))$-utility gap for the class $\mathsf{P}^{\mathsf{NEXP}[\alpha(n)]}$. This protocol works for any function $\alpha(n)$ which (1) is a positive integer for all n, (2) is upper-bounded by a polynomial in n, and (3) is polynomial-time computable.[7]

▶ **Lemma 11.** *Any language $L \in \mathsf{P}^{\mathsf{NEXP}[\alpha(n)]}$ has a 3-prover 5-round ncRIP protocol that has a utility gap of $6/(5\alpha(n))$.*

The ncRIP protocol for any $L \in \mathsf{P}^{\mathsf{NEXP}[\alpha(n)]}$ is in Figure 3. It is fairly intuitive – V simulates the polynomial-time Turing machine directly, and uses the ncRIP protocol for NEXP for the oracle queries.

For any input x of length n, the protocol $(V, \vec{P})$ works as follows.

1. P_1 sends $(c, c_1, \ldots, c_{\alpha(n)}) \in \{0,1\}^{\alpha(n)+1}$ to V. V outputs c at the end of the protocol.
2. V simulates M on x using the bits $c_1, \ldots, c_{\alpha(n)}$ as answers to NEXP queries $\phi_1, \ldots, \phi_{\alpha(n)}$ generated by M respectively. If M accepts and $c = 0$ or M rejects and $c = 1$, then the protocol ends and $R_1 = -1, R_2 = R_3 = 0$.
3. V picks a random index i' from $\{1, \ldots, \alpha(n)\}$ and sends $(i', \phi_{i'})$ to P_2 and P_3.
4. V runs the 2-prover 3-round $O(1)$-gap ncRIP protocol for NEXP (Figure 2) with P_2 and P_3 on ϕ_i. P_2 and P_3 get payments R_2 and R_3 based on the protocol. Let $c^*_{i'}$ be the answer bit in the NEXP protocol. If $c^*_{i'} \neq c_{i'}$, then $R_1 = 0$; otherwise $R_1 = 1$.

Figure 3 An $O(\alpha(n))$-utility gap ncRIP protocol for $\mathsf{P}^{\mathsf{NEXP}[\alpha(n)]}$.

We argue the correctness of this protocol at a high-level. Under any strategy of P_1, the resulting NEXP queries in the protocol in Figure 3 are the roots of non-trivial subforms. Which of these subforms are reachable under a strategy profile s is determined solely by the strategy of P_1. However, because weak dominance is imposed on all subforms in a bottom-up fashion, P_2 and P_3 must play their optimal strategy in these subforms regardless of their reachability – and therefore, they must play optimally for any strategy of P_1. (This is one example of why ruling out weakly-dominated strategies in subforms in the definition of dominant SSEs is crucial to arguing correctness.) From the correctness of the NEXP protocol in Figure 2, we know that the optimal strategy of P_2 and P_3 is to compute the NEXP queries correctly. Given that the best response of P_2 and P_3 is to solve the NEXP queries correctly, and given that V randomly verifies 1 out of $\alpha(n)$ queries, P_1 must commit to correct answer bits in the first round, or risk losing a $O(1/\alpha(n))$ amount from his expected payment.

[7] For Theorem 1 and Theorem 2, $\alpha(n)$ need only be a constant or polynomial in n. However, Lemma 11 holds for all $\alpha(n)$'s that are polynomial-time computable (given 1^n) and polynomially bounded, such as $\log n$, $\sqrt{n}$, etc.

If P_1 gives the correct answer bits in step 1, but P_2 or P_3 deviate within a subform corresponding to an NEXP query ϕ_q, then with probability $1/\alpha(n)$, V simulates the protocol in Figure 3 on ϕ_q, in which case they lose a constant amount of their expected payment.

4 Upper Bounds: ncRIP Protocols with Utility Gap

In this section, we prove matching upper bounds on the classes of ncRIP protocols with constant and polynomial utility gaps. In particular, we show that any language in O(1)-ncRIP (or poly(n)-ncRIP) can be decided by a polynomial-time Turing machine with a constant (resp. polynomial) number of queries to an NEXP oracle.

To simulate an ncRIP protocol, we need to find a strategy profile "close enough" to the dominant SSE so that the answer bit is still correct, i.e. a strategy profile that satisfies the utility-gap guarantee. We formalize this restatement of Definition 9 below.

▶ **Observation 12.** *Given input x and an ncRIP protocol $(V, \vec{P})$ with $\alpha(n)$-utility gap, let s be a strategy profile such that for all reachable subforms H_I and all provers P_j acting in H_I,*

$$u_j(x, r, (V, \vec{P}), (s_{-I}, s_I^*)) - u_j(x, r, (V, \vec{P}), (s_{-I}, s_I)) < \frac{1}{\alpha(n)},$$

where s^* is a dominant SSE. Then, the answer bit c under s must be correct.

There are several challenges involved in finding a strategy profile satisfying Observation 12.

First, the size of the game tree of any ncRIP protocol – small gap notwithstanding – can be exponential in n. Even if the polynomial-time machine considers a single strategy profile s at a time, since V can flip polynomially many coins, the part of the tree "in play" – the number of decision nodes reached with positive probability under s – can be exponential in n.

The second (and related) challenge is that of verifying whether a strategy profile is a dominant SSE. While the NEXP oracle can guess and verify an SSE, it cannot directly help with dominant SSEs. The polynomial-time machine must check using backward induction if an SSE is dominant on all its reachable subforms, which can again be exponential in n.

Finally, the polynomial-time machine needs to search through the exponentially large strategy-profile space in an efficient way to find one which leads to the correct answer.

In the remainder of the section we address these challenges. In Lemma 13 we show that we can prune the game tree, resolving the first two challenges. Then in Lemmas 17 and 18, we show how to efficiently search through the strategy-profile space.

Pruning Nature moves in ncRIP protocols

We now give our main technical lemma for the upper bound, which shows that we can limit ourselves to examining protocols with bounded game trees without loss of generality.

Recall that a verifier's coin flips in an ncRIP protocol represent Nature moves in the resulting game. The problem is that a polynomial-time verifier can result in Nature moves that impose nonzero probabilities over exponentially many outcomes.

We prune the Nature moves of a verifier so that a polynomial-time Turing machine simulating an $\alpha(n)$-utility-gap protocol can traverse the game tree reachable under a given s. This pruning operation takes exponential time (linear in the size of the game tree), and can be performed by the NEXP oracle.

▶ **Lemma 13** (Pruning Lemma)**.** *Let $L \in \alpha(n)$-ncRIP and let $(V, \vec{P})$ be an ncRIP protocol for L with $\alpha(n)$ utility gap and $p(n)$ provers. Given an input x and a strategy s, the protocol $(V, \vec{P})$ can be transformed in exponential time to a new protocol $(V', \vec{P})$, where*

- *the probability distribution on the outcomes imposed by the Nature moves of V' for input x has $O(\alpha(n))$ support,*
- *if s is a dominant SSE of $(V, \vec{P})$, then s induces a dominant SSE in $(V', \vec{P})$,*
- *$|u_j(x, s, (V, \vec{P})) - u_j(x, s, (V', \vec{P}))| < 1/(4\alpha(n))$ for all $j \in \{1, \ldots, p(n)\}$, and*
- *the utility gap guarantee is preserved, that is, if the answer bit under s is wrong, then there exists a subform H_I in the game $(V', \vec{P})$ (reachable under s) and a prover P_j acting at H_I, such that P_j loses a $1/(2\alpha(n))$ amount in his expected payment compared to a strategy profile where s_I (induced by s on H_I) is replaced by s_I^* (the dominant SSE on H_I), keeping the strategy profile outside H_I, s_{-I}, fixed.*

We prove Lemma 13 in several parts. First, given an input x and a strategy s of the provers, we show how to transform any verifier V that imposes a probability distribution over outcomes with exponential support into a verifier V' that imposes a probability distribution with $O(\alpha(n))$ support.

Let $(V, \vec{P})$ use $p(n)$ provers and let the running time of V be n^k for some constant k. There can be at most 2^{n^k} different payments that V can generate for a particular prover given input x. Given x and s, fix a prover index $j \in \{1, \ldots, p(n)\}$. Let $R_1, R_2, \ldots, R_m$ be the payments generated by V on s for P_j. Let V's randomness assign probability distribution $\mu = (p_1, p_2, \ldots, p_m)$ to $R_1, R_2, \ldots, R_m$ respectively. Then, the expected payment of P_j under s is $u_j(x, s, (V, \vec{P})) = \sum_{i=1}^{m} p_i R_i$.

Recall that $u_j(x, s, (V, \vec{P})) \in [-1, 1]$ for all $1 \leq j \leq p(n)$. For each prover P_j, divide the interval $[-1, 1]$ into $4\alpha(n)$ intervals, each of length $1/(2\alpha(n))$. In other words, prover P_j's ith interval is $[i/2\alpha(n), (i+1)/2\alpha(n))$,[8] for each $i \in \{-2\alpha(n), \ldots, 2\alpha(n) - 1\}$.

We round the possible payments for P_j to a representative of the their corresponding interval. Specifically, we map each payment R_i to r_j as described in Equation 1. There

$$r_j = \begin{cases} \frac{4\ell+1}{4\alpha(n)} & \text{if } R_i \in \left[\frac{\ell}{2\alpha(n)}, \frac{2\ell+1}{4\alpha(n)}\right) \\ \frac{4\ell+3}{4\alpha(n)} & \text{if } R_i \in \left[\frac{2\ell+1}{4\alpha(n)}, \frac{\ell+1}{2\alpha(n)}\right) \end{cases} \quad (1) \qquad p'_i = \begin{cases} \sum_{k \in T_j} p_k & \text{if } i = f(S(i)) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

are potentially exponentially many different payments R_i, and only polynomially many different payments r_j, so several R_i must map to the same r_j. Let $T_j = \{i : R_i \text{ maps to } r_j\}$. Let $\mathcal{T} = \cup_j \{T_j\}$. Thus the total number of distinct r_j is $8\alpha(n)$, so $|\mathcal{T}| = O(\alpha(n))$. Let $S : \{1, \ldots, m\} \to \mathcal{T}$ such that $S(i) = T_j$ if and only if $i \in T_j$.

For each $T_j \in \mathcal{T}$, let $f(T_j)$ denote a unique index in the set T_j. Without loss of generality, let $f(T_j)$ be the lowest index in T_j. We define a new probability distribution $\mu' = (p'_1, \ldots, p'_h)$ over the payments $R_1, \ldots, R_h$ respectively, given by Equation 2. In particular, for every $T_j \in \mathcal{T}$, assign $R_{f(T_j)}$ probability $\sum_{k \in T_j} p_k$ and for every other index $\ell \in T_j$, $\ell \neq f(T_j)$, assign R_ℓ probability 0.

Define V' as a polynomial-time verifier that simulates all deterministic computation of V. For a fixed input x, V' imposes a probability distribution μ' with $O(\alpha(n))$ support for any probability distribution μ imposed by V. For other inputs, V' simulates V without any modification.

[8] To include 1 as a possible payment, interval $2\alpha(n) - 1$ should be closed on both sides; we ignore this for simplicity.

Note that given input x, a strategy profile s and the protocol $(V, \vec{P})$, transforming the distribution μ to μ' takes time linear in the size of the game tree, and thus exponential in n. (This means that an NEXP oracle, given x, can guess a particular s and perform the transformation.)

The remainder of the proof of Lemma 13 consists of the following three claims. We argue their correctness at a high-level and defer formal proofs to the full version.

First, we show that if the strategy profile s is a dominant SSE of $(V, \vec{P})$, then s restricted to the pruned game tree of $(V', \vec{P})$ imposes a dominant SSE on $(V', \vec{P})$ as well.

▶ **Claim 14.** *Any dominant SSE s of the game formed by $(V, \vec{P})$ induces a dominant SSE in the game formed by $(V', \vec{P})$.*

First, we prove that s is an SSE of $(V', \vec{P})$. Suppose by contradiction that s is not a SSE of $(V', \vec{P})$. Then there exists an information set I, such that, conditioned on reaching I, the prover acting at I can improve his expected payment by deviating (given his belief u'_I at I if I is reachable under s and for any belief he may hold at I if I is unreachable under s). Writing out their expected payments, accounting for the probabilistic transformation between V and V', in both cases leads to a contradiction to the assumption that s was an SSE in $(V, \vec{P})$. We then argue that a similar contradiction holds for proving that s is a dominant SSE of $(V', \vec{P})$.

The following claim states that for a given s, the expected payments of the provers under $(V, \vec{P})$ and under $(V', \vec{P})$ are not too far off. This claim is one of the bullet points in Lemma 13, and will be used to prove Claim 16.

▶ **Claim 15.** *For all $j \in \{1, \ldots, p(n)\}$, $|u_j(x, s, (V, \vec{P})) - u_j(x, s, (V', \vec{P}))| < 1/(4\alpha(n))$.*

With the above, we show that $(V', \vec{P})$ preserves utility gap guarantees.

▶ **Claim 16.** *Given input x, if the answer bit under s is wrong, then there exists a subform H_I reachable under s in $(V', \vec{P})$ and P_j acting at H_I, such that P_j's expected payment under s is $\frac{1}{2\alpha(n)}$ less than his expected payment under (s_{-I}, s^*_I), where s^*_I is a dominant SSE on H_I.*

Consider a strategy profile s^* that is a dominant SSE in the game tree of $(V, \vec{P})$. Since s gives the wrong answer bit, from the $\alpha(n)$-utility gap guarantee of $(V, \vec{P})$ and Definition 9, there exists a subform H_I reachable under s, such that a prover P_j acting in H_I loses $1/\alpha(n)$ in his expected payment under s compared to the strategy profile (s_{-I}, s^*_I).

Using Claim 14, s^* also induces a dominant SSE in the game tree of $(V', \vec{P})$. And since H_I is reachable under s in $(V, \vec{P})$, it is reachable under s in $(V', \vec{P})$ as well. Finally, Claim 16 follows by applying Claim 15 twice: once to show that payments under V and V' are similar under s, and once to show that the payments are similar under (s_{-I}, s^*_I). In the worst case this leads to a payment difference of $1/(4\alpha(n)) + 1/(4\alpha(n)) = 1/(2\alpha(n))$.

Using Lemma 13, given an $O(\alpha(n))$-gap ncRIP protocol (where $\alpha(n)$ is constant or polynomial), a polynomial-time oracle Turing machine can use its NEXP oracle to guess a strategy profile s, prune the verifier's Nature moves, and report the resulting $O(\alpha(n))$-support distribution bit-by-bit. Thus, it can simulate the new distribution and find the decision nodes that are reachable under s.

Searching through the strategy-profile space efficiently

The next question is: how should the polynomial-time Turing machine navigate the potential strategy-profile space (in polynomial time) to find the strategy profile that satisfies Observation 12? To cut down on the search space, we invoke a recurring idea: divide each prover's expected payment interval $[-1, 1]$, evenly into $8\alpha(n)$ **subintervals** of length $1/(4\alpha(n))$, and consider **subinterval profiles** (a tuple of subintervals, one for each prover).

▶ **Lemma 17.** *Given an input x and an ncRIP protocol $(V, \vec{P})$ with $\alpha(n)$-utility gap, consider a subinterval profile $(L_1, \ldots, L_{p(n)})$, where each $L_i = [k/(4\alpha), (k+1)/(4\alpha+1))$ denotes a subinterval of prover P_i in $[-1, 1]$, for some $k \in \{-2\alpha(n), \ldots, 2\alpha(n)-1\}$. Let s be an SSE that has an expected payment profile $\tilde{u}(x, s)$ such that $u_i(x, s) \in L_i$ for all $1 \leq i \leq p(n)$, and s does not satisfy Observation 12. Then the expected payment profile $\tilde{u}(x, s^*)$ under a dominant SSE s^* cannot lie in the same subinterval profile, that is, there exists a prover index j such that $u_j(x, s^*) \notin L_j$.*

Using Lemma 17, if the polynomial-time Turing machine is able to test *any* SSE s with $\tilde{u}(x, s)$ in a subinterval profile, for all subinterval profiles, then it is guaranteed to find one that satisfies Observation 12. This is because a dominant SSE of an ncRIP protocol is guaranteed to exist and its expected payment profile must belong to some subinterval profile.

However, there are still $O(\alpha(n))$ subintervals for each prover, and thus $O(\alpha(n)^{p(n)})$ total subinterval profiles. A polynomial-time machine cannot test SSEs for each of them.

To reduce the search space further, we show that it is sufficient to consider subintervals of the total expected payment rather than individual and test an SSE s for each of them. Recall that a SSE s is weakly dominant if for any player i and SSE s', $u_i(s) \geq u_i(s')$.

▶ **Lemma 18.** *If a weakly-dominant SSE exists, then a strategy profile s is a weakly-dominant SSE if and only if s is an SSE and s maximizes the sum of utilities of all players among all SSEs.*

We are now ready to prove the upper bound for ncRIP classes with constant and polynomial utility gap. We defer formal details of the proof to the full version of the paper.

Constant utility gap

Using Lemma 13 and Lemma 18, simulating a constant-gap protocol using a $\mathsf{P}^{\mathsf{NEXP}[\mathsf{O}(1)]}$ machine M is straightforward. We give a high-level overview below.

There are at most $O(1)$ subforms that are reachable under any strategy profile s, and the total expected payment of the provers conditioned on reaching these subforms will be in one of the $O(1)$ subintervals. Thus, there are $O(1)$ combinations of total expected payments on all subforms (including the whole game). M queries its NEXP oracle whether there exists an SSE that achieves that combination of total expected payments on those subforms, for all combinations. Then, M finds the maximum among all of the combinations that got a "yes." Such a maximum is guaranteed to exist for an ncRIP protocol. Finally, M queries the oracle for the answer bit of the corresponding SSE by giving the dominant profile of total expected payments over the subgames.

▶ **Lemma 19.** $\mathsf{O}(1)\text{-ncRIP} \subseteq \mathsf{P}^{\mathsf{NEXP}[\mathsf{O}(1)]}$.

Polynomial utility gap

To simulate a polynomial-utility gap ncRIP protocol $(V, \vec{P})$, using a $\mathsf{P}^{\mathsf{NEXP}}$ machine M, we put to use all the structure we have established in this section.

For each of the $O(\alpha(n))$ total payment subintervals of the interval $[-1, 1]$ that correspond to an SSE, M does a recursive search to find an exact total expected payment $u(x, s)$ that is generated by an SSE. (We can restrict ourselves to $O(\alpha(n))$ oracle queries due to Lemma 18.) In particular, M queries the NEXP oracle: *Does there exist an SSE with total expected payment in the first half of the ith interval?*. If the answer is *yes* then M recurses on the first half of the ith interval; M does not need to search the second half by Lemma 17. Otherwise

(if the answer is *no*) then M recurses on the second half. Thus, in polynomial time and with polynomial queries, M can find an exact $u(x, s)$ for an SSE s in the subinterval using the power of its *adaptive* queries.

Next, M simulates the protocol $(V, \vec{P})$ with the help of the oracle, under the SSE s for a given subinterval. Lemma 13 is crucial for M to simulate the verifier's moves, because V in general can induce exponential-size distributions. M traverses the tree reachable under s "top-down" using the oracle to learn the pruned distributions and provers' moves. Finally, M goes "bottom-up" to test whether s satisfies Observation 12 on all its reachable subgames.

▶ **Lemma 20.** $\mathsf{poly}(n)\text{-ncRIP} \subseteq \mathsf{P}^{\mathsf{NEXP}}$.

References

1 Pablo Daniel Azar and Silvio Micali. Rational proofs. In *Proceedings of the Forty-Fourth Annual Symposium on Theory of Computing (STOC)*, pages 1017–1028, 2012.

2 Pablo Daniel Azar and Silvio Micali. Super-efficient rational proofs. In *Proceedings of the Fourteenth Annual ACM conference on Electronic Commerce (EC)*, pages 29–30, 2013.

3 László Babai. Trading group theory for randomness. In *Proceedings of the Seventeenth annual ACM symposium on Theory of Computing (STOC)*, pages 421–429, 1985.

4 László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational complexity*, 1(1):3–40, 1991.

5 Jeffrey S Banks and Joel Sobel. Equilibrium selection in signaling games. *Econometrica: Journal of the Econometric Society*, pages 647–661, 1987.

6 Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing (STOC)*, pages 113–131, 1988.

7 Nir Bitansky and Alessandro Chiesa. Succinct arguments from multi-prover interactive proofs and their efficiency benefits. In *Advances in Cryptology–CRYPTO 2012*, pages 255–272. Springer, 2012.

8 Andrew J Blumberg, Justin Thaler, Victor Vu, and Michael Walfish. Verifiable computation using multiple provers. *IACR Cryptology ePrint Archive*, 2014:846, 2014.

9 Benjamin Braun, Ariel J Feldman, Zuocheng Ren, Srinath Setty, Andrew J Blumberg, and Michael Walfish. Verifying computations with state. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, pages 341–357, 2013.

10 Jin-yi Cai, Anne Condon, and Richard J Lipton. On games of incomplete information. *Theoretical Computer Science*, 103(1):25–38, 1992.

11 Matteo Campanelli and Rosario Gennaro. Sequentially Composable Rational Proofs. In *International Conference on Decision and Game Theory for Security*, pages 270–288, 2015.

12 Matteo Campanelli and Rosario Gennaro. Efficient Rational Proofs for Space Bounded Computations. In *International Conference on Decision and Game Theory for Security*, pages 53–73, 2017.

13 Ran Canetti, Ben Riva, and Guy N Rothblum. Practical delegation of computation using multiple servers. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 445–454, 2011.

14 Ran Canetti, Ben Riva, and Guy N Rothblum. Two 1-round protocols for delegation of computation. In *International Conference on Information Theoretic Security*, pages 37–61, 2012.

15 Ran Canetti, Ben Riva, and Guy N Rothblum. Refereed delegation of computation. *Information and Computation*, 226:16–36, 2013.

16 Ashok K Chandra and Larry J Stockmeyer. Alternation. In *17th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 98–108, 1976.

17 Jing Chen, Samuel McCauley, and Shikha Singh. Rational Proofs with Multiple Provers (Full Version). *arXiv preprint arXiv:1504.08361*, 2015. `arXiv:1504.08361`.

18 Jing Chen, Samuel McCauley, and Shikha Singh. Rational Proofs with Multiple Provers. In *Proceedings of the Seventh Innovations in Theoretical Computer Science Conference (ITCS)*, pages 237–248, 2016.

19 Jing Chen, Samuel McCauley, and Shikha Singh. Efficient Rational Proofs with Strong Utility-Gap Guarantees. In *International Symposium on Algorithmic Game Theory (SAGT)*, pages 150–162. Springer, 2018.

20 In-Koo Cho and David M Kreps. Signaling games and stable equilibria. *The Quarterly Journal of Economics*, 102(2):179–221, 1987.

21 John Duggan. An extensive form solution to the adverse selection problem in principal/multi-agent environments. *Review of Economic Design*, 3(2):167–191, 1998.

22 Uriel Feige and Joe Kilian. Making games short. In *Proceedings of the Twenty-Ninth Annual ACM Symposium On Theory of Computing (STOC)*, pages 506–516, 1997.

23 Uriel Feige and László Lovász. Two-prover one-round proof systems: their power and their problems. In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing (STOC)*, pages 733–744, 1992.

24 Uriel Feige and Adi Shamir. Multi-oracle interactive protocols with constant space verifiers. *Journal of Computer and System Sciences*, 44(2):259–271, 1992.

25 Uriel Feige, Adi Shamir, and Moshe Tennenholtz. The noisy oracle problem. In *Proceedings of the Tenth Annual Conference on Advances in Cryptology (CRYPTO)*, pages 284–296, 1990.

26 Joan Feigenbaum, Daphne Koller, and Peter Shor. A game-theoretic classification of interactive complexity classes. In *Proceedings of Tenth Annual IEEE Structure in Complexity Theory Conference*, pages 227–237, 1995.

27 Lance Fortnow, John Rompel, and Michael Sipser. On the power of multi-prover interactive protocols. *Theoretical Computer Science*, 134(2):545–557, 1994.

28 Jacob Glazer and Motty Perry. Virtual implementation in backwards induction. *Games and Economic Behavior*, 15(1):27–32, 1996.

29 S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM J. Comput.*, 18(1), 1989.

30 Shafi Goldwasser, Yael Tauman Kalai, and Guy N Rothblum. Delegating computation: interactive proofs for muggles. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing (STOC)*, pages 113–122, 2008.

31 Siyao Guo, Pavel Hubáček, Alon Rosen, and Margarita Vald. Rational arguments: single round delegation with sublinear verification. In *Proceedings of the Fifth Annual Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 523–540, 2014.

32 Siyao Guo, Pavel Hubáček, Alon Rosen, and Margarita Vald. Rational sumchecks. In *Theory of Cryptography Conference*, pages 319–351, 2016.

33 Tsuyoshi Ito and Thomas Vidick. A multi-prover interactive proof for NEXP sound against entangled provers. In *53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 243–252, 2012.

34 Gillat Kol and Ran Raz. Competing provers protocols for circuit evaluation. In *Proceedings of the Fourth Annual Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 473–484, 2013.

35 Daphne Koller and Nimrod Megiddo. The complexity of two-person zero-sum games in extensive form. *Games and economic behavior*, 4(4):528–552, 1992.

36 David M Kreps and Robert Wilson. Sequential equilibria. *Econometrica: Journal of the Econometric Society*, pages 863–894, 1982.

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

38 Andrew McLennan. Justifiable beliefs in sequential equilibrium. *Econometrica: Journal of the Econometric Society*, pages 889–904, 1985.

39 Martin J Osborne and Ariel Rubinstein. *A course in game theory*. MIT press, 1994.

40 John H Reif. The complexity of two-player games of incomplete information. *Journal of Computer and System Sciences*, 29(2):274–301, 1984.

41 Guy N Rothblum, Salil Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 793–802, 2013.

42 Reinhard Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International journal of game theory*, 4(1):25–55, 1975.

43 Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

44 Madhu Sudan. Probabilistically checkable proofs. *Communications of the ACM*, 52(3):76–84, 2009.

45 Justin Thaler, Mike Roberts, Michael Mitzenmacher, and Hanspeter Pfister. Verifiable Computation with Massively Parallel Interactive Proofs. In *HotCloud*, 2012.

46 Hannu Vartiainen. Subgame perfect implementation of voting rules via randomized mechanisms. *Social Choice and Welfare*, 29(3):353–367, 2007.

47 Michael Walfish and Andrew J Blumberg. Verifying computations without reexecuting them. *Communications of the ACM*, 58(2):74–84, 2015.