ON INTERACTIVE PROOFS WITH A LACONIC PROVER

Oded Goldreich, Salil Vadhan, and Avi Wigderson

Abstract. We continue the investigation of interactive proofs with bounded communication, as initiated by Goldreich and Håstad (IPL 1998). Let L be a language that has an interactive proof in which the prover sends few (say b) bits to the verifier. We prove that the complement $\bar{L}$ has a *constant-round* interactive proof of complexity that depends only exponentially on b. This provides the first evidence that for **NP**-complete languages, we cannot expect interactive provers to be much more "laconic" than the standard **NP** proof. When the proof system is further restricted (*e.g.,* when $b = 1$, or when we have perfect completeness), we get significantly better upper bounds on the complexity of $\bar{L}$.

Keywords. Interactive Proof systems, Arthur-Merlin games, **NP**, sampling protocols, statistical zero-knowledge, game theory.

Subject classification. 68Q05, 68Q10, 68Q15, 91A05.

1. Introduction

Interactive proof systems were introduce by Goldwasser, Micali and Rackoff [GMR89] in order to capture the most general way in which one party can *efficiently verify* claims made by another, more powerful party.[1] That is, interactive proof systems are two-party randomized protocols through which a computationally unbounded prover can convince a probabilistic polynomial-time verifier of the membership of a common input in a predetermined language. Thus, interactive proof systems generalize and contain as a special case the traditional "**NP**-proof systems" (in which verification is deterministic and "non-interactive").

It is well-known that this generalization buys us a lot: The *IP Characterization Theorem* of Lund, Fortnow, Karloff, Nisan and Shamir [LFKN92, Sha92]

[1]Arthur-Merlin games, introduced by Babai [Bab85], are a special type of interactive proofs in which the verifier is restricted to send the outcome of each coin it tosses. Such proof systems are also called *public coin*, and are known to be as expressive as general interactive proofs [GS89]. We warn that the latter assertion refers to the entire class but not to refined complexity measures such as the total number of bits sent by the prover (considered below).

states that every language in **PSPACE** has an interactive proof system (and it is easy to see that only languages in **PSPACE** have interactive proof systems).

It is well-known that the strong expressive power of interactive proofs is largely due to the presence of interaction. In particular, interactive proofs in which a single message is sent (like in **NP**-proofs) yield a complexity class (known as **MA**) that seems very close to **NP**. It is interesting to explore what happens between these extremes of unbounded interaction and no interaction. That is, *what is the expressive power of interactive proofs that utilize a bounded, but nonzero, amount of interaction?*

1.1. Prior work regarding interactive proofs with bounded interaction.

Interactive Proofs with Few Messages. The earliest investigations of the above question examined the *message complexity* of interactive proofs, *i.e.,* the number of messages exchanged. (Sometimes, we refer to *rounds*, which are a pair of verifier-prover messages.) The Speedup Theorem of Babai and Moran [BM88] (together with [GS89]) shows that the number of messages in an interactive proof can be always reduced by a constant factor (provided the number of messages remains at least 2). On the other hand, there is a large gap between constant-round interactive proofs and unrestricted interactive proofs. As mentioned above, all of **PSPACE** has a general interactive proof [LFKN92, Sha92]. In contrast, the class **AM** of problems with constant-round interactive proofs is believed to be relatively close to **NP**. Specifically, **AM** lies in the second level of the polynomial-time hierarchy [BM88], cannot contain **coNP** unless the polynomial-time hierarchy collapses [BHZ87], and actually equals **NP** under plausible circuit complexity assumptions [AK97, KvM99, MV99].

Laconic Provers. A more refined investigation of the above question was initiated by Goldreich and Håstad [GH98], who gave bounds on the complexity of languages possessing interactive proofs with various restrictions on the *number of bits* of communication (and/or randomness) used. One of the restrictions they considered, and the main focus of our investigation, limits the number of bits sent from the prover to the verifier by some bound b. That is, what languages can be proven by "laconic" provers?

Since the prover is trying to convey something to the verifier, this seems to be the most interesting direction of communication. Moreover, for applications of interactive proofs (*e.g.,* in cryptographic protocols), it models the common situation in which communication is more expensive in one direction (*e.g.,* if the prover is a hand-held wireless device).

On one hand, we know of interactive proofs for several "hard" problems (*e.g.,* Quadratic Nonresiduosity [GMR89], Graph Nonisomorphism [GMW91], and others [GK93, GG00, SV03]) in which the communication from the prover to the verifier is severely bounded (in fact, to one bit). On the other hand, laconic provers exist only for problems in $\mathbf{BPP}^{\mathbf{NP}}$ (resp., **BPP** in case the proof system is of the public-coin type) [GH98]. Furthermore, it was conjectured that **NP**-complete problems cannot have general interactive proofs with laconic provers (but the results in [GH98] fall short of supporting this conjecture). In this work, we provide strong support for this conjecture.

1.2. New results regarding interactive proofs with bounded interaction. Our main focus is on laconic provers; that is, on interactive proofs in which the total number of bits sent by the prover is bounded.

Laconic Provers. Consider interactive proofs in which the prover sends at most $b = b(n)$ bits to the verifier on inputs of length n. Goldreich and Håstad [GH98, Thm. 4] placed such languages in $\mathbf{BPTIME}(T)^{\mathbf{NP}}$, where $T = \mathrm{poly}(n) \cdot 2^{\mathrm{poly}(b)}$, which clearly implies nothing for languages in **NP**. In contrast, we show that the *complements* of such languages have *constant-round* interactive proofs of complexity T (*i.e.,* the verifier's computation time and the total communication is bounded by T). In particular, **NP**-complete problems cannot have interactive proofs in which the prover sends poly-logarithmically many bits to the verifier, unless **coNP** is in the quasi-polynomial analogue of **AM**. In fact, assuming **NP** has constant-round interactive proofs with logarithmic prover-to-verifier communication we conclude $\mathbf{coNP} \subseteq \mathbf{AM}$. As mentioned above, this is highly unlikely. We obtain stronger results in two special cases:

1. We show that if a language has an interactive proof of perfect completeness (*i.e.,* zero error probability on yes instances) in which the prover sends at most $b(n)$ bits, then it is in $\mathbf{coNTIME}(T)$, where $T(n) = 2^{b(n)} \cdot \mathrm{poly}(n)$. Thus, unless $\mathbf{NP} = \mathbf{coNP}$, NP-complete languages cannot have interactive proof systems of perfect completeness in which the prover sends logarithmically many bits.

2. We show that if a language has an interactive proof in which the prover sends a single bit (with some restrictions on the error probabilities), then it has a statistical zero-knowledge interactive proof; that is, is in the

class **SZK**. This is a stronger conclusion than our main result because $\mathbf{SZK} \subseteq \mathbf{AM} \cap \mathbf{coAM}$, as shown by Fortnow [For89] and Aiello and Håstad [AH91]. Recalling that Sahai and Vadhan [SV03] showed that any language in **SZK** has an interactive proof in which the prover sends a single bit, we obtain a surprising equivalence between these two classes.

Interactive Proofs with Few Messages. We obtain one (apparently) new result regarding message complexity. A question that is left open by the results mentioned earlier is what happens "in between" constant rounds and polynomially many rounds. Phrased differently, can the Speedup Theorem of Babai and Moran be improved to show that $m(n)$-message interactive proofs can be emulated by (and hence are no more powerful than) $m'(n)$-message interactive proofs for some $m' = o(m)$? By combining careful parameterizations of [LFKN92, Sha92] and [BM88], we observe that such an improvement speedup is unlikely. More precisely, for every nice function m, we show that there is a language which has an $m(n)$-message interactive proof but not an $o(m(n))$-message one, provided that #SAT is not contained in the sub-exponential analogue of **coAM**.

1.3. Additional related work. We note that Goldreich and Håstad [GH98] have presented significantly stronger results regarding interactive proofs with laconic provers when further restrictions are imposed on the interactive proof. In particular, they obtain an upper bound of $\mathbf{BPTIME}(T)$ (rather than $\mathbf{BPTIME}(T)^{\mathbf{NP}}$), with $T = 2^{\mathrm{poly}(b)} \cdot \mathrm{poly}(n)$, for languages possessing either of the following kinds of interactive proofs: (1) *public-coin* proofs in which the prover sends at most b bits, (2) proofs in which the communication *in both directions* is bounded by b.

Multi-prover interactive proofs and PCP. The expressive power of *multi-prover interactive proofs* (MIP's) and *probabilistically checkable proofs* (PCP's) with low communication has been the focus of extensive research. Much of this research is motivated by the importance of the communication parameter in the applications of MIP/PCP to inapproximability. In particular, Bellare, Goldreich, and Sudan [BGS98] give negative results about the expressive power of "laconic" PCP's and MIP's. Since one-query PCP's are equivalent to interactive proofs in which the prover sends a single message, our results provide bounds on the former.

Knowledge complexity of interactive proofs. Our work is also related to work on *knowledge complexity.* Knowledge complexity, proposed by [GMR89], aims to measure how much "knowledge" is leaked from the prover to the verifier in an interactive proof. Several measures of knowledge complexity were proposed by Goldreich and Petrank [GP99], and a series of works provided upper bounds on the complexity of languages having interactive proofs with low knowledge complexity (see [GOP98, PT96] for results regarding the main notion of knowledge complexity and [GP99, ABV95, SV03] for results regarding alternative notions). These results are related to, but incomparable to ours.

For example, Petrank and Tardos [PT96] showed that languages having knowledge complexity $k = O(\log n)$ are contained in $\mathbf{AM} \cap \mathbf{coAM}$. While it is true that the knowledge complexity of an interactive proof is bounded by the amount of prover-to-verifier communication, their result does not yield anything interesting for laconic interactive proofs. The reason is that their result only applies to interactive proofs with error probabilities significantly smaller than 2^{-k}, and it is easy to see that interactive proofs with prover-to-verifier communication $k = O(\log n)$ and error probability noticeably smaller than 2^{-k} only capture **BPP** (and hence are uninteresting). In contrast, our results apply even for constant error probabilities.

Sahai and Vadhan [SV03] (improving [GP99]) showed that languages with logarithmic knowledge complexity in the "hint sense" collapse to **SZK**, and their result applies even if the error probabilities are constant. However, this is also incomparable to ours, because the "hint sense" is the one measure of knowledge complexity which is *not* bounded by the prover-to-verifier communication. (Indeed, the "hint sense" formulation was dismissed as a satisfactory definition of knowledge complexity by Goldreich and Petrank [GP99] because of the above and related issues. Still knowledge complexity in the "hint sense" yields an interesting extension of zero-knowledge.)

Computationally-sound interactive proofs. Finally, it is important to note that the situation is dramatically different for *argument systems* [BCC88] (also known as *computationally sound proofs*). These are like interactive proofs, but the soundness condition is restricted to polynomial-time provers. Kilian [Kil92] showed that **NP** has laconic argument systems if strong collision-resistant hash functions exist. Specifically, under a strong enough (but still plausible) assumption, **NP** has *public-coin* arguments in which the verifier's randomness and the communication in both directions is *polylogarithmic*. Combined with [GH98], this provides a strong separation between the efficiency of arguments versus interactive proofs for **NP**. Our results extend this separation to the case that only the prover-to-verifier communication is counted (and the interactive proof is not required to be public coin).

1.4. Organization and techniques. In Section 2 we recall some relevant definitions, notations and results of prior work. Using these notations, in Section 3, we state our results and compare them to prior work. Directions for further research are suggested in Section 9.

In Sections 4 and 5, we study laconic provers who send only one message (or even a single bit). The main technical contribution of these sections is a sequence of reductions among various forms of proof systems with the end result being a statistical zero-knowledge proof system.

In Section 6 we consider laconic provers of perfect completeness. We reduce the analysis of such proof systems to a classical result in game theory.

The main result of this paper is proven in Section 7. The main technical contribution is a proof system for proving a (quite tight) lower bound on the sum of exponentially many (*e.g.,* 2^n) quanities, where each quantity is easily verifiable. The basic idea is to cluster these quantities according to their approximate magnitude, randomly select a few clusters (with probability proportional to the cluster's "weight") and sample each selected cluster via an adequate protocol. We stress that the novelty of this proof system is in establishing quite tight lower bounds (e.g., tight up to a factor of $1 \pm o(1)$) rather than lower bounds that may be off by a much larger factor (*e.g.,* a factor of n).

In Section 8, we present a message complexity hierarchy (based on a reasonable conjecture regarding #SAT). The result follows immediately from refined versions of known results; specifically, the interactive proof for #SAT of Shamir [Sha92] (following [LFKN92]) and the Speedup Theorem of Babai and Moran [BM88].

For sake of self-containment and clarity, we provide detailed analysis of refined variants of two known results: the set sampling/lower-bound protocol of [BM88, GS89, AH91] and the Speedup Theorem of [BM88]. We beleive that these refined variants, provided in Appendices A and B, may be useful for future reference.

2. Preliminaries

We assume that the reader is familiar with the basic concepts underlying interactive proofs (and public-coin interactive proofs); see, *e.g.,* [Sip97, Gol99, Vad00]. Throughout, we work with interactive proofs for *promise problems* rather than languages. A promise problem $\Pi = (\Pi_Y, \Pi_N)$ is a pair of disjoint sets of strings, corresponding to YES and NO instances, respectively. In other words, a promise problem is simply a decision problem in which some inputs are excluded. The definition of interactive proofs is extended to promise problems in the natural way: we require that when the input is a YES instance, the prover convinces the verifier to accept with high probability (*completeness*); and when the input is a NO instance, the verifier accepts with low probability no matter what strategy the prover follows (*soundness*). Working with promise problems rather than languages only makes our results stronger (except for one direction of Theorem 4.6).

2.1. Notation. We denote by $\mathbf{IP}(b, m)$ (resp., $\mathbf{AM}(b, m)$) the class of problems having interactive proofs (resp., public-coin interactive proofs) in which the prover sends a total of at most b bits, and the total number of messages exchanged (in both directions) is at most m. Note that b and m are integer functions of the common input length, denoted n. When b is not polynomial in n, it will be understood that we talk of a generalization in which the verifier is allowed time polynomial in b and n (rather than just in n). Unless specified differently, we refer to proof systems with completeness probability-bound $2/3$ and soundness probability-bound $1/3$.

We denote $\mathbf{IP}(b) = \mathbf{IP}(b, 2b)$; that is, making only the trivial bound on the number of messages exchanged. We denote by $\mathbf{IP}^+$ the analogue of $\mathbf{IP}$ when the proof system has perfect completeness (*i.e.,* completeness probability 1). The class of problems with constant-round interactive proofs is denoted $\mathbf{AM} \stackrel{\text{def}}{=} \mathbf{AM}(\text{poly}(n), 2) = \mathbf{IP}(\text{poly}(n), O(1))$. (The second equality is by Theorems 2.3 and 2.4 below.) When we wish to specify the completeness probability-bound $c = c(n)$ and soundness probability-bound $s = s(n)$ we will use subscripts: $\mathbf{IP}_{c,s}$ and $\mathbf{AM}_{c,s}$. Unless otherwise specified, we always assume that $c(n) > s(n) + 1/\text{poly}(n)$.

Comments: Our notations are not universally acceptable. In some other works $\mathbf{IP}(r)$ denotes $\mathbf{IP}(\text{poly}(n), r)$. Furthermore, whereas the definition $\mathbf{AM} \stackrel{\text{def}}{=} \mathbf{AM}(\text{poly}(n), 2)$ is standard, the class $\mathbf{IP}$ is typically defined as $\mathbf{IP}(\text{poly}(n), \text{poly}(n))$. Finally, sometimes (e.g., see above) we refer to hierarchies such as $\mathbf{IP}(\cdot, \cdot)$ by the notation $\mathbf{IP}$.

2.2. Interactive proofs with bounded interaction. Using the above notations, we recall some results that are relevant to our study.

Our starting point. The main results of Goldreich and Håstad are the starting point (and point of reference) for our work.

THEOREM 2.1 ([GH98]). $\mathbf{AM}(b, m) \subseteq \mathbf{BPTIME}(\mathrm{poly}(2^b, m^m, n))$.

THEOREM 2.2 ([GH98]). $\mathbf{IP}(b, m) \subseteq \mathbf{BPTIME}(\mathrm{poly}(2^b, m^m, n))^{\mathbf{NP}}$.

Theorem 2.1 is stated merely for sake of perspective. Our results relate to and improve upon Theorem 2.2 (which relates to general interactive proofs rather than to public-coin ones). We stress that the transformation from general interactive proofs to public-coin ones (see Theorem 2.4) does not preserve the total number of bits sent by the prover. In fact, very laconic provers (i.e., in which the prover sends a single bit) are known for several problems that are widely believed not to be in **BPP**. (Examples of such problems include QUADRATIC NONRESIDUOSITY [GMR89], GRAPH NONISOMORPHISM [GMW91], and the DISCRETE LOGARITHM PROBLEM [GK93].)

Results used. We will use some (parametrized) extensions of known results. Except for the second inclusion in Theorem 2.3 (which is justified in Appendix B), all the extensions (or parameterized versions) are straightforward from the corresponding original work.

THEOREM 2.3 (Speedup Theorem [BM88]).

$$\mathbf{AM}(b, m) \subseteq \mathbf{AM}(b^2 \cdot \mathrm{poly}(m), \lceil m/2 \rceil) \subseteq \mathbf{AM}((b \cdot m)^{O(m)}, 2)\,.$$

THEOREM 2.4 (AM emulation of IP [GS89]). $\mathbf{IP}(b, m) \subseteq \mathbf{AM}(\mathrm{poly}(b, n), m+2)$.

THEOREM 2.5 ([BHZ87]). *If* $\mathbf{coNP} \subseteq \mathbf{AM}(b, 2)$, *then* $\mathbf{\Sigma_2} \subseteq \mathbf{\Pi_2}(\mathrm{poly}(n, b))$. *In particular, if* $\mathbf{coNP} \subseteq \mathbf{AM}$, *then the polynomial-time hierarchy collapses to* $\mathbf{PH} = \mathbf{\Sigma_2} = \mathbf{\Pi_2}$.

Above and throughout the paper, $\mathbf{\Sigma_i}(t(n))$ (resp., $\mathbf{\Pi_i}(t(n))$) denotes the class of problems accepted by $t(n)$-time alternating Turing machines with i alternations beginning with an existential (resp., universal) quantifier. Thus, $\mathbf{\Sigma_i} \stackrel{\mathrm{def}}{=} \mathbf{\Sigma_i}(\mathrm{poly}(n))$ and $\mathbf{\Pi_i} \stackrel{\mathrm{def}}{=} \mathbf{\Pi_i}(\mathrm{poly}(n))$ comprise the i^{th} level of the polynomial-time hierarchy.

2.3. Statistical Zero Knowledge (SZK). We will also consider **SZK**, the class of problems possessing statistical zero-knowledge interactive proofs. Rather than reviewing the definition of **SZK** here, we will use a recent characterization of **SZK** in terms of complete problems. For random variables X and Y, let $\Delta(X,Y)$ denote their *statistical difference* (or *variation distance, i.e.,* $\Delta(X,Y) = \max_S |\Pr[X \in S] - \Pr[Y \in S]|$, which equals $\frac{1}{2} \cdot \sum_z |\Pr[X = z] - \Pr[Y = z]|$). We consider distributions (or random variables) specified by ("sampling") circuits; that is, a circuit with m input gates and n output gates can be viewed as a sampling algorithm for the distribution on $\{0,1\}^n$ that is induced on the output gates by evaluating the circuit on m random input bits. Statistical Difference is the promise problem $\mathrm{SD} = (\mathrm{SD}_Y, \mathrm{SD}_N)$, where

$$\begin{aligned} \mathrm{SD}_Y &= \{(X,Y) \,:\, \Delta(X,Y) \geq 2/3\} \\ \mathrm{SD}_N &= \{(X,Y) \,:\, \Delta(X,Y) \leq 1/3\}\,. \end{aligned}$$

Recall that X and Y represent probability distributions specified by corresponding sampling circuits (*i.e.,* circuits C_1 and C_2 such that evaluating C_1 (resp., C_1) on the uniform distribution yields a random variable distributed as X (resp., Y)). More generally, for any $1 \geq \alpha > \beta \geq 0$, we will consider variants $\mathrm{SD}^{\alpha,\beta}$, where the thresholds of $2/3$ and $1/3$ are replaced with α and β respectively.

Theorem 2.6 (Complete Problem for **SZK** [SV03]). *For any constants* $1 > \alpha^2 > \beta > 0$, *the problem* $\mathrm{SD}^{\alpha,\beta}$ *is complete for* **SZK**. *That is,* $\mathrm{SD}^{\alpha,\beta}$ *is in* **SZK**, *and every problem in* **SZK** *is reducible to* $\mathrm{SD}^{\alpha,\beta}$ *via a polynomial-time (many-one) reduction.*

Thus, instead of placing certain problems in the class **SZK** (resp., showing that **SZK** has certain interactive proofs), we may reduce these problems to SD (resp., show that SD has such an interactive proof). Indeed, we use the fact that **SZK** is closed under many-one reductions [SV03].

Other results used. The following results about **SZK** are also relevant to us:

Theorem 2.7 ([For89, AH91]). $\mathbf{SZK} \subseteq \mathbf{AM} \cap \mathbf{coAM}$.

Theorem 2.8 ([Oka00]). **SZK** *is closed under complement.*

THEOREM 2.9 ([SV03]). $\mathbf{SZK} \subseteq \mathbf{IP}_{1-2^{-n},1/2}(1)$.

2.4. Probabilistically Checkable Proofs (PCP). As stated in the introduction, some of our results can be viewed in terms of probabilistically checkable proofs. Loosely speaking, a probabilistically checkable proof system consists of a probabilistic polynomial-time verifier having access to an oracle which represents a proof in redundant form. Typically, the verifier accesses only few of the oracle bits, and these bit positions are determined by the outcome of the verifier's coin tosses. For completeness and soundness bounds c and s, it is required that the verifier accepts any YES instance x with probability at least $c(|x|)$ (i.e., when given access to an adequate oracle), whereas it accepts any NO instance x with probability at most $s(|x|)$ no matter which oracle is used. Whenever this holds and if the verifier uses at most $r(|x|)$ random bits and makes at most $q(|x|)$ boolean queries, we say that the problem is in $\mathbf{PCP}_{c,s}(r, q)$. For logarithmically bounded q, we will also say that the problem has amortized query complexity $\frac{q}{\log_2(c/s)}$, and denote the class of problems having amortized query complexity $\overline{q}$ (and randomness complexity r) by $\overline{\mathbf{PCP}}(r, \overline{q})$. (For further discussion of these notions, see [BGS98].) It will be interesting to contrast our results with the following known results:

THEOREM 2.10 (Sec. 10.2 in [BGS98]).

(i) $\mathbf{PCP}_{c,s}(\mathrm{poly}(n), 1) \subseteq \mathbf{AM}$, *for any functions* c, s.

(ii) $\overline{\mathbf{PCP}}(O(\log n), 1-\epsilon) \subseteq \mathbf{P}$, *for every constant* $\epsilon > 0$.

We also consider free-bit complexity of PCP systems. Loosely speaking, here one distinguishes queries for which the verifier compares the answer against a value determined by previously obtained answers, from queries in which the verifier only records the answer for future usage. The latter queries are called *free* (as the "acceptable answers" to them are not determined). By $\mathbf{FPCP}_{c,s}(r, f)$ we denote the class of problems having a $\mathbf{PCP}_{c,s}(r, q)$-system in which at most $f \leq q$ queries are free. (Actually, the definition of free-bit complexity also requires polynomial-time reconstruction of the 2^f acceptable answers.)

THEOREM 2.11 (Sec. 10.3 in [BGS98]). $\mathbf{FPCP}_{1,s}(\mathrm{poly}(n), 0) \subseteq \mathbf{coNP}$, *for any function* $s < 1$.

3. Formal Statement of Results

We improve over Theorem 2.2, and address most of the open problems suggested in [GH98, Sec. 3]. Our main results are listed below.

3.1. On provers that send only one bit. For one bit of prover-to-verifier communication, we obtain a collapse to **SZK**.

THEOREM 3.1. *For every pair of constants c, s such that $1 > c^2 > s > c/2 > 0$, $\mathbf{IP}_{c,s}(1) = \mathbf{SZK}$.*

Viewed in terms of PCP systems, this says that $\mathbf{PCP}_{c,s}(\text{poly}(n), 1) = \mathbf{SZK}$, for any $1 > c^2 > s > c/2 > 0$. For this range of c and s, the latter improves over the bound provided by Part i of Theorem 2.10. Combining Theorems 3.1 and 2.8, we get:

COROLLARY 3.2. *For every c, s as in Theorem 3.1, $\mathbf{IP}_{c,s}(1)$ is closed under complement.*

Theorem 3.1 can be generalized to non-constant completeness and soundness bounds as follows.

THEOREM 3.3. *For every constant $\delta > 0$, and every pair of functions c, s such that $c(n)^{2+\delta} > s(n)$, $\mathbf{IP}_{c,s}(1) \subseteq \mathbf{SZK}$. In fact, this holds even for non-constant $\delta = \Omega(1/\log n)$.*

3.2. On provers that send only one message. We are sometimes able to reduce proof systems with a laconic prover that sends a single message to the above case (of provers that send only one bit).

THEOREM 3.4. *For every $b = b(n) = O(\log n)$, $c = c(n)$, and $s = s(n)$ satisfying $s < 2^{-b/2}$, $\mathbf{IP}_{c,s}(b, 2) \subseteq \mathbf{IP}_{c,s'}(1)$ where $s' = 1 - \exp\left(-O\left(s2^b/(1-s^2 2^b)^2\right)\right)$.*

Applying Theorem 3.3, this gives:

COROLLARY 3.5. $\mathbf{IP}_{c,s}(b, 2) \subseteq \mathbf{SZK}$, *provided the following conditions hold:*

(i) $b = O(\log n)$ *and* $s < 2^{-b/2}$.

(ii) $s2^b/(1-s^2 2^b)^2 = O(\log n)$.

(iii) $c > 1 - \exp\left(-\kappa s 2^b/(1-s^2 2^b)^2\right)$, *where κ is a universal constant.*

(Condition (ii) guarantees that $1 - s' = \exp(-O(s2^b/(1-s^2 2^b)^2))$ is at least $1/\text{poly}(n)$, and Condition (iii) guarantees that $c^2 > s'$.) In particular, the above conditions are satisfied in the following two cases:

1. $b = O(\log n)$, $s = O(2^{-b}) = 1 - \Omega(1)$ and $c = 1 - o(1)$.

2. $b \leq 2\log_2 \log_2 n$, $s = (1-\Omega(1)) \cdot 2^{-b/2}$, and $c \geq 1 - \exp(-\omega(2^{b/2}))$.

Viewed in terms of PCP systems, the above results refer to a generalization of PCP in which non-boolean queries are allowed. Specifically, the above results refer to a PCP system in which a single query is made and is answered by a b-bit long string. The amortized query complexity of such a scheme may be viewed as $\frac{b}{\log_2(c/s)}$, and so the setting in Item 2 asserts that 1-query PCP with polynomial randomness, constant (or even double-logarithmic) answer size, perfect completeness, and amortized query complexity below 2 is in **SZK**. This is slightly related to Part ii of Theorem 2.10 that refers to amortized query complexity below 1 but in a different PCP model (which, on one hand, allows many Boolean queries and arbitrary completeness bound, but on the other hand allows only logarithmic randomness).

3.3. On provers that send a bounded number of bits. For more bits of communication, we first obtain the following result for interactive proofs with perfect completeness (denoted by $\mathbf{IP}^+$):

THEOREM 3.6. $\mathbf{IP}^+(b) \subseteq \mathbf{coNTIME}(2^b \cdot \mathrm{poly}(n))$. *In particular,* $\mathbf{IP}^+(O(\log n)) \subseteq \mathbf{coNP}$.

In the general case (*i.e.,* with imperfect completeness), we prove:

THEOREM 3.7. $\mathbf{IP}(b, m) \subseteq \mathbf{coAM}(2^b \cdot \mathrm{poly}(m^m, n), O(m))$. *In particular,* $\mathbf{IP}(O(\log n), m) \subseteq \mathbf{coAM}(\mathrm{poly}(n), O(m))$, *for* $m = O(\log n / \log\log n)$,

The above theorems provide first evidence that **NP**-complete problems cannot have interactive proof systems in which the prover sends very few bits. Further evidence toward this claim is obtained by applying Theorems 2.3 and 2.5:

COROLLARY 3.8. $\mathbf{IP}(b, m) \subseteq \mathbf{coAM}(\mathrm{poly}(2^b, m^m, n)^m, 2)$. *In particular,* $\mathbf{IP}(O(\log n), O(1)) \subseteq \mathbf{coAM}$ *and* $\mathbf{IP}(\mathrm{polylog}\, n) \subseteq \widetilde{\mathbf{coAM}}$.

COROLLARY 3.9. $\mathbf{NP} \not\subseteq \mathbf{IP}(O(\log n), O(1))$ *unless the polynomial-time hierarchy collapses (to* $\mathbf{\Sigma_2} = \mathbf{\Pi_2}$*).* $\mathbf{NP} \not\subseteq \mathbf{IP}(\mathrm{polylog}\, n)$ *unless* $\mathbf{\Sigma_2} \subseteq \widetilde{\mathbf{\Pi}}_2$.

Above, $\widetilde{\mathbf{coAM}}$ and $\widetilde{\mathbf{\Pi}}_2$ denote the quasipolynomial-time ($2^{\mathrm{polylog}\, n}$) analogues of **coAM** and $\mathbf{\Pi_2}$.

3.4. On provers that send a bounded number of messages. Finally, we mention our result on message complexity. (A more precise statement is contained in Section 8.)

Theorem 3.10. *Let $m(n) \leq n/\log n$ be any "nice" growing function and suppose that $\#\mathrm{SAT} \notin \mathbf{AM}(2^{o(n)}, 2)$. Then $\mathbf{AM}(\mathrm{poly}(n), m(n)) \neq \mathbf{AM}(\mathrm{poly}(n), o(m(n)))$.*

Theorem 3.10 asserts that a stronger round-reduction than the one provided by the Speedup Theorem of [BM88] (*i.e.,* Theorem 2.3) is unlikely, unless one can present an unexpectedly-short two-message proof for non-satisfiability. Note that, by Theorem 2.4, it is irrelevant whether we use **IP** or **AM** in Theorem 3.10.

4. On Extremely Laconic Provers (Saying Only One Bit)

In this section, we prove Theorem 3.1 (*i.e.,* relate $\mathbf{IP}(1)$ and **SZK**). The proof is based on the following lemma, along with previous results.

Lemma 4.1. *For every two constants c, s, every problem in $\mathbf{IP}_{c,s}(1)$ reduces to $\mathrm{SD}^{c,s}$.*

Recall that $\mathrm{SD}^{c,s}$ is the promise problem Statistical Difference (as defined in Section 2.3).

Proof. Let (P, V) be an interactive proof for some problem so that the prover sends a single bit during the entire interaction. Thus, on input x and internal coin tosses r, the verifier first sends a message, denoted $y = V_x(r)$, the prover answers with a bit, denoted $\sigma \in \{0,1\}$, and the verifier decides whether to accept or reject by evaluating the predicate $V_x(r, \sigma) \in \{0,1\}$.

To demonstrate the main idea, we consider first the natural case in which for every pair (x, r) there exists *exactly one* σ such that $V_x(r, \sigma) = 1$. (Note that otherwise the interaction on input x and verifier's internal coin tosses r is redundant, because the verifier's final decision is unaffected by it.) Actually, the lemma is proved by reducing the general case to this special case (see Claim 4.4) and treating the special case (Claim 4.2).

The special case — unique (acceptable) answers. A proof system is said to have unique answers if for every pair (x, r) there exists a single σ such that $V_x(r, \sigma) = 1$.

Claim 4.2. *If a problem has an $\mathbf{IP}_{c,s}(1)$ proof system with unique answers, then it reduces to $\mathrm{SD}^{2c-1,2s-1}$.*

Note that the hypothesis can be satisfied only if $s \geq 1/2$.

PROOF. Let $\sigma_x(r)$ denote the unique σ satisfying $V_x(r, \sigma) = 1$. The prover's ability to convince the verifier is related to the amount of information regarding $\sigma_x(r)$ that is revealed by $V_x(r)$. For example, if for some x and random r, the value of $\sigma_x(r)$ is determined by $V_x(r)$ then the prover can convince the verifier to accept x with probability 1 (by replying with $\sigma_x(r)$). If, on the other hand, for some x and random r, the value of $\sigma_x(r)$ is statistically independent of $V_x(r)$ (and unbiased), then there is no way for the prover to convince the verifier to accept x with probability higher than $1/2$. This suggests the reduction $x \mapsto (C_x^1, C_x^2)$, where $C_x^1(r) \stackrel{\text{def}}{=} (V_x(r), \sigma_x(r))$ and $C_x^2(r) \stackrel{\text{def}}{=} (V_x(r), \overline{\sigma_x}(r))$, where $\overline{b}$ denotes the complement of a bit b.

Now we relate the statistical difference between the distributions sampled by C_x^1 and C_x^2 to the maximum acceptance probability of the verifier. Since the first components of C_x^1 and C_x^2 are distributed identically, their statistical difference is exactly the average over the first component $V_x(r)$ of the statistical difference between the second components conditioned on $V_x(r)$. That is,

$$\Delta(C_x^1, C_x^2) = \mathop{\mathrm{E}}_{y \leftarrow V_x} \left[\Delta\left(\sigma_x|_y, \overline{\sigma_x}|_y\right)\right],$$

where $\sigma_x|_y$ denotes the distribution of $\sigma_x(r)$ when r is uniformly distributed among $\{r' : V_x(r') = y\}$. For any y and $b \in \{0, 1\}$, let $q_{b|y}$ denote the probability that $\sigma_x|_y = b$. Then, for any fixed y, $\Delta\left(\sigma_x|_y, \overline{\sigma_x}|_y\right) = |q_{1|y} - q_{0|y}| = 2q_y - 1$, where $q_y \stackrel{\text{def}}{=} \max_{b \in \{0,1\}}\{q_{b|y}\} \geq \frac{1}{2}$. So, we have:

$$\Delta(C_x^1, C_x^2) = \mathop{\mathrm{E}}_{y \leftarrow V_x} [2q_y - 1].$$

On the other hand, the optimal prover strategy in (P, V) is: upon receiving y, respond with b that maximizes $q_{b|y}$. When the prover follows this strategy, we have

$$\Pr[V \text{ accepts } x] = \mathop{\mathrm{E}}_{y \leftarrow V_x} [q_y].$$

Putting the last two equations together, we conclude that $\Delta(C_x^1, C_x^2) = 2 \cdot \Pr[V \text{ accepts } x] - 1$. (Recall that under the hypothesis of the special case, for every x the prover may convince the verifier to accept x with probability at least $1/2$ (and so such a proof system must have soundness bound at least $1/2$).) Thus if the proof system has completeness and soundness bounds c and s, respectively, then the reduction maps instances to pairs having distance bounds $2c - 1$ and $2s - 1$, respectively. This establishes Claim 4.2. ■

Remark 4.3. *The result of Claim 4.2 can be reversed in the sense that* $\mathrm{SD}^{2c-1,2s-1} \in \mathbf{IP}_{c,s}(1)$. *Specifically, consider the interactive proof system for* $\mathrm{SD}^{\alpha,\beta}$ *in which the verifier selects at random a single sample from one of the two distributions and asks the prover to guess which of the distributions this sample came from. If the distributions are at distance* δ *then the prover succeeds with probability* $\frac{1}{2}+\frac{\delta}{2}$. *Thus, applying this proof system to* $\mathrm{SD}^{2c-1,2s-1}$, *we obtain completeness and soundness bounds* $\frac{1}{2}+\frac{2c-1}{2}=c$ *and* $\frac{1}{2}+\frac{2s-1}{2}=s$, *respectively.*

The general case. We now proceed to deal with the general case in which there may exist pairs (x,r) so that either both σ's or none of them satisfy $V_x(r,\sigma)=1$. We do so by reducing the general case to the special case.

Claim 4.4. *If a problem is in* $\mathbf{IP}_{c,s}(1)$, *then it has an* $\mathbf{IP}_{(1+c)/2,(1+s)/2}(1)$ *proof system with unique answers.*

Clearly, Lemma 4.1 follows by combining Claims 4.2 and 4.4.

Proof. Let (P,V) be a general $\mathbf{IP}_{c,s}$ proof system. Consider the following modified verifier strategy, denoted V'.

1. Generate coin tosses r for the original verifier V.
2. Depending on the number j of possible prover responses σ for which $V_x(r,\sigma)=1$, proceed as follows:

   Case $j=2$: Send the prover a special "`respond with 1`" message, and accept if and only if the prover responds with 1.

   Case $j=1$: Randomly do one of the following (each with probability 1/2):
   - Send the prover $y=V_x(r)$ and accept if and only if the prover responds with the unique σ such that $V_x(r,\sigma)=1$.
   - Send the prover a special "`respond with 1`" message, and accept if and only if the prover responds with 1.

   Case $j=0$: Choose a random bit σ. Send the prover a special "`guess my bit`" message, and accept if and only if the prover responds with σ.

For all possible choices of the coin tosses of V', there is exactly one prover response that will make V' accept. Hence V' satisfies the conditions of the special case. To establish Claim 4.4, we show that if an optimal prover makes

V accept with probability δ, then an optimal prover makes V' accept with probability $(1+\delta)/2$. To see this, observe that an optimal prover strategy P' for V' consists of always responding "1" to the special messages, and otherwise responding as an optimal prover P for V. It can be verified by inspection that, conditioned on each value of j, if P makes V accept with probability δ_j, then P' makes V' accept with probability $(1+\delta_j)/2$. (That is, δ_j is the probability that V accepts when interacting with an optimal prover, conditioned on V selecting a random r for which there are j accepting answers (i.e., $j = |\{\sigma : V_x(r,\sigma) = 1\}|$). Indeed, $\delta_0 = 0$, $\delta_2 = 1$, and $\delta_1 \geq 1/2$.) ■

Combining Claims 4.2 and 4.4, the lemma (i.e., Lemma 4.1) follows. Specifically, by Claim 4.4, any problem in $\mathbf{IP}_{c,s}(1)$ has a unique-answer (1-bit) interactive proof with completeness and soundness bounds $c' = (1+c)/2$ and $s' = (1+s)/2$, respectively. By Claim 4.2, the latter interactive proof system implies that the problem is reducible to $\mathrm{SD}^{2c'-1,2s'-1} = \mathrm{SD}^{c,s}$ (since $2c'-1 = (1+c)-1 = c$ and $2s'-1 = (1+s)-1 = s$). ■

PROOF OF THEOREM 3.1. Let c and s satisfy the conditions in Theorem 3.1 (which asserts that $\mathbf{IP}_{c,s}(1)$ equals $\mathbf{SZK}$). The inclusion of $\mathbf{IP}_{c,s}(1)$ in $\mathbf{SZK}$ follows by combining Lemma 4.1 and Theorem 2.6: That is, $\mathbf{IP}_{c,s}(1)$ reduces to $\mathrm{SD}^{c,s}$, which (for $1 > c^2 > s > 0$) resides in $\mathbf{SZK}$.
The opposite inclusion (i.e., of $\mathbf{SZK}$ in $\mathbf{IP}_{c,s}(1)$) follows from Theorem 2.9. Specifically, recall that $c < 1$ and $s > c/2$, and let $\epsilon > 0$ be such that $c+\epsilon \leq 1$ and $s \geq (c/2)+\epsilon$. For any problem in $\mathbf{SZK}$, consider a verifier that executes the $\mathbf{IP}_{1-2^{-n},1/2}(1)$ proof system of Theorem 2.9 with probability $c+\epsilon \leq 1$ and otherwise rejects without any interaction. This yields a proof system with completeness $(c+\epsilon)\cdot(1-2^{-n}) > c$ (for sufficiently large n), and soundness $(c+\epsilon)\cdot(1/2) < s$. ■

To generalize the above to non-constant completeness and soundness bounds and prove Theorem 3.3, we use the following transformation.

LEMMA 4.5 (Polarization Lemma [SV03]). *There is an algorithm that takes as input a quintuple (X, Y, α, β, k), where X and Y are random variables specified by circuits and $\alpha^2 > \beta$, and outputs a pair of random variables (X', Y') such that:*

$$\Delta(X,Y) \geq \alpha \;\Rightarrow\; \Delta(X',Y') \geq 1-2^{-k}$$
$$\Delta(X,Y) \leq \beta \;\Rightarrow\; \Delta(X',Y') \leq 2^{-k}.$$

The running time of the algorithm is polynomial in the description of X and Y as well as in the $k \cdot \exp(1/\delta)/(\alpha-\beta)$, where $\delta = \frac{\log\beta}{\log\alpha} - 2 > 0$.

Proof of Theorem 3.3. Let c, s be as in the theorem and consider any problem Π in $\mathbf{IP}_{c,s}(1)$. The proof of Lemma 4.1 shows how from any instance x of Π, we can construct in polynomial time a pair of random variables (X, Y) whose statistical difference is at least $c(|x|)$ (resp., at most $s(|x|)$) when x is a YES instance (resp., NO instance). Applying the Polarization Lemma to (X, Y) with $\alpha = c(|x|)$, $\beta = s(|x|)$, and $k = 2$, gives a reduction from Π to $\mathrm{SD}^{3/4,1/4}$, which is in **SZK**. This reduction is computable in polynomial time because $1/(\alpha - \beta) = 1/(c - s) \leq \mathrm{poly}(|x|)$ (by the definition of **IP**) and $\exp(1/\delta) \leq \mathrm{poly}(|x|)$ since $\delta = \Omega(1/\log|x|)$ (by hypothesis). ■

On the limitations regarding c and s. The $c^2 > s$ constraint in Theorem 3.1 is due to the analogous constraint in Theorem 2.6 (which in turn stems from the limitation in Lemma 4.5). Recall that, for every $1 > \alpha > \beta > 0$, every problem in **SZK** reduces to $\mathrm{SD}^{\alpha,\beta}$ (cf. [SV03]). However, it is not known whether $\mathrm{SD}^{\alpha,\beta}$ is in **SZK** for every $1 > \alpha > \beta > 0$ (rather than for every $1 > \alpha^2 > \beta > 0$ as in Theorem 2.6). In fact, the latter is an intriguing open problem, and we establish its equivalence to a question regarding $\mathbf{IP}_{c,s}(1)$ (for arbitrary $1 > c > s > c/2 > 0$).

Theorem 4.6. *The following hypotheses are equivalent.*

(i) For all α, β such that $1 > \alpha > \beta > 0$, $\mathrm{SD}^{\alpha,\beta}$ is in **SZK**.

(ii) For all constants c, s such that $1 > c > s > c/2 > 0$, $\mathbf{IP}_{c,s}(1) \subseteq$ **SZK**.

Recall that $\mathbf{SZK} \subseteq \mathbf{IP}_{c,s}(1)$, for every c, s such that $1 > c > s > c/2 > 0$. (Note that this was actually established in the above proof of Theorem 3.1, since the actual conditions used were $c < 1$ and $s > c/2$.)

Proof. The direction (i)⇒(ii) is proven in the same way as Theorem 3.1, except that we now use Hypothesis (i) instead of Theorem 2.6: Specifically, $\mathbf{IP}_{c,s}(1)$ reduces to $\mathrm{SD}^{c,s}$ (for every c, s by Lemma 4.1), and Hypothesis (i) asserts that the latter resides in **SZK**.
The direction (ii)⇒(i) is proven by recalling that $\mathrm{SD}^{\alpha,\beta}$ is in $\mathbf{IP}_{(1+\alpha)/2,(1+\beta)/2}(1)$ (see [SV03] and Remark 4.3), which by Hypothesis (ii) is contained in **SZK** (since $(1+\alpha)/2 > (1+\beta)/2 > (1+\alpha)/4$ holds for any $1 > \alpha > \beta > 0$). ■

Finally, we remark that the condition $s > c/2$ in Theorem 3.1 (or, more generally, for $\mathbf{SZK} \subseteq \mathbf{IP}_{c,s}(1)$) seems necessary becuase of the following result.

Proposition 4.7 (*cf.*, [Vad99, Prop 4.1.2]). *For every c, s such that $s < c/2$,* $\mathbf{IP}_{c,s}(1) = \mathbf{BPP}$.

5. On Laconic Provers That Send One Message

In this section, we prove Theorem 3.4, which reduces 2-message proof systems with a laconic prover to proof systems in which the prover sends only one bit. Let (P, V) be an $\mathbf{IP}_{c,s}(b, 2)$ proof system, with $s \leq 2^{-b/2}$. As in Section 4, we may assume that on input x and internal coin tosses r, the verifier first sends a message $y = V_x(r)$, the prover answers with a string $z \in \{0,1\}^b$, and the verifier decides whether to accept or reject by evaluating the predicate $V_x(r, z) \in \{0, 1\}$. We obtain a new proof system (P', V') by randomly "hashing" the prover's responses to one bit.

Construction 5.1 (Modified Proof System (P', V')). *On input x, the parties behave as follows:*

(i) V': Choose r uniformly, and let $y = V_x(r)$. Choose a random function $h : \{0,1\}^b \to \{0,1\}$. Send y and h to P'.

(ii) P': Let $z = P(x, y)$, and $\sigma = h(z)$. Send σ to V'.

(iii) V': Accept if there exists a $w \in \{0,1\}^b$ such that $h(w) = \sigma$ and $V_x(r, w) = 1$.

Clearly, the prover-to-verifier communication of (P', V') is one bit, and the verifier's program can be implemented $\text{poly}(n) \cdot 2^b$. Also, it is clear that the modified prover can convince the modified verifier to accept any input with probability that is lower bounded by the corresponding probability in the original proof system. Our focus is thus on analyzing the soundness of the modified proof system.

The basic intuition is that the impossibility of determining a good prover answer for the verifier's message y in (P, V) means that it is hard to predict the hash-value of such a good answer (under a random hash function). This intuition is very clear in case the original system has unique acceptable answers, but it holds also in general. Specifically, consider a typical message y, and two random r_i's that may lead to it (i.e., $y = V_x(r_i)$). Assuming that x is a NO-instance, in case of unique acceptable answers, it is likely that the (unique) good answer for r_1 differs from the (unique) good answer for r_2, and furthermore (with probability 1/2) these different good answers have different hash-values under a random hash function. This contributes to the rejection probability of $V'(x)$. In the general case, when x is a NO-instance, it is unlikely that the set of good answers for r_1 has a non-empty intersection with the set of good answer for r_2 (or else P could make V accept). Furthermore, with positive probability (which is exponential in the cardinality of these sets), a random hash function maps the two sets to different values, which contributes to the rejection probability of $V'(x)$. Lastly, the expected cardinality of these sets can be upper-bounded (by $s2^b$, due to the soundness of V). The actual analysis follows, where we first handle the (easy) special case of unique acceptable answers.

5.1. Acceptance probabilities — unique answers. Even more than in Section 4, it is illuminating to first analyze the natural special case of *unique answers*. That is, we assume that for every pair (x, r) there exists *exactly one* z such that $V_x(r, z) = 1$. We refer to the proof system (P', V') as derived by Construction 5.1.

CLAIM 5.2. *If (P, V) has unique answers, then (P', V') has completeness $c' = (1 + c)/2$ and soundness $s' = (1 + \sqrt{s})/2$. Moreover, (P', V') also has unique answers.*

Note that $c' \geq c$ and $1 - s' \geq \frac{1-s}{4}$. On the other hand, $s' < c'$ if and only if $s < c^2$.

PROOF. We start by establishing the completeness bound, letting x be an arbitrary YES-instance. Note that whenever z succeeds in making V accept, it is the case that $b = h(z)$ succeeds in making V' accept. (That is, if V accepts z on coins r then V' accepts $b = h(z)$ on coins (r, h), for any h.) On the other hand, if V does not accept z on coins r, then V' accepts $b = h(z)$ on coins (r, h) with probability $1/2$ for a uniformly chosen h. Specifically, V' accepts $b = h(z)$ on coins (r, h) if the unique $w \neq z$ that is accepted by V on coins r satisfies $h(w) = h(z)$. Thus, V' accepts $P'(V'_x(r, h)) \stackrel{\text{def}}{=} h(P(V_x(r)))$ with probability

$$\begin{aligned}
&\Pr_r[V_x(r, P(V_x(r))) = 1] + \frac{1}{2} \cdot \Pr_r[V_x(r, P(V_x(r))) \neq 1] \\
&= \frac{1}{2} \cdot (1 + \Pr_r[V_x(r, P(V_x(r))) = 1]) \\
&\geq \frac{1 + c}{2}.
\end{aligned}$$

This establishes the claimed completeness bound. (We comment that uniqueness of the acceptable answer was not important above; what we actually need and use is that for every r there exists a w such that $V_x(r, w) = 1$.)

Establishing the soundness bound is (as usual) more involved. We fix an arbitrary NO-instance x (which we will hereafter drop from the notation). For a

V message y and a P response z, let $q_{z|y}$ denote the probability that V accepts the prover response z given that V's message is y. That is,

$$q_{z|y} \stackrel{\text{def}}{=} \frac{|\{r : V(r) = y \text{ and } V(r,z) = 1\}|}{|\{r : V(r) = y\}|}.$$

The optimal prover strategy (for convincing V) is to respond with z that maximizes the above probability, and this strategy succeeds with probability $q_y = \max_z\{q_{z|y}\}$. By the soundness (of V), we have $\mathrm{E}_y[q_y] \le s$, where here and below the distribution of y is as induced by V_x (when applied to a random r).

Similarly, for a V' message (y,h) and a P' response $\sigma \in \{0,1\}$, let $q'_{\sigma|y,h}$ denote the probability that V' accepts the prover response σ given that V' sent (y,h). Using the unique answers hypothesis, observe that the prover response 0 makes V' accept iff the response 1 makes V' reject. Thus, $q'_{0|y,h} = 1 - q'_{1|y,h}$. It follows that the optimal strategy (for convincing V') succeeds with probability

$$q'_{y,h} = \max\{q'_{0|y,h}, q'_{1|y,h}\} = \frac{1}{2} + \left| q'_{0|y,h} - \frac{1}{2} \right|.$$

We will now relate $s' = \mathrm{E}_{y,h}[q'_{y,h}]$ to $\mathrm{E}_y[q_y] \le s$. Using the unique answers hypothesis, note that $q'_{0|y,h} = \sum_{z \in h^{-1}(0)} q_{z|y} = \sum_z q_{z|y} \cdot \chi_z(h)$, where $\chi_z(h)$ is a random variable (defined over the space of h's) indicating the event $h(z) = 0$. Over the choice of the totally random function h, the χ_z's are independent random variables, each with expectation $1/2$ and variance $1/4$. Thus,

$$\mathop{\mathrm{E}}_h \left[q'_{0|y,h}\right] \;=\; \mathop{\mathrm{E}}_h \left[\sum_z q_{z|y} \cdot \chi_z(h)\right] \;=\; \sum_z q_{z|y} \cdot \frac{1}{2} \;=\; \frac{1}{2}.$$

and

$$\begin{aligned} \mathrm{Var}_h \left[q'_{0|y,h}\right] &= \mathrm{Var}_h \left[\textstyle\sum_z q_{z|y} \cdot \chi_z(h)\right] \;=\; \textstyle\sum_z q_{z|y}^2 \cdot \frac{1}{4} \\ &\le \tfrac{\max_z\{q_{z|y}\}}{4} \cdot \textstyle\sum_z q_{z|y} \;=\; \tfrac{q_y}{4}\,. \end{aligned} \tag{5.3}$$

Combining (5.3) with the fact that $\mathrm{E}(X)^2 \le \mathrm{E}(X^2)$ for every random variable X, we get

$$\mathop{\mathrm{E}}_{y,h}\left[\left|q'_{0|y,h} - \frac{1}{2}\right|\right]^2 \;\le\; \mathop{\mathrm{E}}_{y,h}\left[\left|q'_{0|y,h} - \frac{1}{2}\right|^2\right] \;=\; \mathop{\mathrm{E}}_{y}\left[\mathop{\mathrm{Var}}_h\left[q'_{0|y,h}\right]\right] \;\le\; \mathop{\mathrm{E}}_{y}\left[\frac{q_y}{4}\right] \;\le\; \frac{s}{4}.$$

This implies that

$$s' = \mathop{\mathrm{E}}_{y,h}[q'_{y,h}] = \frac{1}{2} + \mathop{\mathrm{E}}_{y,h}\left[\left|q'_{0|y,h} - \frac{1}{2}\right|\right] \le \frac{1+\sqrt{s}}{2}.$$

and the claim follows. ∎

REMARK 5.4. *When $s > 1/2$, the soundness bound stated above can be improved to $(1 + \sqrt{1 - 2s(1 - s)})/2$. This is obtained by replacing (5.3) with* $\mathrm{Var}_h[q'_{0|y,h}] = \frac{1}{4}\sum_z q^2_{z|y} \leq (q_y^2 + (1 - q_y)^2)/4$, *and obtaining* $s' = \mathrm{E}_{y,h}[q'_{y,h}] = \frac{1}{2} + \mathrm{E}_y[\mathrm{Var}_h[q'_{0|y,h}]]^{1/2}$.

REMARK 5.5. *The above analysis only requires the values of the hash function h to be pairwise independent, so V' can restrict its choice of h to any pairwise independent family (e.g., inner product modulo 2 with a random vector). This can eliminate the exponential dependence on b in the running-time of V' if the original protocol has the property that the unique accepting prover response can be computed from V's coin tosses r in polynomial time. As pointed out by an anonymous referee, the modified construction corresponds to having P' send the value of a random location in the Hadamard encoding of the unique acceptable answer, whereas Construction 5.1 corresponds to using the Long-Code (of [BGS98]).*

Combining Claims 5.2 and 4.2, we get

COROLLARY 5.6. *For constants $c > s$, if a problem has an $\mathbf{IP}_{c,s}(O(\log n), 2)$ proof system with unique answers, then it reduces to $\mathrm{SD}^{c,\sqrt{s}}$. Hence, if $c^4 > s$ then this problem is in* **SZK**.

By Remark 5.5, the above extends also to $\mathbf{IP}_{c,s}(\mathrm{poly}(n), 2)$ proof system with unique answers, provided that the unique accepting prover response can be computed in polynomial-time (from the common input and V's coin tosses).

PROOF. By Claim 5.2, for $c' = (1 + c)/2$ and $s' = (1 + \sqrt{s})/2$, the problem has an $\mathbf{IP}_{c',s'}(1, 2)$ proof system with unique answers. By Claim 4.2, any such problem reduces to $\mathrm{SD}^{2c'-1,2s'-1}$. Recalling that $(2c' - 1, 2s' - 1) = (c, \sqrt{s})$, the first claim follows. The second claim follows by Theorem 2.6. ■

REMARK 5.7. *We note that the unique answers property has a "zero-knowledge" flavor. Specifically, consider a simulator that executes the verifier strategy and uses the unique accepting answer as the simulated prover message. The statistical difference between this simulation and the (honest) verifier's view is at most the completeness error $1-c$. If the completeness error is negligible, membership in* **SZK** *follows immediately. Thus, what is interesting about Corollary 5.6 is that it applies even when the completeness error is* constant.

The PCP perspective. Observe that $\mathbf{IP}(b, 2)$ systems with unique answers correspond to PCP systems with zero free-bit complexity in which a single (non-Boolean) query is made and is answered by a b-bit string. Viewed in these terms, Corollary 5.6 asserts that, for $c^4 > s$, $\mathbf{PCP}_{c,s}$ schemes with zero free-bit complexity in which a single (non-Boolean) query is made (and is answered by a logarithmically-long bit string) exist only for problems in $\mathbf{SZK}$. This is slightly related to Theorem 2.11 that refers to arbitrary $\mathbf{PCP}_{1,s}$ schemes with free-bit complexity zero (which are placed in $\mathbf{coNP}$). Note that the hypotheses of the two results are incomparable: here we allow arbitrary $c > s^{1/4}$ but require a single (non-Boolean) query, whereas Theorem 2.11 requires $c = 1$ but allows an arbitrary number of (Boolean) queries.

Generalized Statistical Difference. We consider the following many-distribution version of STATISTICAL DIFFERENCE. For random variables $X_1, ..., X_t$, define

$$D(X_1, ..., X_t) \stackrel{\text{def}}{=} \frac{1}{t} \cdot \sum_x \max\{\Pr[X_1 = x], \Pr[X_2 = x], ..., \Pr[X_t = x]\} \in \left[\frac{1}{t}, 1\right] . \tag{5.8}$$

For $t = 2$, the function D is related to the statistical difference between the two distributions: $\Delta(X, Y) = 2 \cdot D(X, Y) - 1$ (*i.e.,* $D(X, Y) = (1 + \Delta(X, Y))/2$). Furthermore, $D(X_1, ..., X_t)$ is the acceptance probability of the verifier in the following interactive proof system, executed on common input $X_1, ..., X_t$:

1. The verifier selects uniformly $i \in [t]$, generates a sample x from X_i (*i.e.,* $x \leftarrow X_i$), and sends x to the prover.

2. The prover tries to guess i; that is, the optimal prover responds with j such that $\Pr[X_j = x] = \max\{\Pr[X_1 = x], \Pr[X_2 = x], ..., \Pr[X_t = x]\}$.

3. The verifier accepts if and only if $i = j$.

Note that the above ($\mathbf{IP}(\log_2 t, 2)$) interactive proof system has unique answers. Thus applying Corollary 5.6 it follows that, for $\alpha^4 > \beta$, the problem of distinguishing between the case that $D(X_1, ..., X_t) \geq \alpha$ from the case that $D(X_1, ..., X_t) \leq \beta$ is in $\mathbf{SZK}$. That is, for $\alpha^4 > \beta$, the promise problem $\mathrm{GSD}^{\alpha,\beta} = (\mathrm{GSD}_Y^{\alpha}, \mathrm{GSD}_N^{\beta})$ is in $\mathbf{SZK}$, where

$$\begin{aligned} \mathrm{GSD}_Y^{\alpha} &= \{(X_1, ..., X_t) \; : \; D(X_1, ..., X_t) \geq \alpha\} \\ \mathrm{GSD}_N^{\beta} &= \{(X_1, ..., X_t) \; : \; D(X_1, ..., X_t) \leq \beta\} . \end{aligned}$$

5.2. Acceptance probabilities — general case. The following lemma establishes the bounds claimed in Theorem 3.4. Again, we refer to the proof system (P', V') as derived by Construction 5.1 (from a proof system (P, V) having completeness and soundness bounds c and s, which (like b) may be functions of n).

LEMMA 5.9. *The proof system (P', V') has completeness $c' = c$ and soundness $s' = 1 - \exp(-O(s2^b/(1-s^2 2^b)^2))$, provided $s < 2^{-b/2}$.*

PROOF. The completeness bound is established similarly to the way this was done in the unique answer case. It still holds (here) that whenever $z = P(x, y)$ succeeds in making V accept (which happens probability at least c), the answer $\sigma = h(z)$ succeeds in making V' accept. However, since we are not guaranteed here that for every r there exists a w that is acceptable by V_x (i.e, that $V_x(r, w) = 1$), we cannot benefit from the cases in which V does not accept z (but does accept w). Thus, we get a completeness bound of c (rather than $(c + 1)/2$).

For the analysis of the soundness bound, we adopt some of the notation used in the unique answers case: that is, $q_{z|y}$, $q_y = \max_z\{q_{z|y}\}$, $q'_{\sigma|y,h}$ and $q'_{y,h} = \max\{q'_{0|y,h}, q'_{1|y,h}\}$ are as in Claim 5.2. Unlike the unique answers case, it is no longer true that $q'_{y,h} = 1/2 + |q'_{0|y,h} - 1/2|$, because it may be the case that both (or neither) of the answers 0 and 1 make V' accept. Instead, let R_y denote the set of coin tosses (of V) leading to message y, and let $A_r \subseteq \{0,1\}^b$ denote the set of P responses making V accept on coin tosses r. (For a set $S \subseteq \{0,1\}^b$, we let $h(S)$ denote the image of S under h; i.e., $h(S) \stackrel{\text{def}}{=} \{h(s) : s \in S\}$.) Then, for any $\sigma \in \{0, 1\}$ (and any y and h),

$$q'_{\sigma|y,h} = \Pr_{r \in R_y}[\sigma \in h(A_r)],$$

since V' accepts σ if there exists an element of $h^{-1}(\sigma)$ that would make V accept (*i.e.,* is in R_y). Observe that for any fixed y and h

$$\begin{aligned}
\max\{q'_{0|y,h}, q'_{1|y,h}\} &= \max\left\{\Pr_{r \in R_y}[0 \in h(A_r)], \Pr_{r \in R_y}[1 \in h(A_r)]\right\} \\
&\le \Pr_{r_0, r_1 \in R_y}[0 \in h(A_{r_0}) \text{ or } 1 \in h(A_{r_1})] \\
&= 1 - \Pr_{r_0, r_1 \in R_y}[0 \notin h(A_{r_0}) \text{ and } 1 \notin h(A_{r_1})]
\end{aligned}$$

Thus, we can bound the soundness of (P', V') as follows:

$$\begin{aligned} s' &= \mathop{\mathrm{E}}_{y,h}[q'_{y,h}] \\ &= \mathop{\mathrm{E}}_{y,h}\left[\max\{q'_{0|y,h}, q'_{1|y,h}\}\right] \\ &\leq 1 - \Pr_{y,h,r_0,r_1\in R_y}[0 \notin h(A_{r_0}) \text{ and } 1 \notin h(A_{r_1})] \end{aligned}$$

Since h is a total function (from $\{0,1\}^b$ to $\{0,1\}$), the sets A_{r_0} and A_{r_1} must be disjoint in order for both $0 \notin h(A_{r_0})$ and $1 \notin h(A_{r_1})$ (since otherwise $h(A_{r_0} \cap A_{r_1}) \subseteq \{0,1\}$ is non-empty and must contain either 0 or 1). Furthermore, if A_{r_0} and A_{r_1} are disjoint, then the probability (over the choice of h) that both $0 \notin h(A_{r_0})$ and $1 \notin h(A_{r_1})$ occurs is exactly $2^{-|A_{r_0}|} \cdot 2^{-|A_{r_1}|} = 2^{-(|A_{r_0}|+|A_{r_1}|)}$. Thus, for any bound B, we get

$$\begin{aligned} &(5.10)\quad s' \leq 1 - \Pr_{y,h,r_0,r_1\in R_y}[0 \notin h(A_{r_0}) \text{ and } 1 \notin h(A_{r_1})] \\ &\qquad = 1 - \sum_{i=0}^{2^b} \Pr_{y,r_0,r_1\in R_y}[A_{r_0} \cap A_{r_1} = \emptyset \text{ and } |A_{r_0}| + |A_{r_1}| = i] \cdot 2^{-i} \\ &(5.11)\quad\ \ \leq 1 - \Pr_{y,r_0,r_1\in R_y}[A_{r_0} \cap A_{r_1} = \emptyset \text{ and } |A_{r_0}| + |A_{r_1}| \leq B] \cdot 2^{-B} \end{aligned}$$

Thus, we now lower bound the probability, that A_{r_0} and A_{r_1} are disjoint and not too large.

CLAIM 5.12. *Let $\delta = 1 - (s^2 2^b)^{1/3} > 0$. Then*

$$\Pr_{y,r_0,r_1\in R_y}\left[A_{r_0} \cap A_{r_1} = \emptyset \text{ and } |A_{r_0}| + |A_{r_1}| \leq \frac{4s2^b}{\delta^2}\right] \geq \frac{\delta^2}{2}.$$

Observe that $s < 2^{-b/2}$ guarantees that $\delta > 0$. Combining Claim 5.12 with the bound (on s') provided by (5.11), implies that the soundness error of (P', V') is at most

$$1 - \left(\frac{\delta^2}{2}\right) \cdot 2^{-\frac{4s2^b}{\delta^2}} \tag{5.13}$$

Using $\delta = 1 - (s^2 2^b)^{1/3} = \Omega(1 - s^2 2^b)$, we get $(\delta^2/2) \cdot 2^{-\frac{4s2^b}{\delta^2}} \geq 2^{-O(s2^b/(1-s^2 2^b)^2)}$, and Lemma 5.9 follows. Thus, we proceed with the proof of Claim 5.12.

For a fixed y and any z, $\Pr_{r \in R_y}[z \in A_r] = q_{z|y} \leq q_y$ (by definition), and $\Pr_{r_0, r_1 \in R_y}[z \in A_{r_0} \cap A_{r_1}] = q_{z|y}^2 \leq q_y^2$. Thus,

$$\begin{aligned}
\Pr_{r_0,r_1}[A_{r_0} \cap A_{r_1} \neq \emptyset] &= \Pr_{r_0,r_1}[|A_{r_0} \cap A_{r_1}| \geq 1] \\
&\leq \mathop{\mathrm{E}}_{r_0,r_1}[|A_{r_0} \cap A_{r_1}|] \\
&= \sum_z \Pr_{r_0,r_1 \in R_y}[z \in A_{r_0} \cap A_{r_1}] \\
&\leq 2^b \cdot q_y^2.
\end{aligned}$$

Since $\mathrm{E}_y[q_y] \leq s$, we have $\Pr_y[q_y \leq s/(1-\delta)] \geq \delta$. Thus,

$$\begin{aligned}
\Pr_{y,r_0,r_1 \in R_y}[A_{r_0} \cap A_{r_1} = \emptyset] &\geq \Pr_y\left[q_y \leq \frac{s}{1-\delta}\right] \cdot \Pr_{y,r_0,r_1 \in R_y}\left[A_{r_0} \cap A_{r_1} = \emptyset \,\middle|\, q_y \leq \frac{s}{1-\delta}\right] \\
&\geq \delta \cdot \left(1 - \left(\frac{s}{1-\delta}\right)^2 \cdot 2^b\right) \\
&= \delta^2
\end{aligned}$$

where the last equality merely uses $\delta = 1-(s^2 2^b)^{1/3}$. Turning to the complement of the second event, we see that

$$\begin{aligned}
&\Pr_{y,r_0,r_1 \in R_y}\left[|A_{r_0}| + |A_{r_1}| \geq \frac{4s2^b}{\delta^2}\right] \\
&\leq \frac{\delta^2}{4s2^b} \cdot \mathop{\mathrm{E}}_{y,r_0,r_1 \in R_y}[|A_{r_0}| + |A_{r_1}|] \\
&= \frac{\delta^2}{4s2^b} \cdot \sum_{z \in \{0,1\}^b} \left(\Pr_{y,r_0 \in R_y}[z \in A_{r_0}] + \Pr_{y,r_1 \in R_y}[z \in A_{r_1}]\right) \\
&\leq \frac{\delta^2}{4s2^b} \cdot 2^b \cdot 2s = \frac{\delta^2}{2}.
\end{aligned}$$

where the last inequality is due to the soundness of V (which implies, as a very restricted case, that any fixed prover strategy z is accepted with probability at most s). This establishes Claim 5.12, and thereby Lemma 5.9. ■

Proof of Theorem 3.4: For $b = O(\log n)$, given an $\mathbf{IP}_{c,s}(b,2)$ proof system (P,V), we modify it into an $\mathbf{IP}_{c',s'}(1,2)$ proof system (P',V') as in Construction 5.1. By Lemma 5.9 (using $s < 2^{-b/2}$), we have $c' = c$ and $s' = 1 - \exp(-O(s2^b/(1-s^2 2^b)^2))$ as required by Theorem 3.4. ■

6. On Laconic Provers with Perfect Completeness

In this section, we prove Theorem 3.6.

THEOREM (Theorem 3.6, restated). *If a problem* Π *has an interactive proof system with perfect completeness in which the prover-to-verifier communication is at most* $b(\cdot)$ *bits then* $\Pi \in \mathbf{coNTIME}(2^{b(n)} \cdot \mathrm{poly}(n))$.

PROOF. We take a slightly unusual look at the interactive proof system for Π, viewing it as a "progressively finite game" between two players P^* and V^*. Player P^* corresponds to the usual prover strategy and its aim is to make the original verifier accept the common input. Player V^* is a "cheating verifier" and its aim is to produce an interaction that looks legal and still makes the original verifier reject the common input.

To make this precise, let $b = b(n)$ be the bound on the prover-to-verifier communication in the original interactive proof (on inputs of length n), and let $m = m(n)$ be the number of messages exchanged. Without loss of generality, we may assume that V sends all its coin tosses in the last message. A *transcript* is a sequence of m strings, corresponding to (possible) messages exchanged between P and V. We call a transcript t *consistent* (for x) if every verifier message in t is the message V would have sent given input x, the previous messages in t, and the coin tosses specified by the last message in t. We call a consistent t *rejecting* if V would reject at the end of such an interaction.

Now, the game between P^*_x and V^*_x has the same structure as the interaction between P and V on input x: a total of m messages are exchanged and P^*_x is allowed to send at most b bits. The game between P^*_x and V^*_x yields a transcript t. We say that V^*_x *wins* if t is consistent and rejecting, and that P^*_x wins otherwise. We stress that V^*_x need not emulate the original verifier nor is it necessarily implemented in probabilistic polynomial time.

The above constitutes a "perfect information finite game in extensive form" (also known as a "progressively finite game") and Zermelo's Theorem (*cf.*, [Tuc95, Sec. 10.2]) says that exactly one of the two players has a *winning strategy* — that is, a (deterministic) strategy that will guarantee its victory no matter how the other player acts.

Using the perfect completeness condition, we infer that if the common input x is a YES instance (of Π) then there exists a winning strategy for P^*_x. (This is because the optimal prover for the original interactive proof wins whenever V^*_x plays in a manner consistent with some sequence of coin tosses for the original verifier, and it wins by definition if the V^*_x plays inconsistently with any such sequence.) On the other hand, by the soundness condition, if the common input is a NO instance then there exists no winning strategy for P_x^*. (This is because in this case no prover strategy can convince the original verifier with probability 1.) By Zermelo's Theorem, it follows that whenever the common input is a NO instance (of Π) there exists a winning strategy for V_x^*.

Thus, a proof that x is a NO instance (of Π) consists of a winning strategy for V_x^*. Such strategy is a function mapping partial transcripts of P_x^* messages to the next V_x^* message. Thus, such a strategy is fully specified by a function from $\cup_{i=0}^{b}\{0,1\}^i$ to $\{0,1\}^{\mathrm{poly}(n)}$, and has description length $\mathrm{poly}(n) \cdot 2^{b(n)+1}$. To verify that such a function constitutes a winning strategy for V_x^*, one merely tries all possible deterministic strategies for P_x^* (*i.e.*, all possible $b(n)$-bit long strings). The theorem follows. ■

REMARK 6.1. *As pointed out by an anonymous referee, Theorem 3.6 can be proven without reference to game theory, however we feel that the game theoretic proof is more insightful. The alternative proof is based on considering the quantified boolean formula that represents the (perfect completeness) acceptance criterion of the original proof system. Next, one observes that negating this formula yields a sequence of polynomially-many Boolean quantifiers with at most b universal quantifiers. Thus, a proof that x is a* NO*-instance consists of an adequate sequence of 2^b assignments to all existentially-quantified variables, where the simplest way of formulating the notion of an adequate sequence is via a b-move game (or a tree of depth b).*

7. On General Laconic Provers

In this section, we prove Theorem 3.7 (*i.e.*, $\mathbf{IP}(b,m) \subseteq \mathbf{coAM}(2^b \cdot \mathrm{poly}(m^m, n), O(m))$). That is, for any problem that has a laconic interactive proof, we will construct an interactive proof of few rounds for its complement.

Conventions. Let (P,V) be an interactive proof for Π so that, on common input x, the prover sends a total of at most $b(|x|)$ bits, and the total number of messages exchanged (in both directions) is at most $m(|x|)$. To simplify the following exposition, we denote by $n = n(|x|)$ the number of coins tossed by V on common input x (so $n = \mathrm{poly}(|x|)$). We adopt several of the conventions from Section 6. Specifically, we assume, without loss of generality, that the last message is by V and it consists of V's entire sequence of coins. Recall that a transcript t of a possible (P,V) interaction is called *consistent* (for x) if every verifier message in t is the message V would have sent given input x, the previous messages in t, and the coin tosses specified by the last message in t.

More generally, we say that a *transcript prefix σ is consistent* if there exists a sequence of verifier coin tosses that would give rise to all the verifier messages contained in σ. We call a full transcript t *rejecting* if it is consistent and V would reject at the end of such an interaction.

For simplicity of exposition, we assume that the length of the next prover message is determined by the transcipt of the interaction so far.

Without loss of generality, we may assume that P is an optimal prover with respect to V; that is, for every x and every prefix σ of a possible transcript (even with suboptimal prover moves), P responds so as to maximize the acceptance probability of V.

The Rejecting Sets. Our aim is to devise an $O(m)$-message interactive proof system for $\overline{\Pi}$ (*i.e.,* the complement of Π). Following the ideas of Goldwasser and Sipser [GS89], for any possible prefix of a (P, V)-interaction, we consider the set of verifier coins that are consistent with this prefix and make V *reject* when interacting with P. For YES instances of $\overline{\Pi}$ (i.e., NO instances of Π), these sets are typically large, whereas for NO instances of $\overline{\Pi}$ (i.e., YES instances of Π) they are typically small.

We devise an interactive proof for proving that such sets are large. As we shall see below, we need to show that the sets corresponding to *all* (i.e., 2^b) possible prover moves are large. (This is in contrast to [GS89], where it was only necessary to consider sets corresponding to the optimal prover moves. This is because the aim in [GS89] was to prove (via a public-coin protocol) membership in Π itself, and so the sets considered there corresponded to verifier coins that are consistent with a given prefix and make V *accept* when interacting with P.)

Specifically, for any fixed common input x and any possible prefix σ of an (P, V)-interaction, let $\mathrm{REJ}_x(\sigma)$ denote the set of verifier coins that are consistent with σ and make V reject when interacting with P. Note that these sets $\mathrm{REJ}_x(\sigma)$ depend on the prover strategy P; there may be several different optimal prover strategies, and each may cause the verifier to accept on different coin tosses. However, it is important to note that the *size* of $\mathrm{REJ}_x(\sigma)$ is the same no matter which *optimal* prover strategy P is used.

We now discuss some basic properties of these "rejecting sets." Recall that, when interacting with the optimal prover P, the verifier V rejects a YES instance (resp., NO instance) of Π with probability at most $\frac{1}{3}$ (resp., at least $\frac{2}{3}$). Letting λ denote the empty prefix, it follows that, depending on x's membership in $\overline{\Pi}$, we have:

$$\text{(7.1)} \qquad \textsc{yes}\ \textbf{instance of}\ \overline{\Pi}\textbf{:} \qquad |\mathrm{REJ}_x(\lambda)| \geq \frac{2}{3} \cdot 2^n.$$

$$\text{(7.2)} \qquad \textsc{no}\ \textbf{instance of}\ \overline{\Pi}\textbf{:} \qquad |\mathrm{REJ}_x(\lambda)| \leq \frac{1}{3} \cdot 2^n.$$

For any possible prover move α following a prefix σ it holds that $|\mathrm{REJ}_x(\sigma)| \leq |\mathrm{REJ}_x(\sigma\alpha)|$ with equality holding for at least one α (*i.e.*, the α chosen by an optimal P to be its next move). Thus,

$$\text{(7.3)} \qquad \textbf{prover move:}\ |\mathrm{REJ}_x(\sigma)| = \min_{\alpha}\{|\mathrm{REJ}_x(\sigma\alpha)|\}.$$

For the next verifier move following a prefix σ it holds that $\mathrm{REJ}_x(\sigma) = \cup_\beta \mathrm{REJ}_x(\sigma\beta)$. Thus,

$$\text{(7.4)} \qquad \textbf{verifier move:}\ |\mathrm{REJ}_x(\sigma)| = \sum_{\beta} |\mathrm{REJ}_x(\sigma\beta)|.$$

7.1. Motivation to the protocol. Fixing a common input x (supposedly a YES instance of $\overline{\Pi}$), our goal is to prove that $|\mathrm{REJ}_x(\lambda)| \geq \frac{2}{3} \cdot 2^n$. This is done recursively following the round structure of (P, V). Suppose that we currently need to prove that $|\mathrm{REJ}_x(\sigma)| \geq N$. We consider three cases.

Case 1: σ is a full transcript. In this case, it is easy to generate the set $\mathrm{REJ}_x(\sigma)$ (which is either an empty or a singleton set) and to compare its size to N.

(Recall that by our conventions, the last verifier message consists of the outcomes of all the coins the verifier has tossed during the interaction. Thus, the latter sequence is easily extracted from σ, and one can easily determine whether or not σ is rejecting.)

Case 2: the next message is by P. Specifically, suppose that the next message is (a prover message) of length ℓ. Then, by Equation (7.3), we just prove recursively that $|\mathrm{REJ}_x(\sigma\alpha)| \geq N$ for every $\alpha \in \{0,1\}^\ell$.

This means branching in parallel to 2^ℓ recursive proofs, yielding a total branching factor of 2^b (in all rounds). Indeed, here is where the bound on the total number of bits sent by P is used.

Case 3: the next message is by V. In case the next message is a verifier message, by Equation (7.4) we need to prove that $\sum_\beta |\mathrm{REJ}_x(\sigma\beta)| \geq N$. Note that the number of possible verifier messages may be huge (*i.e.,* exponential in n), and thus we cannot afford to examine each term in the sum. Instead, we let the prover supply a *succinct representation* of a sequence $\{N_\beta\}$ such that $\sum_\beta N_\beta \approx N$ and $|\mathrm{REJ}_x(\sigma\beta)| \geq N_\beta$ for every β. This succinct representation should allow the new verifier to verify that both conditions hold. The verification will use parallel executions of a constant-round *sampling protocol* as well as poly(m) parallel recursive calls (*i.e.,* to verify $|\mathrm{REJ}_x(\sigma\beta)| \geq N_\beta$ for poly(m)-many β's).

This means branching in parallel to poly(m) recursive proofs, yielding a total branching factor of $\mathrm{poly}(m)^{m/2} = m^{O(m)}$ (in all rounds).

Further details regarding the implementation of Case 3 are indeed in place. As a warm-up, suppose that all non-empty $\mathrm{REJ}_x(\sigma\beta)$'s are of the same size. In such a case, the prover can state this size, denoted N', and prove that there are at least N/N' non-empty $\mathrm{REJ}_x(\sigma\beta)$'s each having size N'. Intuitively, the prover can prove this claim by employing a (standard) set lower-bound protocol (cf. [BM88, GS89, AH91]). Such a protocol has constant number of rounds, and produces a β for which the prover has to recursively prove that $|\mathrm{REJ}_x(\sigma\beta)| \geq N'$. Unfortunately, things are not that simple, because it is not necessarily the case that all non-empty $\mathrm{REJ}_x(\sigma\beta)$'s are of the same size. Consequently, a more refined approach seems to be necessary.

The way Goldwasser and Sipser [GS89] dealt with this difficulty (*i.e.,* that not all the sets are the same size) was to group the sets into clusters according to their approximate size; say, the i^{th} cluster contains all sets of size between 2^i and 2^{i+1}. Since there are only n such clusters, at least one of them must account for at least a $1/n$ fraction of the total sum, and we can recursively proceed with just that one cluster using the approach above. Clearly, such an approach incurs at least a factor n loss of accuracy with each round. To compensate for this loss, [GS89] first reduced the error of the proof system dramatically (to increase the gap in the set sizes that is guaranteed between YES and NO instances). However, we cannot afford such an error reduction because it blows up the prover-to-verifier communication.

Wishing to avoid the corresponding cost, we do not apply any error reduction on the interactive proof (P, V), but rather use it directly. Instead of focusing on one cluster (*i.e.,* the "heaviest" one), we simultaneously consider all clusters. Towards the recursive calls, we select a sample of poly(m)-many clusters (according to their weights) and generate poly(m)-many elements in each selected cluster. Loosely speaking, in the recursive calls, we shall verify that each of these elements is indeed in the corresponding cluster.

To summarize, the succinct representation used in implementing Case 3 consists of a sequence of sizes of the corresponding clusters, where we use a more refined clustering than [GS89]; that is, the i^{th} cluster contains all sets of size between $(1+\epsilon)^i$ and $(1+\epsilon)^{i+1}$, where $\epsilon = \Theta(1/m)$. In other words, we are clustering all β_j's having $|\text{REJ}_x(\sigma\beta_j)| \approx (1+\epsilon)^i$ into the i^{th} cluster, and the prover only provides the number of such β_j's. Letting c_i be the claimed size of the i^{th} cluster, we need to verify that $\sum_i c_i \cdot (1+\epsilon)^i \geq N$, and check recursively that these claimed sizes are essentially correct. The latter check is performed by selecting a weighted sample of clusters and sampling elements from each selected cluster.

The fact that we select a sample of clusters rather than working on all of them allows the complexity of our protocol to relate to $\text{poly}(m)^m$ rather than to n^m. (Recall that $n = \text{poly}(|x|)$, whereas m may be very small (e.g., $m = \log\log|x|$).)

The analysis of our protocol relies on a delicate combinatorial lemma regarding the clustering of sets by their size (Lemma 7.8 below), rather than on much simpler versions that are quite straightforward.

7.2. The actual protocol. Recall that the size of $|\text{REJ}_x(\lambda)|$ depends on whether x is a YES instance or a NO instance of $\overline{\Pi}$, and that the ratio between these two cases is at least a factor of 2. Let $\rho \stackrel{\text{def}}{=} 2^{1/(m+2)} = 1 + 1/O(m)$. We start the protocol with the aim to prove that $|\text{REJ}_x(\lambda)| \geq \frac{2}{3} \cdot 2^n$ (which indeed holds in case $x \in \overline{\Pi}_{\text{YES}}$), whereas in case $x \in \overline{\Pi}_{\text{NO}}$ the size of $\text{REJ}_x(\lambda)$ is off by a factor of ρ^{m+1}. We hope that after $i = 1, ..., m$ iterations, the relevant sets in case $x \in \overline{\Pi}_{\text{NO}}$ will be off by a factor of ρ^{m+1-i}. The discrepancy will be easily detectable at the end of the last iteration. (In the description that follows, $\rho = (1+\epsilon)^2$.)

Our protocol utilizes a constant-round (public-coin) protocol for sampling in arbitrary sets. The protocol is invoked so to enable a probabilistic polynomial-time player (called the *verifier*) to sample in a set, which is implicitly defined via some common input, and this player will be assisted by a computationally unbounded player (called the *prover*) that the first player does not trust. The first player will be given an integer, denoted N, that is supposed to be a valid lower-bound on the size of the set, denoted S. The names given above to the two parties fit the standard conventions regarding interactive proofs as well as fit our application (in which the high-level verifier will play the role of the verifier in the sampling protocol). The sampling protocol satisfies the following two properties:

1. If both players are honest, agree to sample a set $S \subseteq \{0,1\}^n$, and the verifier has a *valid* lower bound on $|S|$, then, with overwhelmingly high probability, the verifier will output an element of S.

2. If both players agree to sample a set $S \subseteq \{0,1\}^n$, and the (honest) verifier has a (possibly *invalid*) lower bound N on $|S|$ (i.e., possibly $|S| < N$), then no matter how the prover behaves, with probability at least $1 - \frac{|S|}{N} - \frac{1}{\mathrm{poly}(n)}$, the verifier will *not* output an element of S.

   (In fact, for any $S' \subseteq S$, the probability that the output is in S' is at most $\frac{|S'|}{N} + \frac{1}{\mathrm{poly}(n)}$.)

Protocols satisfying the above properties are implicit in the literature (cf., [BM88, GS89, AH91]). For sake of self-containment, we present such a protocol in Appendix A.

CONSTRUCTION 7.5 (main and recursive protocols). *The common input to the main protocol is a string* x (supposedly in $\overline{\Pi}_{\mathrm{YES}}$). *Let* $b = b(|x|)$, $m = m(|x|)$, $n = n(|x|)$ *and* ρ *be as above. Let* $\epsilon = 1/\Theta(m)$ *and* $t = n/\log(1+\epsilon) = \Theta(n/\epsilon)$.

Main protocol: *Invoke the recursive protocol* $(\overline{P}, \overline{V})$ *on input* $(x, \lambda, \frac{2}{3} \cdot 2^n)$. *The verifier accepts if and only if* $\overline{V}$ *returns* `true`.

(Motivation: If $x \in \overline{\Pi}_{\mathrm{YES}}$ then $|\mathrm{REJ}_x(\lambda)| \geq \frac{2}{3} \cdot 2^n$).

Recursive protocol $(\overline{P}, \overline{V})$: *On input* (x, σ, N), *depending on* σ, *perform one of the following:*

Case of full transcript: *In this case,* σ *is a full transcript of* (P, V). *If* σ *is a consistent transcript that makes* V *reject and* $N = 1$ *then the verifier* $\overline{V}$ *returns* `true`. *Otherwise,* (*i.e.,* $N \neq 1$ or σ does not make V reject), *the verifier* $\overline{V}$ *returns* `false`. (Note that the higher level never invokes the protocol with $N < 1$.)

Case of next move by P: *In this case, the next message w.r.t.* σ *is a message by* P. *Let us denote the length of this message by* ℓ. *Here the parties invoke* 2^ℓ *parallel executions of* $(\overline{P}, \overline{V})$, *with inputs* $(x, \sigma\alpha, N)$, *corresponding to all possible* $\alpha \in \{0,1\}^\ell$. *The verifier* $\overline{V}$ *returns* `true` *if and only if all these executions return* `true`.

Case of next move by V: *In this case, the next message w.r.t.* σ *is a verifier message.*

(i) Prover's initial message: *The prover* $\overline{P}$ *computes* $s_0, ..., s_t$ *such that* $s_i = |C_i|$, *where the* message class C_i *is defined as follows:*

$$C_i \stackrel{\text{def}}{=} \{\beta : |\text{REJ}_x(\sigma\beta)| \geq (1+\epsilon)^i\}. \tag{7.6}$$

The prover $\overline{P}$ *sends* $s_0, ..., s_t$ *to* $\overline{V}$.
(Motivation: $C_i \supseteq C_{i+1}$ and thus s_i should be greater or equal to s_{i+1}. Similarly, $\sum_{i=0}^{t} |C_i \setminus C_{i+1}| \cdot (1+\epsilon)^{i+1} > |\text{REJ}_x(\sigma)|$, and thus $\sum_{i=0}^{t}(s_i - s_{i+1}) \cdot (1+\epsilon)^{i+1}$ should be greater than N.)

(ii) Verifier's initial checks: *If* $s_i < s_{i+1}$, *for some* i, *then the verifier* $\overline{V}$ *aborts with output* `false`. *If* $\sum_{i=0}^{t}(s_i - s_{i+1}) \cdot (1+\epsilon)^{i+1} \leq N$ *then the verifier* $\overline{V}$ *aborts with output* `false`.

(iii) Verifier's selection of classes: *The verifier randomly selects a sequence of* $w = \text{poly}(m)$ *indices* $i_0, i_1, ..., i_{w-1}$ *such that* $i_0 = 0$ *and for each* $j \geq 1$ *the index* i_j *is selected independently according to the following distribution* $\mathcal{I}$ *that assigns* $i \in [t]$ *probability proportional to* $(1+\epsilon)^i s_i$. *That is,*

$$\Pr[\mathcal{I} = i] = \frac{(1+\epsilon)^i s_i}{\sum_{k=1}^{t}(1+\epsilon)^k s_k}. \tag{7.7}$$

(iv) Sampling in (the selected) classes: *In parallel, for all* $j = 0, 1, ..., w-1$, *the parties run a sampling protocol to obtain* w *samples* (*supposedly*) *in* C_{i_j}, *where the verifier enters* s_{i_j} *as input to this sampling protocol. All invocations are with deviation parameter* $\epsilon/16$ *and probability parameter* 2^{-b^2} (see Appendix A). *Denote the* w^2 *samples obtained by* $(\beta_{j,k})_{j,k}$, *where* $\beta_{j,k}$ *is the* k^{th} *sample generated supposedly in* C_{i_j}.
(Motivation: If $\beta_{j,k}$ is indeed in C_{i_j} then $|\text{REJ}_x(\sigma\beta_{j,k})| \geq (1+\epsilon)^{i_j}$.)

(v) Recursive calls: *The parties invoke* $W \stackrel{\text{def}}{=} w^2$ *parallel executions of* $(\overline{P}, \overline{V})$, *with corresponding inputs* $(x, \sigma\beta_{j,k}, (1+\epsilon)^{i_j})$. *The verifier* $\overline{V}$ *returns* `true` *if and only if all these executions return* `true`.

Since the body of the recursive protocol (i.e., without the recursive calls) can be implemented by a constant-round (public-coin) protocol, our main protocol has $O(m)$ messages (and is of the public-coin type). The total number of bottom-level recursive calls invoked by the main protocol is $2^b \cdot W^m = 2^b \cdot m^{O(m)}$, and so the overall complexity is $2^b \cdot m^{O(m)} \cdot \text{poly}(n, m, 1/\epsilon) = 2^b \cdot m^{O(m)} \cdot \text{poly}(n)$.

Motivation to the analysis: Assuming that the sampling protocol works perfectly (in case both parties are honest), it follows that $x \in \overline{\Pi}_{\mathrm{YES}}$ is always accepted by $\overline{V}$. (Unfortunately, the sampling protocol does carry a small probability of error, and so the actual analysis of this case is also postponed to the next subsection.) On the other hand, if $x \in \overline{\Pi}_{\mathrm{NO}}$ and $\overline{P}$ wishes not to fail $\overline{V}$'s initial checks (of Step ii), then $\overline{P}$ must provide many over-estimated s_i's in each recursive call in which it is asked to prove an over-estimated size bound. Furthermore, the probability mass of these over-estimated s_i's (w.r.t. the distribution in (7.7)) is at least $1/O(m)$, and so some over-estimated s_i will be selected w.h.p. (in Step iii). (The different treatment of s_0 is due to some technicality.) For each such over-estimated s_i, taking a large sample is likely to yield a β for which $(1+\epsilon)^i$ is also an over-estimation. Thus, an over-estimation for some claim at some recursive level is propagated to next recursive level. Needless to say, the above is merely a very rough sketch; the actual analysis is provided in the next subsection.

7.3. Analysis. The following lemma plays a key role in our analysis.

LEMMA 7.8. *Let $S \subseteq \{0,1\}^n$ be a nonempty set, $\epsilon > 0$, $t = n/\log(1+\epsilon)$, and $\{S_\beta\}$ be a partition of S. For every integer i, define*

$$C_i \stackrel{\mathrm{def}}{=} \{\beta : |S_\beta| \geq (1+\epsilon)^i\} \tag{7.9}$$

Indeed, C_i is defined also for $i < 0$, and in this case it equals C_0. Then:

(i) There exist $s_0 \geq s_1 \geq \cdots \geq s_t \geq s_{t+1} = 0$ such that $s_i \leq |C_i|$ for all $i = 0, ..., t$ and

$$\sum_{i=0}^{t} (s_i - s_{i+1}) \cdot (1+\epsilon)^{i+1} > |S|.$$

Furthermore, setting $s_i = |C_i|$, for all i's, will do.

(ii) Let $\epsilon' > 0$ and $\ell \in \mathbb{N}$. Suppose that $s_0 \geq s_1 \geq \cdots \geq s_t \geq s_{t+1} = 0$ and that

$$\sum_{i=0}^{t} (s_i - s_{i+1}) \cdot (1+\epsilon)^{i+1} > \frac{(1+\epsilon)^{\ell+1}}{(1-\epsilon')^2} \cdot |S|\,. \tag{7.10}$$

Let $\mathcal{I}$ be the probability distribution on $[t]$ that assigns $i \in [t]$ probability mass proportional to $(1+\epsilon)^i s_i$ (as in Equation 7.7). Then either $|C_{-\ell}| < (1-\epsilon') \cdot s_0$ or

$$\Pr_{i \leftarrow \mathcal{I}} [|C_{i-\ell}| < (1-\epsilon') \cdot s_i] > \epsilon'.$$

We apply Lemma 7.8 with $S = \mathrm{REJ}_x(\sigma)$ and $S_\beta = \mathrm{REJ}_x(\sigma\beta)$. Part i implies that if $|\mathrm{REJ}_x(\sigma)| \geq N$ and the prover sets the s_i's as directed by the protocol (i.e., to equal the $|C_i|$'s defined in (7.6)) then the verifier does not abort in Step ii. Furthermore, with overwhelmingly high probability, the recursive calls will be invoked with valid lower-bound claims (i.e., $|\mathrm{REJ}_x(\sigma\beta_{j,k})| \geq (1+\epsilon)^{i_j}$). On the other hand, Part ii asserts that if $\frac{(1+\epsilon)^\ell}{(1-\epsilon')^2} \cdot |\mathrm{REJ}_x(\sigma)| \leq N$ and the verifier does not abort in Step ii, then in Step iii the verifier is likely to select an index in $I \stackrel{\mathrm{def}}{=} \{i : |C_{i-\ell}| < (1-\epsilon') \cdot s_i\}$, where the $|C_i|$'s are as in (7.6). Specifically, either $0 \in I$ in which case i_0 is always in I or each i_j hits I with probability at least ϵ' (which will be set to equal $1/O(m)$). Loosely speaking, for each $i_j \in I$, with probability at least ϵ', each sample β that is generated with size parameter s_{i_j} is not in $C_{i_j-\ell}$; that is, with probability at least ϵ', $|\mathrm{REJ}_x(\sigma\beta)| < (1+\epsilon)^{i_j-\ell}$, in contrast to the recursive call that uses a size lower-bound of $(1+\epsilon)^{i_j}$. Thus, in such a case, we started with an over-estimate factor of $\frac{(1+\epsilon)^{\ell+1}}{(1-\epsilon')^2} = \frac{1+\epsilon}{(1-\epsilon')^2} \cdot (1+\epsilon)^\ell$, and invoke a recursive call with an over-estimate factor of $(1+\epsilon)^\ell$. But before applying Lemma 7.8, let us establish its correctness.

PROOF. First, we note that

$$(7.11) \quad \sum_{i=-\infty}^{+\infty} (|C_i| - |C_{i+1}|) \cdot (1+\epsilon)^i \;\leq\; |S| \;<\; \sum_{i=0}^{t} (|C_i| - |C_{i+1}|) \cdot (1+\epsilon)^{i+1},$$

because $(1+\epsilon)^i \leq |S_\beta| < (1+\epsilon)^{i+1}$ for every $\beta \in C_i \setminus C_{i+1}$. Thus, Part i follows by setting $s_i = |C_i|$, for $i = 0, ..., t$.

Part ii is established by the following claim and an application of Markov's inequality (*i.e.,* for $X = |C_{i-\ell}|/s_i$, which is non-negative, it holds that $\Pr[X \geq 1-\epsilon'] \leq \mathrm{E}[X]/(1-\epsilon') < 1-\epsilon'$).

CLAIM 7.12. *Suppose that* (7.10) *holds and* $|C_{-\ell}| \geq (1-\epsilon') \cdot s_0$. *Then*

$$\mathop{\mathrm{E}}_{i \leftarrow \mathcal{I}} \left[\frac{|C_{i-\ell}|}{s_i} \right] < (1-\epsilon')^2.$$

To prove this claim, we first expand the expectation:

$$\begin{aligned} \mathop{\mathrm{E}}_{i \leftarrow \mathcal{I}} \left[\frac{|C_{i-\ell}|}{s_i} \right] &= \sum_{i=1}^{t} \frac{(1+\epsilon)^i \cdot s_i}{\sum_{k=1}^t (1+\epsilon)^k \cdot s_k} \cdot \frac{|C_{i-\ell}|}{s_i} \\ &= \frac{\sum_{i=1}^t (1+\epsilon)^i \cdot |C_{i-\ell}|}{\sum_{i=1}^t (1+\epsilon)^i \cdot s_i}. \end{aligned}$$

We rewrite both the numerator and denominator using the following identity, which holds for any sequence of numbers $x_0, ..., x_t$ and $x_{t+1} = 0$:

$$\sum_{i=1}^{t}(1+\epsilon)^i \cdot x_i = \frac{1+\epsilon}{\epsilon} \cdot \left(-x_0 + \sum_{i=0}^{t}(x_i - x_{i+1}) \cdot (1+\epsilon)^i \right).$$

This gives:

$$\text{(7.13)} \qquad \mathop{\mathrm{E}}_{i \leftarrow \mathcal{I}} \left[\frac{|C_{i-\ell}|}{s_i} \right] = \frac{-|C_{-\ell}| + \sum_{i=0}^{t}(|C_{i-\ell}| - |C_{i+1-\ell}|) \cdot (1+\epsilon)^i}{-s_0 + \sum_{i=0}^{t}(s_i - s_{i+1}) \cdot (1+\epsilon)^i}.$$

We upper-bound the numerator as follows:

$$\begin{aligned} & -|C_{-\ell}| + \sum_{i=0}^{t}(|C_{i-\ell}| - |C_{i+1-\ell}|) \cdot (1+\epsilon)^i \\ \leq \quad & -|C_{-\ell}| + (1+\epsilon)^\ell \cdot |S| \\ < \quad & -(1-\epsilon')^2 \cdot s_0 + (1+\epsilon)^\ell \cdot |S| \\ < \quad & -(1-\epsilon')^2 \cdot s_0 + (1-\epsilon')^2 \cdot \sum_{i=0}^{t}(s_i - s_{i+1}) \cdot (1+\epsilon)^i \end{aligned}$$

where the first inequality is due to (7.11), the second inequality is due to the claim's second hypothesis (*i.e.*, $|C_{-\ell}| \geq (1-\epsilon') \cdot s_0 > (1-\epsilon')^2 \cdot s_0$), and the third inequality is due to the claim's first hypothesis (*i.e.*, (7.10)). Substituting this upper-bound into (7.13) establishes Claim 7.12 and thereby Lemma 7.8. ■

Proof of Theorem 3.7. Construction 7.5 yields a (public-coin) protocol which satisfies the complexity bounds asserted in Theorem 3.7 (*i.e.*, it exchanges $O(m)$ messages and the total complexity is at most $2^b \cdot \mathrm{poly}(n, m^m)$). It is left to show that this protocol constitutes an interactive proof system for $\overline{\Pi}$. This fact is established in the following two claims.

Claim 7.14 (completeness). *If $x \in \overline{\Pi}_{\mathrm{YES}}$ and the prover plays as directed, then $\overline{V}$ accepts with probability at least* $2/3$.

Proof. We will show (below) that, with probability at least $2/3$, all recursive calls are with inputs (x, σ, N) satisfying $|\mathrm{REJ}_x(\sigma)| \geq N$. Applying Part i of Lemma 7.8 with $S = \mathrm{REJ}_x(\sigma)$ and $S_\beta = \mathrm{REJ}_x(\sigma\beta)$, it then follows that (when the prover sets the s_i's as directed by the protocol) the verifier does not abort in Step ii (because $\sum_{i=0}^{t}(s_i - s_{i+1}) \cdot (1+\epsilon)^{i+1} > |\mathrm{REJ}_x(\sigma)| \geq N$). Furthermore, in this case, all the bottom-level recursive calls are with inputs (x, σ, N) that satisfy $|\text{REJ}_x(\sigma)| \geq N \geq 1$ (because $N \geq 1$ in each call), and since such σ fully specifies V's coins it must be that $|\text{REJ}_x(\sigma)| = 1 = N$ (because for such a full transcript σ it must be that $|\text{REJ}_x(\sigma)| \leq 1$). Thus, all the bottom-level calls return `true`, and thus *all* recursive calls return `true`.

We now show that, with probability at least $2/3$, all calls at each level of the recursion are with inputs (x, σ, N) satisfying $|\text{REJ}_x(\sigma)| \geq N$. This is shown by induction on the recursion level, using the fact that the number of recursive calls (in each level) is less than $(2m)^{O(b)}$. The basis of the induction holds because at the top level the input is $(x, \lambda, \frac{2}{3} \cdot 2^n)$ and $|\text{REJ}_x(\lambda)| \geq \frac{2}{3} \cdot 2^n$ holds (since $x \in \overline{\Pi}_{\text{YES}}$). For the induction step we show that if $|\text{REJ}_x(\sigma)| \geq N$ for some recursive execution with input (x, σ, N), then, for each recursive call that is directly invoked by the former execution, with probability at least $1 - 2^{-b^2}$, the input (x, σ', N') associated with the recursive call satisfies $|\text{REJ}_x(\sigma')| \geq N'$. Thus, if the induction hypothesis holds for some level, then, with probability at least $1 - (2m)^{O(b)} \cdot 2^{-b^2} > 1 - \frac{1}{3m}$, it holds also for the next level.

We consider two cases. In case the next message is by P, we have $|\text{REJ}_x(\sigma\alpha)| \geq N$ for every possible α (by (7.3)), and so the recursive calls $(x, \sigma\alpha, N)$ satisfy the condition. In case the next message is by V, it holds that the s_i's sent by $\overline{P}$ satisfy $s_i = |C_i|$, where the C_i's are as in (7.6). Thus, for every i_j selected in Step iii, it holds that $|C_{i_j}| \geq s_{i_j}$. Thus, each invocation of the sampling protocol in Step iv, is likely to return a sample in the corresponding C_{i_j}; specifically (by Part i of Lemma A.2), with probability at least $1 - 2^{-b^2}$, the sampled $\beta_{j,k}$ is in C_{i_j}. In this case, the resulting recursive call with input $(x, \sigma\beta_{j,k}, (1+\epsilon)^{i_j})$ satisfies $|\text{REJ}_x(\sigma\beta_{j,k})| \geq (1+\epsilon)^{i_j}$. ■

CLAIM 7.15 (soundness). *If $x \in \overline{\Pi}_{\text{NO}}$ then, no matter how the prover plays, the verifier $\overline{V}$ accepts with probability at most* $1/3$ *(provided $\epsilon < 1/cm$ for a sufficiently large constant c).*

PROOF. We may assume, without loss of generality, that in each recursive call the prover supplies a list of s_i's that pass the verifier's initial check (of Step ii). We will show, by induction on the recursion depth $d = 0, 1, ..., m$, that with high probability, one of the recursive calls at level d is with an input (x, σ, N) that satisfies $|\text{REJ}_x(\sigma)| \leq (1+\epsilon)^{2d-(2m+1)} \cdot N$. Thus, the last recursion level has a call an input (x, σ, N) that satisfies $|\text{REJ}_x(\sigma)| \leq (1+\epsilon)^{-1} \cdot N$. If $N > 1$ then such a call returns `false`, causing the verifier to reject (i.e., return `false` to the main protocol). Otherwise, it must be that $N = 1$ and $\text{REJ}_x(\sigma) = \emptyset$, which means that σ is not a consistent rejecting transcript, and so this call also returns `false`. Thus, we may focus on proving the above inductive claim.

The induction basis holds because it refers to the main protocol's call to the recursive protocol, a call that is with input $(x, \lambda, \frac{2}{3}\cdot 2^n)$ that satisfies $|\mathrm{REJ}_x(\lambda)| \le \frac{1}{3}\cdot 2^n \le (1+\epsilon)^{-(2m+1)} \cdot \frac{2}{3}\cdot 2^n$ (since $x \in \overline{\Pi}_{\mathrm{NO}}$ and $\epsilon < 1/cm$). Specifically, we may use $\epsilon \approx (\ln 2)/(2m+1)$ so that $(1+\epsilon)^{2m+1} = 2$ holds.

We now turn to the induction step. We assume that there is a level d recursive-call with input (x, σ, N) that satisfies $|\mathrm{REJ}_x(\sigma)| \le (1+\epsilon)^{2d-(2m+1)} \cdot N$. We will show that, with probability at least $1 - 2^{-\operatorname{poly}(m)} > 1 - 1/3m$, this level d recursive-call invokes a level $d+1$ call with an input (x, σ', N') that satisfies $|\mathrm{REJ}_x(\sigma')| \le (1+\epsilon)^{2(d+1)-(2m+1)} \cdot N'$. We consider two cases. In case the next message is by P, for some α, we have $|\mathrm{REJ}_x(\sigma\alpha)| = |\mathrm{REJ}_x(\sigma)| \le (1+\epsilon)^{2d-(2m+1)} \cdot N < (1+\epsilon)^{2(d+1)-(2m+1)} \cdot N$ (where the equality is due to (7.3)), and so the recursive calls $(x, \sigma\alpha, N)$ satisfy the condition.

The more involved case is when the next message is by V. Using the hypothesis that the list of s_i's (supplied by the prover) passes the verifier's initial check (of Step ii), we may invoke Part ii of Lemma 7.8 with $S = \mathrm{REJ}_x(\sigma)$,[2] $S_\beta = \mathrm{REJ}_x(\sigma\beta)$, $\epsilon' \approx \epsilon/2$ (so that $(1-\epsilon')^{-2} = 1+\epsilon$) and $\ell = (2m+1) - 2d - 2$ (so that $(1+\epsilon)^{(2m+1)-2d} = (1+\epsilon)^{\ell+1}/(1-\epsilon')^2$). For C_i's as in (7.6) and $I \stackrel{\text{def}}{=} \{i : |C_{i-\ell}| < (1-\epsilon')\cdot s_i\}$, it follows that either $i_0 = 0 \in I$ or for every $j = 1, ..., w-1$, the index i_j selected in Step iii is in I with probability at least ϵ'. Thus, with probability at least $1 - 2^{-\operatorname{poly}(m)}$, one of the i_j's (possibly i_0) selected in Step iii is in I. For the rest of the argument, let us fix a j such that $i_j \in I$. By the definition of I and C_i, it follows that

$$|\{\beta : |\mathrm{REJ}_x(\sigma\beta)| \ge (1+\epsilon)^{i_j-\ell}\}| \;=\; |C_{i_j-\ell}| \;<\; (1-\epsilon')\cdot s_{i_j}\,.$$

Thus, in each of the w corresponding invocations of the sampling protocol (in Step iv), with probability at least $\epsilon' - 4\cdot\frac{\epsilon}{16} = \Omega(1/m)$, we generate $\beta \notin C_{i_j-\ell}$; this follows by using Part sampling.lem.2 of Lemma A.2, with $N = s_{i_j}$ and $S' = C_{i_j-\ell}$, and observing that $1 - \frac{|S'|}{N} > \epsilon'$ (and that the deviation parameter equals $\epsilon/16$). It follows that (in each of these w invocations), with probability at least $\Omega(1/m)$, the generated β is such that

$$|\mathrm{REJ}_x(\sigma\beta)| \;<\; (1+\epsilon)^{i_j-\ell} \;=\; (1+\epsilon)^{-((2m+1)-2d-2)}\cdot(1+\epsilon)^{i_j}\,.$$

Thus, with probability at least $1-2^{-\operatorname{poly}(m)}$, one of the $\beta_{j,k}$'s generated in Step iv satisfies $|\mathrm{REJ}_x(\sigma\beta_{j,k})| \;<\; (1+\epsilon)^{2(d+1)-(2m+1)}\cdot(1+\epsilon)^{i_j}$, which implies that the

[2] Lemma 7.8 requires that $S \neq \emptyset$, but if $\mathrm{REJ}_x(\sigma) = \emptyset$ then $\mathrm{REJ}_x(\sigma\beta_{j,k}) = \emptyset$, for all recursive calls, and the induction step trivially holds.

corresponding recursive call is with input $(x, \sigma\beta_{j,k}, (1+\epsilon)^{i_j})$ that satisfies the induction claim. This establishes the induction step, and the claim follows. ■

8. A Message Complexity Hierarchy

In this section, we give evidence that the Speedup Theorem (Thm. 2.3) cannot be improved. To do so, for every "nice" function $m()$, we give a problem that has an interactive proof with m messages but is unlikely to have an interactive proof with $o(m)$ messages.

First, we formalize what we mean by a "nice" function. For a function $f : \mathbb{N} \to \mathbb{N}$, let $f^{-1}(n)$ be the least m such that $f(m) \geq n$. We say that f is nice if (a) $f(n)$ and $f^{-1}(n)$ are computable in time poly(n) (b) f is monotone increasing (not necessarily strict), and (c) $f(f^{-1}(n)) = O(n)$. Note that these conditions are satisfied by functions such as $\log n$, $\log^c n$, n^ϵ, and n.

The problems we consider are variants of #SAT, which was shown to be in **IP** in [LFKN92]. Recall that the decisional version of the counting problem #SAT is

$$\#\mathrm{SAT} \stackrel{\mathrm{def}}{=} \{(\varphi, k) : \varphi \text{ has at most } k \text{ satisfying assignments}\}.$$

For a nice function $v : \mathbb{N} \to \mathbb{N}$ satisfying $v(n) \leq n$, we define

$$\#\mathrm{SAT}_v \stackrel{\mathrm{def}}{=} \{(\varphi, k) \in \#\mathrm{SAT} : \varphi \text{ has at most } v(|\varphi|) \text{ variables}\}.$$

By refining the standard proof system for #SAT, we have:

THEOREM 8.1 (refining [LFKN92, Sha92]). *For every nice function $v(n)$,*

$$\#\mathrm{SAT}_v \in \mathbf{AM}(\mathrm{poly}(n), m)\,, \text{ where } m(n) = v(n)/\log_2 n.$$

PROOF SKETCH. We begin by sketching what the standard interactive proof for #SAT (*e.g.,* as presented in [Sip97, Gol99, Vad00]) gives for an instance of $\#\mathrm{SAT}_v$. The common input is a pair (φ, k), where φ is of length n and has $v = v(n)$ variables. The prover sends the verifier the number k' of satisfying assignments of φ, and the verifier checks that $k' \leq k$. Then, the prover and verifier extend $\varphi : \{0,1\}^v \to \{0,1\}$ to a polynomial $\tilde{\varphi} : \mathbb{F}^v \to \mathbb{F}$ of degree at most n over some sufficiently large finite field $\mathbb{F}$ and the problem is reduced to proving a statement of the form:

$$\sum_{x_1 \in \{0,1\}} \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_v \in \{0,1\}} \tilde{\varphi}(x_1, \ldots, x_v) = k', \tag{8.2}$$

In each round of the protocol, one variable of $\tilde{\varphi}$ is "eliminated". More precisely, in the i^{th} round, the prover sends the verifier a univariate polynomial

$$p_i(x) \stackrel{\text{def}}{=} \sum_{x_{i+1}\in\{0,1\}} \cdots \sum_{x_v\in\{0,1\}} \tilde{\varphi}(\alpha_1,\ldots,\alpha_{i-1},x,x_{i+1},\ldots,x_v),$$

where $\alpha_1,\ldots,\alpha_{i-1}$ are elements of the field determined by earlier rounds of the protocol. Then the verifier checks that $p_i(0)+p_i(1)=p_{i-1}(\alpha_{i-1})$, and chooses α_i uniformly from $\mathbb{F}$. (p_0 is defined to be constant polynomial k', and at the end, the verifier checks that $p_v(\alpha_v)=\tilde{\varphi}(\alpha_1,\ldots,\alpha_v)$.)

To reduce the message complexity of the proof system, we instead work with $\ell=\Theta(\log n)$ variables at a time, like done in [BFLS91, AS98]. Let H be any canonical subset of $\mathbb{F}$ of size 2^ℓ, and let $\pi=(\pi_1,\ldots,\pi_\ell)$ be a bijection from H to $\{0,1\}^\ell$. By interpolation, each π_i can be extended to a degree $|H|=\text{poly}(n)$ polynomial $\tilde{\pi}_i:\mathbb{F}\to\mathbb{F}$ which agrees with π_i on H. Consider the polynomial $f:\mathbb{F}^{v/\ell}\to\mathbb{F}$ defined by

$$f\left(y_1,\ldots,y_{v/\ell}\right)=\tilde{\varphi}\left(\tilde{\pi}_1(y_1),\ldots,\tilde{\pi}_\ell(y_1),\ldots,\tilde{\pi}_1(y_{v/\ell}),\ldots,\tilde{\pi}_\ell(y_{v/\ell})\right).$$

Two key points are that f is still of degree poly(n) (because $\tilde{\varphi}$ and the $\tilde{\pi}_i$'s have degree poly(n)) and that f is just as easy to evaluate as $\tilde{\varphi}$. Now proving Equation 8.2 becomes equivalent to proving

$$\sum_{y_1\in H}\sum_{y_2\in H}\cdots\sum_{y_{v/\ell}\in H} f(y_1,\ldots,y_{v/\ell})=k'. \tag{8.3}$$

This can be done in almost exactly the same way as before, eliminating one variable at a time, except that instead of checking $p_i(0)+p_i(1)=p_{i-1}(\alpha_{i-1})$, the verifier must check that $\sum_{\beta\in H}p_i(\beta)=p_{i-1}(\alpha_{i-1})$. The representation of the p_i's and the evaluation of this sum are still feasible because the degree of f is poly(n) and H is of size poly(n). The analysis of the new proof system is identical to that of the original, with a slight loss in the soundness due to the fact that the degree of f is larger than that of $\tilde{\varphi}$. ■

Now we observe that it is unlikely that $\#\text{SAT}_v$ has a proof system with substantially less rounds, because this would yield shorter than expected (two-message) proofs for non-satisfiability:

PROPOSITION 8.4. *Let v be any nice function satisfying $\omega(\log n)\le v(n)\le n$. If $\#\text{SAT}\notin\mathbf{AM}(2^{o(n)},2)$, then for every $m:\mathbb{N}\to\mathbb{N}$ such that $m(n)=o(v(n)/\log_2 n)$:*

$$\#\text{SAT}_v\notin\mathbf{IP}(\text{poly}(n),m)$$

PROOF. Suppose $\#\mathrm{SAT}_v \in \mathbf{IP}(\mathrm{poly}(n), m)$, where $m(n) = o(v(n)/\log n)$. Combining Theorems 2.4 and 2.3, we have

$$\begin{aligned}\#\mathrm{SAT}_v &\in \mathbf{IP}\left(\mathrm{poly}(n), m\right)\\ &\subseteq \mathbf{AM}\left(\mathrm{poly}(n), m+2\right)\\ &\subseteq \mathbf{AM}\left(\mathrm{poly}(n)^{O(m)}, 2\right)\\ &= \mathbf{AM}\left(2^{O(m\cdot\log n)}, 2\right)\\ &= \mathbf{AM}\left(2^{o(v)}, 2\right).\end{aligned}$$

Now we can obtain a proof system for #SAT by padding. Given an instance (φ, k) of #SAT of length n, if we pad it to length $N = v^{-1}(n)$, then it has at most $n \leq v(N)$ variables. So we can view it as an N-bit long instance of $\#\mathrm{SAT}_v$ and execute the above $\mathbf{AM}(2^{o(v(N))}, 2)$ proof system on it. This gives a 2-message proof system for #SAT that on instances of length n has bit complexity

$$\mathrm{poly}(N, 2^{o(v(N))}) = 2^{o(v(N))} = 2^{o(n)},$$

where the first equality is because $v(N) = \omega(\log N)$, and the second because $N = v^{-1}(n)$. ■

PROOF OF THEOREM 3.10: Combining Theorem 8.1 and Proposition 8.4 (and assuming $\#\mathrm{SAT} \notin \mathbf{AM}(2^{o(n)}, 2)$), we have for every nice and super-logarithmic $v : \mathbb{N} \to \mathbb{N}$ (such that $v(n) \leq n$)

$$\#\mathrm{SAT}_v \in \mathbf{AM}(\mathrm{poly}(n), m) \setminus \mathbf{AM}(\mathrm{poly}(n), o(m))$$

where $m(n) = v(n)/\log_2 n$. Theorem 3.10 follows. ■

9. Directions for Further Work

There are clearly several places where quantitative improvements to our results would be desirable. As discussed in Section 4, it would be very interesting to remove the $c^2 > s$ constraint in Theorem 3.1 (or to give evidence that it is necessary). The constraints on the completeness and soundness in our results for general 1-message proof systems (in Section 5) are even more severe, and do not stem solely from constraints in previous results about **SZK**. Another place where it is not clear that our bounds are quantitatively optimal involves the complexity bounds in our results for general $\mathbf{IP}(b)$. Specifically, it is unclear whether the m^m complexity in Theorem 3.7 and the additional exponent of

m incurred in Corollary 3.9 are necessary. (In fact, removing the m^m from Theorems 2.1 and 2.2 was stated as an open problem in [GH98].)

Another direction for further work is to unify these results with those which bound the complexity of interactive proofs with low knowledge complexity. As mentioned in Section 1.3, those works are incomparable to ours. For example, the results of [PT96] require that the error probabilities are exponentially small in the knowledge complexity, and the results of [SV03] only apply to knowledge complexity in the "hint sense" (which is not bounded by the prover-to-verifier communication). Can one give evidence that **NP** does not have interactive proofs of low knowledge complexity k (in the usual sense) where the error probabilities are larger than 2^{-k}? The strongest imaginable statement of this form would say that interactive proofs with logarithmic knowledge complexity and constant error probabilities capture exactly **SZK**; such a result would simultaneously subsume all of our results and those mentioned above.

Acknowledgements

Oded Goldreich was supported by the MINERVA Foundation, Germany. Work done while Salil Vadhan was at the Institute for Advanced Study, Princeton, NJ, and was supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship. Avi Wigderson was partially supported by NSF grants CCR-9987845 and CCR-9987077. The authors are grateful to the anonymous referees for their comments.

References

[ABV95] William Aiello, Mihir Bellare, and Ramarathnam Venkatesan. Knowledge on the average—perfect, statistical and logarithmic. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing*, pages 469–478, Las Vegas, Nevada, 29 May–1 June 1995.

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42(3):327–345, June 1991.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, January 1998.

[AK97] V. Arvind and J. Köbler. On resource-bounded measure and pseudorandomness. In *Proceedings of the 17th Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 235–249. LNCS 1346, Springer-Verlag, 1997.

[BFLS91] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the Twenty Third Annual ACM Symposium on Theory of Computing*, pages 21–31, New Orleans, Louisiana, 6–8 May 1991.

[Bab85] László Babai. Trading group theory for randomness. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, pages 421–429, Providence, Rhode Island, 6–8 May 1985.

[BM88] László Babai and Shlomo Moran. Arthur-Merlin games: A randomized proof system and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.

[BGS98] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM Journal on Computing*, 27(3):804–915 (electronic), 1998.

[BHZ87] Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25:127–132, 1987.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, October 1988.

[For89] Lance Fortnow. The complexity of perfect zero-knowledge. In Silvio Micali, editor, *Advances in Computing Research*, volume 5, pages 327–343. JAC Press, Inc., 1989.

[Gol99] Oded Goldreich. *Modern Cryptography, Probabilistic Proofs, and Pseudorandomness*. Number 17 in Algorithms and Combinatorics. Springer-Verlag, 1999.

[GG00] Oded Goldreich and Shafi Goldwasser. On the limits of nonapproximability of lattice problems. *Journal of Computer and System Sciences*, 60(3):540–563, 2000.

[GH98] Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Information Processing Letters*, 67(4):205–214, 1998.

[GK93] Oded Goldreich and Eyal Kushilevitz. A perfect zero-knowledge proof system for a problem equivalent to the discrete logarithm. *Journal of Cryptology*, 6:97–116, 1993.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, 1991.

[GOP98] Oded Goldreich, Rafail Ostrovsky, and Erez Petrank. Computational complexity and knowledge complexity. *SIAM Journal on Computing*, 27(4):1116–1141, August 1998.

[GP99] Oded Goldreich and Erez Petrank. Quantifying knowledge complexity. *Computational Complexity*, 8(1):50–98, 1999.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, February 1989.

[GS89] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In Silvio Micali, editor, *Advances in Computing Research*, volume 5, pages 73–90. JAC Press, Inc., 1989.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *Proceedings of the Twenty-Fourth Annual ACM Symposium on the Theory of Computing*, pages 723–732, Victoria, British Columbia, Canada, 4–6 May 1992.

[KvM99] Adam R. Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. In *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*, pages 659–667, Atlanta, 1–4 May 1999.

[LFKN92] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, October 1992.

[MV99] Peter Bro Miltersen and N.V. Vinodchandran. Derandomizing Arthur–Merlin games using hitting sets. In *40th Annual Symposium on Foundations of Computer Science*, New York, NY, 17–19 October 1999. IEEE.

[Oka00] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *Journal of Computer and System Sciences*, 60(1):47–108, February 2000.

[PT96] Erez Petrank and Gábor Tardos. On the knowledge complexity of $\mathcal{NP}$. In *37th Annual Symposium on Foundations of Computer Science*, pages 494–503, Burlington, Vermont, 14–16 October 1996. IEEE.

[SV03] Amit Sahai and Salil Vadhan. A complete problem for statistical zero knowledge. *Journal of the ACM*, 50(2):196–249, March 2003.

[Sha92] Adi Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, October 1992.

[Sip97] Michael Sipser. *Introduction to the Theory of Computation.* PWS Publishing, 1997.

[Tuc95] Alan Tucker. *Applied combinatorics.* John Wiley & Sons Inc., New York, third edition, 1995.

[Vad00] Salil Vadhan. Probabilistic proof systems I: Interactive and zero-knowledge proofs. Lecture Notes from the *IAS/PCMI Graduate Summer School on Computational Complexity*, August 2000. Available from `http://eecs.harvard.edu/~salil/`.

[Vad99] Salil P. Vadhan. *A Study of Statistical Zero-Knowledge Proofs.* PhD thesis, Massachusetts Institute of Technology, August 1999.

A. Sampling Sets of Known Cardinality

The following protocol is a variant of known protocols for sampling NP-sets of known cardinality. (By sampling an NP-set, we actually mean sampling a "slice" of such a set; that is, for some $n \in \mathbb{N}$, the slice is a subset S_α of $\{0,1\}^n$, specified by a string $\alpha \in \{0,1\}^{\mathrm{poly}(n)}$, where the set $\{(\alpha, x) : x \in S_\alpha\}$ is an NP-set.) In fact, we will present a two-party protocol for sampling in arbitrary sets, while ignoring the issue of verifying membership in the set. In correspondence with common applications as well as with the respective computing powers of the parties, we call the parties *verifier* and *prover*. Indeed, the verifier strategy presented below is implementable in probabilistic polynomial-time, whereas this is not necessarily the case for the prover.

Standard sampling protocol utilize a family of pairwise independent hash functions (*cf.,* [BM88, GS89, AH91]). In order to obtain improved performance (in Item i of Lemma A.2 below), we use in our protocol hash functions that are $2k$-wise independent. More precisely, a set of hash functions $H_{n,\ell}$ is $2k$-*wise independent* if, for every $2k$ distinct preimages (in $\{0,1\}^n$), the corresponding images, under a function uniformly selected in $H_{n,\ell}$, are independent and uniformly distributed in the range ($\{0,1\}^\ell$). One efficient construction of such hash functions is obtained by considering the set of all $2k-1$ degree univariate polynomials over $GF(2^n) \equiv \{0,1\}^n$, and taking the ℓ-bit prefix of the value of such polynomials (on an evaluation point).

Construction A.1 (a sampling protocol). *The common input is an integer n, a string $\alpha \in \{0,1\}^{\mathrm{poly}(n)}$ specifying a set $S \stackrel{\text{def}}{=} S_\alpha \subseteq \{0,1\}^n$, and an integer N which is supposed to equal $|S|$.*

Error parameters: *a* deviation *error $\epsilon > 0$ and a* probability *error $\delta > 0$.*

Protocol: *The parties set $k = \log_2(1/\delta)$ and $\ell \stackrel{\text{def}}{=} \lfloor \log_2(\epsilon^3 N/2k^2) \rfloor$. Thus, $2^{-\ell}N \approx \frac{2k^2}{\epsilon^3}$.*

Assuming that $\ell > 0$, we denote by $H_{n,\ell}$ a family of efficient $2k$-wise independent hashing functions. Otherwise (i.e., for $\ell \leq 0$), we redefine $\ell \stackrel{\text{def}}{=} 0$ and let $H_{n,\ell}$ be the singleton set containing the function h mapping $\{0,1\}^n$ to the all-zero string (and so satisfying $h^{-1}(0^\ell) = \{0,1\}^n$).

(i) The verifier uniformly selects $h \in H_{n,\ell}$, and sends it to the prover.

(ii) Upon receiving $h \in H_{n,\ell}$, the prover responds with a list of $t \stackrel{\text{def}}{=} (1-\epsilon) \cdot \frac{N}{2^\ell}$ strings in $S \cap h^{-1}(0^\ell)$. Denote the list sent to the verifier $y_1, ..., y_t$.

(iii) The verifier performs a superficial examination of the list and produces a corresponding output. That is:

(a) Reject illegal lists: *The verifier checks that all the y_i's are distinct, and that $h(y_i) = 0^\ell$ for every $i = 1, ..., t$. If any of these conditions is not satisfied, the verifier outputs a special error symbol, denoted $\perp$.*

(b) Act on legal lists: *Assuming that the above conditions hold, the verifier selects uniformly $i \in \{1, ..., t\}$, and outputs y_i.*

The above protocol exchanges two messages, and the messages being sent have length $\mathrm{poly}(\frac{n}{\epsilon} \cdot \log(1/\delta))$. The verifier is of the public-coin type and can be implemented in probabilistic polynomial-time. Clearly, the verifier's output is either an n-bit long string or the error symbol $\perp$. A cheating prover may easily cause the verifier to always output $\perp$, but this means that the verifier detects that the prover is cheating. In case S is a slice of an NP-set, the protocol can be augmented with NP-witnesses, and the verifier may avoid outputting elements not in S (and output $\perp$ whenever such an element is presented to it). As we shall show, the essential feature of the above protocol is that the prover cannot restrict the output to a too small set (see Item ii below).

Lemma A.2 (analysis of the sampling protocol). *Suppose that $\delta \leq \epsilon < 1/3$ and that the verifier follows the prescribed program.*

(i) If $N = |S|$ and the prover follows the prescribed program then, with probability at least $1 - \delta$, the verifier outputs an element of S.

(ii) For every set $S' \subseteq \{0,1\}^n$ and every $N \leq 2^n$, no matter what the prover does, the probability that the verifier output resides in S' is at most $\frac{|S'|}{N} + 4\epsilon$.

Proof. We start with Part i, and ignore the case $\ell = 0$ (which is obvious). In case $\ell > 0$, we define 0-1 random variables ζ_x such that $\zeta_x \stackrel{\mathrm{def}}{=} 1$ if $h(x) = 0^\ell$ and $\zeta_x \stackrel{\mathrm{def}}{=} 0$ otherwise. Clearly, for every $x \in S$, it holds that $\mathrm{E}(\zeta_x) = \Pr[\zeta_x = 1] = 2^{-\ell}$, and the ζ_x's are $2k$-wise independent. Denoting $\overline{\zeta}_x \stackrel{\mathrm{def}}{=} \zeta_x - 2^{-\ell}$ and employing standard analysis (using $|S| > 2^{\ell+1}$) we get

$$\begin{aligned}
\Pr_h\left[\left|\frac{\sum_{x\in S}\zeta_x}{|S|} - 2^{-\ell}\right| > \epsilon \cdot 2^{-\ell}\right] &< \frac{\mathrm{E}\left[\left(\sum_{x\in S}\overline{\zeta}_x\right)^{2k}\right]}{(\epsilon 2^{-\ell}\cdot|S|)^{2k}} \\
&= \frac{\mathrm{E}\left[\sum_{x_1,\dots,x_{2k}\in S}\prod_{i=1}^{2k}\overline{\zeta}_{x_i}\right]}{\epsilon^{2k}\cdot(2^{-\ell})^{2k}\cdot|S|^{2k}} \\
&< \frac{|S|^k\cdot k^{2k}\cdot(2^{-\ell})^k}{\epsilon^{2k}\cdot(2^{-\ell})^{2k}\cdot|S|^{2k}} \\
&= \left(\frac{k^2\cdot 2^\ell}{\epsilon^2\cdot|S|}\right)^k .
\end{aligned}$$

Using $|S| = N$ and $2k^2 \cdot 2^\ell \leq \epsilon^3 N < \epsilon^2 N$ (where the first inequality is due to $\ell \leq \log_2(\epsilon^3 N/2k^2)$), we get:

$$\begin{aligned} \Pr_h \left[\big| \, |\{x \in S : h(x) = 0^\ell\}| \, - 2^{-\ell} \cdot N \big| \, > \, \epsilon \cdot 2^{-\ell} \cdot N \right] \; &< \; \left(\frac{k^2 \cdot 2^\ell}{\epsilon^2 \cdot |S|} \right)^k \\ &< \; 2^{-k} \; = \; \delta \, . \end{aligned}$$

Thus, with probability at least $1 - \delta$, the cardinality of the set $S \cap h^{-1}(0^\ell)$ is at least $t = (1 - \epsilon) \cdot 2^{-\ell} \cdot N$ (and at most $(1 + \epsilon) \cdot 2^{-\ell} \cdot N$). Part i follows.

We now turn to Part ii. Suppose that $|S'| \geq \epsilon \cdot N$ (otherwise, if $|S'| < \epsilon \cdot N$, we may augment S' with $\epsilon N - |S'|$ elements of $\{0,1\}^n \setminus S'$). Applying a similar argument as above to the set S', we conclude that with probability at least $1 - \left(\frac{k^2 \cdot 2^\ell}{\epsilon^2 \cdot |S'|} \right)^k$, the set $h^{-1}(0^\ell)$ contains at most $(1+\epsilon) \cdot 2^{-\ell} \cdot |S'|$ members of S'. Then, using $\epsilon^2 \cdot |S'| \geq \epsilon^3 N \geq 2k^2 \cdot 2^\ell$, it follows that with probability at least $1 - \delta$:

$$\begin{aligned} |S' \cap h^{-1}(0^\ell)| \; &\leq \; (1 + \epsilon) \cdot 2^{-\ell} \cdot |S'| \\ &= \; \frac{1+\epsilon}{1-\epsilon} \cdot \frac{|S'|}{N} \cdot t \end{aligned}$$

where the equality is due to $t = (1 - \epsilon) \cdot 2^{-\ell} \cdot N$. Thus, the probability that the output resides in the set S' is bounded by $\delta + \frac{1+\epsilon}{1-\epsilon} \cdot \frac{|S'|}{N}$, where the first term accounts for the probability that $t' \stackrel{\text{def}}{=} |S' \cap h^{-1}(0^\ell)|$ is greater than $\frac{1+\epsilon}{1-\epsilon} \cdot \frac{|S'|}{N} \cdot t$, and the second term is an upper bound on $\frac{t'}{t}$ (which holds otherwise). Using $\delta \leq \epsilon < 1/3$ and recalling that the original S' was possibly augmented so that $|S'| \geq \epsilon N$, the probability that the output resides in S' is upper-bounded by

$$\delta + \frac{1+\epsilon}{1-\epsilon} \cdot \frac{\max(|S'| \, , \, \epsilon \cdot N)}{N} \; < \; \epsilon + (1 + 3\epsilon) \cdot \max\left(\frac{|S'|}{N} \, , \epsilon \right) \; < \; 4\epsilon + \frac{|S'|}{N}$$

and the lemma follows. ■

B. Some Comments regarding Theorem 2.3

Recall that Theorem 2.3 is equivalent to the following two claims:

$$\mathbf{AM}(b, m) \; \subseteq \; \mathbf{AM}(b^2 \cdot \mathrm{poly}(m), \lceil m/2 \rceil) \, , \tag{B.1}$$

$$\mathbf{AM}(b, m) \; \subseteq \; \mathbf{AM}((b \cdot m)^{O(m)}, 2) \, . \tag{B.2}$$

As stated in the main text, (B.1) is implicit in the work of Babai and Moran [BM88]. However, (B.2) does not follow by merely applying (B.1) for $\lceil \log_2 m \rceil$ times (unless m is a constant), because such a sequence of applications does not allow to keep track of the computational complexity of the verifier. The problem is that (B.1) does not assert that the computational complexity of the new verifier is polynomial in the computational complexity of the original verifier (but rather that if the latter is polynomial in $n+b$ then the former is polynomial in $n+b' = n+b^2 \cdot \mathrm{poly}(m)$). Indeed, by going into the original proof of (B.1), one may verify that the computational complexity of the new verifier is polynomially related to that of the original verifier, because the new verifier just manipulates the new messages, derives one set of original messages and applies the original verifier to it. Still, it seems nicer (and more convincing) to present a direct proof of (B.2). This is done by "unraveling"the recursion, and "optimizing" things a little (as done below).

We assume that the reader is familiar with the terminology of public-coin (a.k.a Arthur-Merlin) interactive proofs, where the verifier is called Arthur and the prover is called Merlin. By possibly using padding, we may assume, without loss of generality, that all Arthur's messages are of the same length n. Starting with an $\mathbf{AM}(b,m)$ system, we modify it so that each Merlin message has length exactly b. (This increases the total number of bits sent by the prover by a factor of m, but we do not care.) Let us denote a generic message of Arthur by $\alpha \in \{0,1\}^n$, and a generic message of Merlin's by $\beta \in \{0,1\}^b$.

For sake of perspective and as a warm-up, we start (see Section B.1) by presenting the main idea of the Babai–Moran transformation [BM88], and recall (in Section B.2) how it is applied in order to cut the number of rounds by half and establish (B.1). However, one may skip these preliminaries and proceed directly to Section B.3, where we prove (B.2).

B.1. The basic switch (from MA to AM). We start by recalling the main idea underlying the transformation of Babai and Moran [BM88]. An Arthur–Merlin proof system can be viewed as a game between an honest Arthur and Merlin that alternate in taking moves such that Arthur takes random moves and Merlin takes optimal ones with respect to a fixed predicate that is evaluated on the full transcript of the game's execution. The value of the game is defined as the expected value of an execution of the game (when played against an optimal Merlin).

The basic idea is to transform an MA-game (*i.e.,* a two-move game in which Merlin moves first and Arthur follows) into an AM-game (in which Arthur moves first and Merlin follows). That is, in the original game Merlin first sends

$\beta \in \{0,1\}^b$, Arthur responds with a random $\alpha \in \{0,1\}^n$, and the value of the game is defined given by $v(\beta,\alpha)$. Then, (for t to be specified) we switch the order of moves by letting Arthur first send a random sequence $(\alpha^1,...,\alpha^t) \in \{0,1\}^{tn}$, then Merlin responds with an $\beta \in \{0,1\}^b$ and the value is defined as the average of the values $v(\beta,\alpha^i)$, over $i = 1,...,t$. Using $t = O(b)$, this guarantees that for every $\beta \in \{0,1\}^b$ with very high probability (*i.e.*, probability at least $1-2^{-b-2}$), the value of the modified game (*i.e.*, $\frac{1}{t}\sum_{i=1}^{t} v(\beta,\alpha^i)$ for random α^i's) approximates the value of the original game (*i.e.*, $\mathrm{E}_\alpha(\beta,\alpha)$) up to an additive constant. Thus:

$$\Pr_{\alpha^1,...,\alpha^t}\left[\forall\beta \in \{0,1\}^b : \left|\frac{1}{t}\sum_{i=1}^{t} v(\beta,\alpha^i) - \mathop{\mathrm{E}}_{\alpha}(\beta,\alpha)\right| > \frac{1}{6}\right] \;<\; 2^b \cdot 2^{-b-3} \;<\; \frac{1}{4}.$$

This immediately implies that the class **MA** is contained in the class **AM**. A similar reasoning can be applied to longer games (by considering the value of the residual game after two moves) implies that the class $\mathbf{A(MA)^j}$ is contained in the class $\mathbf{AAM(MA)^{j-1}} = \mathbf{A(MA)^{j-1}}$. This implies $\mathbf{AM}(poly, O(1)) = \mathbf{AM}$ and, more generally, $\mathbf{AM}(poly, b+O(1)) = \mathbf{AM}(poly, b)$ (for any $b \geq 2$).

B.2. Concurrent switches in mid-game ($[MAMA]^r$ to $[AMMA]^r = [AM]^r A$). Sequential applications of the "MA-to-AM switch" allow to reduce the number of rounds by any additive constant. In order to cut the number of rounds by a constant, one may apply the "MA-to-AM switch" concurrently to disjoint segments of the game. That is, suppose that the original game proceeds in r stages, where the i^{th} stage ($i \in [r]$) is as follows:

1. Merlin selects $\beta_{2i-1} \in \{0,1\}^b$,
2. Arthur responds with a random $\alpha_{2i-1} \in \{0,1\}^n$,
3. Merlin selects $\beta_{2i} \in \{0,1\}^b$,
4. Arthur responds with a random $\alpha_{2i} \in \{0,1\}^n$,

The value of the corresponding execution of the game is defined as $v(\beta_1,\alpha_1,\beta_2,\alpha_2,...,\beta_{2r-1},\alpha_{2r-1},\beta_{2r},\alpha_{2r})$. For $t = \mathrm{poly}(r)\cdot b$, we transform the above game into the following corresponding r-stage game, where the i^{th} stage ($i \in [r]$) is as follows:

1. Arthur selects a random sequence $(\alpha^1_{2i-1},...,\alpha^t_{2i-1}) \in \{0,1\}^{tn}$,
2. Merlin responds with a single $\beta_{2i-1} \in \{0,1\}^b$,

3. Merlin further selects and sends a sequence $(\beta_{2i}^1, ..., \beta_{2i}^t) \in \{0,1\}^{tb}$,

4. Arthur responds with a random $c_i \in [t]$ and a random $\alpha_{2i} \in \{0,1\}^n$,

The value of the corresponding execution of the game is defined as $v(\beta_1, \alpha_1^{c_1}, \beta_2^{c_1}, \alpha_2, ..., \beta_{2r-1}, \alpha_{2r-1}^{c_r}, \beta_{2r}^{c_r}, \alpha_{2r})$. Observe that, starting from a game of the form $(MA)^{2r} = (MAMA)^r$, we have obtained a game of the form $(AMMA)^r = A(MA)^r$.

Each of the r switches is analyzed by first considering a random choice of Arthur's first move and the average over its choices of $c_i \in [t]$ in its second move. Specifically, for $\vec{\gamma} = (\beta_1, \alpha_1, \beta_2, \alpha_2, ..., \beta_{2(i-1)-1}, \alpha_{2(i-1)-1}, \beta_{2(i-1)}, \alpha_{2(i-1)})$,

$$\begin{aligned}
& v_{\vec{\gamma}}(\beta_{2i-1}, \alpha_{2i-1}, \beta_{2i}, \alpha_{2i}, ..., \beta_{2r-1}, \alpha_{2r-1}, \beta_{2r}, \alpha_{2r}) \\
& \quad \stackrel{\text{def}}{=} v(\vec{\gamma}, \beta_{2i-1}, \alpha_{2i-1}, \beta_{2i}, \alpha_{2i}, ..., \beta_{2r-1}, \alpha_{2r-1}, \beta_{2r}, \alpha_{2r}), \\
\overline{v}(\vec{\gamma}) & \stackrel{\text{def}}{=} \max_{\beta_{2i-1}} \left\{ \mathop{\mathrm{E}}_{\alpha_{2i-1}} \left[\max_{\beta_{2i}} \left\{ \mathop{\mathrm{E}}_{\alpha_{2i}} \left[\overline{v}(\vec{\gamma}, \beta_{2i-1}, \alpha_{2i-1}, \beta_{2i}, \alpha_{2i}) \right] \right\} \right] \right\}.
\end{aligned}$$

That is, $\overline{v}(\vec{\gamma})$ is the value of the game conditioned on the $2i-2$ first messages having transcript $\vec{\gamma}$. The key observation is that for any $\vec{\gamma}$ and every β_{2i-1}, with probability at least $1 - (1/10r) \cdot 2^{-b}$ over the choice random sequence $(\alpha_{2i-1}^1, ..., \alpha_{2i-1}^t) \in \{0,1\}^{tn}$, we have

$$\begin{aligned}
& \frac{1}{t} \cdot \sum_{j=1}^{t} \max_{\beta_{2i}^j} \left\{ \mathop{\mathrm{E}}_{\alpha_{2i}} \left[\overline{v}(\vec{\gamma}, \beta_{2i-1}, \alpha_{2i-1}^j, \beta_{2i}^j, \alpha_{2i}) \right] \right\} \\
& \quad = \mathop{\mathrm{E}}_{\alpha_{2i-1}} \left[\max_{\beta_{2i}} \left\{ \mathop{\mathrm{E}}_{\alpha_{2i}} \left[\overline{v}(\vec{\gamma}, \beta_{2i-1}, \alpha_{2i-1}, \beta_{2i}, \alpha_{2i}) \right] \right\} \right] \pm \frac{1}{10r}.
\end{aligned}$$

Applying the same reasoning to each possible $\beta_{2i-1} \in \{0,1\}^b$, we conclude that with probability at least $1 - (1/10r)$ over the choice of the α_{2i-1}^js,

$$\max_{\beta_{2i-1}, \beta_{2i}^1, ..., \beta_{2i}^t} \left\{ \mathop{\mathrm{E}}_{c_i, \alpha_{2i}} \left[\overline{v}(\vec{\gamma}, \beta_{2i-1}, \alpha_{2i-1}^{c_i}, \beta_{2i}^{c_i}, \alpha_{2i}) \right] \right\} = \overline{v}(\vec{\gamma}) \pm \frac{1}{10r}.$$

In the actual analysis we consider $p \stackrel{\text{def}}{=} \text{poly}(r)$ parallel executions of each of the games, and define the value of each parallel game to be the average of the values of the corresponding copies.[3] One may show that each of the r switches

[3] This part of the analysis is different from the analysis in [BM88]. In [BM88] one first reduces the error probability of the original game (also by parallel executions), and argues that each of the residual values to be considered is very likely to be very close to either 0 or 1. Here by considering the value of the average of p copies, we can relate the likely value of this average to its expected value.

approximately maintains the value of the original game. That is, for every $i = 0, ..., r$, consider the value of the (p-parallel) game obtained by performing only the first i switches. Denote these (p-parallel) games by $G_0, ..., G_r$, and note that G_0 is the (p-parallel version of the) original $(MAMA)^r$ game, and G_r is the resulting (p-parallel) $A(MA)^r$ game. For every $i = 1, ..., r$, we consider the difference between the value of G_{i-1} and the value of G_i. For any fixed transcript of the first $i-1$ stages, with probability at least $1-(1/10r)$ the values of the residual executions of G_{i-1} and G_i differ by at most $1-(1/10r)$. Thus, with probability at least $9/10$, the value of a random execution of G_r is within 0.1 of the (expected) value of G_0 (which equals the expected value of the original game).

B.3. A direct approach (to placing $(\mathbf{AM})^{\mathbf{r}}$ in AM). Think of the original $2r+1$-message $(MA)^r M$ game as a tree of depth r with nodes being labeled by Merlin moves (each in $\{0,1\}^b$) and edges being labeled by Arthur moves (each in $\{0,1\}^n$). Thus, the tree has $(2^n)^r$ leaves. Each Merlin strategy corresponds to a different node-labeling of the tree, whereas the edge labels are fixed. Such a vertex-labeling assigns Boolean values to the leaves (in correspondence to A's decision), and by this to all internal nodes such that the value of an internal node is the average of the value of its 2^n children. The value of a specific Merlin strategy is just the value of the root under the corresponding vertex-labeling.

Following [GH98], we consider selecting a random subtree of the above tree so that for each internal node we select at random $t = \text{poly}(r) \cdot b$ children. Again, each specific Merlin strategy used as vertex-labeling (of the random sub-tree) defines a value of the root, a value that corresponds to a new game in which Arthur's moves are restricted to this subtree. We shall prove that, with high probability over the choice of the random subtree, for each specific Merlin strategy (*i.e.,* a labeling of all vertices in the full tree), the value of the subtree approximates the value of the full tree. This leads to the following new (2-message) AM game:

1. Arthur selects and sends Merlin a random subtree.

2. Merlin provides a labeling of the vertices in this subtree.

Arthur computes the value of the root of the subtree, under the vertex-labeling (provided by Merlin), and decides accordingly. Note that all complexities (*i.e.,* the number of bits sent by Merlin as well as the computational complexity of the new Arthur) are related to the size of the subtree, which equals $t^r = (\text{poly}(r) \cdot b)^r = b^{O(r)}$ (since $b \geq r$).

Analysis: We consider hybrid tree distributions in which the first i top levels are as in a random subtree, and the bottom $r-i$ levels are as in the full tree (i.e., span all $(2^n)^{r-i}$ leaves). The zero hybrid (i.e., $i=0$) corresponds to the full tree, and the r^{th} hybrid is a random subtree. We will show that for every $i=0,...,r-1$, with probability $1-(1/10r)$, for every vertex-labeling (i.e., Merlin strategy) of levels $0,...,i$ and the best possible Merlin strategy for levels $i+1,...,r$, the (random) value of the $i+1^{\text{st}}$ hybrid approximates the value of the i^{th} hybrid up to an additive term of $1/10r$. It follows that, with probability at least 0.9, the value of a random subtree (under the best labeling) approximates the value of the full tree (under the best labeling) up to an additive term of 0.1.

Consider any fixed tree T of the i^{th} hybrid, and the random $i+1^{\text{st}}$ hybrid trees obtained by selecting a sample of t vertices out of the 2^n children of each level i node v in T. (Recall: The root is zero level, and the leaves are at level r.) For each vertex-labeling of levels $0,...,i$, we consider the best possible Merlin strategy for levels $i+1,...,r$. Such strategy assigns values to all vertices of level $r,...,i+1$ of T, and the value of any level i node vertex is merely the average of the value of its children. Specifically, the value at a leaf is determined by the path to the leaf (which correspond to the edge labels) and by the labels of the vertices on this path, where the first $i+1$ vertex-labels are determined by the fixed labeling of levels $0,...,i$, and the labels of vertices at levels $i+1,...,r$ are determined by the optimal Merlin moves. The values of all other internal nodes are determined recursively as the average of the values of their children, where nodes at levels $i+1,...,r-1$ have 2^n children (as in the original tree), nodes at level i have t random children, and nodes at levels $0,...,i-1$, have t children as determined by T.

Our aim is to prove that, with high probability over the choice of children for the i^{th} level nodes, the value of each of these nodes under any labeling of the vertices in levels $0,...,i$ is approximately the average of the values of all its 2^n children. Thus, the $i+1^{\text{st}}$ hybrid approximates the i^{th} hybrid.

Fixing any level i node vertex, denoted v, and any vertex-labeling for levels $0,...,i$, we consider the value of v in the random ($i+1^{\text{st}}$) hybrid tree (which extends T). Actually, we only fix the vertex-labeling of v and its ancestors, because only these labels affect the value of the vertices in the subtree rooted at v. For each such labeling, with probability at least $1-2^{-\Omega(t\epsilon^2)}$, the average value of t random children of v approximates the average value of all 2^n children of v up to an additive $\epsilon = 1/10r$. Since there are at most $2^{(i+1)b} \leq 2^{rb}$ possible labelings to the vertices along this path, with probability $1-2^{rb}\cdot 2^{-\Omega(t/r^2)}$, for every vertex-labeling of levels $0,...,i$, the value of v in T is within $1/10r$ of its value in a random $i+1^{\text{st}}$ hybrid tree obtained from T (by sampling its level $i+1$ nodes). In case the above holds, we call v a good vertex, otherwise we call it bad; that is, v is good with probability at least $1-2^{rb}\cdot 2^{-\Omega(t/r^2)}$. Using $t \geq Cr^2\cdot(b+r\log t+\log 10r)$ for a sufficiently large constant C (*e.g.*, $t=\Theta(r^4 b)$), we conclude that each level i vertex is bad with probability at most $2^{br-t/Cr^2} < 2^{-r\log_2 t-\log_2 10r} = t^{-r}/10r$. Thus, with probability at least $1-(1/10r)$, all vertices of the i^{th} level are good. This means that, with probability at least $1-(1/10r)$, for every vertex-labeling of levels $0,...,i$, the values of all level i nodes in the $i+1^{\text{st}}$ hybrid tree obtained from T is within $1/10r$ of their corresponding values in T.

Considering all possible T's and doing the same for all neighboring hybrid pairs it follows that (as claimed above), with probability at least 0.9, the value of a random subtree (under the best labeling) approximates the value of the full tree (under the best labeling) up to an additive term of 0.1. Hence we obtain $\mathbf{AM}(b,4r) \subseteq \mathbf{AM}(t^r\cdot b,2)$, and (B.2) follows (since $t=O(r^4 b)=b^{O(1)}$). (If one cares then $\mathbf{AM}(b,4r)\subseteq \mathbf{AM}(b^{r+1}\cdot O(r)^{4r},2)$.)

Manuscript received June 2, 2002

Oded Goldreich
Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot, Israel.
oded@wisdom.weizmann.ac.il
http://www.wisdom.weizmann.ac.il/~oded

Salil Vadhan
Division of Engineering and Applied Sciences
Harvard University
Cambridge, MA, USA.
salil@eecs.harvard.edu
http://eecs.harvard.edu/~salil

Avi Wigderson
Institute for Advanced Study
Princeton, NJ, USA.
avi@ias.edu

Institute for Computer Science
Hebrew University
Jerusalem, Israel.